



**DIGITAL SYSTEM DESIGN FINAL PROJECT REPORT  
DEPARTMENT OF ELECTRICAL ENGINEERING  
UNIVERSITAS INDONESIA**

**PROJECT TITLE**

**GROUP B8**

<b>Bisma Alif Alghifari</b>	<b>2106731402</b>
<b>Daffa Anis Fahrizi</b>	<b>2106731365</b>
<b>Muhammad Rizky Utomo</b>	<b>2106731320</b>
<b>Muhammad Salman Sadad</b>	<b>2106731371</b>

## **PREFACE**

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa yang telah membantu kami untuk menyelesaikan proyek ini. Jika kami mengerjakan ini tanpa bantuan-Nya, dipastikan kami tidak akan bisa menyelesaikan mata kuliah Praktikum Perancangan Sistem Digital.

Pada masa kini, teknologi berada mengelilingi kita. Dengan teknologi, pekerjaan yang repetitif, atau melelahkan bisa diringankan dengan adanya teknologi. Pada akhirnya, kualitas kehidupan manusia menjadi lebih baik. Kami berharap proyek akhir ini bisa ikut membantu meringankan dan meningkatkan kualitas kehidupan manusia, dari penggunaanya ataupun yang mendapatkan hasil dari penggunaanya.

Dan juga, kami mengucapkan terima kasih kepada semua pihak yang telah membantu kami dalam menyelesaikan proyek ini. Dari orang tua kami, yang telah membantu kami dari dahulu kala sampai sekarang dan seterusnya, dan juga kepada Aslab kami, Bang Raihan Azhari, yang telah membimbing kami dalam penyusunan proyek ini dan bersedia memberi waktu luang untuk melancarkan praktikum kami dari modul ke-1 sampai proyek akhir ini. Kami juga mengucapkan terima kasih untuk kelompok B8 yang berhasil menyusun laporan ini.

Penulis laporan ini berharap para pembaca bisa mendapatkan wawasan baru dalam perancangan sistem digital setelah membaca laporan kami.

Depok, December 10, 2022

Group B8

## **TABLE OF CONTENTS**

### **CHAPTER 1: PENDAHULUAN**

- 1.1 Latar Belakang
- 1.2 Deskripsi Proyek
- 1.3 Tujuan
- 1.4 Peran dan Tanggung Jawab

### **CHAPTER 2: IMPLEMENTASI**

- 2.1 Peralatan
- 2.2 Implementasi

### **CHAPTER 3: PENGUJIAN DAN ANALISIS**

- 3.1 Pengujian
- 3.2 Hasil
- 3.3 Analisis

### **CHAPTER 4: KESIMPULAN**

### **REFERENSI**

### **LAMPIRAN**

LAMPIRAN A: Project Schematic

LAMPIRAN B: Documentation

# CHAPTER 1

## PENDAHULUAN

### 1.1 Latar Belakang

Pada zaman sekarang ini kita masih sering sekali menemukan sistem antrian suatu kantor pelayanan masyarakat seperti perbankan dan perpajakan yang masih sangat konvensional, yaitu hanya dengan mengambil nomor dan menunggu penjaga untuk memanggil atau meneriakkan nomor antrian yang akan dipanggil. Sistem antrian konvensional ini sangat tidak efisien karena bisa menyebabkan alur antrian yang berantakan, pengalaman mengantri yang kurang menyenangkan, dan memeras tenaga para pekerja.

Masalah dan kekurangan yang terjadi jika menggunakan antrian konvensional membuat kelompok kami berdiskusi dan menginisiasikan sebuah ide untuk mendigitalisasi serta mengotomasi sistem antrian tersebut. Sistem yang kami gunakan akan tetap menggunakan untuk kertas sebagai penanda antrian akan tetapi cara memanggil akan melalui sebuah display nomor antrian yang tertera di kertas. Sistem ini juga akan memperlihatkan berapa banyak orang yang sedang mengantri dan orang yang sedang berada pada teller.

Perangkat dengan sistem yang kami buat ini akan membawakan manfaat dari segi kenyamanan antrian, sistem alur antrian yang teratur untuk mencegah kesalahan dalam pemanggilan nomor antrian dan tidak memeras tenaga para pekerja untuk mengatur antrian secara konvensional.

### 1.2 Deskripsi Proyek

*Queue System* (QueueS) merupakan sistem antrian yang digunakan untuk mengelola dan mengatur jalannya antrian di suatu kantor pelayanan masyarakat seperti bank, kantor perpajakan, atau tempat publik. *Queue system* dapat membantu mengurangi kemacetan dan kebingungan atau kesalahan dalam mengantri sehingga proses pelayanan menjadi lebih cepat dan efisien.

Pada suatu bank, *queue system* dapat digunakan untuk melakukan pengelolaan antrian di meja teller. Setiap nasabah yang akan datang akan mendapatkan nomor antrian secara otomatis agar nasabah tidak perlu menunggu di antrian yang panjang dan tidak teratur.

*Queue system* juga dapat menampilkan nomor antrian yang sedang dipanggil di layar monitor untuk memudahkan nasabah dalam mengetahui nomor antrian yang akan dipanggil selanjutnya. Dengan pengimplementasian *queue system*, proses pelayanan di teller bank dapat berjalan lebih cepat, efisien, dan teratur. Hal tersebut akan meningkatkan kepuasan nasabah. Sistem yang kami buat ini juga akan menampilkan jumlah antrian yang ada (indikasi berada pada nomor antrian yang terakhir dipanggil) maka dari itu orang akan bisa mengestimasi berapa lama dia harus menunggu antrian tersebut.

### 1.3 Tujuan

The objectives of this project are as follows:

1. Melakukan penerapan *Finite State machine (FSM)* dalam bentuk VHDL dan dieksekusi dengan FPGA untuk menjalankan suatu perangkat digital
2. Memanfaatkan VHDL kedalam kegiatan sehari hari
3. Membuat sistem antrian berbasis digital yang dapat digunakan dalam kantor pusat pelayanan masyarakat perbankan dan perpajakan.
4. Meningkatkan efisiensi antrian untuk mencegah adanya kesalahan dan ketidaknyamanan jika dijalankan secara konvensional (nondigital) dalam pelayanan masyarakat

### 1.4 Peran dan Tanggung Jawab

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
	Membuat display <i>seven segment</i> (VHDL Component), Membuat Laporan Proyek, dan membuat README github.	Bisma Alif Alghifari

	Membuat Memori (VHDL Component) dan Laporan Proyek	Daffa Anis Fahrizi
Ketua	Mendesain rangkaian secara keseluruhan dan Memimpin alur pengerjaan	Muhammad Rizky Utomo
	Membuat <i>Finite State Machine</i> (FSM) dan membuat TestBench	Muhammad Salman Sadad

Table 1. Roles and Responsibilities

## CHAPTER 2

### IMPLEMENTASI

#### 2.1 Peralatan

Berikut adalah alat yang akan digunakan untuk mengerjakan proyek ini :

- Visual Studio Code (Extension VHDL dan VHDL Lens)
- ModelSim (Simulasi Rangkaian)
- Draw.io (Finite State Machine)
- Finite State Machine Designer - Evan Wallace (Perancangan FSM)
- Google Drive (Kolaborasi kelompok)

#### 2.2 Implementasi

QueueS merupakan program sistem digital untuk mengatur antrian pada bank atau tempat pelayanan publik lainnya. QueueS dapat dikostumisasi untuk memiliki banyak konter teller dalam mempresentasikannya, dengan pengaturan default sebanyak 4 konter dan maksimum konter sebanyak 8. Sistem ini mengimplementasikan *queue* untuk menampung antrian dengan pengaturan default sebanyak 250 orang. Akan tetapi, secara teoritis dapat lebih dari 250 orang.

Program ini memiliki beberapa input dan output untuk menunjang fungsionalitasnya. Inputnya antara lain, *clock signal*, *reset signal*, dan *request ticket signal*. Program ini memiliki output antara lain, *current queue counter value*, *queue display*, dan status dari tiap counter nya.

Program QueueS ini memiliki dua buah komponen yang dihubungkan di bagian main. Komponen yang digunakan antara lain, *memory\_management* dan *bin\_to\_bcd*. Komponen *memory\_management* mengimplementasikan package di dalamnya. Pada komponen *memory-management* berisi beberapa function yang digunakan untuk mengatur antrian. Function tersebut antara lain, ‘tambahAntrian’ untuk melakukan penambahan orang pada antrian, ‘hapusAntrian’ untuk menghapus orang pertama pada antrian ketika orang tersebut dipanggil ke konter teller, ‘cekKosong’ untuk melakukan pengecekan status antrian (antara kosong dan tidak). Komponen *bin\_to\_bcd* digunakan untuk melakukan konversi nilai

biner menjadi BCD yang nantinya akan dihubungkan ke *seven segment display* untuk menampilkan nomor antrian.

Program QueueS juga mengimplementasikan *finite state machine*. Pada *state* S0, ketika sinyal reset bernilai 1 diterima maka akan memasuki *state* S3. Pada *state* ini *counter* dan isi memori akan diatur ulang. Ketika sinyal resetnya bernilai 0 maka akan memasuki *state* S1. Pada *state* ini, jika ada permintaan tiket, *counter* akan dinaikkan dan elemen baru akan ditambahkan ke antrian pelanggan. Jika, tidak ada permintaan tiket, *state machine* akan berpindah ke *state* S2. Pada *state* ini, jika ada konter yang kosong, orang pertama dari antrian pelanggan (elemen pertama) akan diambil dari antrian, seven segment akan menampilkan antrian, dan kemudian kembali ke *state* S0.

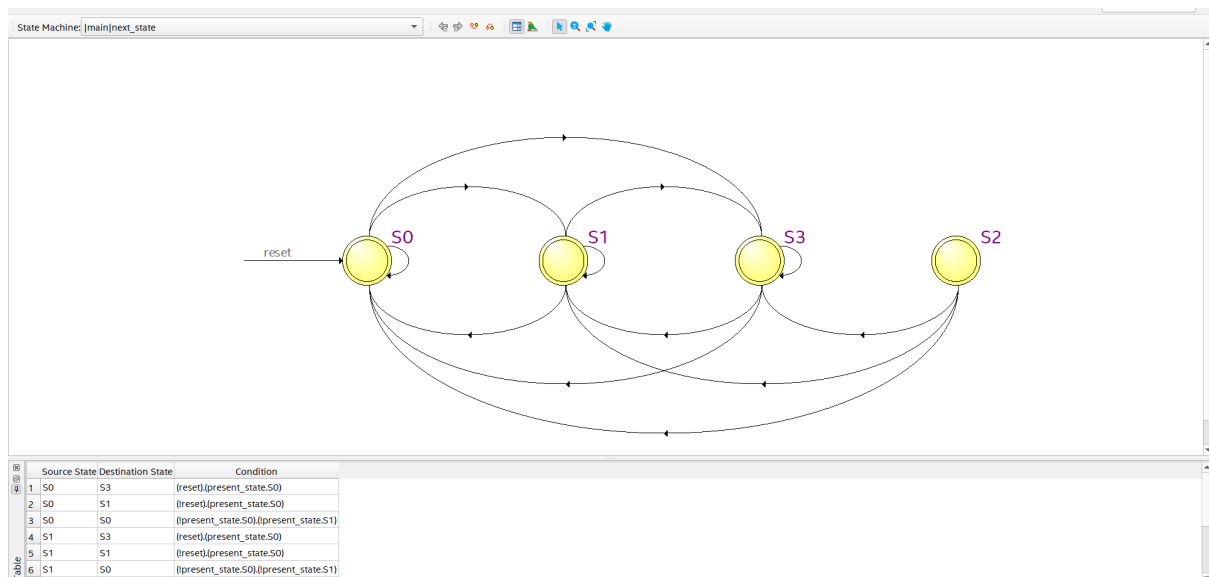


Fig 1. Finite State Machine



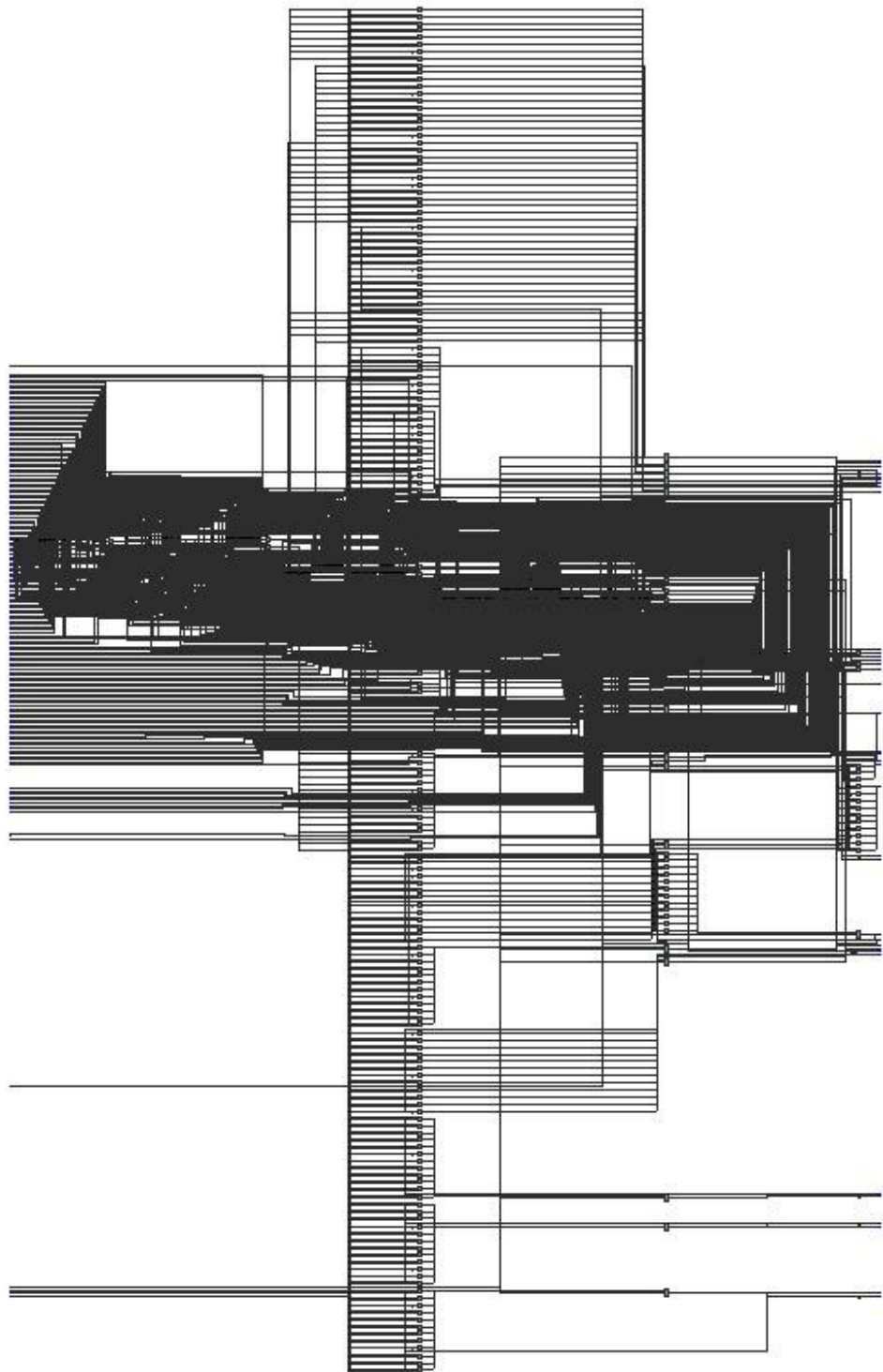


Fig 2. Schematic

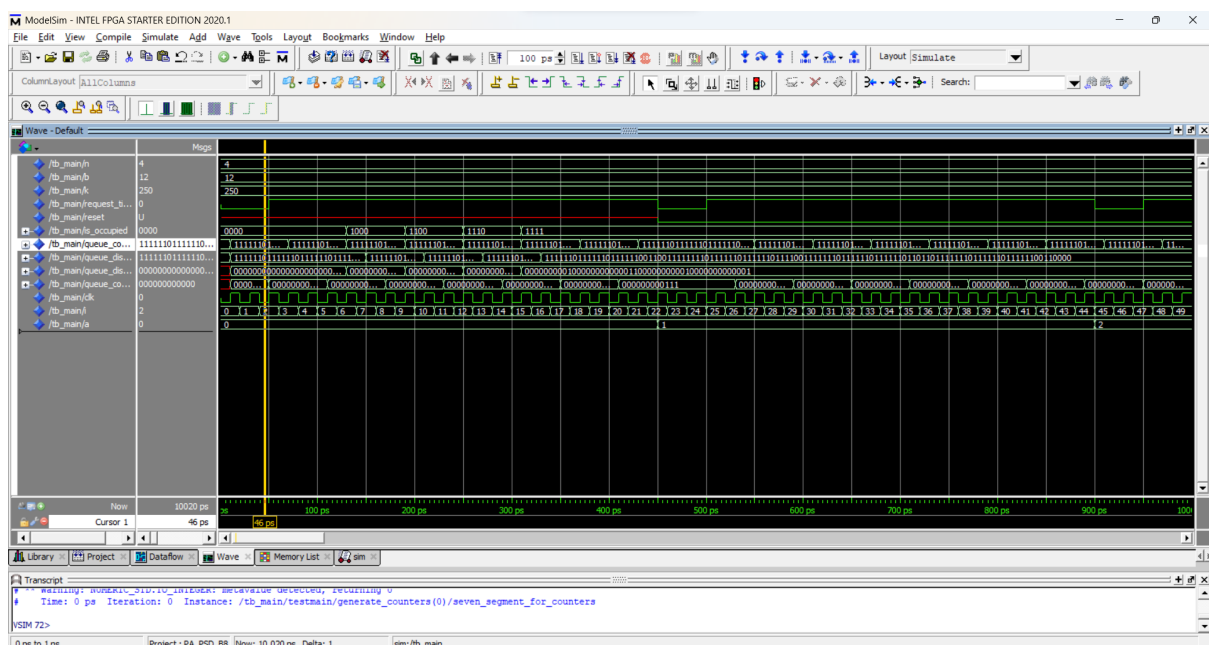
## CHAPTER 3

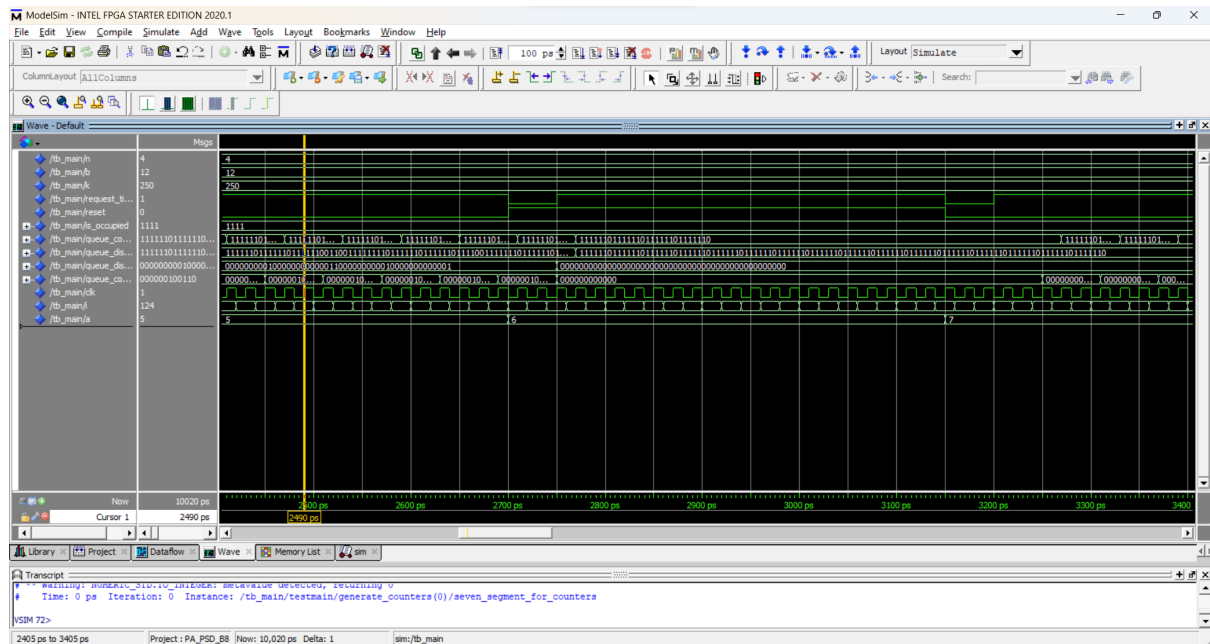
### PENGUJIAN DAN ANALISIS

#### 3.1 Pengujian

Proses pengujian dilakukan dengan membuat program VHDL testbench dengan melakukan port map kepada component main.vhd menggunakan signal-signal yang memiliki tipe data serupa. Kemudian dalam program testbench diberikan beberapa command process yang akan mengendalikan value input dan clock, serta berapa lama simulasi VHDL ingin dijalankan. Untuk proyek akhir ini kami atur supaya simulasi berjalan sebanyak 500 pola pulsa clock high ('1') dan low ('0'), serta jumlah cycle dari QueueS disesuaikan dengan pola clock. Dalam pengujian, program akan dibuat dengan memberi value '1' ke request\_ticket yang akan membuat perpindahan state yang dimana program akan menyimpan dan mengolah data antrian. Dan ada kalanya request ticket diatur valuenya menjadi '0' yang sehingga nomor antrian tidak bertambah. Lalu ada kalanya pula input reset di-set nilainya menjadi '1' yang membuat nomor antrian dihapus menjadi 0.

#### 3.2 Hasil





Dari hasil simulasi yang ditampilkan bisa dilihat bahwa yang proses pengujian yang telah dijelaskan sebelumnya berhasil. Hal itu ditunjukkan dengan nomor urut antrian yang akan

Fig 2. Testing Result

Donec at iaculis leo. Integer congue sed lacus suscipit iaculis. Nulla a augue ut sapien rutrum consectetur. Sed ac dignissim lorem. Maecenas hendrerit nisl a metus posuere, vel vehicula metus eleifend. Mauris blandit, dolor nec malesuada tempor, purus nibh aliquet nibh, at faucibus leo felis a nisi. Donec pharetra leo risus, in vestibulum dui laoreet in. Nulla facilisi. Etiam nec consequat justo. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Etiam pharetra eleifend hendrerit.

### 3.3 Analisis

## CHAPTER 4

## KESIMPULAN

Dari penjelasan di atas, dapat disimpulkan bahwa Queue System (QueueS) adalah sistem antrian yang digunakan untuk mengelola dan mengatur jalannya antrian di suatu kantor pelayanan masyarakat seperti bank, kantor perpajakan, atau tempat publik. Sistem ini dapat membantu mengurangi kemacetan dan kebingungan atau kesalahan dalam mengantri sehingga proses pelayanan menjadi lebih cepat dan efisien. Queue System digunakan untuk melakukan pengelolaan antrian di sebuah counter atau teller. Setiap pengunjung yang akan datang akan mendapatkan nomor antrian secara otomatis agar pengunjung tidak perlu menunggu di antrian yang panjang dan tidak teratur.

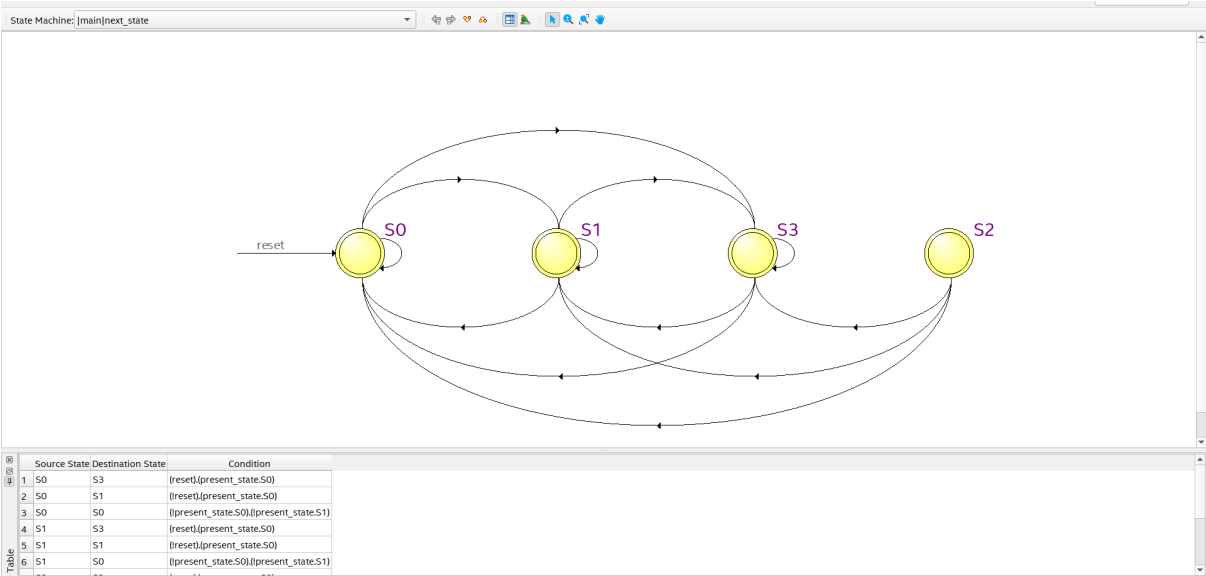
Dari proyek ini kelompok kami juga belajar bahwa dengan VHDL, kita dapat mendeskripsikan isi hardware-hardware yang akan digunakan dalam perancangan sistem digital tanpa perlu membuat real dengan komponen-komponen asli nya atau menggunakan proteus. Cukup menggunakan IDE untuk pemrograman VHDL dan kita dapat dengan mudah menggambarkan komponen apa yang ingin kita pakai dan bagaimana mereka saling terhubung satu sama lain

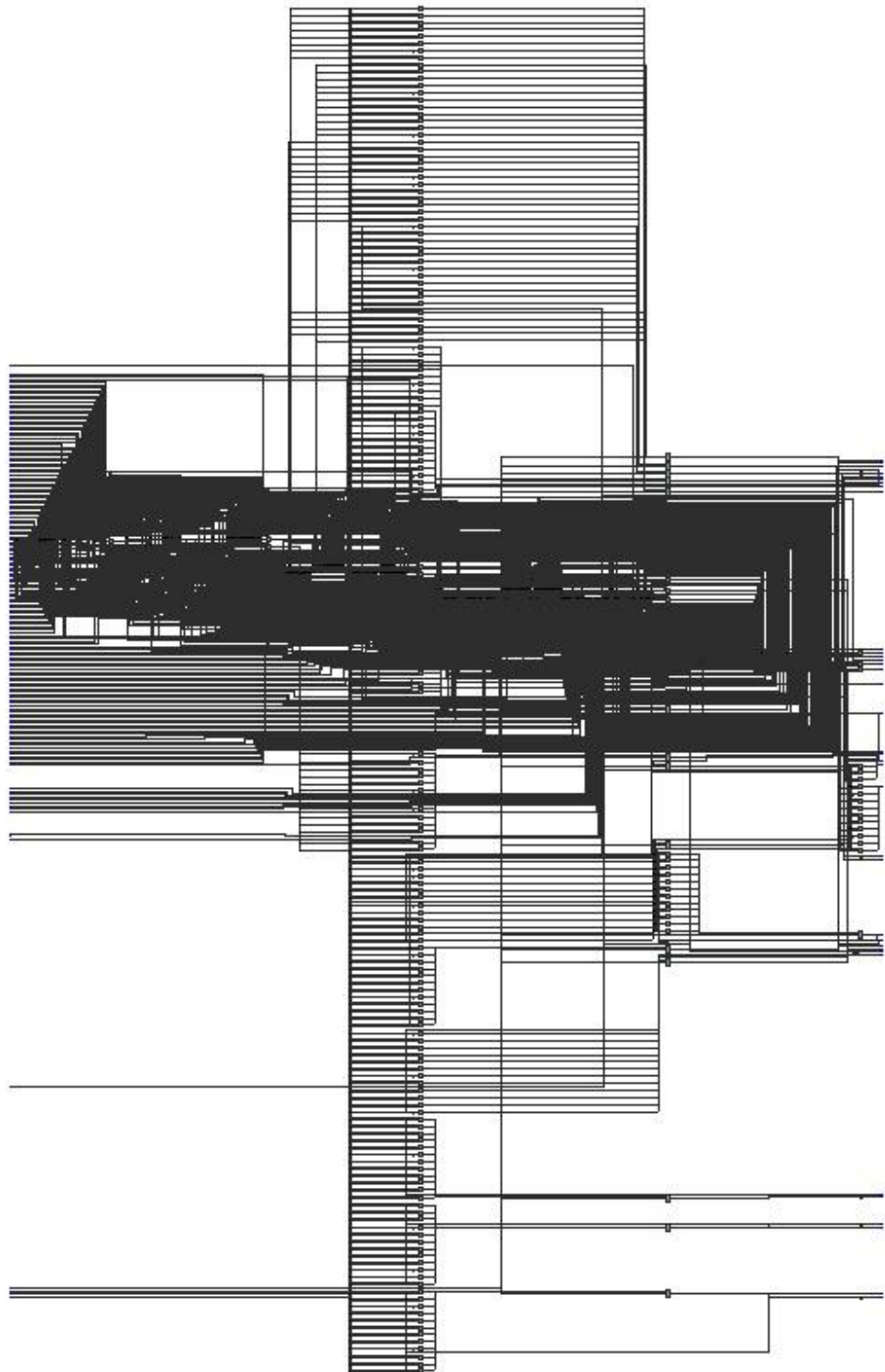
## **REFERENSI**

- [1] Eda Playgrounds, “Eda Playground,” *Generic Package Example*. [Online]. Available: <https://www.edaplayground.com/x/6Mpm>. [Accessed: 9-Dec-2022, 20.00]
- [2] Digital Laboratory, *Tutorial praktikum PSD modul 6: Testbench*, 30-Mar-2021. [Online]. Available: <https://www.youtube.com/watch?v=3Gld-kKh41o>. [Accessed: 10-Dec-2022, 16:00].
- [3] N. A. N. D. L. A. N. D. Russell, “*Function – VHDL Example*,” 30-Jun-2022. [Online]. Available: <https://nandland.com/function-3/#:~:text=Functions%20are%20part%20of%20a,always%20use%20a%20return%20statement>. [Accessed: 7-Dec-2022, 20.00].
- [4] Reference 2
- [5] Reference 3
- [6] Reference 4
- [7] Reference 5
- [8] Reference 6
- [9] And so on

## LAMPIRAN

# Lampiran A: Project Schematic





**Lampiran B: Dokumentasi**

The image shows a dual-monitor setup. The left monitor displays a Verilog HDL code editor with the following code:

```
1 module
2
3 component bin2bcd is
4 generic(
5     N: positive := 32
6 )
7 port(
8     clk, reset: in std_logic;
9     binary_in: in std_logic_vector(N-1 downto 0);
10    bcd_out: out std_logic_vector(N*4 downto 0);
11 )
12 end component;
13
14 type states is (S0, S1, S2, S3);
15 signal present_state, next_state: states := S0;
16
17 --package mem is new work.memory_management generic map (
18     N => 32, K => 4
19 );
20
21 type storage is array (0 to k) of std_logic_vector(N-1 downto 0);
22
23 function lsb4toBin(q: storage; queue_size: positive; value: std_logic_vector(k-1
24     downto 0): storage := q;
25     variable state: storage := q;
26     variable size: std_logic_vector(N-1 downto 0) := (others => "0");
27 begin
28     for i in 0 to queue_size-1 loop
29         if N < 32 then queue_size := (i+4) downto 32 := (others => "0"); and if;
30         if queue_out(i) = reset then
31             queue_out(i) := value; exit;
32         end if;
33     end loop;
34     return queue_out;
35 end lsb4toBin;
36
37 function un4toBin(q: storage; queue_size: positive; value: std_logic_vector(k-1
38     downto 0): storage := q;
39     variable state: std_logic_vector(N-1 downto 0) := (others => "0");
40     for i in 0 to queue_size-1 loop
41         queue_out(i) := queue_out(i);
42         if i < queue_size-1 then queue_out(i+1) := (others => "0"); and if;
43     end loop;
44     return queue_out;
45 end un4toBin;
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

The right monitor displays a logic simulator with the following code:

```
1 -- Created and revised by Alina Allif
2
3 library IEEE;
4 use IEEE.std_logic_1164.all;
5 use IEEE.numeric_std.all;
6 use IEEE.math_real.all;
7
8
9
10 entity bin2bcd is
11 generic(
12     N: positive := 32
13 );
14 port(
15     clk, reset: in std_logic := '0';
16     binary_in: in std_logic_vector(N-1 downto 0);
17     bcd_out: out std_logic_vector(N*4 downto 0) := "000000";
18 )
19 and bin2bcd is
20
21 architecture behavior of bin2bcd is
22     signal bcd0, bcd1, bcd2, bcd3: std_logic_vector(i downto 0);
23 begin
24
25     bcd0 <= std_logic_vector(to_integer(to_integer(binary_in) and 10, bcd0'length));
26     bcd1 <= std_logic_vector(to_integer(to_integer(binary_in) and 100 / 10, bcd1'length));
27     bcd2 <= std_logic_vector(to_integer(to_integer(binary_in) and 1000 / 100, bcd2'length));
28     bcd3 <= std_logic_vector(to_integer(to_integer(binary_in) and 10000 / 1000, bcd3'length));
29
30     --end of
31
32     --end of
33
34     --end of
35
36     --end of
37
38     --end of
39
40     --end of
41
42     --end of
43
44     --end of
45
46     --end of
47
48     --end of
49
50     --end of
51
52     --end of
53
54     --end of
55
56     --end of
57
58     --end of
59
60     --end of
61
62     --end of
63
64     --end of
65
66     --end of
67
68     --end of
69
70     --end of
71
72     --end of
73
74     --end of
75
76     --end of
77
78     --end of
79
80     --end of
81
82     --end of
83
84     --end of
85
86     --end of
87
88     --end of
89
90     --end of
91
92     --end of
93
94     --end of
95
96     --end of
97
98     --end of
99
100
```

The bottom of the image shows a Windows taskbar with various icons and a system clock displaying 10:58 AM on 11/11/2023.