FUNDAMENDTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

Data Compressor Circuit

GROUP B7

GROUP MEMBER 1	**Ahmad Rifqi**
GROUP MEMBER 2	**Michael Winston Tjahaja**
GROUP MEMBER 3	**Laode Alif Masum**
GROUP MEMBER 4	**Mochammad Dyenta**

PREFACE

The main goal of this project was to be able to pass our mandatory Digital System Design Practicum. However, we also wanted to push ourselves and use the knowledge and skills that we have gained throughout the semester to be able to create something that we can be proud of and show the progress of our growth as well.

The design we have made for this project is a reflection of our combined efforts and represents the culmination of our learning. We hope this report will not serve only as evidence of our achievement overall, but also as a helpful resource for others who are going to or currently are learning this subject. By sharing our experience and insights through this project, we hope to be able to contribute onto the ongoing conversation and the further progression of Digital System Design.

Depok, December 07, 2022

Group B7

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Data bits often contain repetitive patterns that often leads into inefficiency and errors when transmitting data. To help counteract this issue, we proposed a Data Compressor Circuit that functions by identifying and removing repeating patterns in order to compress data.

Our main goal is to improve the efficiency and reliability of data transmission by reducing the amount of data that is transmitted normally. In this paper, we describe the design and implementation of the Data Compressor Circuit, as well as evaluating its performance by running a series of experiments.

1.2 PROJECT DESCRIPTION

Our program has two main functions:

1) To count the number of repetitions of a given data string,
2) to reset the count when the data string changes.

The program runs by incrementing a counter whenever it receives an input that matches the current data bit. The counter will reset to 0 when the input string changes, and the program starts counting repetitions of the new input. This allows the program to efficiently compress data by identifying and removing any repetitive patterns.

1.3 OBJECTIVES

The objectives of this project are as follows:

1. To reduce the size of data that is passed through a network
2. To increase the success rate transmission of data
3. To make the process of transmitting data more efficient

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Role 1	Main Coding of the project	Ahmad Rifqi, Laode Alif Masum, Michael Winston Tjahaja
Role 2	Test Bench Creation	Ahmad Rifqi, Laode Alif Masum
Role 3	ModelSim Simulation	Michael Winston Tjahaja, Mochammad Dyenta
Role 4	Report	Michael Winston Tjahaja, Ahmad Rifqi
Role 5	Quartus Prime Model	Ahmad Rifqi
Role 6	Power Point Presentation	Mochammad Dyenta

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

The tools that are going to be used in this project are as follows:

- VSCode
- ModelSim – Intel FPGA Starter Edition Model (Quartus Prime 20.1)
- Microsoft Word
- Quartus Prime 21.1

2.2 IMPLEMENTATION

In making this project, practitioners used the following applications:

Image 1. Application Logos

VSCode was used to create both the base code for the project and the test benches needed to test the results. ModelSim was used to simulate and prove that the project had reached the desired outcome. Quartus Prime was used to create the real-life structure of the line of coding that we had made. Microsoft Word was used to make the report for this project.

The encoder has four states: INIT, CMP, MRK, and FIN. In the INIT state, the incoming input is placed in the buffer and the buffer frequency is reset. In the CMP state, the new input is compared to the previous input, and if the input is the same, the count is incremented and the next state is CMP. If the sequence detects a change in input, the CMP state will enter the MRK state. The MRK state is used for processing before the sequence outputs the input. After the MRK state, the sequence enters the FIN state. In the FIN state, the initial sequence is connected to the frequency and produces a 16-bit output for transmission to the destination device.

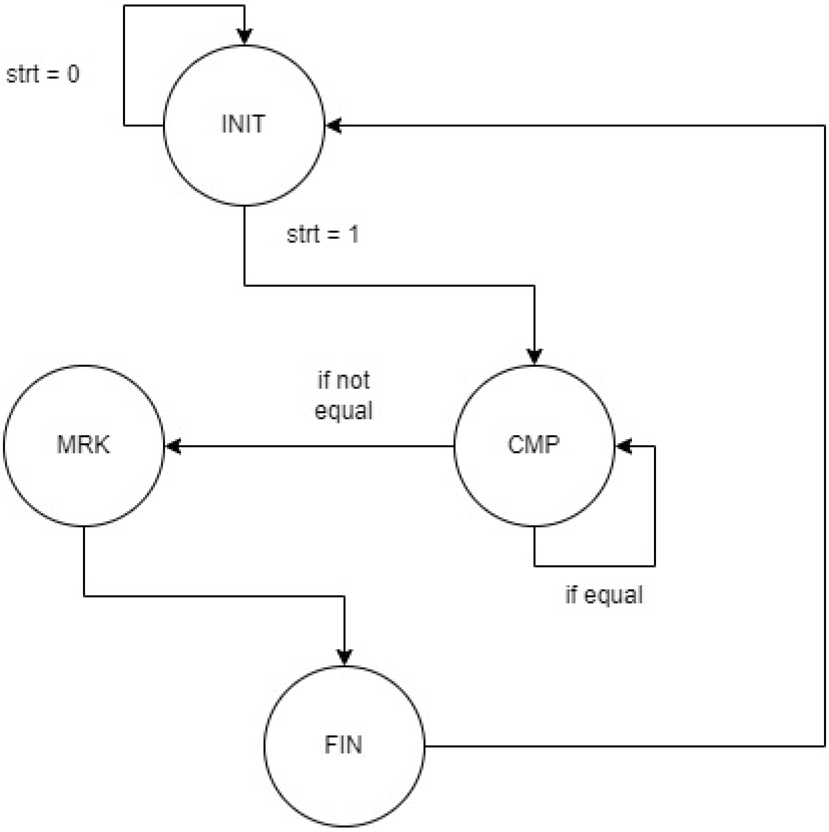


Fig 1. Encoder Schematic

The decoder also has 4 states; INT, CNT, RST and FIN. In the INIT state, the sequence receives a 16-bit input that contains an 8-bit initial sequence which will be copied as many times as the frequency within the last 7 digits. Once the decoder changes states to CNT, the sequence will be copied once and every time said sequence is copied, the frequency will be decremented. If the frequency reaches 0, the circuit will enter the FIN state. In the FIN state, the circuit continues the decompression result to be read by the computer, then the circuit will enter the RST state where it resets the frequency and input buffer.

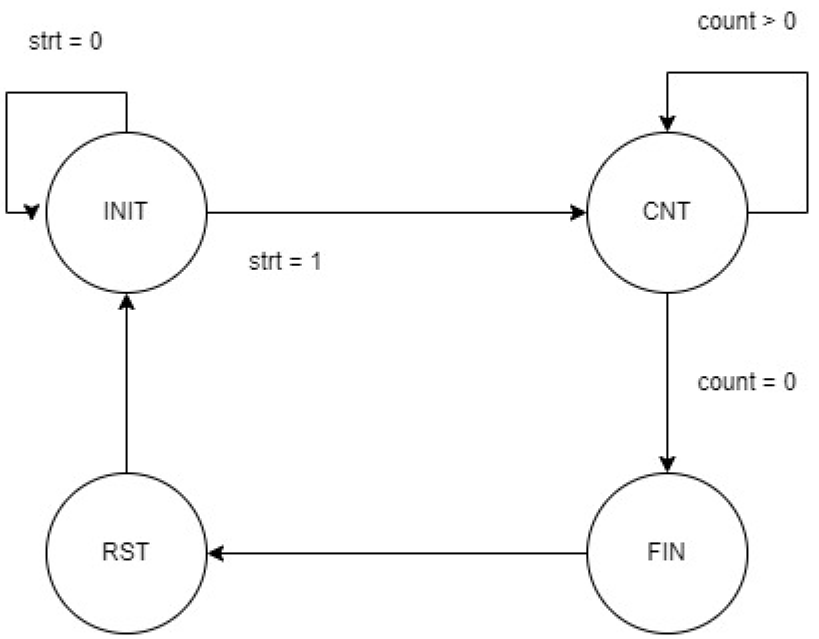


Fig 2. Decoder Schematic

CHAPTER 3

TESTING AND ANALYSIS

3.1 TESTING

For this project, we implemented an encoder and a decoder to evaluate the result of the circuit we had created. We designed two types of testbenches to generate the necessary test inputs and assure that the outputs it provided matched with the results we were expecting. The test inputs for both testbenches were made as vectors, pre-determined before hand to provide controlled conditions and produce the desired results. The encoder used 8-bit inputs and produced 16-bit outputs, whereas the decoder testbench used 16-bit inputs and produced 8-bit outputs.

3.2 RESULT

The encoder testbench operates by assigning values to 8-bit inputs which are repeated multiple times. When the sequence detects a change within the input values it concatenates the input with the number of repetitions of the input. This allows the testbench to evaluate the beahviour of the sequence under different input patterns.

For the first testcase (encoder), the sequence input appears 24 times in a row, which means the size of input would be 8 x 24 bits equaling in a large size of 192 bits. After encoding the sequence using the encoder testbench, the input changes to the size of 2 bytes or 16 bits which results in a significant reduction in data size from before and after encoding.

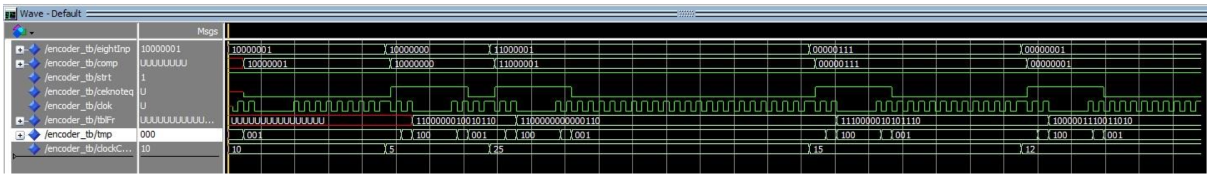


Image 2. Wave Simulation for Encoder

For the second testcase (decoder), the input consists of 16 bits that are divided into the original sequence bits and their frequency. When the circuit receives said input, it copies the original sequence of the specified number of times that input had been repeated to reconstruct the compressed data. This allows the decoder to recover the original data from the compressed representation that had been produced by the encoder.

The results that we found from the simulation is according to what we had wanted since the decoder had successfully returned all the compressed data back into its original form.

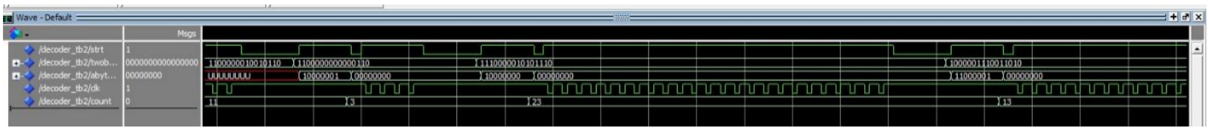


Image 3. Wave Simulation for Decoder

3.3 ANALYSIS

The main circuit has 4 states where it will go through each state and be able to decompress data bits according to the sequence it presents itself in. Within the INIT state, the count will be set to 1 whenever a new data bit is sent into the input. An input of 8 bits will be assigned momentarily into the memory. Then it moves to a new state, CMP, where the new bit of data will be compared with the input within the buffer and compared. If the second bit combination matches with the input within the buffer, the count variable will be incremented. The circuit will then detect whether a repetition occurs once or more within the MRK state. If it only occurs once, the tag value will be 0, and if it does occur more than once, the tag value will be 1. The tag is used to prevent wasting 8 bits of space if the sequence frequency only occurs once. The circuit will then finally enter the FIN state where the tag (1 bit), the original sequence (8 bits), and the frequency (7 bits) are concatenated to form a 16-bit output that will be transmitted towards the internet.

CHAPTER 4

CONCLUSION

The encoder and decoder circuit functions as desired and meets the goals of its creation. There are some improvements that can be made for this circuit including increasing the flexibility of the decoder input length to accommodate graphic based data like photos or videos, implementing parallelism to reduce the time needed to process data, as well as other improvements.

REFERENCES

[1] Prasetyo, H. (2022, November 13). Algoritma run length encoding compression. Hendro Prasetyo. Retrieved December 10, 2022, from https://hendroprasetyo.com/memahami-algoritma-run-length-encoding-compressi/

[2] Charles H. Roth, L.K. John, Digital System Design Using VHDL, 2nd edition. Austin,TX: Thomson, 2008

[3] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 9: Finite State Machine. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

[4] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 8: Looping Constuct. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

[5] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 6: Testbench. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

[6] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 5: Structural Circuit Design. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

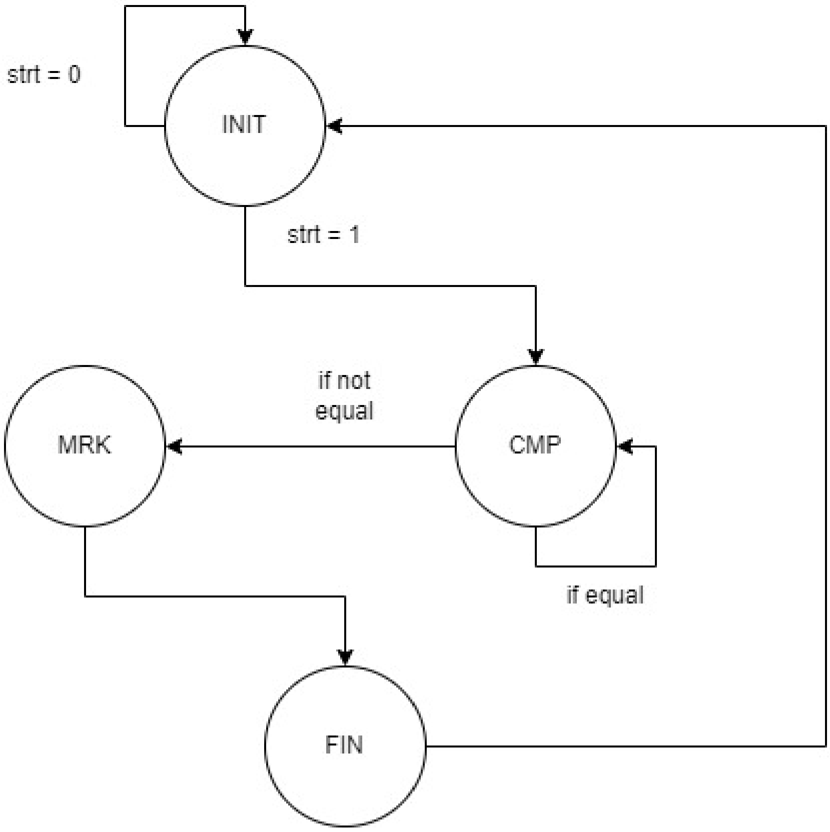
[7] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 4: Behavioral Circuit Design. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

[8] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 3: Concurrent Circuit Design. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

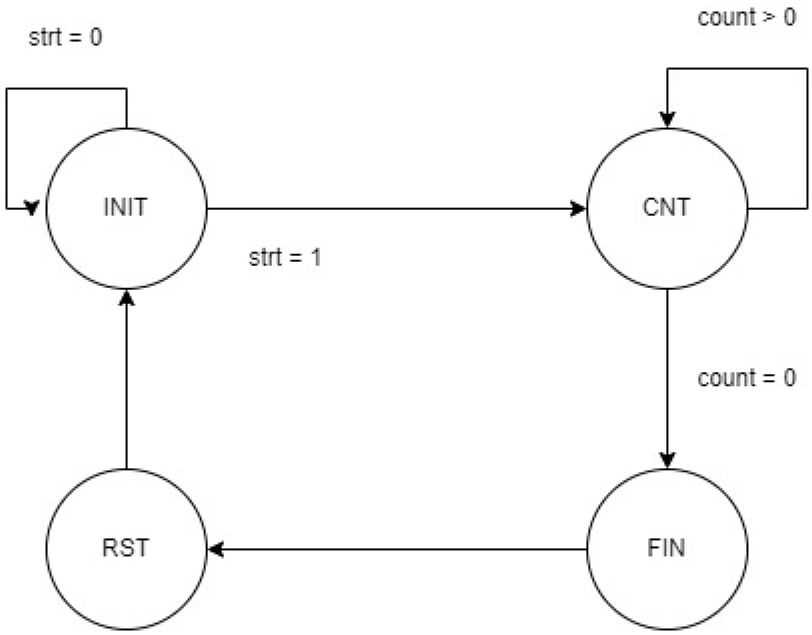
[9] Universitas Indonesia. (2021). Perancangan Sistem Digital Modul 2: VHDL Introduction. [PDF File]. Available: https://emas.ui.ac.id/mod/resource/view.php?id=962413

APPENDICES

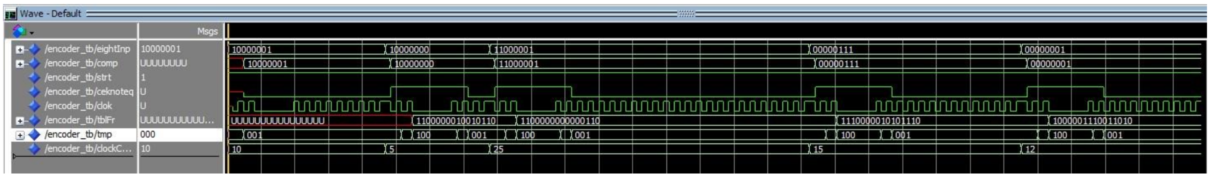
Appendix A: Project Schematic



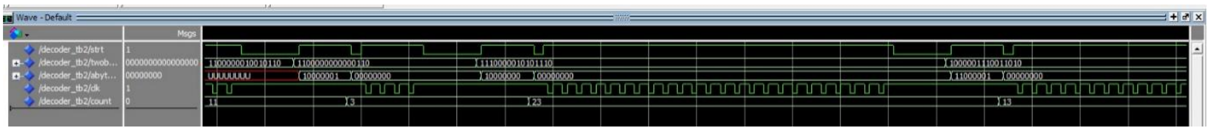
Encoder FSM



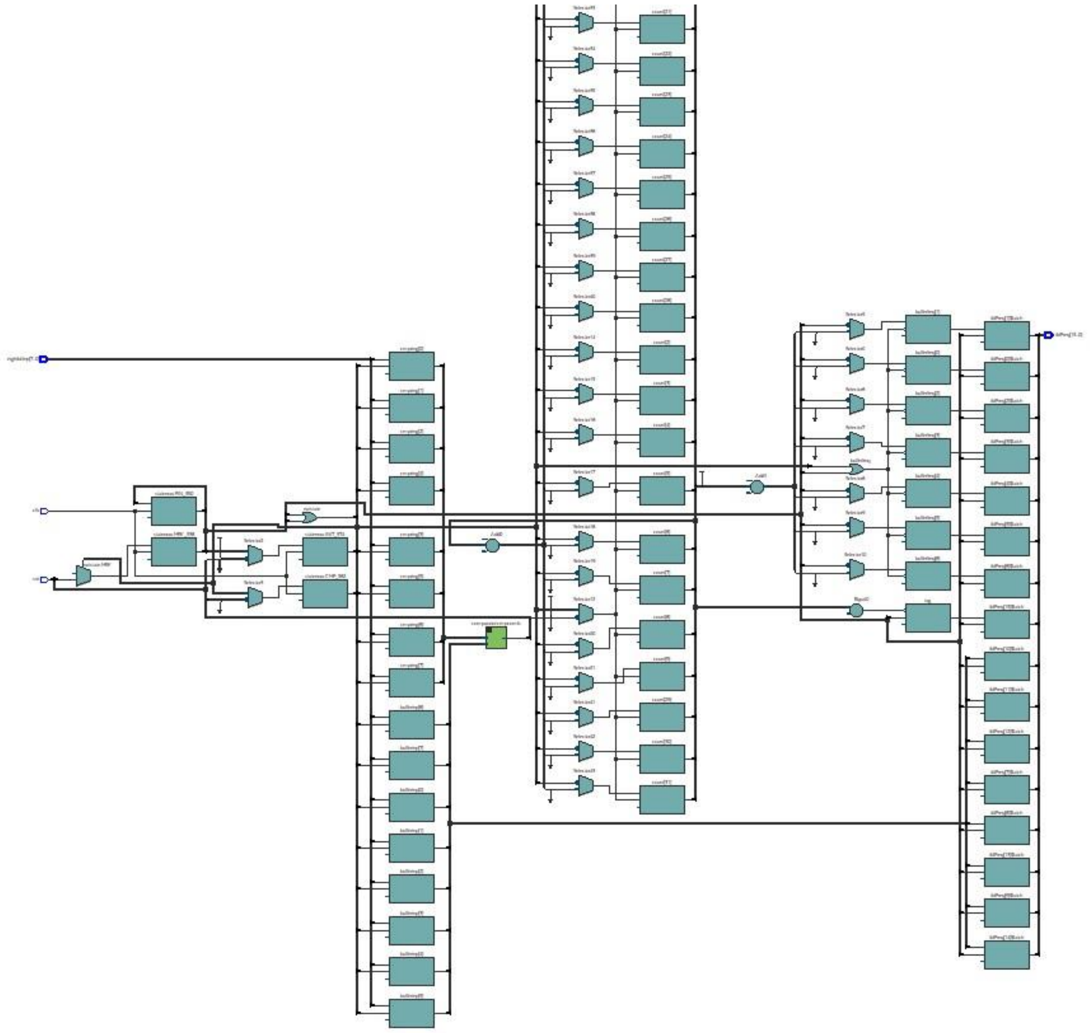
Decoder FSM



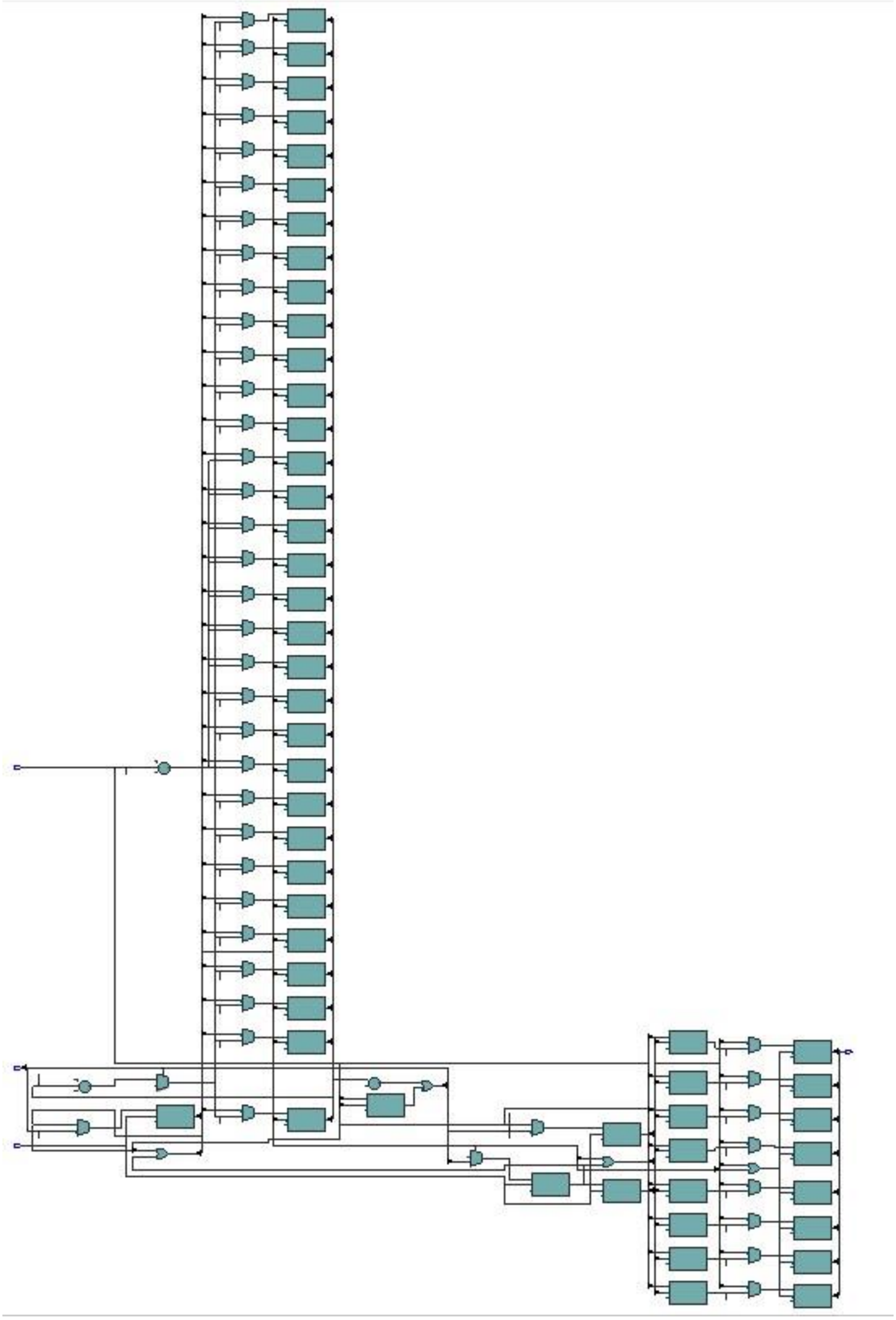
Wave simulation for Encoder



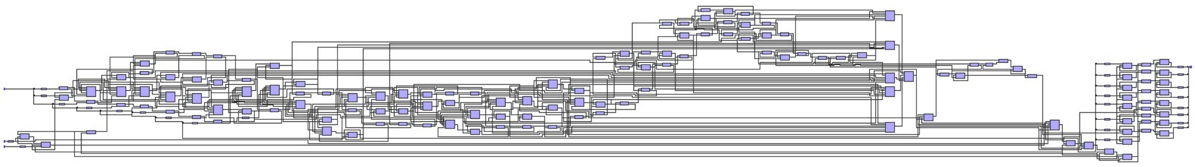
Wave simulation for Decoder



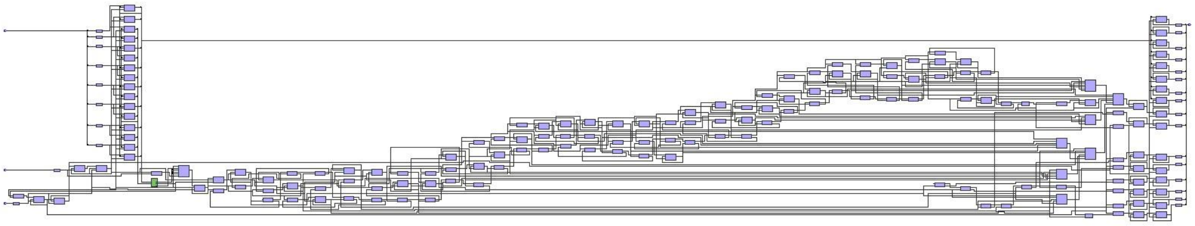
Actual Encoder circuit simulated within Quartus



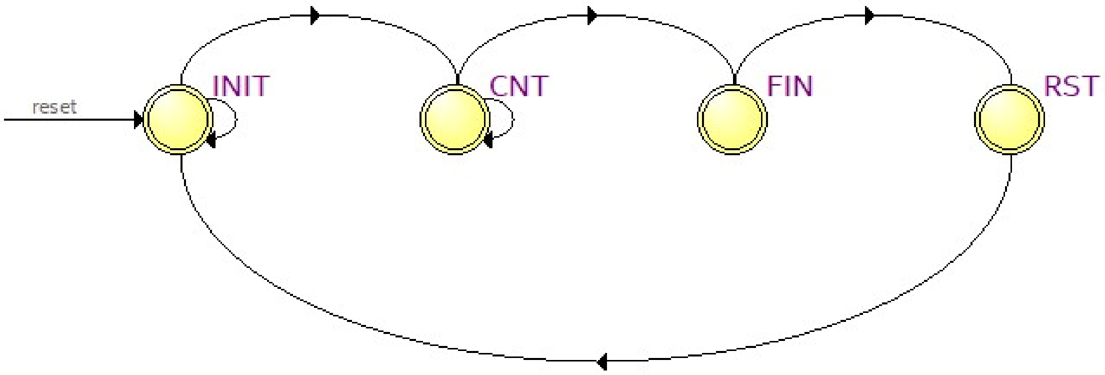
Actual Decoder circuit simulated within Quartus (RTL Viewer)



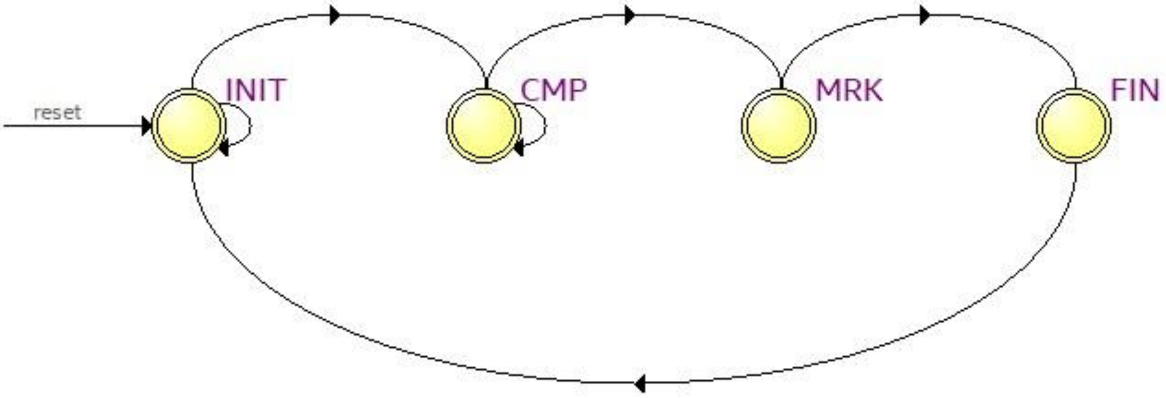
Actual Decoder circuit simulated within Quartus (Technology Map Viewer)



Actual Encoder circuit simulated within Quartus (Technology Map Viewer)

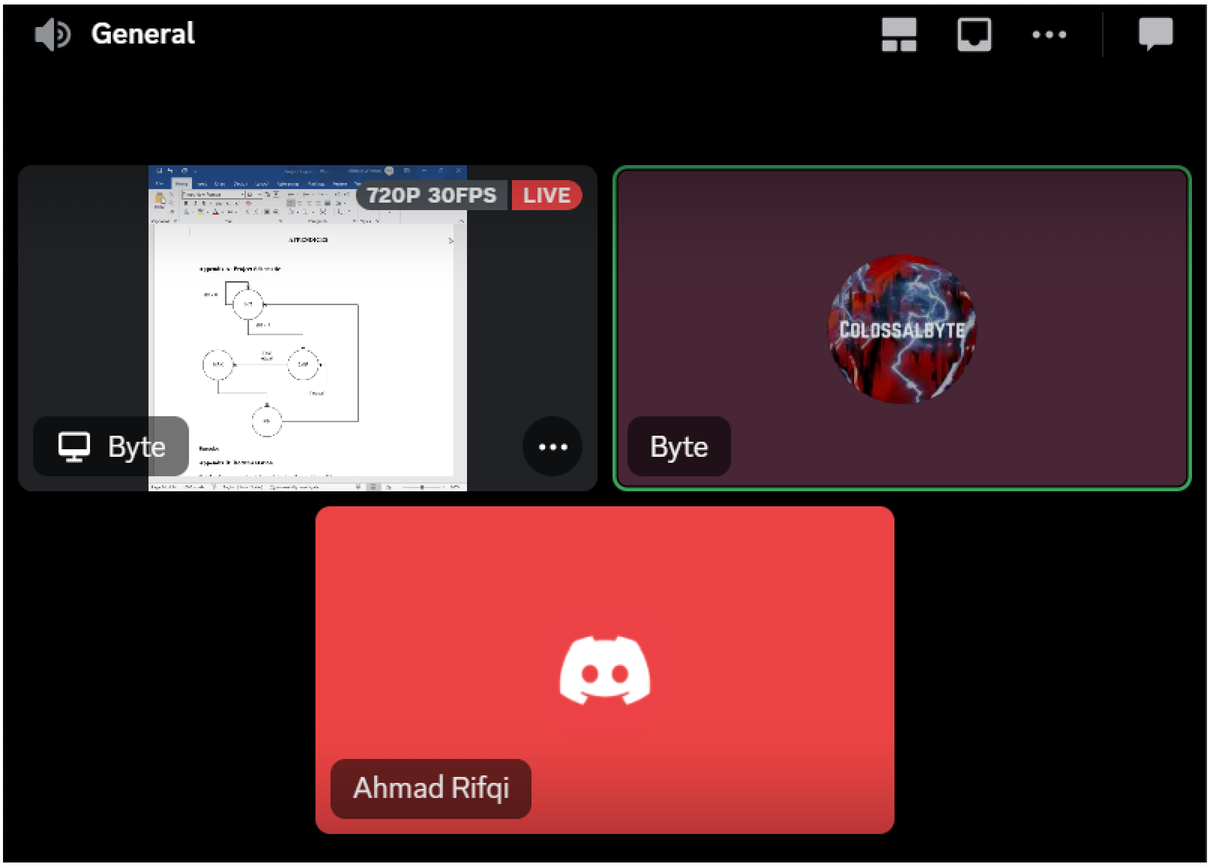


State Diagram for Decoder circuit simulated within Quartus



State Diagram for Encoder circuit simulated within Quartus

Appendix B: Documentation



Working within the discord call for the final report

7:40 PM kio 5x muncul 12 kali?

Michael Winston

Sehrsny iya 7:41 PM

Alif Ma'sum

ahmad_rifqi

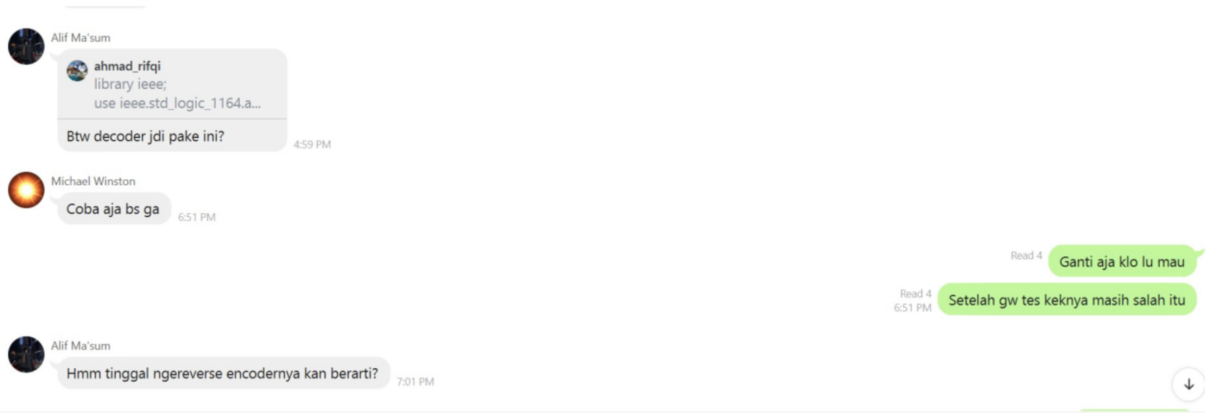
yg 10000001 berarti keluar sebanyak 32 kali?

32 kali apanya? 7:44 PM

Read 4 frekuensi nya

Read 4 7:44 PM kan formatnya tag(1 bit), bit string(8 bit), frekuensi(7 bit)

Discussions within the LINE Group (1)



Alif Ma'sum

ahmad_rifqi
library ieee;
use ieee.std_logic_1164.a...

Btw decoder jdi pake ini? 4:59 PM

Michael Winston

Coba aja bs ga 6:51 PM

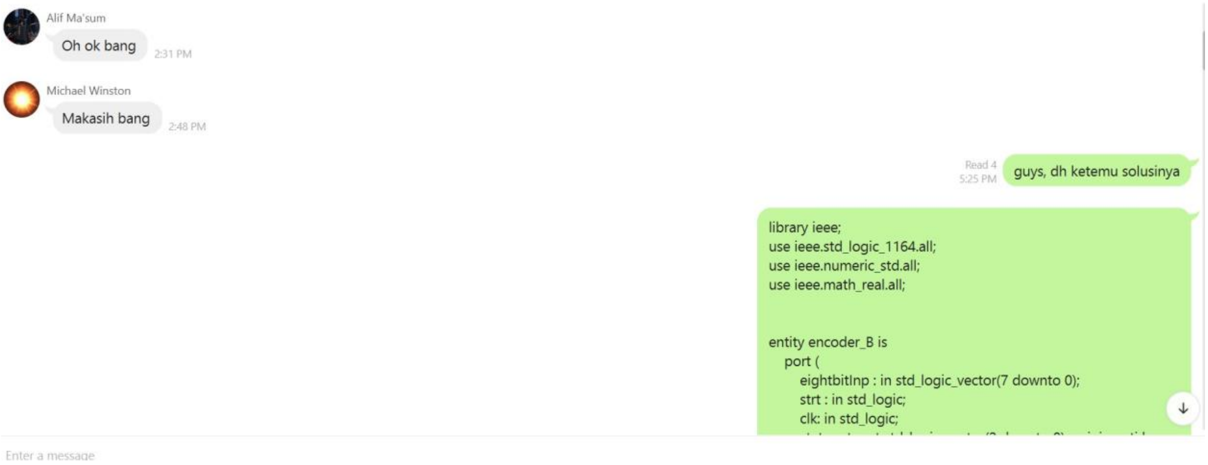
Read 4 Ganti aja klo lu mau

Read 4
6:51 PM Setelah gw tes keknya masih salah itu

Alif Ma'sum

Hmm tinggal ngereverse encodernya kan berarti? 7:01 PM

Discussions within the LINE Group (2)



Alif Ma'sum

Oh ok bang 2:31 PM

Michael Winston

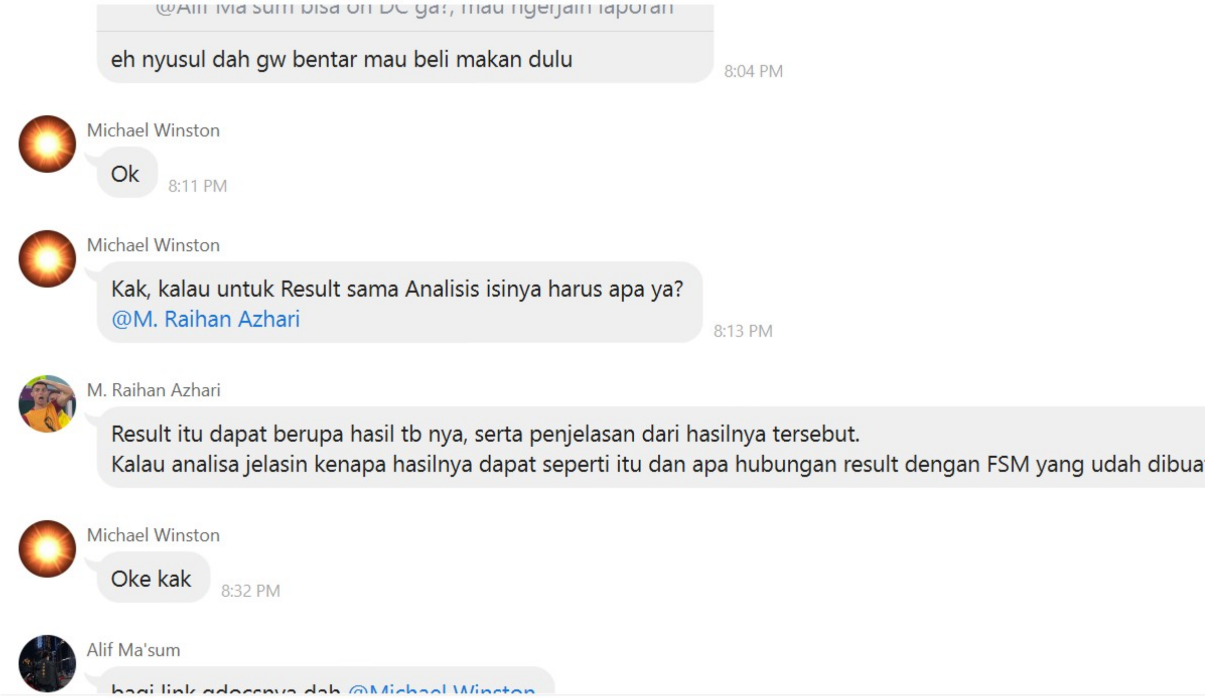
Makasih bang 2:48 PM

Read 4
5:25 PM guys, dh ketemu solusinya

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.all;


entity encoder_B is
   port (
      eightbitInp : in std_logic_vector(7 downto 0);
      strt : in std_logic;
      clk: in std_logic;
```

Discussions within the LINE Group (3)



@Alif Ma'sum bisa on DC ga?, mau ngerjain laporan

eh nyusul dah gw bentar mau beli makan dulu 8:04 PM

Michael Winston

Ok 8:11 PM

Michael Winston

Kak, kalau untuk Result sama Analisis isinya harus apa ya?
@M. Raihan Azhari 8:13 PM

M. Raihan Azhari

Result itu dapat berupa hasil tb nya, serta penjelasan dari hasilnya tersebut.
Kalau analisa jelasin kenapa hasilnya dapat seperti itu dan apa hubungan result dengan FSM yang udah dibua

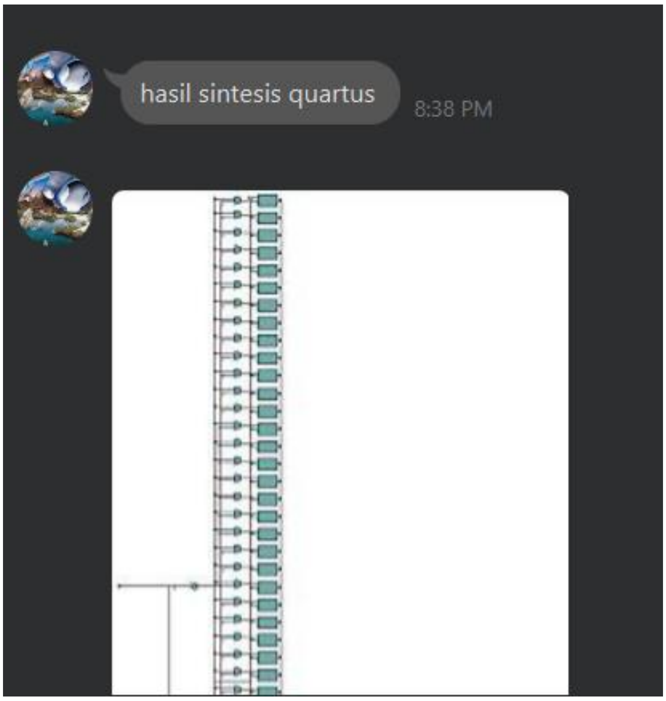
Michael Winston

Oke kak 8:32 PM

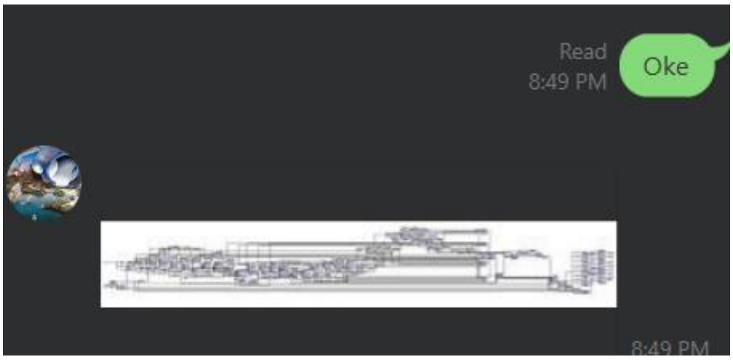
Alif Ma'sum

bagi link gdocsnya dah @Michael Winston

Discussions within the LINE Group (4)



hasil sintesis quartus 8:38 PM



Read
8:49 PM

Oke

8:49 PM

Discussions for Quartus Prime model creation