UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECR1013 – DIGITAL LOGIC

SEMESTER I SESSION 2024/2025

Project Digital Logic

Network Packet Transmission Monitoring System

Lecturer: Dr. Farkhana Muchtar

Section: 03

NAME	MATRIC NO
ANGELA NGU XIN YI	A24CS0226
TAN XIN TIAN	A24CS0198
NIVEETHITA A/P PANDIA RAJAN	A24CS0148
NUR ALIA ATHIRAH BINTI SUZUDDIN	A24CS0153

Dedication & Acknowledgement

This project stands as a reflection of the collaboration, dedication, and determination shared by our group members. We dedicate it to the pursuit of learning, the spirit of innovation, and the value of teamwork that have guided us throughout this journey. We also dedicate this work to our families and friends, whose unwavering support, patience, and encouragement have helped us overcome challenges along the way.

First and foremost, we would like to extend our heartfelt gratitude to our esteemed lecturer, Dr. Farkhana Muchtar, for her invaluable guidance in shaping our understanding of Digital Logic systems and their practical applications. Her unwavering support, insightful feedback, and constructive suggestions motivated us to think creatively and address challenges effectively. Her expertise and encouragement have been instrumental in helping us explore new ideas and refine our problem-solving approach.

We also wish to express our sincere appreciation to the Faculty of Computing for fostering an excellent learning environment, providing essential resources, and giving us the opportunity to work on a project that connects theoretical concepts with real-world implementation. The faculty's structured guidance and emphasis on creativity have greatly enriched our learning experience.

Furthermore, we are deeply grateful to our team members for their commitment and dedication. Each member has contributed unique ideas and efforts, ensuring the success of this project through collaboration and mutual support.

Additionally, we would like to thank our families and friends for their patience, understanding, and encouragement throughout this project. Their support has been a constant reminder of the importance of perseverance and teamwork.

Lastly, we would like to extend our thanks to everyone who, directly or indirectly, contributed to the success of this project. Your support and efforts have been crucial in helping us complete this work successfully.

CONTENTS

1.0 Introduction

Digital Logic is one of the most important branches of the Electronic and Telecommunication Science sector. Digital logic is mainly used for data representation, manipulation and processing of using discrete signals or binary digits (bits). It can perform logical operations, data retrieval or storing and data transformation by analyzing logical circuit design. This project focuses on the practical application of Digital Logic concepts by designing and simulating a Network Packet Transmission Monitoring System. The aim of the project is to transmit a specified number of data packets from one source computer in Lab A to a destination computer in Lab B through a single cable connection or vice versa. The user can define the number of packet data and it includes features such as monitoring, control and fault detection.

This project aims to address real-world challenges by bridging theoretical knowledge with practical implementation. The task involves developing a simplified system architecture using components such as multiplexers (MUX), demultiplexers (DEMUX), counters, comparators, logic gates, and various Digital Logic elements.

A structured methodology will guide the design, simulation, and presentation processes. The project begins with a thorough analysis of the case study to define the key requirements. Based on this understanding, the fundamental functionalities of the system will be implemented, with the option to integrate advanced features like password protection. Additionally, a user-friendly dashboard will be created to monitor and display the system's status, enabling users to clearly understand the packet transmission process.

In this project, students develop different skills in teamwork, creativity and problem-solving by gaining practical experience with circuit simulation tools such as DEEDS. The project not only focuses on the theoretical concepts of Digital Logic but it also focuses on innovation by encouraging students to design the solutions based on the given scenario.

The project report will document each phase, from identifying the problem context to designing and implementing the solution. It will detail the system requirements, design process, and implementation. Finally, the report will conclude by summarizing the project, reflecting on accomplishments, identifying strengths and weaknesses, and proposing improvements for future iterations.

2.0 The Problem Background

As technology advances rapidly, it is important to have efficient systems for sending and monitoring data to ensure smooth communication between devices. Modern networks need clear methods to manage and track data transfer when multiple computers are connected. Problems like data loss, errors, and inefficiencies can affect performance.

This project focuses on connecting computers in Lab A to computers in Lab B using a single cable. The goal is to send a specific number of data packets, chosen by the user, from one computer in Lab A to a target computer in Lab B. However, not having a system to monitor and control the process causes several problems.

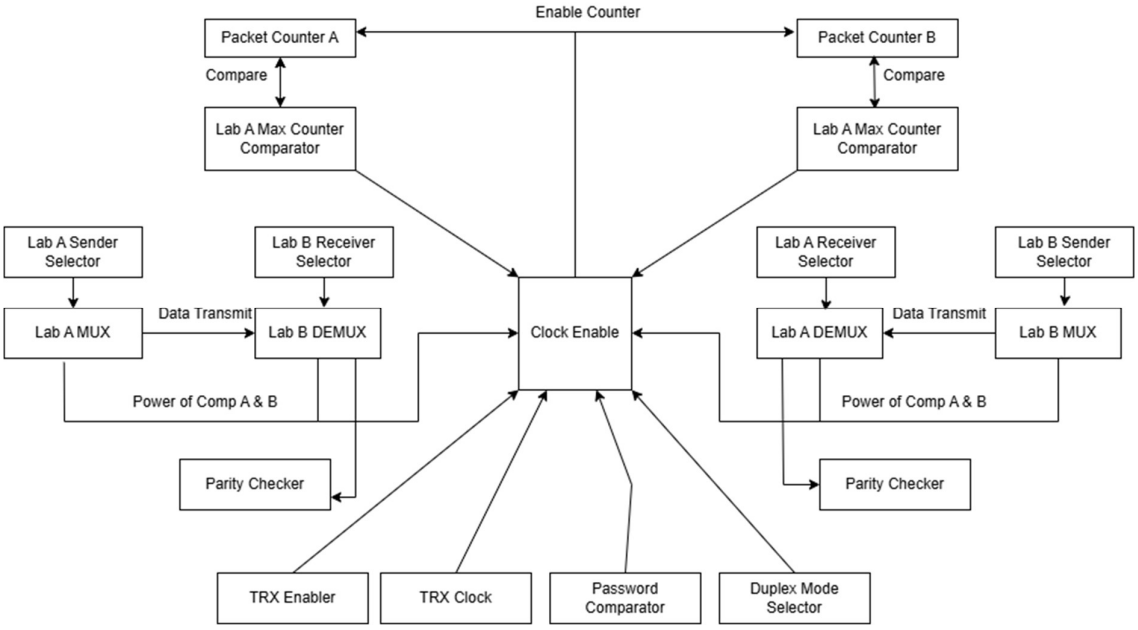
1. **Efficient Circuit Design :**
   - We need to create a circuit that incorporates key components such as counters, comparators, and others, all connected correctly.

2. **Data Packet Transmission Control :**
   - We must ensure accurate data packet transmission while staying within the user-specified maximum limit.

3. **Fault Detection and Monitoring :**
   - We need to detect and display potential system issues, such as failure to start the computer or incorrect passwords.

4. **System Security and Adaptability :**
   - We should improve the transmission system by adding optional features like password authentication and mode selection to enhance security and adaptability.

3.0 Suggested Solution



1. Password Comparator :

The password ensures security by verifying user identity. It serves as a protective measure to make sure only authorized individuals can access and manage data transmission.

- Users are required to enter a 4-digit password correctly before the system begins functioning.
- The system checks the entered password against a preset password to approve or deny access to a particular user.

2. TRX Enabler :

TRX Enable (Transmission Enable) manages the operational status of the system.

- Users can activate or deactivate the entire data transmission system by modifying the TRX Enable state.
- When TRX Enable is set to the disabled state (value = 0), no transmission operations can occur, ensuring that no data is sent or received by any computer. This feature can be used for maintenance or security purposes.

3. TRX Clock :

- The TRX Clock functions as a trigger to initiate the data transmission process.

- It is a pulse-activated input, requiring users to press the button each time to enable the system to transmit a single bit of data.
- This can be implemented as either a physical button or a digital signal within the system interface.

4. Duplex Mode Selector :

- The Duplex Mode Selector determines the specific target destination for the data being transmitted. Data can be sent from Lab A to Lab B, Lab B to Lab A, or in both directions simultaneously (Full Duplex Mode).

5. Max Counter Comparator (Lab A / Lab B) :

- The Max Counter Comparator determines and sets the maximum number of data bits that can be received by Lab A and Lab B.
- Users must enter the maximum number of bits that can be received by Lab A and Lab B before the system begins operating.
- It compares the user-defined maximum bit count with the actual number of bits received by Lab A and Lab B to decide whether the receiving computer can continue to accept additional data bits.

6. Counter (Lab A / Lab B) :

- The counter will track the total number of data bits received by Lab A and Lab B.
- It is a synchronous counter that increments based on the triggered edge of the clock. As a result, it will only become active when the Clock Enabler is activated.

7. Sender Selector & Receiver Selector (Lab A / Lab B) :

- The **Sender Selector** and **Receiver Selector** determine which computers are involved in the data transmission.
- Data can be transmitted from Lab A to Lab B or vice versa. Thus, the user must choose the computer for each lab using the Sender Selector and Receiver Selector to identify the devices that will send and receive the data.
- The **Lab A Receiver Selector** selects the computer that will receive data sent from Lab B.
- The **Lab B Sender Selector** selects the computer that will send data to Lab A.
- The **Lab A Sender Selector** selects the computer that will send data to Lab B.

- The **Lab B Receiver Selector** selects the computer that will receive data sent from Lab A.
- All four components can be activated simultaneously when the system is in Full Duplex Mode.

8. Clock Enabler :

- The Clock Enabler will activate the clock only when the password is correct, the TRX Enable is active, the Duplex mode is set, the Sender Selector and Receiver Selector are configured and the Power of both Computer in Lab A and Lab B is on, the Max Counter Comparator is assigned and the TRX Clock is triggered by the user.
- It controls and activates the system clock, influencing the timing of data transmission. Once the Clock Enabler is activated, the Counters and MUX units in both Lab A and Lab B will become active and operational.

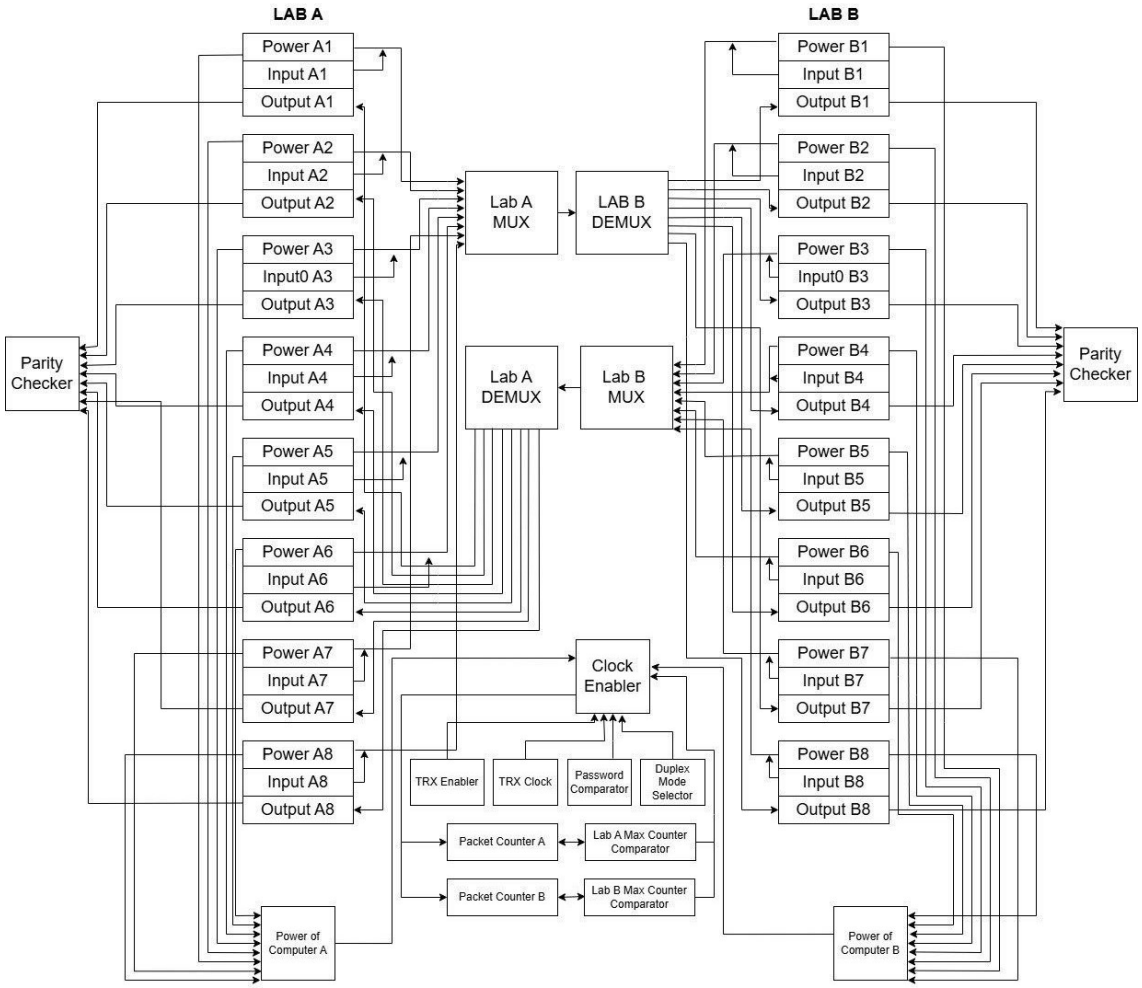
9. MUX & DEMUX (Lab A / Lab B) :

- The MUX (Multiplexer) is a data selector that enables digital information from multiple sources to be sent over a single line to a shared destination.
- The DEMUX (Demultiplexer) is another type of data selector that takes digital information from one line and distributes it to a specific number of output lines.
- When the Clock Enabler is activated, the MUX in one lab can transmit data to the DEMUX in the other lab.
- The MUX determines which computer is authorized to send data based on the choice made in the Sender Selector.
- The DEMUX determines which computer is authorized to receive data based on the choice made in the Receiver Selector.

10. Parity Checker :

- The parity checker is a device used to detect errors in data transmission
- It checks if the number of 1's in a binary data set is even or odd, based on a predefined parity rule (even or odd). If the data does not match the expected parity, the checker will signal an error, indicating that something went wrong during transmission.

3.1 Block Diagram



4.0 The Requirements

The requirements for the Network Packet Transmission Monitoring System are as follows:

1. **Password Verification :**

   Ensures secure system access by requiring users to enter a 4-digit password before operation begins.

2. **TRX Enabler** :

   Manages the system's data transmission status, enabling or disabling transmission.

3. **TRX Clock :**

   Acts as a trigger to start data transmission, requiring users to press a button or send a digital signal to transmit each bit of data.

4. **Duplex Mode Selector :**

   Serves as the trigger for initiating data transmission, requiring users to press a button or send a digital signal to transmit each bit of data.

5. **Maximum Counter Comparator :**

   Compares the user-defined maximum data limit with the actual data received by Lab A and Lab B.

6. **4-bit Counters :**

   A synchronous counter tracks the total bits of data received by Lab A and Lab B, counting based on the clock's triggered edge and only activates when the Clock Enabler is active.

7. **Sender Selector & Receiver Selector (Lab A / Lab B) :**

   Lets users select the sender and receiver computers in Lab A and Lab B for data transmission

8. Clock Enabler :

Manages system's timing and synchronization, enabling the clock only when all required conditions are met

9. MUX and DEMUX :

MUX combines data from multiple sources into a single line, while DEMUX splits data from one line into multiple outputs, allowing efficient data flow between Lab A and Lab B.

10. Parity Checker :

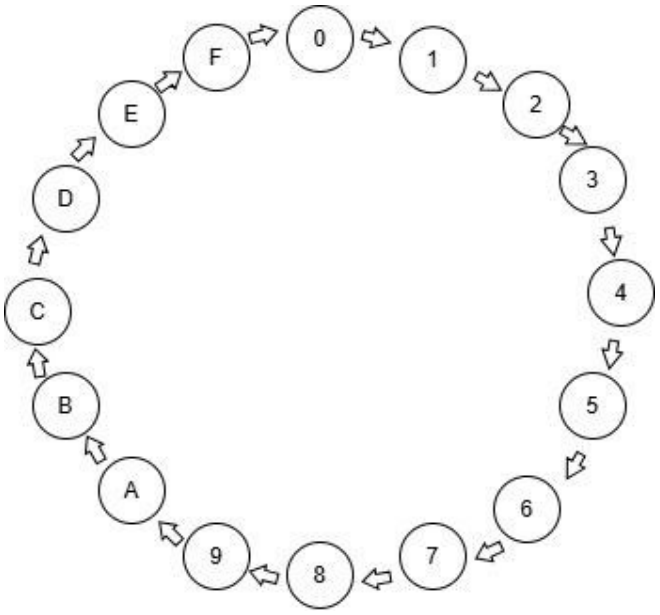
To detect errors in data by verifying if the number of 1's in a binary sequence matches the expected parity which is even parity. If it doesn't, an error is flagged.

Truth Table D Flip-Flop (Not include PR' & CL')

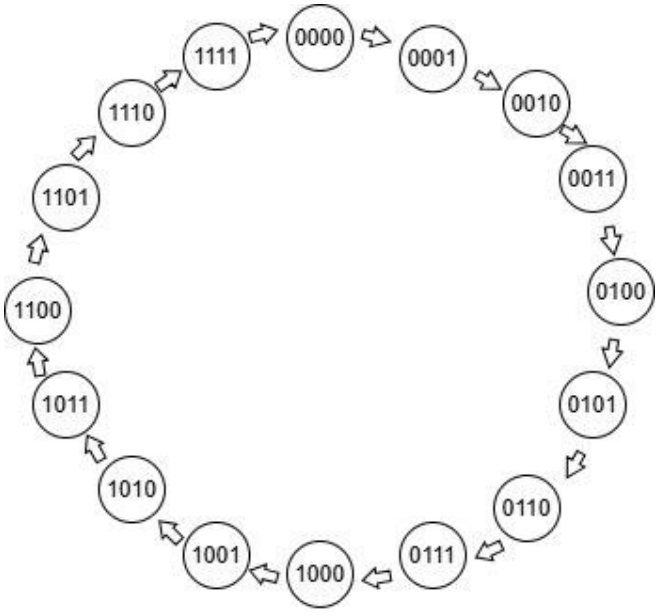
Input		**Output**		**Mode**
Clock input	**D input**			
Ck	**D**	**Q**	**Q'**	
0	X	Q	Q'	No Change
1	1	1	0	Set
1	0	0	1	Reset

Truth Table D Flip-Flop

Input				Output		Mode
Clock Input	Mode Input		D Input			
Ck	PR’	CL’	D	Q	Q’	
0	1	1	X	Q	Q’	No Change
1	1	1	1	1	0	Set
1	1	1	0	0	1	Reset
0	0	1	X	1	0	Set
1	0	1	X	1	0	Set
0	1	0	X	0	1	Reset
1	1	0	X	0	1	Reset
0	0	0	X	Q	Q’	Invalid
1	0	0	X	1	0	Invalid



State Diagram (Hexadecimal Digit)



State Diagram (Binary Digit)

Next State Table

Present State				Next State			
Q3	**Q2**	**Q1**	**Q0**	**Q3+**	**Q2+**	**Q1+**	**Q0+**
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Excitation Table

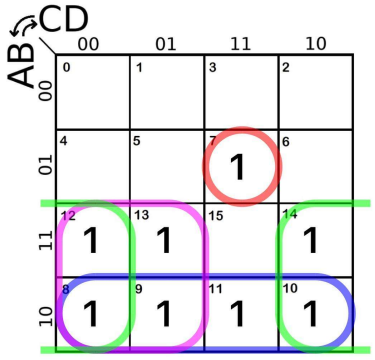
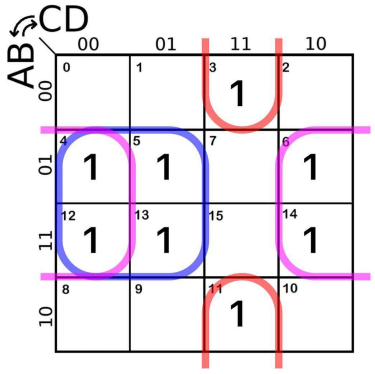
Present State	Next State	Flip-Flop State
Qn	**Qn+1**	**D**
0	0	0
0	1	1
1	0	0
1	1	1

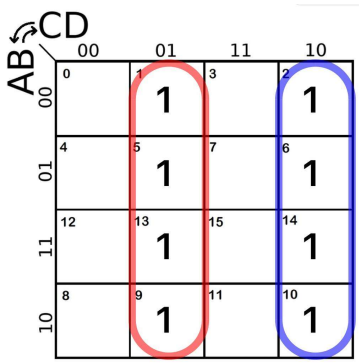
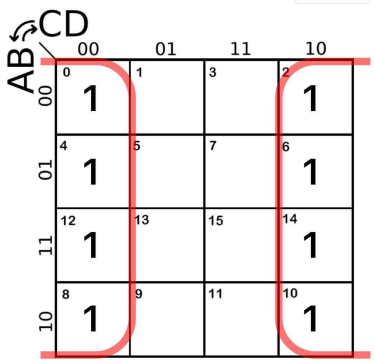
Flip-Flop Transition Table

Present State				Next State				D Transition			
Q3	**Q2**	**Q1**	**Q0**	**Q3+**	**Q2+**	**Q1+**	**Q0+**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1

1	0	1	1	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0

K-Map

Input	K-Map	Logic Equation
D3		D3 = A'BCD + AB' + AC' + AD' D3 = Q3'Q2Q1Q0 + Q3Q2' + Q3Q1' + Q3Q0' = Q3'Q2Q1Q0 + Q3(Q2' + Q1' Q0') = Q3(Q2Q1Q0)' + Q3'(Q2Q1Q0) = Q3 ⊕ Q2Q1Q0
D2		D2 = B'CD + BC' + BD' D2 = Q2'Q1Q0 + Q2Q1' + Q2Q0' = Q2'Q1Q0 + Q2(Q1' + Q0') = Q2 ⊕ Q1Q0

D1		D1 = C'D + CD' D1 = Q1'Q0 + Q1Q0' = Q1 ⊕ Q0
D0		D0 = D' D0 = Q0'

5.0 System Implementation

1. Password Comparator

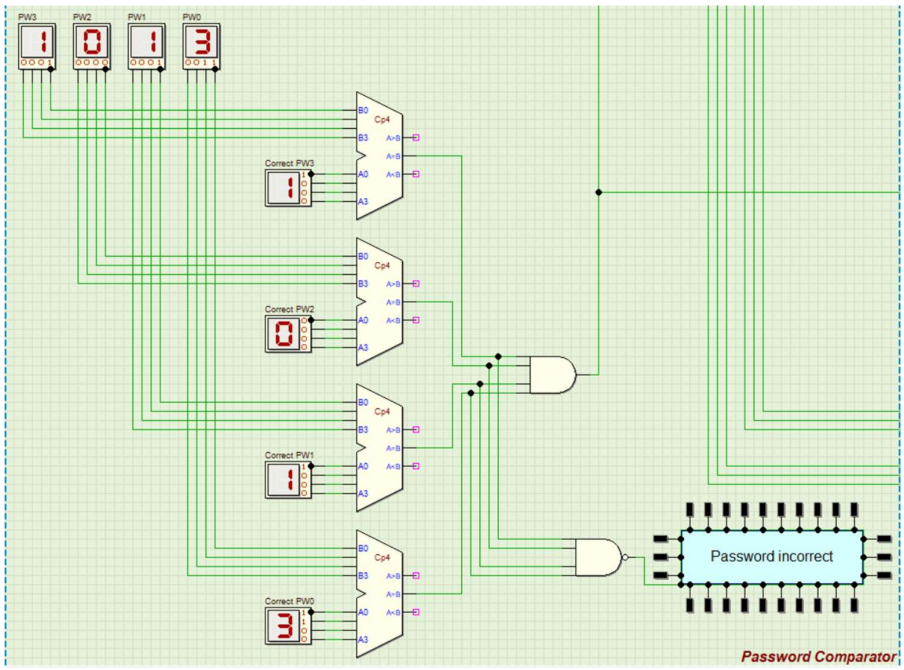


Figure 5.1.1 : Password Correct

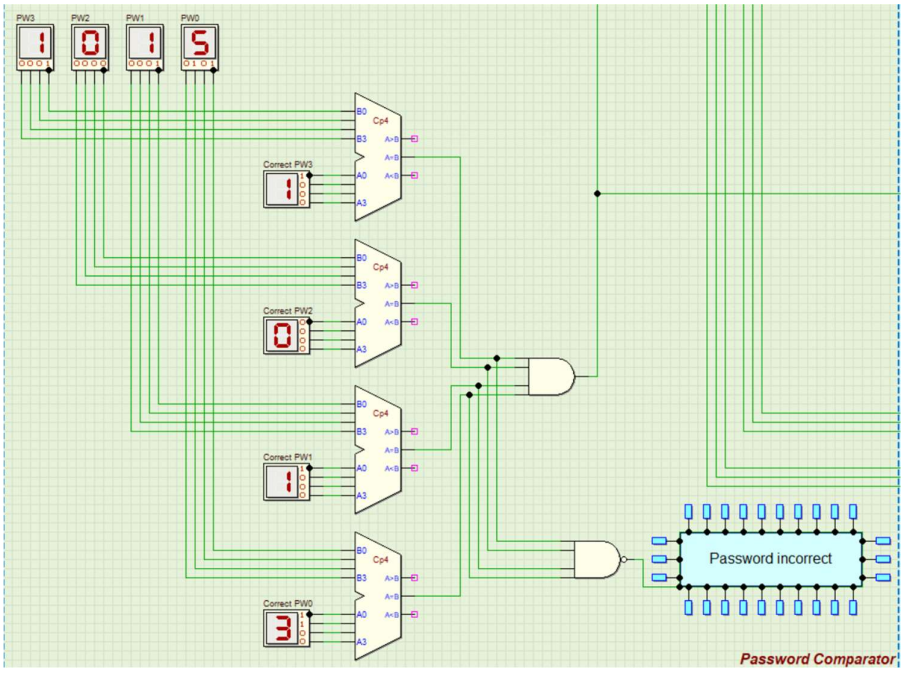


Figure 5.1.2 : Password Incorrect

Inputs:

- The user inputs a 4-digit password (shown as PW0, PW1, PW2, and PW3).

- The predefined password (stored in the system) is displayed as Correct PW0, Correct PW1, Correct PW2, and Correct PW3.

Comparison Process (with 4-bit comparator):

- Each digit of the input password is compared with the corresponding digit of the predefined password using 4-bit comparators.
- For a more detailed example, the comparator will compare the PW3 (A0 until A3) with the Correct PW3 (B0 until B3) and one of the outputs which is the A < B, A = B and A > B will become 1 based on the result.
- Since we just required the result of A = B in our operation, we only connect the wire from the output A = B to the next part.
- A = B will be in a high state if the password is correct. Otherwise, the output will be 0.

AND Gates:

- The outputs of the comparators are combined using AND gates to ensure that all digits match simultaneously.
- If even one comparator output is incorrect, the final AND gate will not activate.

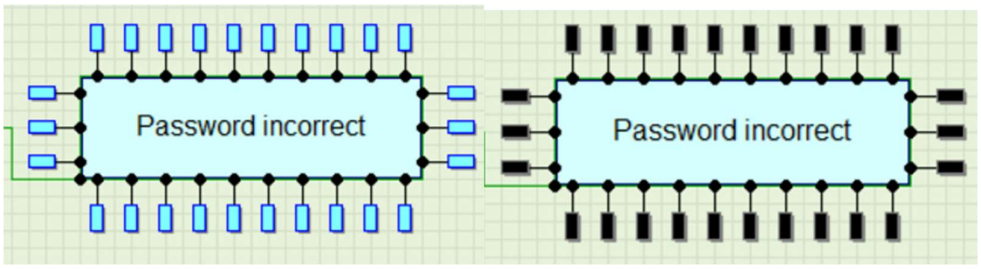


Figure 5.1.3 : Fun Display (Password Incorrect / Password Correct)

Fun Display (with LED test):

- Additionally, the four outputs from the A = B comparison of all digits are connected to a 4-input NAND gate at the bottom.
- The NAND gate produces an output of 1 if not all inputs are 1, indicating that not all password digits match (Password Incorrect). Conversely, if all inputs are 1, the NAND gate outputs 0 (Password Correct).
- The result of the NAND Gate connects to the test LEDs.
    - The test LED will turn black (off) when receiving input 0 which means that the password is not wrong.

- The test LED will turn to blue (on) when receiving input 1 which means that the password is wrong.

2. MUX & DEMUX

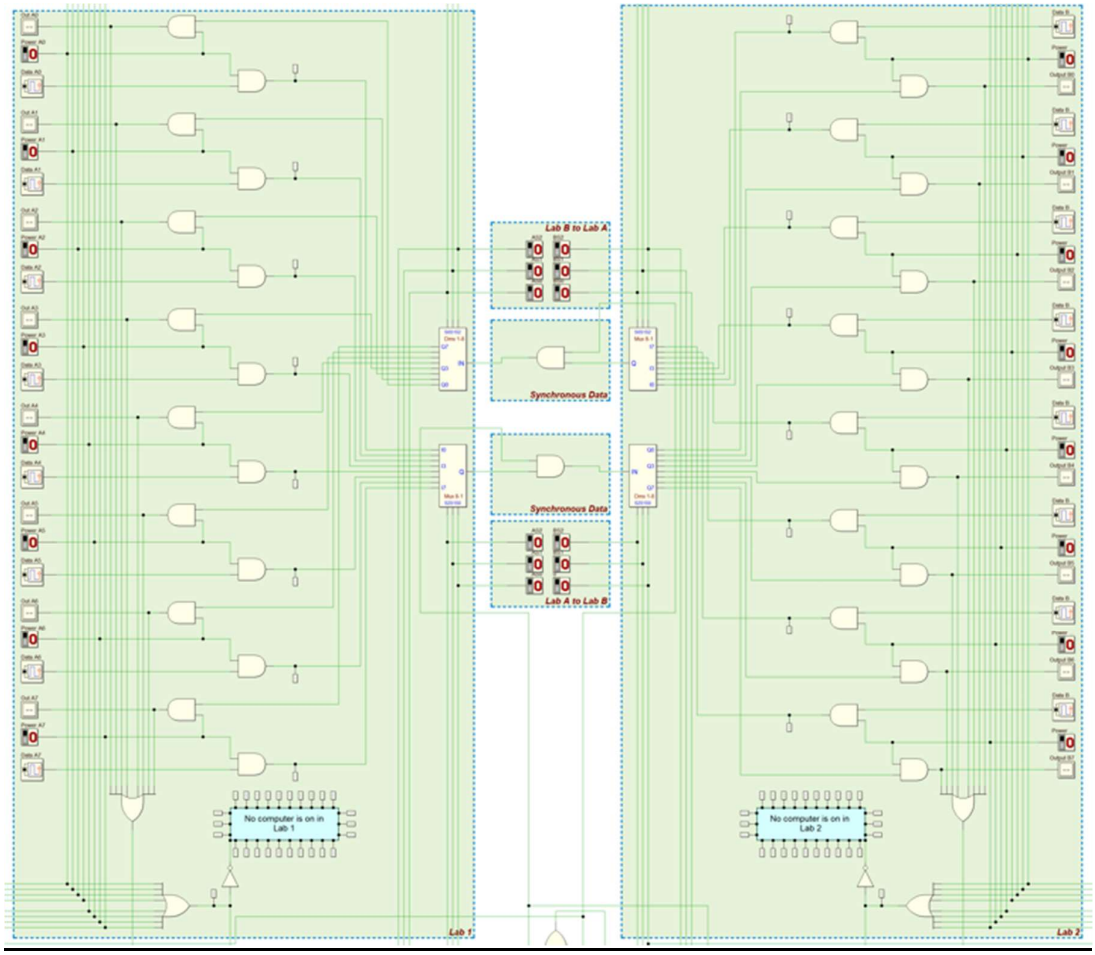


Figure 5.2.1 : Data Transmit From Lab 1 to Lab 2 (vice versa) using MUX & DEMUX

- Each Lab consists 8 computer

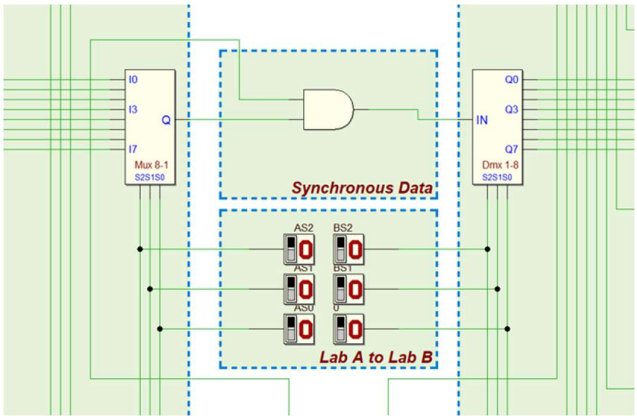


Figure 5.2.2 : Lab A MUX & Lab B DEMUX

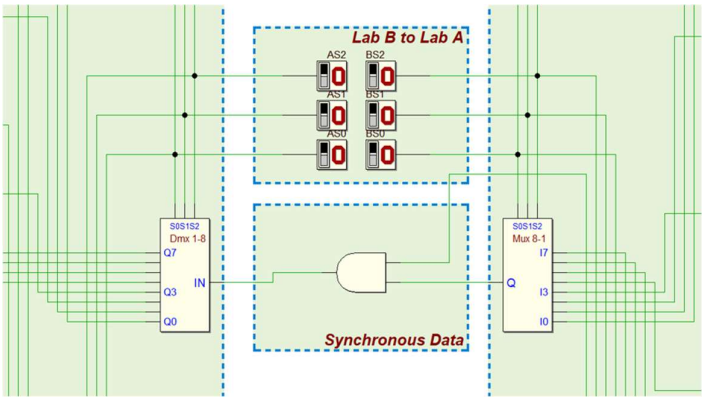


Figure 5.2.3 : Lab B MUX & Lab A DEMUX

- MUX and DEMUX play crucial roles in facilitating data transmission between two devices located in different labs.
- The system includes two MUX and two DEMUX: one MUX and DEMUX for Lab A and one MUX and DEMUX for Lab B.
- This setup enables data transmission not only from Lab A to Lab B but also from Lab B to Lab A.

Input:

Figure 5.2.4 : Power On Button & Clock Data Transmission

- The Lab A MUX section is composed of several smaller components, as illustrated in Figure 5.2.4.
- For instance, **Data A3** represents the clock signal, which serves as the source data from Computer 3 in Lab A.
- **Power A3** refers to the power-on switch for Computer 3 in Lab A. When Power A3 is high, it means Computer 3 in Lab A is powered on, and when it is low, the computer is powered off.
- Both **Data A3** and **Power A3** are connected to a 2-input AND gate on the right side. The AND gate outputs 1 only if data is being transmitted from Computer 3 in Lab A and if the power for Computer 3 in Lab A is on.
- The same principle is applied in the Lab B MUX section.

Output:

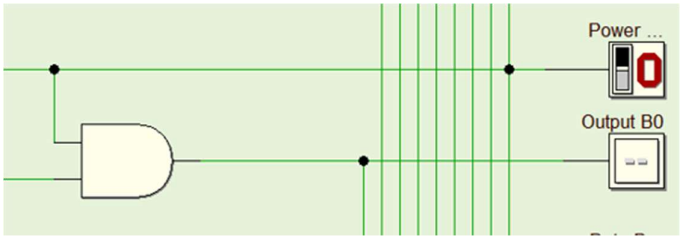


Figure 5.2.5 : Power On Button & Data Output

- The Lab B DEMUX section is made up of multiple smaller components, as shown in Figure 5.2.5.
- **Power B0** refers to the power-on switch for Computer 0 in Lab B. When Power B0 is set to 1, Computer 0 in Lab B is powered on, and when it is set to 0, the computer is powered off.
- Both **Power B0** and the output **Q** from the DEMUX are connected to a 2-input AND gate on the right side. The AND gate produces an output of 1 only if data is sent from the **Q** output of the DEMUX and Computer 0 in Lab B is powered on.
- The same principle applies to the Lab A DEMUX section.

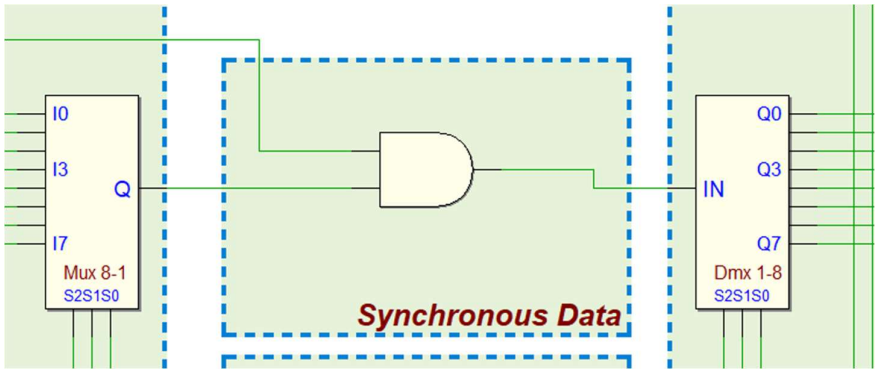


Figure 5.2.6 : (Lab A 8x1 MUX) & (Lab B 1x8 DEMUX)

Lab A MUX:

- The 8-to-1 multiplexer (MUX) at Lab A takes data from the selected computer in Lab A through its input (IN).
- The selectors, **S0 to S2**, decide which input, from **I0 to I7**, will be routed to the output (Q).

Lab B DEMUX:

- The 1-to-8 demultiplexer (DEMUX) at Lab B receives data from the output (Q) of the 8-to-1 MUX at Lab A through its input (IN).
- The selectors, **S0 to S2**, determine which output, from **Q0 to Q7**, will carry the data received at its input (IN).
- The same logic applies to the MUX in Lab B and the DEMUX in Lab A.

Table 5.2.1 : Truth Table MUX

Input											Output
Data Select			Data Input								Data Output
S2	S1	S0	I0	I1	I2	I3	I4	I5	I6	I7	Q
0	0	0	1	X	X	X	X	X	X	X	1
0	0	1	X	1	X	X	X	X	X	X	1
0	1	0	X	X	1	X	X	X	X	X	1
0	1	1	X	X	X	1	X	X	X	X	1
1	0	0	X	X	X	X	1	X	X	X	1

1	0	1	X	X	X	X	X	1	X	X	1
1	1	0	X	X	X	X	X	X	1	X	1
1	1	1	X	X	X	X	X	X	X	1	1

X = Do Not Care

Table 5.2.2 : Corresponding Computer For Data Select & Data Input Lab A

Select Computer			Corresponding Data Input	Corresponding Computer
S2	S1	S0		
0	0	0	I0	1st Computer
0	0	1	I1	2nd Computer
0	1	0	I2	3rd Computer
0	1	1	I3	4th Computer
1	0	0	I4	5th Computer
1	0	1	I5	6th Computer
1	1	0	I6	7th Computer
1	1	1	I7	8th Computer

Table 5.2.3 : Truth Table DEMUX

Input				Output							
Data Select			Data Input	Data Output							
S2	S1	S0	IN	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	1	X	X	X	X	X	X	X
0	0	1	1	X	1	X	X	X	X	X	X
0	1	0	1	X	X	1	X	X	X	X	X
0	1	1	1	X	X	X	1	X	X	X	X
1	0	0	1	X	X	X	X	1	X	X	X
1	0	1	1	X	X	X	X	X	1	X	X
1	1	0	1	X	X	X	X	X	X	1	X
1	1	1	1	X	X	X	X	X	X	X	1

X = Do Not Care

Table 5.2.4 : Corresponding Computer For Data Select & Data Output Lab B

Select Computer			Corresponding Data Output	Corresponding Computer
S2	S1	S0		
0	0	0	Q0	1st Computer
0	0	1	Q1	2nd Computer
0	1	0	Q2	3rd Computer
0	1	1	Q3	4th Computer
1	0	0	Q4	5th Computer
1	0	1	Q5	6th Computer
1	1	0	Q6	7th Computer
1	1	1	Q7	8th Computer

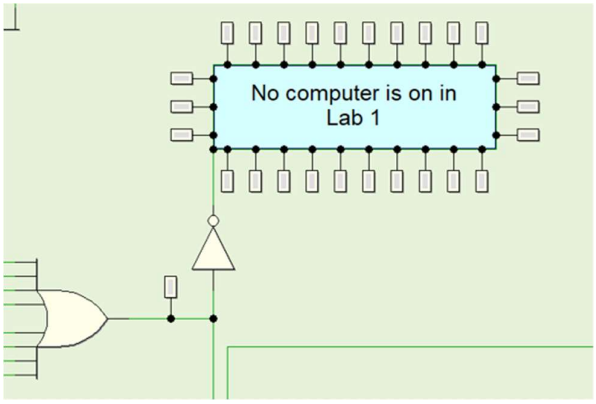


Figure 5.2.7 : Fun Display

- Additionally, all power-on buttons from Lab A are connected to an 8-input OR gate, corresponding to the 8 computers in the lab.
- The OR gate outputs 1 if any of the inputs is 1, indicating that at least one computer in Lab A is powered on.
- If all inputs are 0, the OR gate outputs 0, meaning no computers in Lab A are powered on.
- The output of the OR gate is connected to a NOT gate, which inverts the signal it receives.
- If at least one computer in Lab A is powered on, the OR gate outputs 1, which the NOT gate inverts to 0.

- If no computers in Lab A are powered on, the OR gate outputs 0, which the NOT gate inverts to 1.
- The same principle is applied on the Lab B side.

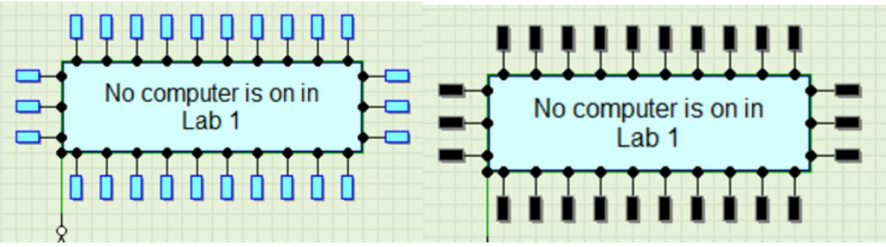


Figure 5.2.8 : Status Test LED

- The NOT gate's output is linked to the test LED.
- The test LEDs remain off (black) when the input is 0, signifying that at least one computer in Lab A is switched on.
- The test LEDs light up (blue) when the input is 1, signifying that no computers in Lab A are switched on.
- The same concept is used on the Lab B side.

3. Sender Selector & Receiver Selector

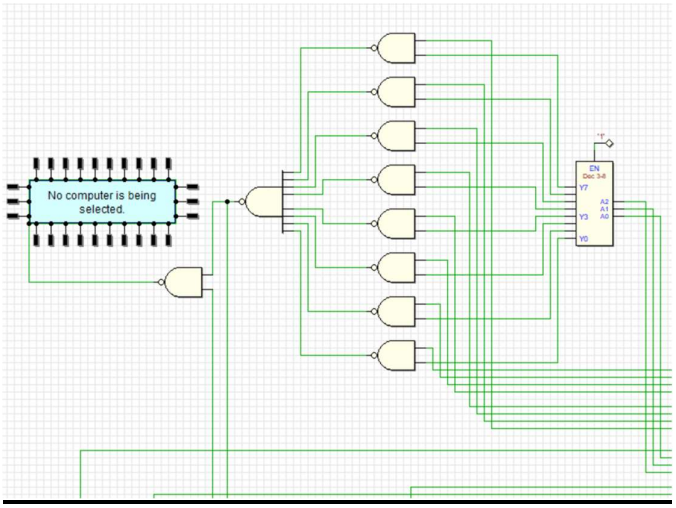


Figure 5.3.1 : Lab A Sender Selector

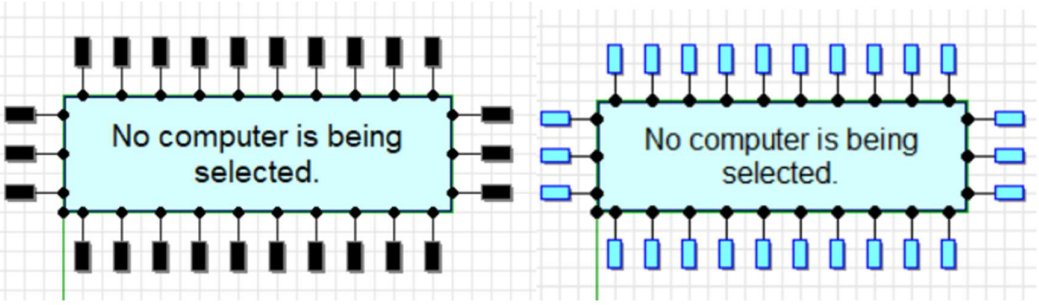


Figure 5.3.2 : Status Test LED

- The NAND gate produces a high output only when the power of the selected computer is switched on and the duplex mode is correctly configured.
- The test LEDs remain off (black) when the input is 0, indicating that at least one computer in Lab A has been selected.
- The test LEDs light up (blue) when the input is 1, signifying that no computers in Lab A are currently selected.
- This same principle is applied to the Lab A Receiver Selector, as well as the Lab B Sender and Receiver Selectors.

4. Max Data Comparator

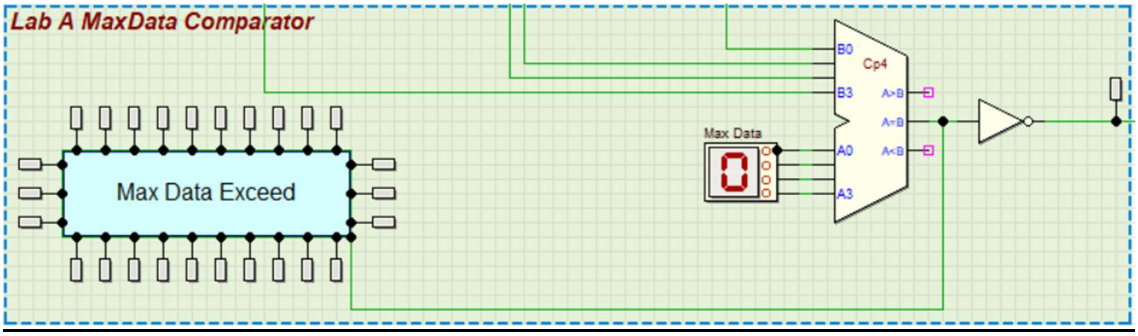


FIGURE 5.4.1 : Lab A Max Data Comparator

- Max Data Comparator is a comparator that compares the actual amount of data bits accepted by a certain computer in Lab A with the predefined maximum data allowed to be accepted.

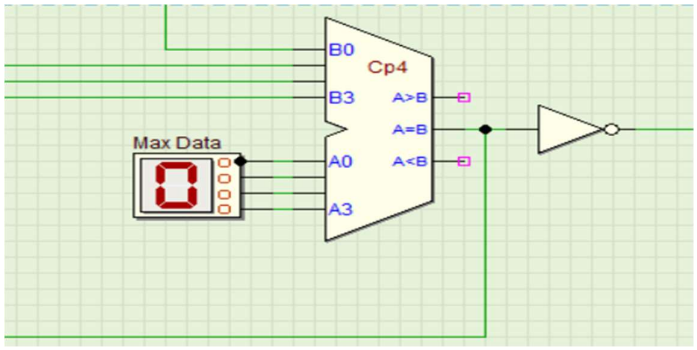


FIGURE 5.4.2 : Predefined Maximum Data Lab A

- The maximum data allowed to be accepted by a computer at Lab A can be defined by the users at the 4-bit hex digit input (Max Data).
- The input of the 4-bit hex digit input (Max Data) is connected to the input of the comparator (A0 until A3). The input B0 to B3 on the other side of the comparator will be connected to the output of the counter that will count the total data received by the computer in Lab A.
- The comparator will compare both data (Actual Received Data & Predefined Max Data) to determine which output of the comparator should output result 1.
- If the actual received data (B0 until B3) reaches the predefined max data, the output at A=B will be 1 and the NOT gate will change the output to 0 so system needs to block the transmission of the data.

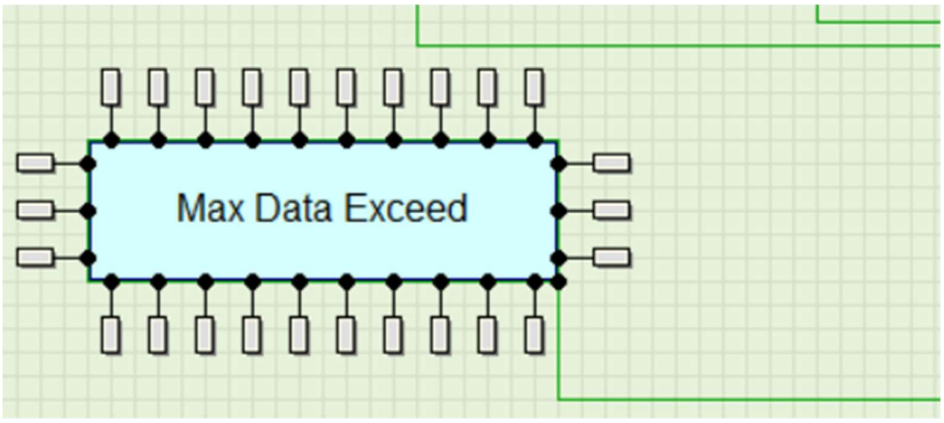


FIGURE 5.4.3 : Fun Display (Max Data Exceed)

- The output of A=B from the comparator will connect to all of the test LEDs above.

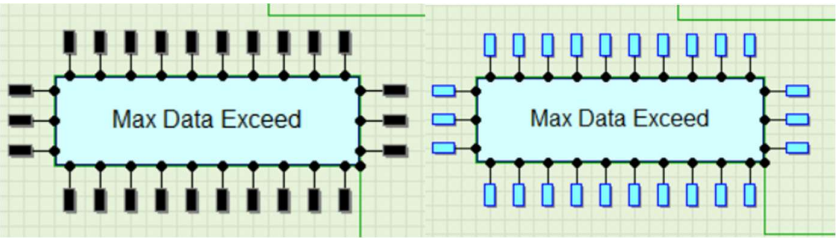


FIGURE 5.4.4 : Status Test LED (Not Max Data Exceed / Max Data Exceed)

- All of the test LEDs will turn black (OFF) when receiving input 0 which means the actual received data does not exceed the predefined max data. (Error not happen)
- All of the test LED will turn blue (ON) when receiving input 1 which means the actual received data exceeds the predefined max data because the users change the value of predefined max data during the process of transmission to a lower value. (Error happen)

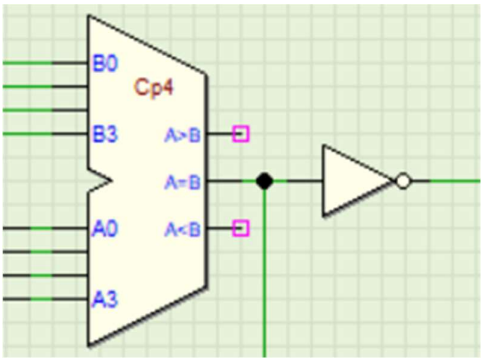


FIGURE 5.4.4 : Clock Enabler Blocker

- The output of A=B from the comparator is connected to a NOT gate. The NOT gate will output the result 1 when all of the received data of computer in Lab A is not equal to the predefined max data of a computer in Lab A. Otherwise, the NOT gate will output the result 0.
- The output of this NOT gate will be used later for the other functions.

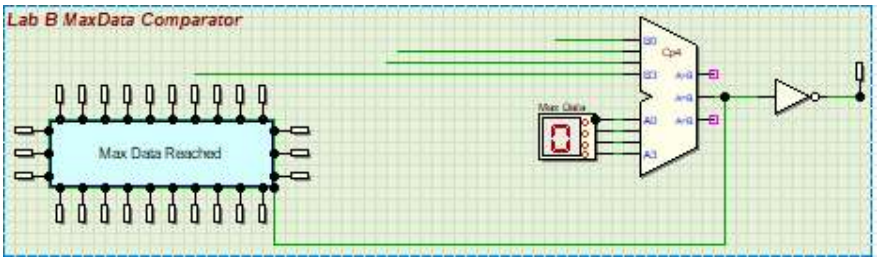


FIGURE 5.4.5 : Lab B Max Data Comparator

- Both circuits of Max Data Comparator at Lab A and Lab B are the same. The only difference between both Max Data Comparator is the input of the actual received data and where the output of the NOT gate at the right side connected to.

5. Clock Enabler

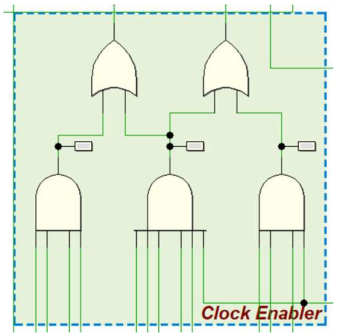


Figure 5.5.1 : Clock Enabler

- There are 3 AND Gates in Clock Enabler, each representing each Duplex Mode: (From left to right)
- First: 4-input AND Gate that allows data from computer in Lab A to transmit to computer in Lab B
- Second: 6-input AND Gate that allows data from computer in Lab A to transmit to computer in Lab B and vice versa (Full Duplex Mode)
- Third: 4-input AND Gate that allows data from computer in Lab B to transmit to computer in Lab A

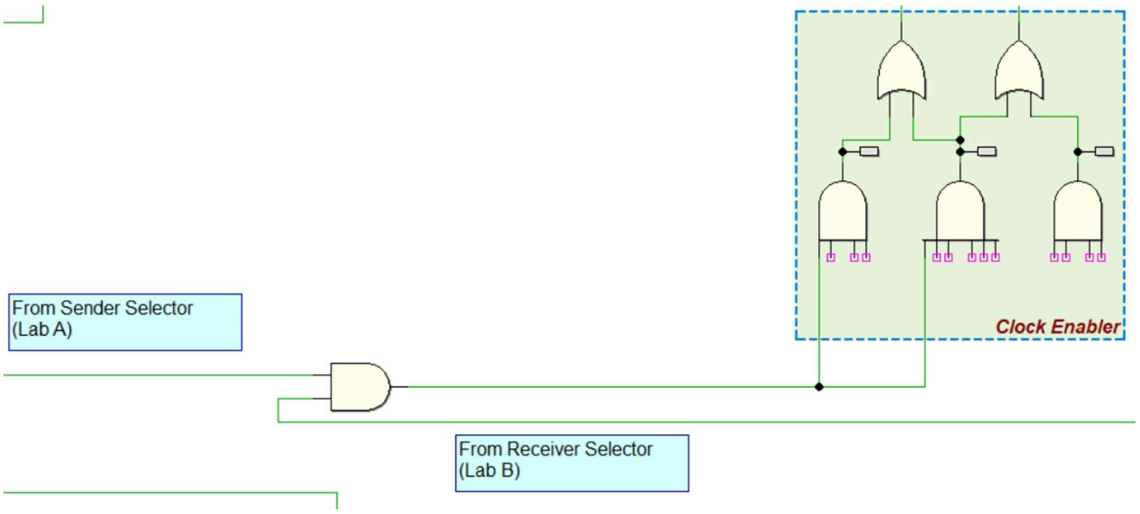


Figure 5.5.2: Sender Selector and Receiver Selector connected to Clock Enabler

- The output of Lab A Sender Selector Checker and output of Lab B Receiver Selector Checker is connected to an AND Gate which is then connected to the First and Second AND Gate of Clock Enabler.
- This is because both first AND gate and second AND gate are involved in data transmission from Lab A to Lab B.
- The AND Gate with sender selector and receiver selector as input will give an output of 1 only if both sender computer and receiver computer is selected (output for sender selector and receiver selector is 1)

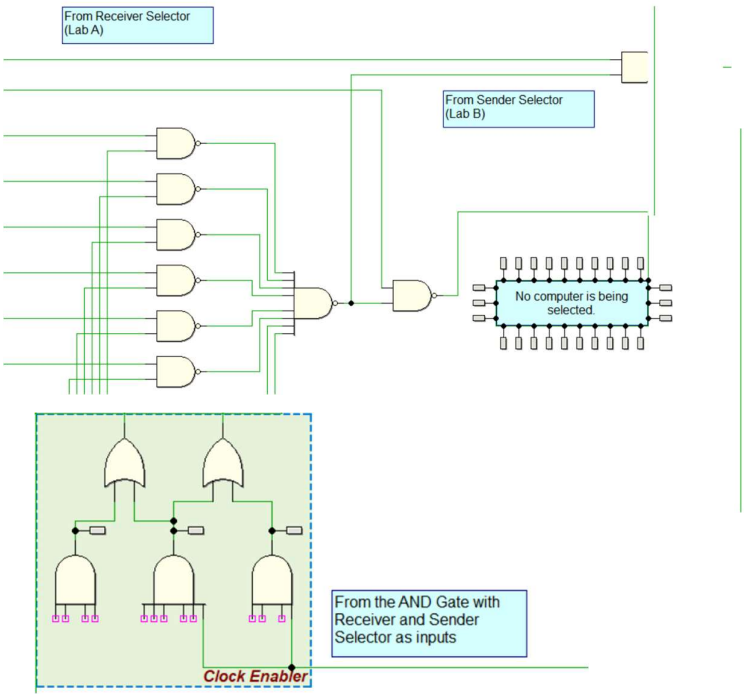


Figure 5.5.3 : Sender Selector (Lab B), Receiver Selector (Lab A), AND Gate to Clock Enabler

- This is the same for Lab B as Sender Selector and Lab A as Receiver Selector.
- The AND Gate connects to the Second and Third AND Gate inside Clock Enabler
- This is because both second AND gate and third AND gate are involved in data transmission from Lab B to Lab A.

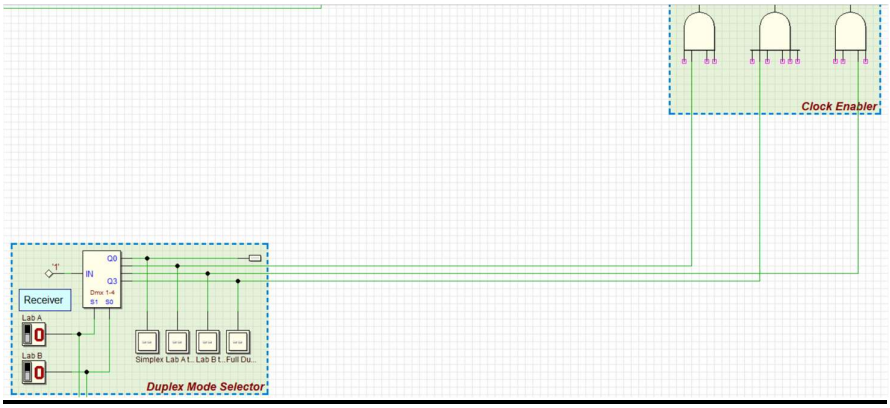


Figure 5.5.4 : Duplex Mode Selector to Clock Enabler

- The Duplex Mode Selector determines the specific target destination for the data being transmitted. The three modes are from Lab A to Lab B, from Lab B to Lab A and both directions.

Table 5.5.1: Truth Table of Duplex Mode Selector

Input			Output				Mode
Data Input	Data Select (Receiver)		Output Data				
	Lab A	Lab B					
IN	Switch2	Switch1	Q0	Q1	Q2	Q3	
1	0	0	1	0	0	0	X (Simplex)
1	0	1	0	1	0	0	Lab A to Lab B
1	1	0	0	0	1	0	Lab B to Lab A
1	1	1	0	0	0	1	Full Duplex

- IN is always 1 whereas the output data is depended on Receiver Switch
- When Switch 2 (Lab A) & Switch 1 (Lab B) is 0, only output Q0 is 1, the other three outputs are 0. The mode is simplex and it is not connected to any of the AND Gate in Clock Enabler.
- When Switch 2 is 0 & Switch 1 is 1, only output Q1 is 1, other three outputs are 0. The mode is from Lab A to Lab B and it is connected to the first AND Gate in Clock Enabler.
- When Switch 2 is 1 & Switch 1 is 0, only output Q2 is 1, other three outputs are 0. The mode is from Lab B to Lab A and it is connected to the third AND Gate in Clock Enabler.
- When Switch 2 & Switch 1is 1, only output Q3 is 1, the other three outputs are 0. The mode is Full Duplex and it is connected to the second AND Gate in Clock Enabler.

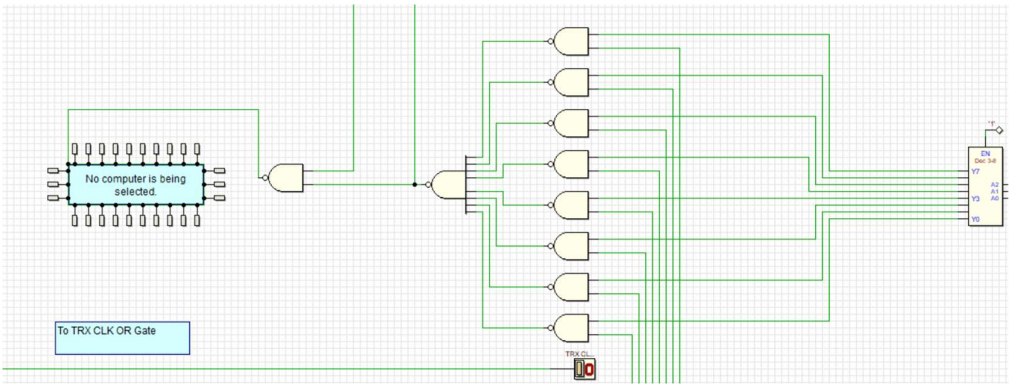


Figure 5.5.5 : TRX CLK located below Lab 1 DEMUX

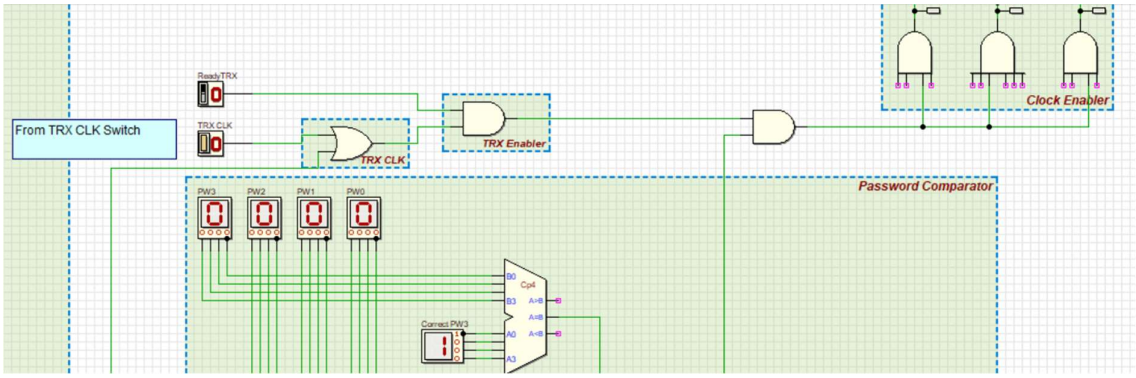


Figure 5.5.6 : TRX CLK, TRX CLK OR Gate, TRX Enabler, TRX Enabler AND Gate, Password Comparator to Clock Enabler

- The TRX CLK OR Gate receives input of 2 TRX CLK Switch (one located below Lab 1 MUX and another located below Lab 1 DEMUX).
- Although both have the same function which is to trigger the initiation of data transmission process it is displaced at different locations for better view and visualisation for the user when button is pushed for different duplex modes.
- The output of TRX CLK is connected to TRX Enabler AND Gate as input together with TRX Enabler (Ready TRX). The TRX Enabler AND Gate is connected to the next level AND Gate which also receives the Password Comparator as input. This next level AND Gate is connected to all AND Gates in Clock Enabler.
- When one of the TRX CLK is pushed and TRX Enabler is at 1, TRX Enabler AND Gates has output of 1 which is passed to the next level AND Gate.
- When the password entered is correct, Password Comparator passes its output of 1 to the AND Gate as input with the input 1 from TRX Enabler AND Gate.
- The AND Gate output 1 is passed to all AND Gates in Clock Enabler.

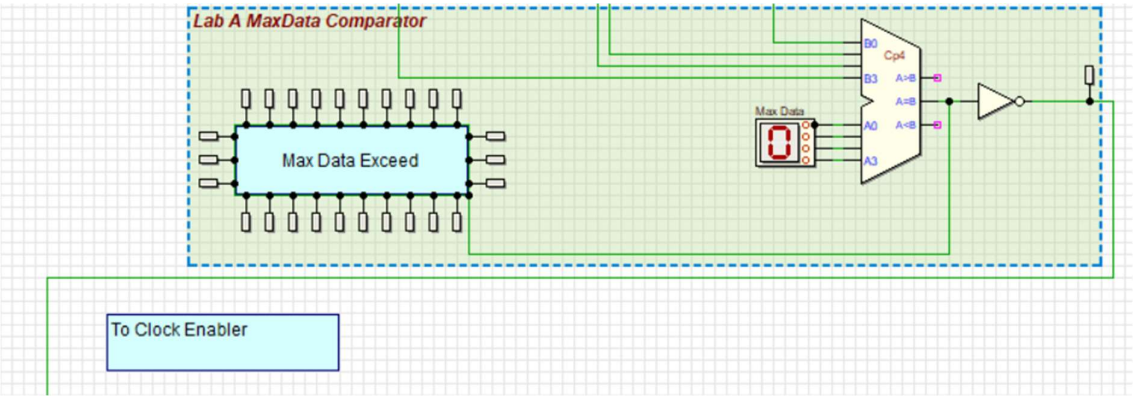


Figure 5.5.7 : Lab A Max Data Comparator to Clock Enabler

- The inverted output from the Lab A Max Data Comparator connected to the second and third AND Gate in the Clock Enabler.
- The inverted output from the Lab B Max Data Comparator is connected to the first and second AND Gate in the Clock Enabler

Table 5.5.2 : Overview of AND Gates in Clock Enabler to be activated

Criteria		First AND Gate (Lab A to Lab B)	Second AND Gate (Full Duplex)	Third AND Gate (Lab B to Lab A)
Lab A Sender Selector		Selected computer is power on (=1)	Selected computer is power on (=1)	X
Lab B Receiver Selector		Selected computer is power on (=1)	Selected computer is power on (=1)	X
Lab A Receiver Selector		X	Selected computer is power on (=1)	Selected computer is power on (=1)
Lab B Sender Selector		X	Selected computer is power on (=1)	Selected computer is power on (=1)
Duplex Mode Selector	**Lab A**	0	1	1
	Lab B	1	1	0
TRX_CLK		Pushed (=1)	Pushed (=1)	Pushed (=1)
TRX Enabler		Enable (=1)	Enable (=1)	Enable (=1)
Password		Correct (=1)	Correct (=1)	Correct (=1)
Lab A Max Data Comparator		X	Data at Lab A is less than or equal to predefined max data (=1)	Data at Lab A is less than or equal to predefined max data (=1)
Lab B Max Data Comparator		Data at Lab B is less than or equal to predefined max data (=1)	Data at Lab B is less than or equal to predefined max data (=1)	X

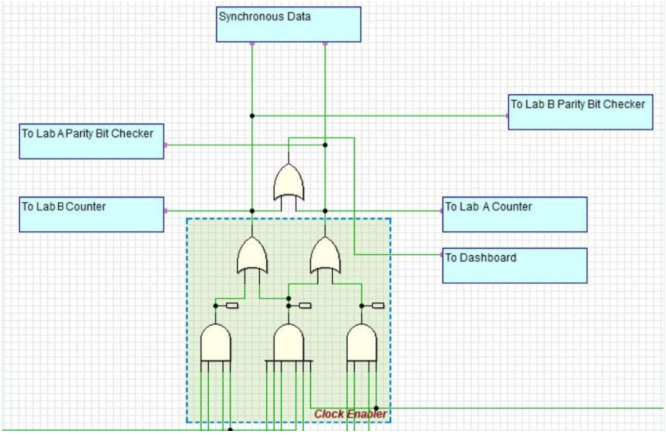


Figure 5.5.8 : Outputs of Clock Enabler

- When either First or Second AND Gate is activated, the first OR gate from the left has the output of 1. The output for the first OR gate is connected to Lab B counter, Lab B Parity Bit Checker, Lab Synchronous Data (between Lab A MUX and Lab A DEMUX) and the next level OR Gate.
- When either Second or Third AND Gate is activated, the second OR gate from the left has the output of 1. The output for the second OR gate is connected to Lab A counter, Lab A Parity Bit Checker, Lab Synchronous Data (between Lab A MUX and Lab A DEMUX) and the next level OR Gate.
- The OR Gate with the previous level OR Gates as input is connected to the Dashboard.

6. Counter

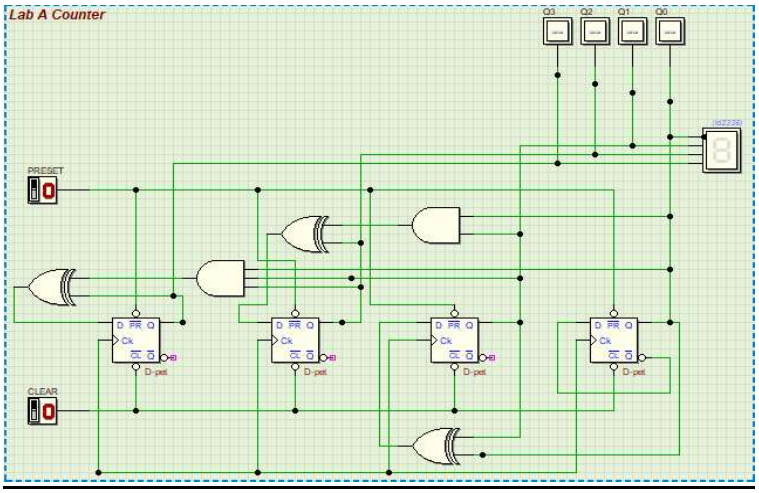


FIGURE 5.6.1 : Lab A Counter

- Lab A counter is a counter which will calculate the total amount of data received by a computer in Lab A. It is a 4-bit synchronous counter which requires clock input to synchronize the circuit operation.

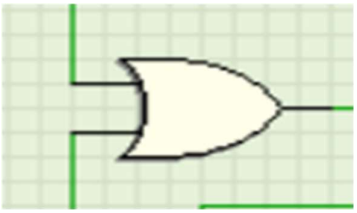


FIGURE 5.6.2 : Lab B to Lab A Mode & Full Duplex Mode

- The operation of the counter depends on the clock. The clock enabler will affect the operation of the counter. All of the clock input of the D Flip-Flops inside the counter is connected by an output of 2-input OR gate.
- The input of 2-input OR gate is connected by the output of a center AND gate (Lab B to Lab A mode) and right side AND gate (Full Duplex mode) at clock enabler which means Lab A counter will function when the system is in Lab B to Lab A mode or Full Duplex mode.

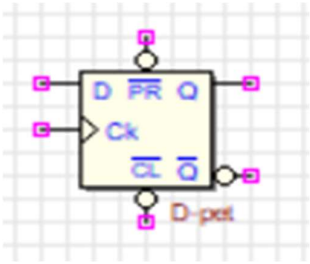


FIGURE 5.6.3 : Positive Edge Triggered D Flip-Flop

- D Flip-Flop is a kind of sequential circuit that is known as “Data Flip-Flop” which is used to store a single bit of data. (GeeksforGeeks, 2023).
- At the left side of D Flip-Flop, it consists of two active high inputs which are D input (D) and Clock input (Ck). D input can affect both active high output of Q and Q’ at the right side of D Flip-Flop.
- At the middle side of D Flip-Flop, it consists of two active low inputs which are Preset input (PR’) and Clear input (CL’). Input PR’ and CL’ can change the mode operation

of D Flip-Flop which are synchronous mode and asynchronous mode. In synchronous mode, the change of output Q and Q' will be changed with the clock signal (Ck). In asynchronous mode, the change of state is not dependent on its clock signal (Ck).

Table 5.6.1 : Truth Table D Flip-Flop (Not include PR' & CL')

Input		Output		Mode
Clock input	D input			
Ck	D	Q	Q'	
0	X	Q	Q'	No Change
1	1	1	0	Set
1	0	0	1	Reset

Table 5.6.2 : Truth Table D Flip-Flop

Input				Output		Mode
Clock Input	Mode Input		D Input			
Ck	PR'	CL'	D	Q	Q'	
0	1	1	X	Q	Q'	No Change
1	1	1	1	1	0	Set
1	1	1	0	0	1	Reset
0	0	1	X	1	0	Set
1	0	1	X	1	0	Set
0	1	0	X	0	1	Reset
1	1	0	X	0	1	Reset
0	0	0	X	Q	Q'	Invalid
1	0	0	X	1	0	Invalid

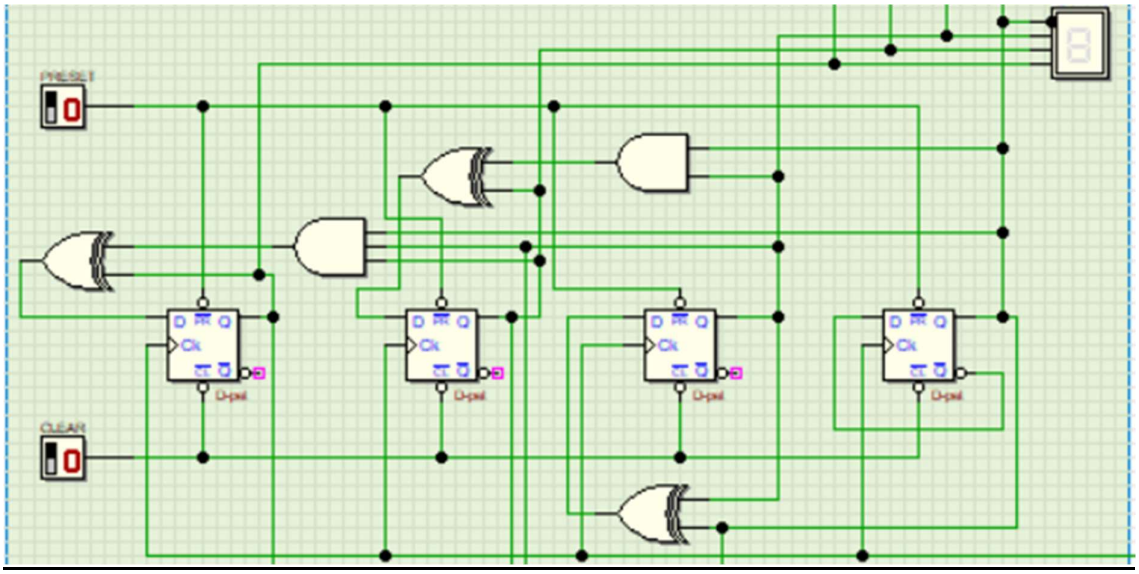


FIGURE 5.6.4 : D Flip-Flop & Connection of Clock, Preset and Clear

- Since this is a 4-bit Synchronous Counter, the circuit requires four D Flip-Flop to store 4-bit digits.
- All of the Clock Input (Ck) of D Flip-Flop is connected to the output of OR gate at Figure 5.6.2 . The clock enabler will decide whether the counter can be functioning or not depending on the condition at Table 5.5.2.
- All of the Preset Input (PR') from every D Flip-Flop is connected to an input (PRESET).
- All of the Clear Input (CL') from every D Flip-Flop is connected to an input (CLEAR).
- The user has to set both Preset Input (PR') and Clear Input (CL') to 1 first before the counter starts to operate to change the mode of counter into synchronous mode. When the counter reaches maximum value, the user should set the Clear Input (CL') to 0 to reset all the values into 0. Then, the user should set the Clear Input (CL') to 1 again to allow the counter to change back to synchronous mode and the counter will be able to start a new round of counting.

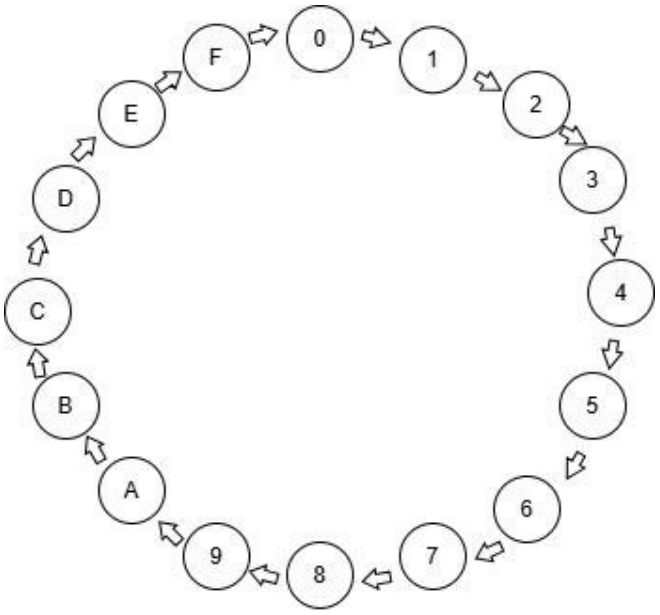


FIGURE 5.6.5 : State Diagram (Hexadecimal Digit)

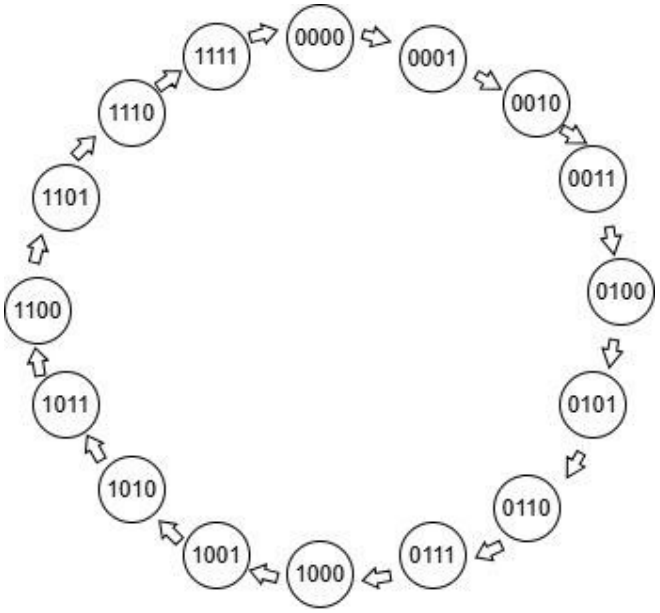


FIGURE 5.6.6 : State Diagram (Binary Digit)

4

Table 5.6.3 : Next State Table

Present State				Next State			
Q3	**Q2**	**Q1**	**Q0**	**Q3+**	**Q2+**	**Q1+**	**Q0+**
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Table 5.6.4 : Excitation Table (Original)

Present State	Next State	Flip-Flop State
Qn	**Qn+1**	**D**
0	0	0
0	1	1
1	0	0
1	1	1

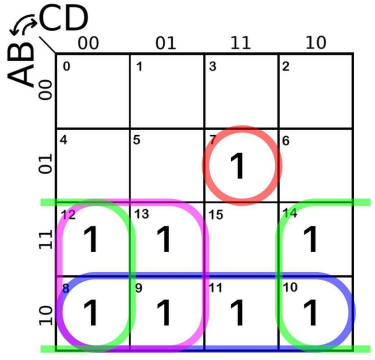
Table 5.6.5 : Flip-Flop Transition Table

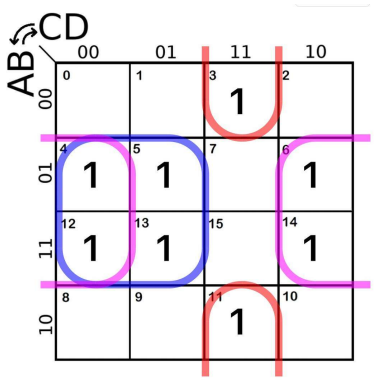
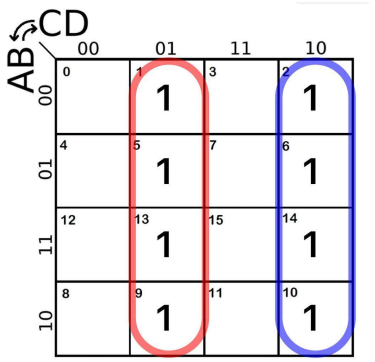
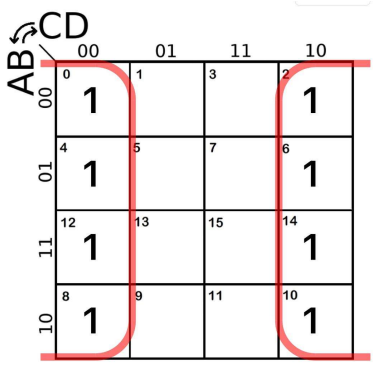
Present State				Next State				D Transition			
Q3	**Q2**	**Q1**	**Q0**	**Q3+**	**Q2+**	**Q1+**	**Q0+**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1

1	0	1	1	1	1	0	0	1	1	0	0
1	1	0	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0

- Before we start connecting the D input, we need to do these steps before we design the connection of the input :
    1. Determine State Diagram (Figure 5.6.4 & Figure 5.6.5)
    2. Determine Next State Table (Table 5.6.2)
    3. Determine Excitation Table of D Flip-Flop (Table 5.6.3)
    4. Determine Flip-Flop Transition Table (Table 5.6.4)
- Finish with the steps, we need to construct the K-Map by using the present state value to get the simplified logic equation before we connect the circuit.

Table 5.6.6 : K-Map

Input	K-Map	Logic Equation
D3		D3 = A'BCD + AB' + AC' + AD' D3 = Q3'Q2Q1Q0 + Q3Q2' + Q3Q1' + Q3Q0' = Q3'Q2Q1Q0 + Q3(Q2' + Q1' Q0') = Q3(Q2Q1Q0)' + Q3'(Q2Q1Q0) = Q3 ⊕ Q2Q1Q0

D2		D2 = B'CD + BC' + BD' D2 = Q2'Q1Q0 + Q2Q1' + Q2Q0' = Q2'Q1Q0 + Q2(Q1' + Q0') = Q2 ⊕ Q1Q0
D1		D1 = C'D + CD' D1 = Q1'Q0 + Q1Q0' = Q1 ⊕ Q0
D0		D0 = D' D0 = Q0'

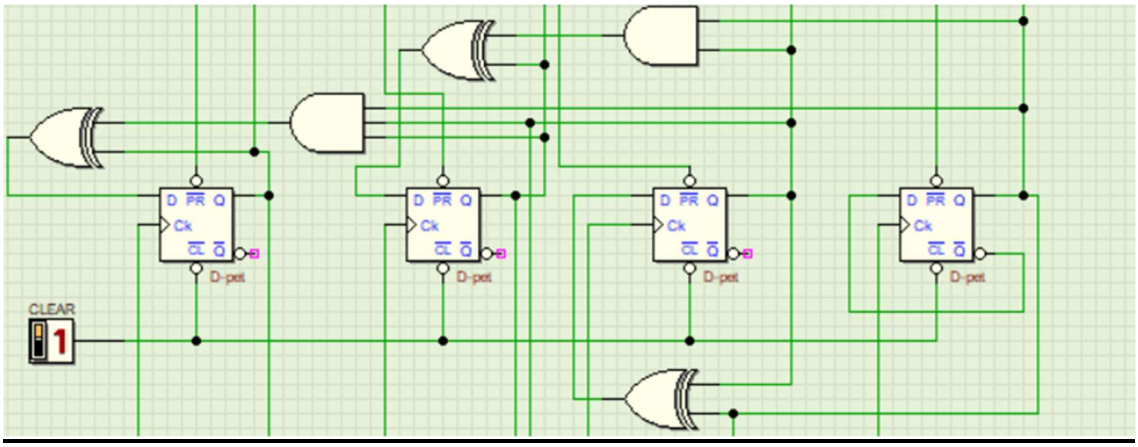


FIGURE 5.6.7 : D Input Connection of Lab A Counter

- Since we had constructed the K-Map and logic equation, we can connect all of the D input based on the result

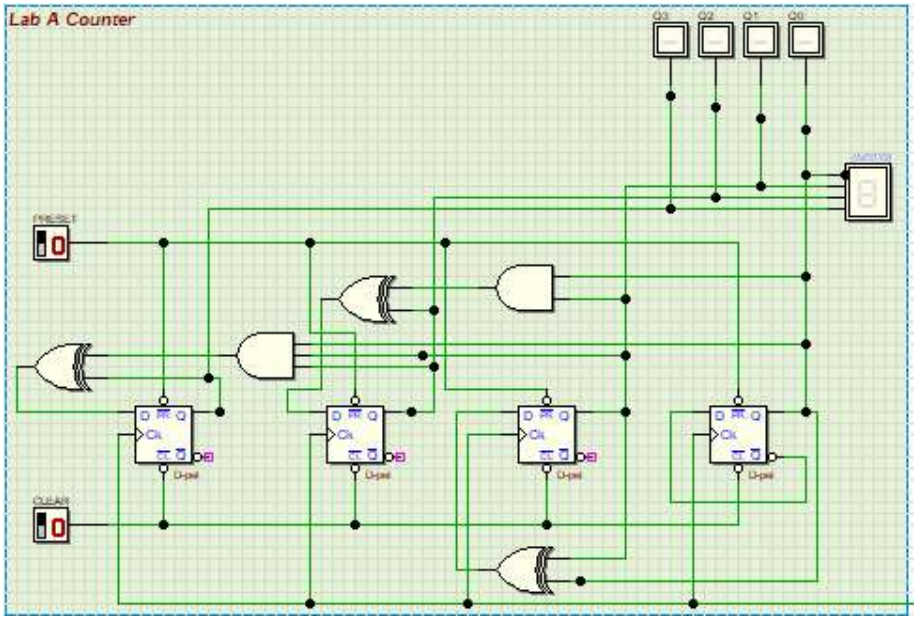


Figure 5.6.8 : Q Output & Actual Received Value

- Each Q output from difference D Flip-Flop is connected to a 1-bit display.
- All of the Q output is connected to a 4-bit hex digit output at the right side.
- The user is able to see the count up process of the counter which will show the result in binary (Four 1-bit Display) or hexadecimal (4-bit Hex Digit Output).

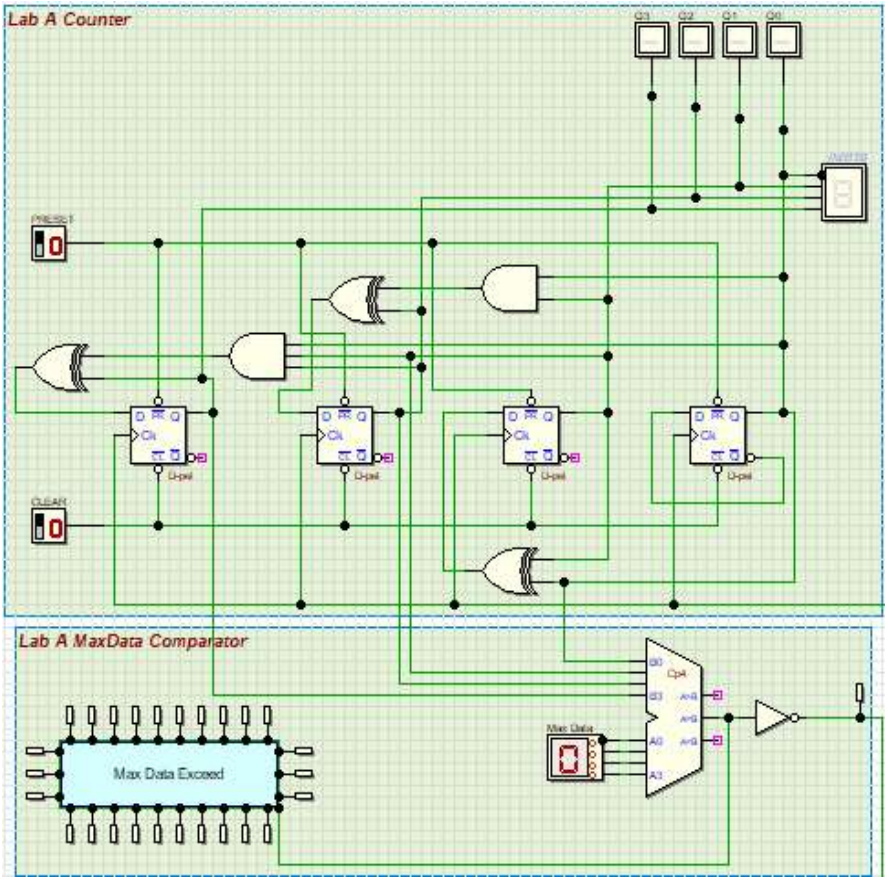


Figure 5.6.9 : Actual Received Data & Predefined Max Data

- All of the output Q from Lab A counter will connect to the input of Comparator (B0 until B3) at the Lab A Max Data Comparator. (Figure 5.4.1)
- The comparator will compare the value of the actual received data by a computer in Lab A with the predefined maximum data that can be accepted by the computer in Lab A to determine whether the data transmission can proceed or not.

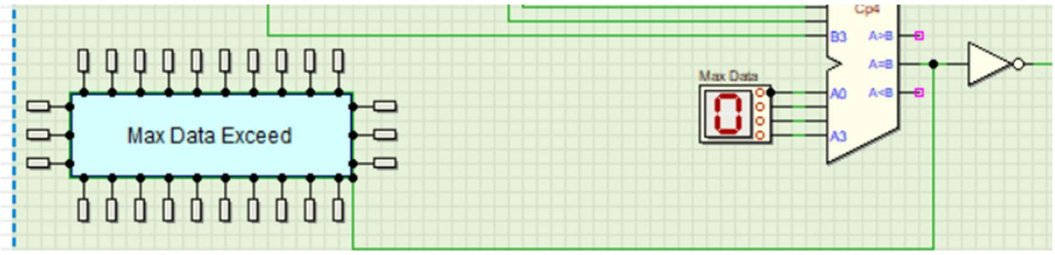


Figure 5.6.10 : Fun Display (Detection For the Maximum Data Exceed or Not)

- We prepared a fun display at Lab A counter which is used to detect whether the actual received data has reached the maximum or not.
- All of the test LED is connected by the output of NOT gate at Lab A Max Data Comparator.(Figure 5.4.4)
- The NOT gate will output result 1 when the actual received data of a computer in Lab A is not equal to the predefined max data of a computer in Lab A . Otherwise, the NOT gate will output the result 0. (Maximum reached)

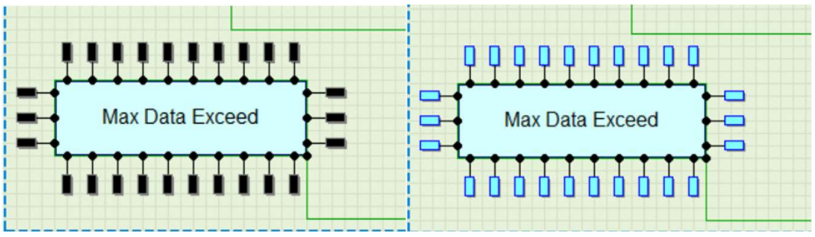


FIGURE 5.6.11 : Status Test LED (Max not Reached/Max Reached)

- All of the test LEDs will turn black (OFF) when receiving input 0 from the comparator which means the maximum is not reached.
- All of the test LEDs will turn blue (ON) when receiving input 1 from the comparator which means the maximum is reached.

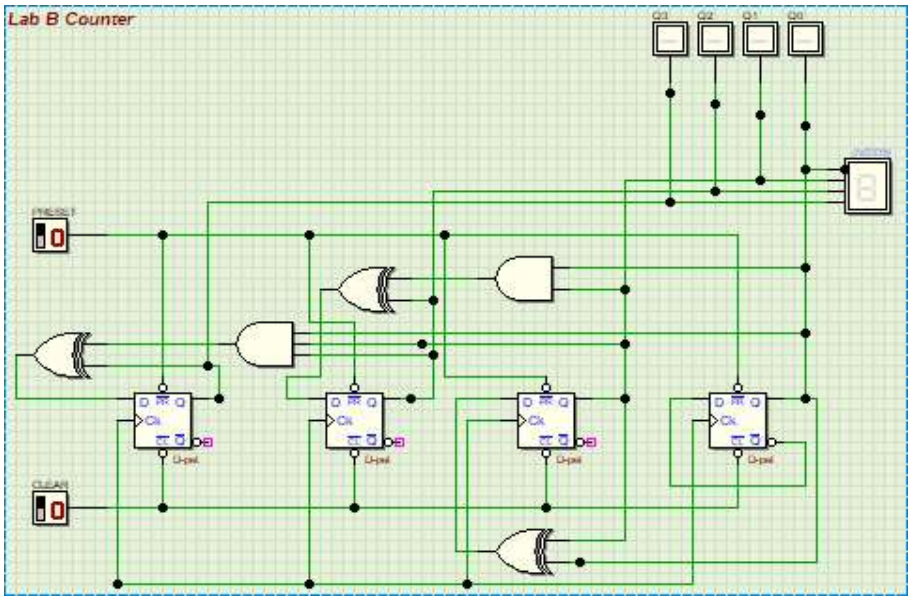


FIGURE 5.6.12 : Lab B Counter

- All of the concepts that are mentioned in Lab A counter are the same as we apply on Lab B counter. The only difference between Lab A counter and Lab B counter is where the input of the clock input (Ck) came from and where the output Q of all D Flip-Flop is connected to.

7. Parity Checker

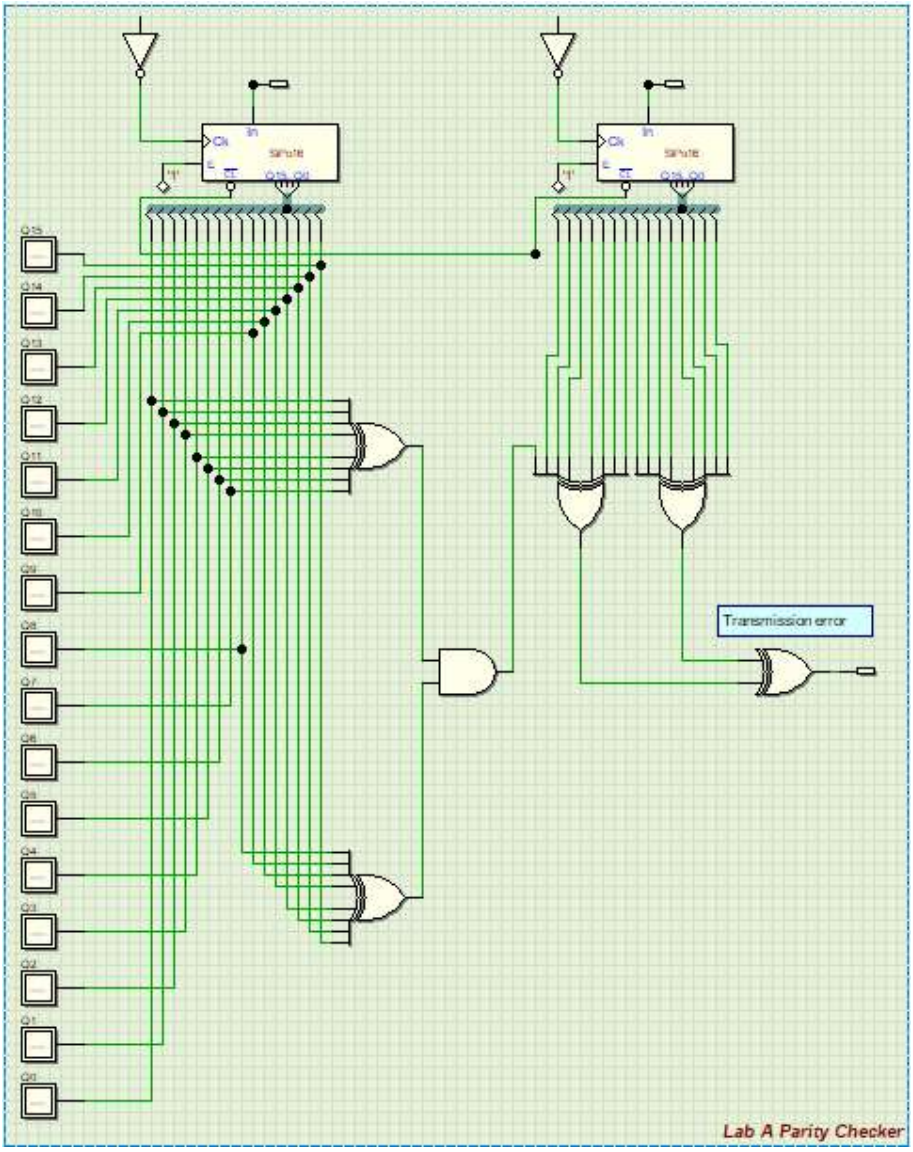


Figure 5.7.1 : Lab A Parity Checker

- Lab A Parity Checker acts as a device that is used to detect errors in data transmission when the data is transferred from computer in Lab B to computer in Lab A.

- This circuit checks whether the received data, including the parity bit conforms to the even parity rule.
- It checks if the number of 1's in a binary data set is following predefined parity rule which is even. If the data does not match the even parity, the checker will signal an error, indicating that something went wrong during transmission.

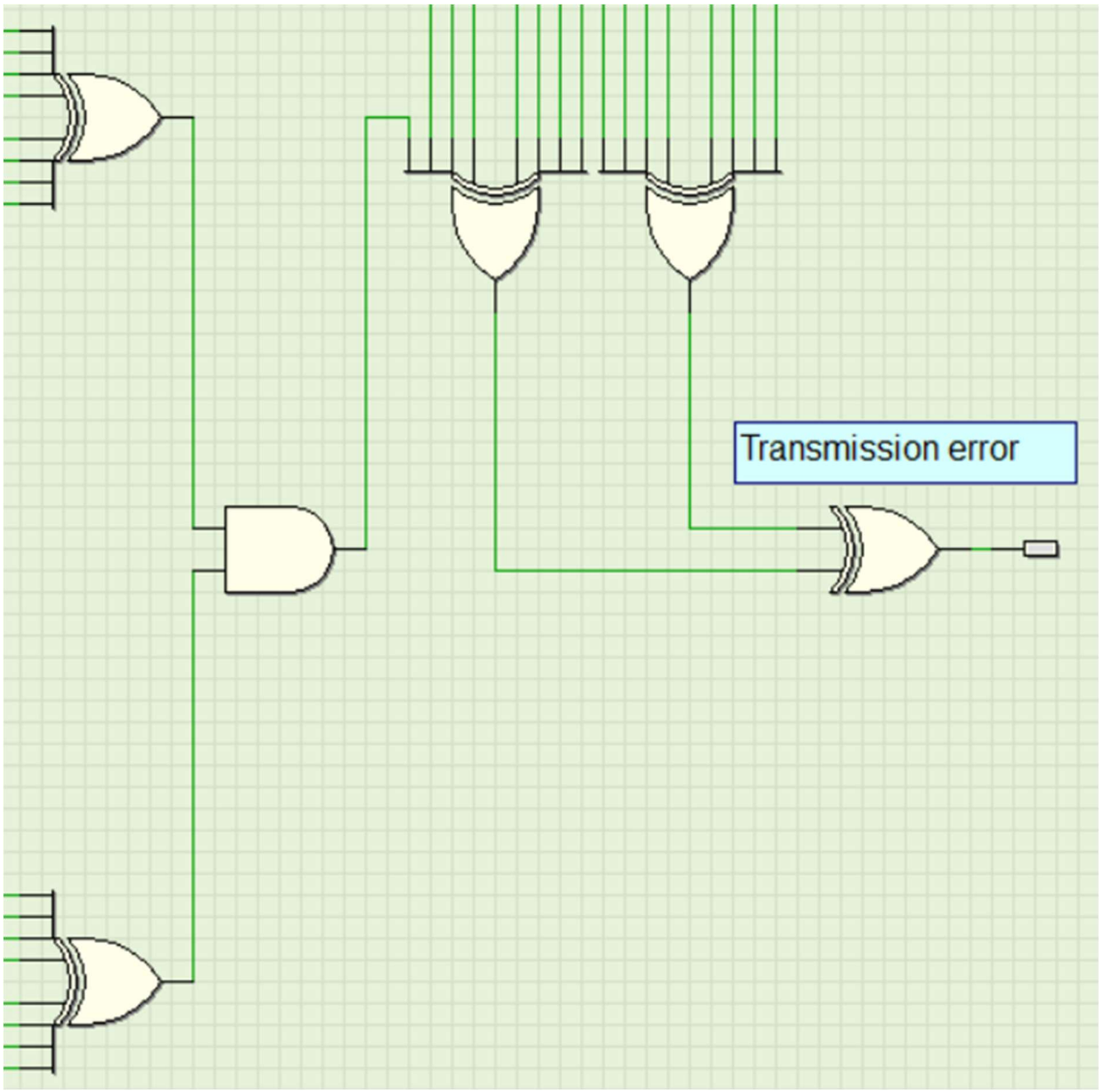


Figure 5.7.2 : XOR Gate

- XOR inherently determines whether there is an even or odd number of 1s.
- If the parity is even, the output is 0 while if the parity is odd, the output is 1.

Table 5.7.1 : Truth table for a two-input XOR gate

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

- The result of A ⊕ B is 0 when the total number of 1 is even while the result turns 1 when the total number of 1 is odd.

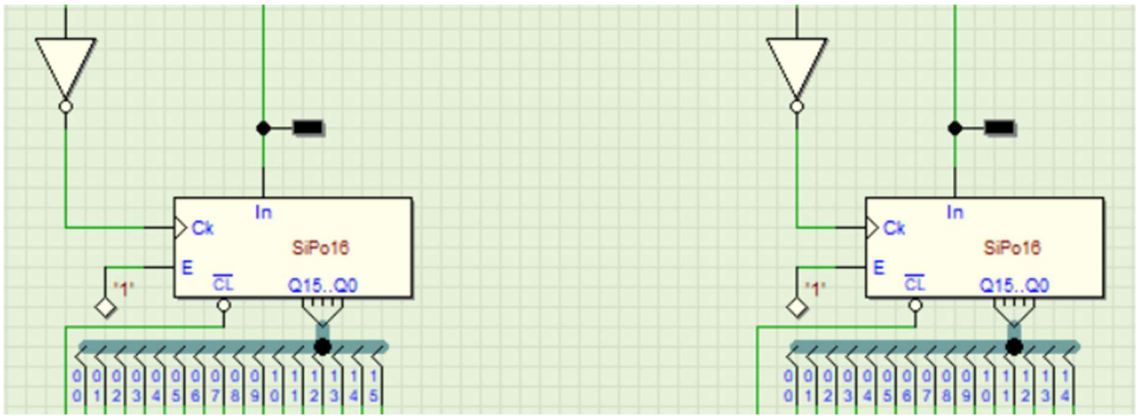


Figure 5.7.3 Serial In Parallel Out (SIPO) Register

- For this 16 bit SIPO, it can store 16 bit data and shift the data at every clock cycle.
- Data entered serially and after the 16 clock cycle, the data entered can be read at the output.

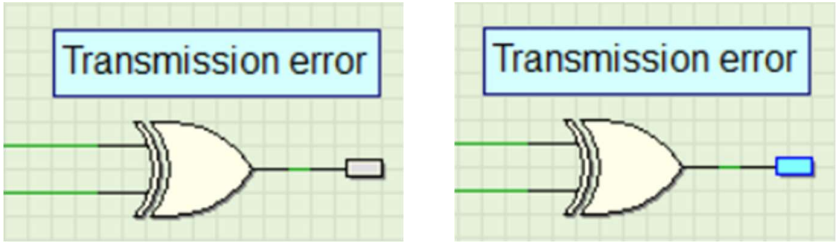


Figure 5.7.4 Status Test LED (No Transmission Error/Transmission Error Exists)

- The LED will not light up when receiving input 0 which means the total number of 1s in the input is even (valid) and there is no transmission error.
- The LED will turn to blue colour when receiving input 1 which means the total number of 1s in the input is odd (invalid) and transmission error exists.

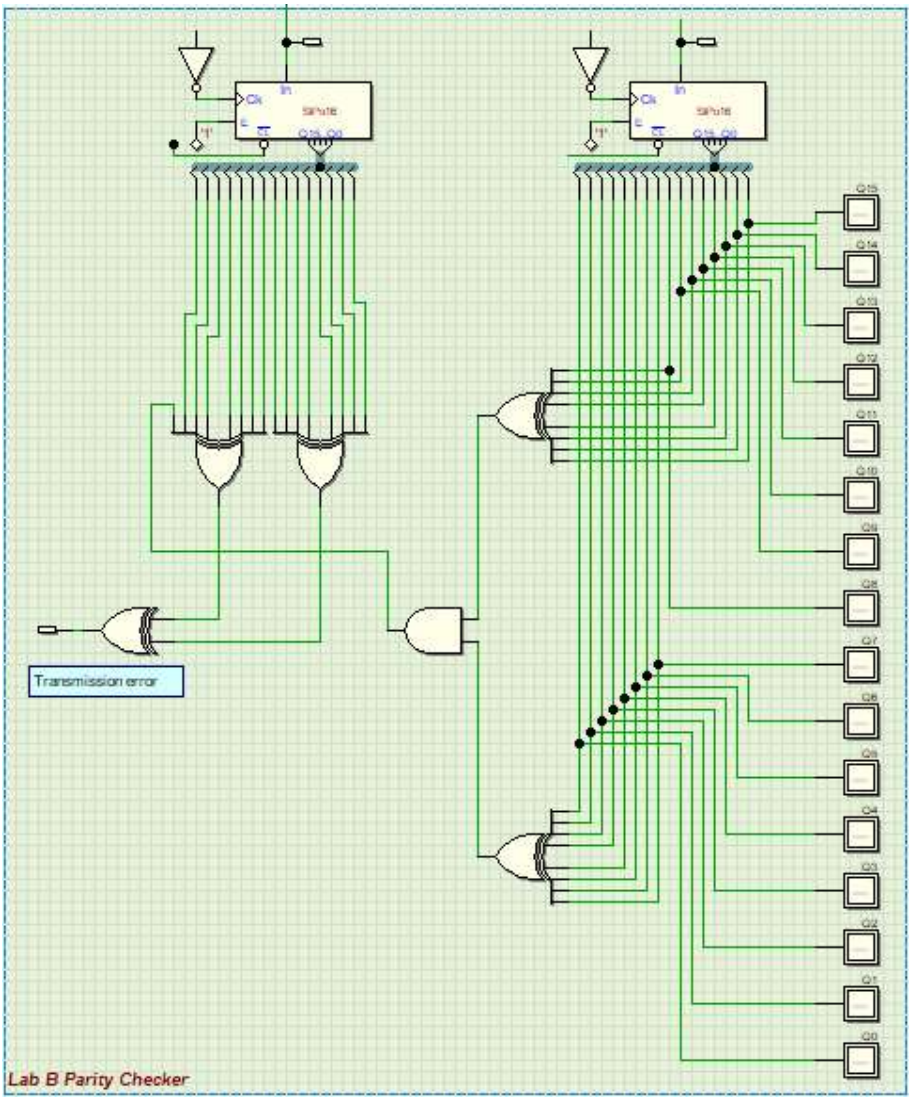


Figure 5.7.5 Lab B Parity Checker

- All of the concepts that are mentioned in Lab A Parity Checker are the same as the concept we apply on Lab B Parity Checker. The only difference between Lab A Parity Checker and Lab B Parity Checker is where the input of the Shift Register came from.
- It acts as a device that is used to detect errors in data transmission when the data is transferred from computer in Lab A to computer in Lab B.

8. Dashboard

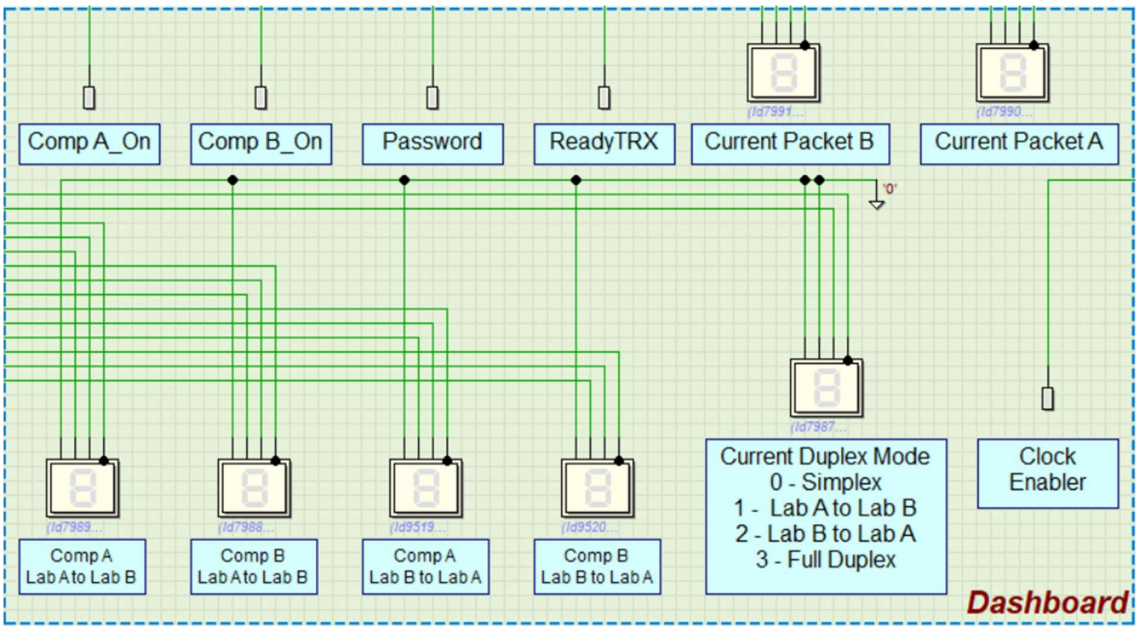


Figure 5.8.1 Dashboard

- This dashboard is designed to make it easier for users to monitor the status output.
- **Row 1**
    - The first LED detects whether any computer in Lab A is powered on. It remains off if no computer in Lab A is powered on, but it turns blue when at least one computer in Lab A is powered on.
    - The second LED detects whether any computer in Lab B is powered on. It remains off if no computer in Lab B is powered on, but it turns blue when at least one computer in Lab B is powered on.
    - The third LED checks the correctness of the password. If the password is correct, it turns blue; otherwise, it remains off.
    - The fourth LED lights up when ReadyTRX is on and remains off when ReadyTRX is off.
    - The first seven-segment display detects the current packet transfer of data from computers in Lab B to Lab A.
    - The second seven-segment display detects the current packet transfer of data from computers in Lab A to Lab B.

- **Row 2**
    - The first seven-segment display checks whether any computer in Lab A is powered on when data is transferred from Lab A to Lab B or during duplex mode.
    - The second seven-segment display checks whether any computer in Lab B is powered on when data is transferred from Lab A to Lab B or during duplex mode.
    - The third seven-segment display checks whether any computer in Lab A is powered on when data is transferred from Lab B to Lab A or during duplex mode.
    - The fourth seven-segment display checks whether any computer in Lab B is powered on when data is transferred from Lab B to Lab A or during duplex mode.
    - The fifth seven-segment display shows the current duplex mode.
        - When it displays 0, the system is in simplex mode.
        - When it displays 1 or 2, the system is in half-duplex mode.
            - 1 indicates data transfer from Lab A to Lab B.
            - 2 indicates data transfer from Lab B to Lab A.
        - When it displays 3, the system is in full-duplex mode.
    - The last LED lights up when the clock enabler is activated.

6.0 Conclusion and Reflection

The Digital Logic Network Packet Transmission Monitoring System provided an excellent opportunity to translate theoretical knowledge into practical application. By leveraging digital logic principles and MSI circuits, we successfully developed a system capable of secure and efficient data transfer between two labs. Key components such as multiplexers (MUX), demultiplexers (DEMUX), encoders, decoders, comparators and shift registers were effectively integrated to enable essential features, including bidirectional communication, user authentication and destination selection.

A key strength of this project was the efficient use of digital circuits to meet system requirements. The inclusion of a seven-segment display for user interaction and the synchronization of data transmission using counters highlighted our focus on creating a functional and user-friendly system. Moreover, we have a dashboard to ease the user to monitor the condition of the system. Additionally, the project demonstrated our ability to work collaboratively and address challenges in digital circuit design.

Despite its success, the system has some limitations. For instance, due to the overall circuit design, our parity checker is unable to transfer the last bit of data. To address this issue, we plan to implement a maximum or optimized data bit transfer mechanism in future iterations to ensure seamless data transmission.

In conclusion, this project successfully demonstrated the practical application of digital logic concepts and provided valuable insights into system design and problem-solving. While there is room for further enhancement, the system lays a solid foundation for future development and real-world applications.

7.0 References

Digital Logic (5th ed.). (2024). Faculty of Computing, Universiti Teknologi Malaysia.

GeeksforGeeks. (2023, July 16). What is Digital Logic ? GeeksforGeeks. https://www.geeksforgeeks.org/what-is-digital-logic/

8.0 Appendices

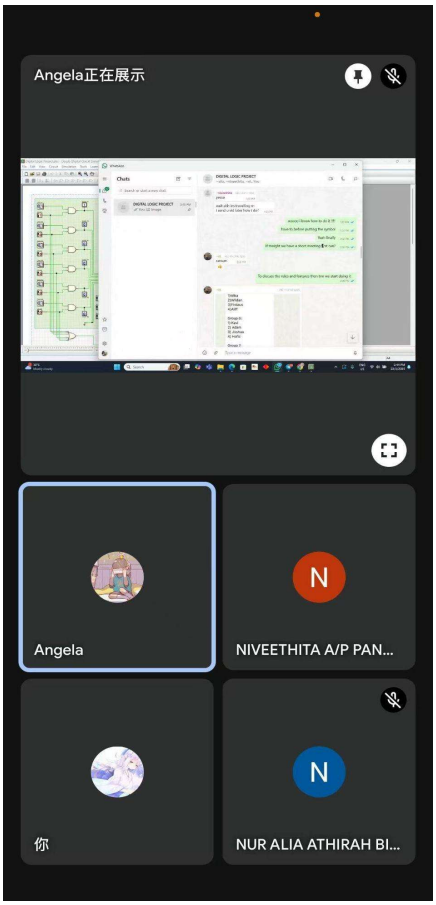
1. Digital Circuit Simulator (Deeds) :
https://www.digitalelectronicsdeeds.com/

2. Block Diagram & Flowchart Generator (draw.io) :
https://app.diagrams.net/

3. K-Map Picture Generator (Karnaugh Kmap Solver) :
https://play.google.com/store/apps/details?id=karnagh.ammsoft.karnagh

4. Presentation Video (Record) :
https://youtu.be/La8l-6TLpho

5. Group Meeting Picture :



6. Task Distribution

1	Angela Ngu Xin Yi	A24CS0226	• Deeds Circuit Implementation • Report Writing • Video Presenting
2	Tan Xin Tian	A24CS0198	• Report Writing • Deeds Circuit Implementation • Video Presenting
3	Niveethita A/P Pandia Rajan	A24CS0148	• Slide Preparation • Video Editing • Report Writing
4	Nur Alia Athirah binti Suzuddin	A24CS0153	• Slide Preparation • Video Editing • Report Writing