Laporan 4 Praktikum Sistem Digital

Sidartha Prastya. P - 13219033

*Institut Teknologi Bandung*

*Jalan Ganesha No.10, Bandung, Indonesia*

13219033@std.stei.itb.ac.id

***Abstract*—Laporan ini berisikan mengenai simulasi rangkaian logika 7-segment LED sederhana untuk FPGA Altera EP4CE6E22C8. Tujuan dari praktikum ini adalah dapat membuat salah satu dari 7-segment LED dapat menyala dan menampilkan angka 0 hingga 9 dengan input berupa push button. Desain dibuat dengan menggunakan VHDL.**

***Keywords*— 7-segment, FPGA, VHDL.**

1. INTRODUCTION

Laporan ini merupakan tugas untuk praktikum 4 Sistem Digital EL2102, yaitu mengenai simulasi penggunaan FPGA Altera Cyclone IV E dengan menggunakan software Intel Quartus.

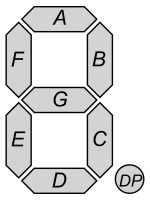
1. SPESIFIKASI

*Hardware* yang digunakan pada praktikum ini adalah satu set FPGA Altera Cyclone EP4CE6E22C8 dengan *software* yang digunakan adalah Intel Quartus. Pada simulasi ini terdapat 4 buah *push button* serta terdapat output berupa 7-segment digital LED. Lalu, dalam satu digit 7-segment LED terdapat 7 buah segment pembentuk angka serta 1 segment titik, akan tetapi pada simulasi ini hanya menggunakan segment pada angka.

Hasil akhir yang diinginkan adalah LED dapat menampilkan angka 0 hingga 9 dengan menekan *push button* yang berperan seperti bilangan biner 4-bit.

1. PERANCANGAN

Pada simulasi ini terdapat 4 input dan 7 output. Input dapat dianggap sebagai bilangan biner 4-bit dan output dapat dibagi per-*segment* sebagai a, b, c, d, e, f, g.



Gambar 1. 7-Segment LED

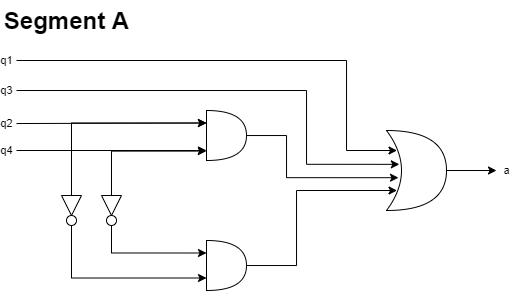
1. *Pembagian Segment setiap Digit*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Angka | a | b | c | d | e | f | g |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Tabel 1. Pembagian Kondisi Segment untuk Setiap Digit

1. *Desain Setiap Segment*

Setiap segment memiliki rangkaian logika masing-masing. Misalkan input yang digunakan adalah q4, q3, q2, dan q1. Dengan menggunakan Karnaugh Map, kita akan mendapatkan logika sebagai berikut.

1. *Segment A* : q1 + q3 + q2 q4 + ­­­­­­­­­¬q2 ¬q4

Gambar 2. Logika Segment A

1. *Diagram, schematic

   Description automatically generatedSegment B* : ¬q2 + ¬q3 ¬q4 + q3 q4

Gambar 3. Logika Segment B

1. *Diagram

   Description automatically generatedSegment C* : q2 + ¬q3 + q4

Gambar 4. Logika Segment C

1. *Segment D* : q1 +¬q2 ¬q4 + ¬q4 q3 + ¬q3 q2 q4 +

*Diagram

Description automatically generated* ¬q2 q3

Gambar 5. Logika Segment D

1. *Diagram

   Description automatically generatedSegment E* : ¬q2 ¬q4 + ¬q4 q3

Gambar 6. Logika Segment E

1. *Diagram

   Description automatically generatedSegment F* : q1 + ¬q3 ¬q4 + ¬q4 q2 + q2 ¬q3

Gambar 7. Logika Segment F

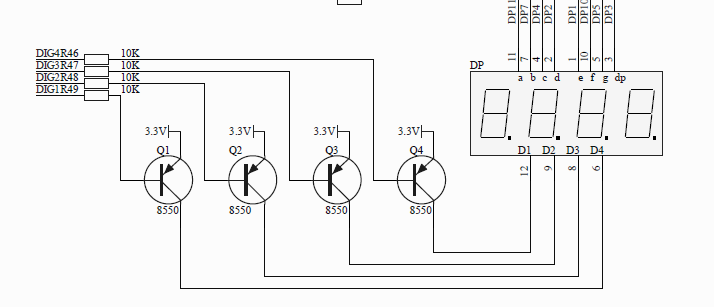
1. *Diagram

   Description automatically generatedSegment G* : q1 + q2 ¬q3 + ¬q3 ¬q2 + q2 ¬q4

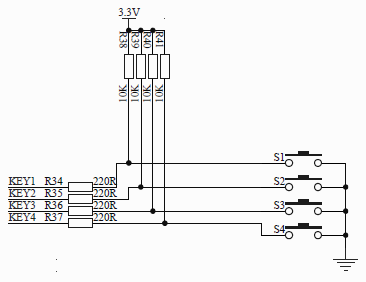
Gambar 8. Logika Segment G

1. *Desain pada FPGA*

Pada rangkaian komponen *push button* akan memberikan input ‘0’ ketika ditekan dan juga sebaliknya. Hal ini juga berlaku untuk LED. Ketika diberikan input berupa ‘0’, maka LED akan menyala, dan juga sebaliknya. Oleh karena itu, semua desain logika yang telah dibuat sebelumnya akan diberikan gerbang “not” dan juga berlaku untuk semua input yang akan diberikan.



Gambar 9. Rangkaian 7-Segment LED pada FPGA



Gambar 10. Rangkaian Push Button pada FPGA

1. IMPLEMENTASI
2. *Program VHDL*

Dalam implementasi meggunakan program VHDL, ditentukan terlebih dahulu entity yang terdiri dari

Input : i4, i3, i2, dan i1

Output : a, b, c, d, e, f, g, dan dig

Lalu, untuk mengubah input menjadi kebalikan dari nilai asli, maka diperlukan suatu signal.

Signal : q4, q3, q2, dan q1

Setelah itu, variable “dig” langsung diinisiasi sebagai ‘0’ dan memasukkan signal ke dalam logika sesuai dengan yang telah didesain sebelumnya dengan ditambahkan gerbang “not” di akhir.

Berikut adalah program VHDL yang telah dibuat.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

Entity seven\_segment is

port(

-- Input

i1 : in std\_logic;

Graphical user interface

Description automatically generated

*B. Pin Planner*

Terakhir, berikut ini adalah hasil dari LED bila memasukkan input lebih dari 9 (1010, 1011, dst.).

Gambar 12. Pengujian FPGA dengan Hasil Salah/Lebih dari 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Node Name* | *Direction* | | *Location* | |
| dig | Output | | PIN\_133 | |
| a | Output | | PIN\_128 | |
| b | Output | | PIN\_121 | |
| c | Output | | PIN\_125 | |
| d | Output | | PIN\_129 | |
| e | Output | | PIN\_132 | |
| f | | Output | | PIN\_126 | |
| g | | Output | | PIN\_124 | |
| i1 | | Input | | PIN\_88 | |
| i2 | | Input | | PIN\_89 | |
| i3 | | Input | | PIN\_90 | |
| i4 | | Input | | PIN\_91 | |

Tabel 2. Daftar Pin pada Program

i2 : in std\_logic;

i3 : in std\_logic;

i4 : in std\_logic;

-- Output

dig: out std\_logic;

a : out std\_logic;

b : out std\_logic;

c : out std\_logic;

d : out std\_logic;

e : out std\_logic;

f : out std\_logic;

g : out std\_logic

);

End entity;

Architecture RTL of seven\_segment is

signal q1,q2,q3,q4 : std\_logic;

Begin

-- Inisiasi dig menjadi '0'

dig <= '0';

-- Inisiasi q1 hingga q4 sebagai kebalikan dari input

q1 <= not(i1);

q2 <= not(i2);

q3 <= not(i3);

q4 <= not(i4);

-- Rangkaian logika dengan

ditambahkan "not"

a <= not(q1 or q3 or (q2 and q4) or

((not q4) and (not q2)));

b <= not((not q2) or ((not q3) and

(not q4)) or (q3 and q4));

c <= not(q2 or (not q3) or q4);

d <= not(((not q2) and (not q4)) or

(q3 and (not q4)) or (q2 and

(not q3) and q4) or ((not q2)

and q3) or q1);

e <= not(((not q2) and (not q4)) or

(q3 and (not q4)));

f <= not(q1 or ((not q3) and (not

q4)) or (q2 and (not q3)) or

(q2 and (not q4)));

g <= not(q1 or (q2 and (not q3)) or

(not q2 and q3) or (q2 and (not

q4)));

End architecture;

V. PENGUJIAN

Graphical user interface

Description automatically generatedA picture containing phone, cellphone, monitor, sitting

Description automatically generatedA picture containing phone, sitting, monitor, computer

Description automatically generatedA circuit board

Description automatically generatedA picture containing circuit, phone, computer

Description automatically generatedA circuit board

Description automatically generatedA circuit board

Description automatically generatedGraphical user interface

Description automatically generatedA circuit board

Description automatically generatedA circuit board

Description automatically generatedBerikut adalah gambar pengujian dari 0 hingga 9 secara beruru

Gambar 11. Pengujian FPGA dari 0 – 9