SECR1013-DIGITAL LOGIC

PROJECT DIGITAL LOGIC

Network Packet Transmission Monitoring System

LECTURER: MR. AHMAD FARIZ

Section : 01

SUBMITTED BY:

STUDENT	NAME	MATRIX NO.	PHONE NO.	E-MAIL
1	ELEANOR TING	A24CS0247	01131780134	eleanorting.pik@graduate.utm.my
2	TAN HUI SHAN	A24CS0197	0102018448	tanhui.shan@graduate.utm.my
3	LEE HUI ZHEN	A24CS0259	0149895814	leehuizhen@graduate.utm.my
4	WONG ZI NING	A24CS0313	01115040506	wongzi.ning@graduate.utm.my

Faculty of Computing

Universiti Teknologi Malaysia, 81310 Skudai,

Johor Bahru, Malaysia

DEDICATION & ACKNOWLEDGEMENT

DEDICATION

This project is humbly dedicated to our parents, mentors, and loved ones, whose unwavering support, encouragement, and sacrifices have been the foundation of our journey. We would like to express our appreciation to our lecturers and mentors for their vital knowledge and inspiration, which helped to mold our abilities and resolve. Lastly, this effort serves as a monument to the boundless potential that perseverance and teamwork may accomplish, for everyone who believes in the transformational power of technology and creativity.

ACKNOWLEDGEMENT

With great appreciation, we would like to thank everyone who helped us finish the Network Packet Transmission Monitoring System project. First and foremost, we thank our lecturer, Mr. Ahmad Fariz bin Ali, for their guidance, patience, and insightful feedback throughout this project. His knowledge and support have been crucial in making this concept a reality. We would especially like to thank our peers and colleagues for their encouragement, cooperation, and helpful critiques, all of which helped us improve our work. This project is the product of teamwork and commitment, and we sincerely thank everyone who helped make it a reality.

Table of Contents

1.0 INTRODUCTION

Digital logic plays a very vital role in modern computing systems today to enable the design and operation of hardware for underpinning communication, computation, and control processes.

This project aims to bridge theoretical knowledge with practical application by simulating a real-world scenario. Students are tasked to design and simulate a digital logic circuit using Deeds software, this allows them to explore creative problem-solving, design methodologies, and the fundamentals of circuit implementation.

This project centers on establishing a functional connection between two computer labs via a simulated single cable. Each lab comprises a group of computers, and the transmission process includes the transfer of packetized data from a source in Lab 1 to a destination in Lab 2. Through collaboration, students will explore synchronous counter designs, multiplexer/demultiplexer configurations and monitoring functions. The project promotes a comprehensive understanding of digital logic principles by integrating a variety of basic and advanced digital components.

2.0 PROBLEM BACKGROUND

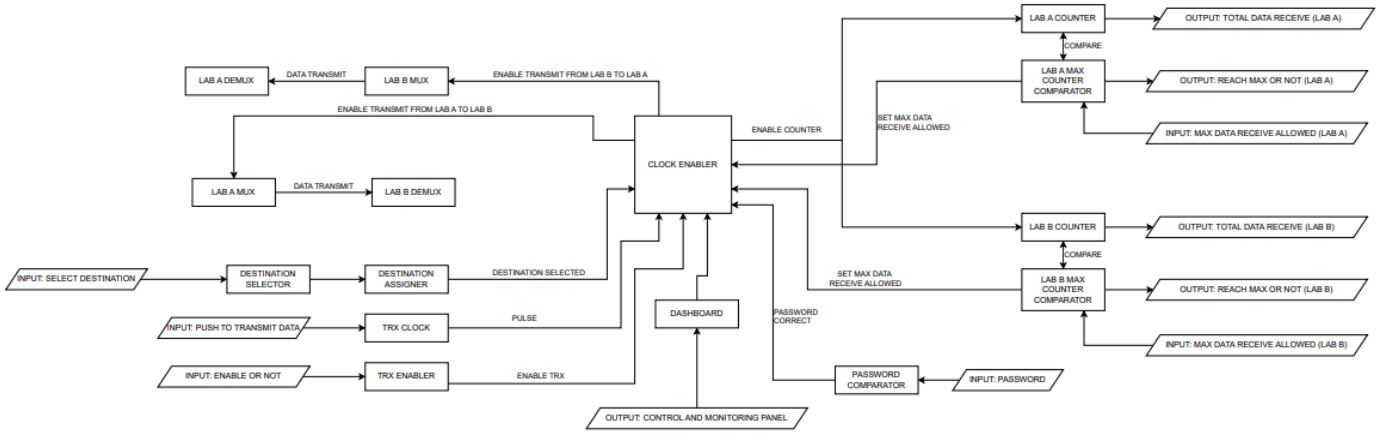
In modern digital systems, reliable and efficient data communication is critical to maintaining the integrity and functionality of interconnected networks. Laboratories, in particular, often rely on communication between multiple computers to enable seamless operations. This project simulates a practical scenario involving the transfer of packetized data between two computer labs, Lab 1 and Lab 2 by using a single cable for communication. The challenge lies in designing a system that not only manages this transmission but also monitors the transfer process and detects potential faults.

The circuit design focuses on implementing synchronous counters for tracking the data packets sent between the labs to ensure accurate operations through the use of flip-flops such as JK, T, or D. Data flow between computers is managed using multiplexers and demultiplexers, facilitating the routing of information from the source to the destination computer. A clock enabler function controls the operation of the counters, ensuring data transfer only occurs under defined conditions. To further enhance functionality, comparators are incorporated to monitor the packet transfer, verifying that the current packet count matches the user-defined maximum.

To provide insights into the system's operation, the design integrates a dashboard that monitors essential parameters such as power status, counter activity and packet completion. For more advanced functionality, error detection mechanisms can be implemented to identify and report potential issues such as power, comparator or clock faults. These elements ensure a robust and reliable system while simulating real-world communication challenges.

By addressing these requirements, this project enables students to apply digital logic concepts practically, fostering creativity and critical problem-solving skills. The integration of these components within the Deeds simulation platform serves as a hands-on approach to exploring the intricacies of digital circuit design and reinforces the relevance of these skills in tackling real-world communication challenges.

3.0 SUGGESTED SOLUTION



1. PASSWORD COMPARATOR

➔ Authentication Check:

- Ensures that only authorized users can proceed with system operations.
- Compares the user-entered password with a stored reference value.

➔ Enabling System Functions:

- If the password is correct, it unlocks and enables turning on computers, starting transmission

➔ Security Control:

- Prevents unauthorized access to data transmission or system controls between Lab A and Lab B.

➔ Signal Processing:

- Likely produces an output signal that acts as a control input for other components

2. DESTINATION SELECTOR & DESTINATION ASSIGNER

➔ Allows the user or system to choose the destination of data transmission(Lab A, Lab B or both).

- ➔ Determines whether data should be sent to Lab A or Lab B based on the selected mode.
- ➔ Assigns the selected destination to the appropriate transmission path.
- ➔ Ensures that data packets are correctly mapped to the chosen destination.

3. MUX & DEMUX(LAB A / LAB B)

- ➔ A Multiplexer (MUX) takes multiple input signals and selects one to send as an output.
- ➔ It reduces the number of required communication lines by allowing multiple signals to share a single transmission path.
- ➔ The MUX selects which computer's data in lab A or lab B gets transmitted to the destination.
- ➔ It helps in controlling which lab's data is currently being sent based on the selected mode.
- ➔ A Demultiplexer (DEMUX) takes a single input signal and routes it to one of multiple outputs.
- ➔ It expands the single transmission path back into multiple destinations
- ➔ When data arrives at the destination, the DEMUX directs it to the correct computer in Lab A or Lab B.
- ➔ It ensures that each computer only gets the data intended for it, avoiding mix-ups.

4. COUNTER(LAB A/LAB B)

- ➔ A counter is a sequential circuit that counts pulses or specific events. It increments its value every time a clock signal is received.
- ➔ Tracking Data Transmission(Using Count up counter):
    - If Lab A sends a message → Lab A Count Up Counter increments.
    - If Lab B sends a message → Lab B Count Up Counter B increments.

5. MAX COUNTER COMPARATOR(LAB A/LAB B)

- ➔ A Counter Comparator is a digital circuit that compares the value of a counter against a predefined reference value. Based on the comparison, it generates an output signal that can trigger specific actions in the system.
- ➔ Transmission Control:
  - If Counter A reaches a maximum packet limit the Max Counter Comparator stops further transmission or triggers a duplex mode change.
- ➔ Prevents overflow errors by enforcing a maximum counter value.

6. 3-BIT INPUT

- ➔ A 3-bit input has 3 individual binary signals (each either 0 or 1).
- ➔ The multiplexer uses the control signals to choose one of the 3-bit wide data inputs and pass it through to the output.
- ➔ The demultiplexer takes in the 3-bit data and based on the selection lines, routes this input to one of the outputs.

7. TRX CLOCK

- ➔ Provides timing signals for synchronization.
- ➔ Ensures accurate timing for data transmission process.
- ➔ It is the pulse-triggered input which users need to push the button each time of the TRX Clock to allow the system to transmit a bit of data.
- ➔ Ensures that data is transmitted and received at the correct intervals.

8. TRX ENABLER

- ➔ Activates and manages transceiver functionalities.
- ➔ Acts as the control unit that manages the operation of the TRX.
- ➔ Controls the operational states of the transceiver.
- ➔ When TRX Enable is at disabled state (= 0), all the transmission operation will not be processed to ensure that no data is transmitted or received by a certain computer.

9. CLOCK ENABLER

- ➔ Generates clock signals required for timing and synchronization within a circuit or system.
- ➔ The Clock Enabler only will enable the clock when the Password(1688) is correct, TRX Enabler is enable, Destination Assigner is assigned, Sender Selector and Receiver Selector is assigned, Max Counter Comparator is assigned and TRX Clock is active by users.

10. DASHBOARD

- ➔ Organise all the output to confirm the data is transmitted.
- ➔ The leds help to check whether the computer A is on, computer B is on, password is correct, TRX Clock is active and the Clock Enabler is enabled.
- ➔ Monitors essential parameters such as power status, counter activity and packet completion.
- ➔ Provides a clear and organized way to monitor the system's status and data flow.

4.0 REQUIREMENT

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

1. **Password authentication:**

Before the system can function, users must provide a four-digit password, ensuring safe access.

2. **Destination selector & Destination Assigner:**

Gives users the option to select Lab A, Lab B, or both (Full Duplex Mode) as the desired destination for data transfer.

3. **MUX and DEMUX:**

To facilitate effective data transfer between Lab A and Lab B, MUX routes data from various sources to a single line, while DEMUX distributes data from one line to several outputs.

4. **Counters:**

Based on the triggered edge of the clock and only active when the Clock Enabler is on, a synchronous 4-bit counter counts the total number of bits of data received by Lab A and Lab B.

5. **Maximum Data Transfer Comparator**:

Determines if data receipt can continue by comparing the user-specified maximum data limit with the actual data received by Lab A and Lab B.

6. **Push-to-Transmit Control (TRX Clock):**

Sends a digital signal or presses a button to initiate data transmission, forcing users to send each bit of data.

7. TRX Enabler:

Regulates the data transmission status of the system, enabling or disabling transmission for security or maintenance reasons.

8. 3-bit Input

In a demultiplexer, the 3-bit input is routed to one of the outputs based on the control signals, whereas in a multiplexer, the control signals select one of the inputs and pass it through to the output.

9. Clock Enabler:

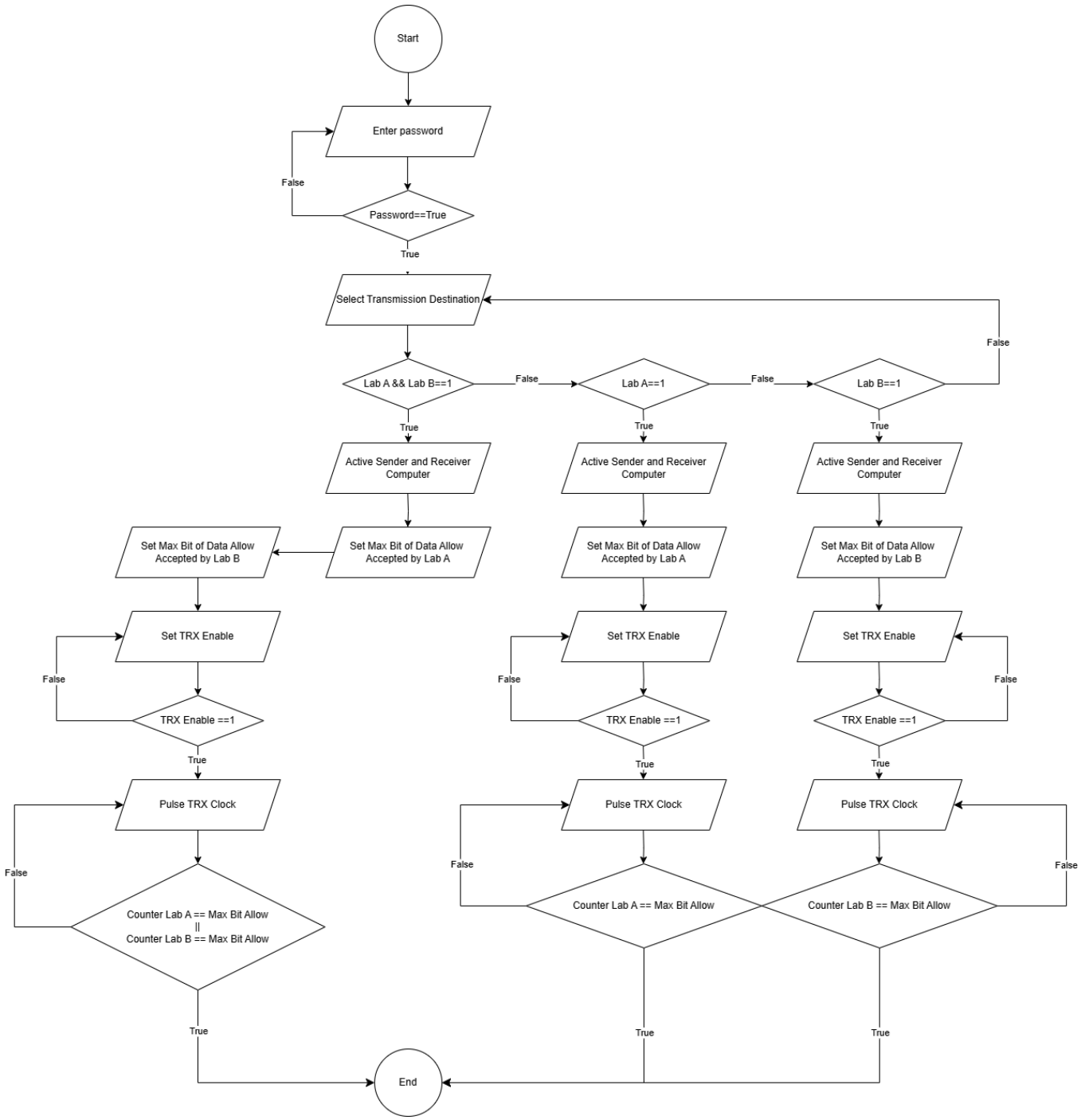
Controls system synchronization and timing, turning on the clock only when certain requirements are satisfied (active TRX Clock, destination assigner, sender/receiver selectors, max counter comparator, correct password, and TRX Enabler).

10. Dashboard

Showing status indicators : LEDs light up when computers A and B are powered on, entering the correct password, the clock signal is active or the TRX button is pressed.

5.0 SYSTEM IMPLEMENTATION

5.1 OVERALL DESIGN



5.2 INTERNAL DESIGN

1. Password Comparator :

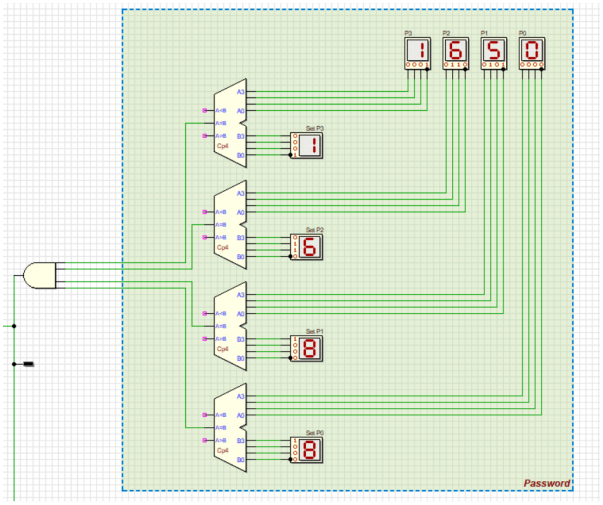


Figure 5.2.1 : Password Incorrect

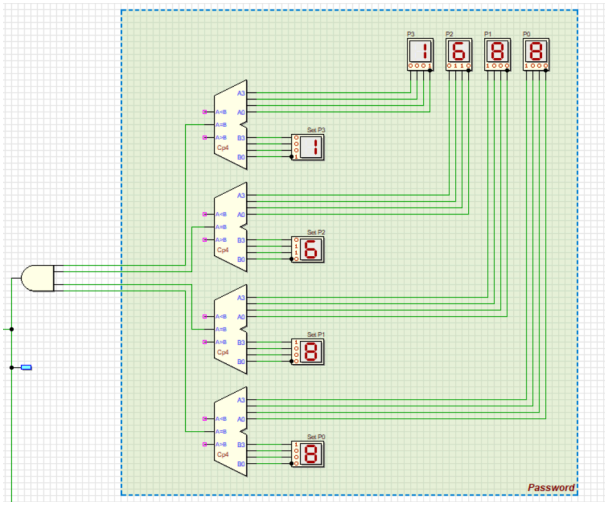


Figure 5.2.2 : Password Correct

- The password comparator consists of four 4-bit comparators which will compare four digits of password input by the user with the password predefined by the system (user).

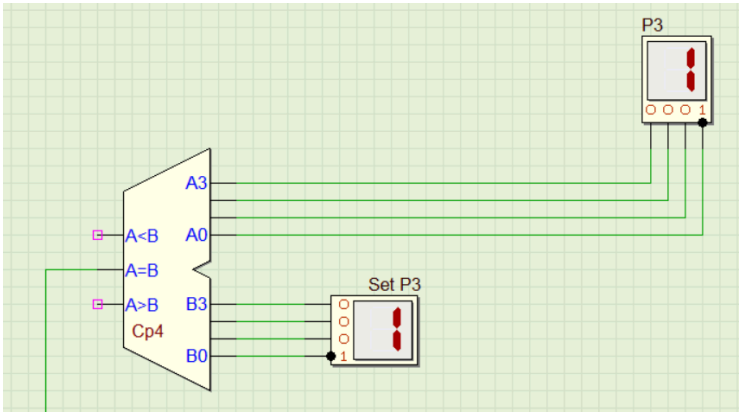


Figure 5.2.3 : 4-Bit Comparator

- P3 (User Input): A 4-bit hex digit input representing the password entered by the user, connected to A0–A3 of the comparator.
- Correct P3 (Stored Password): The system's stored password, connected to B0–B3 of the comparator.
- Comparator Function: Compares P3 (A0–A3) with Correct P3 (B0–B3) and produces three possible outputs:
    - A < B (User input is smaller than the stored password)
    - A = B (User input matches the stored password)
    - A > B (User input is greater than the stored password)
- Required Output: Only A = B is used in the circuit.
- Password Validation:
    - If A = B outputs 1, the password is correct.
    - If A = B outputs 0, the password is incorrect.

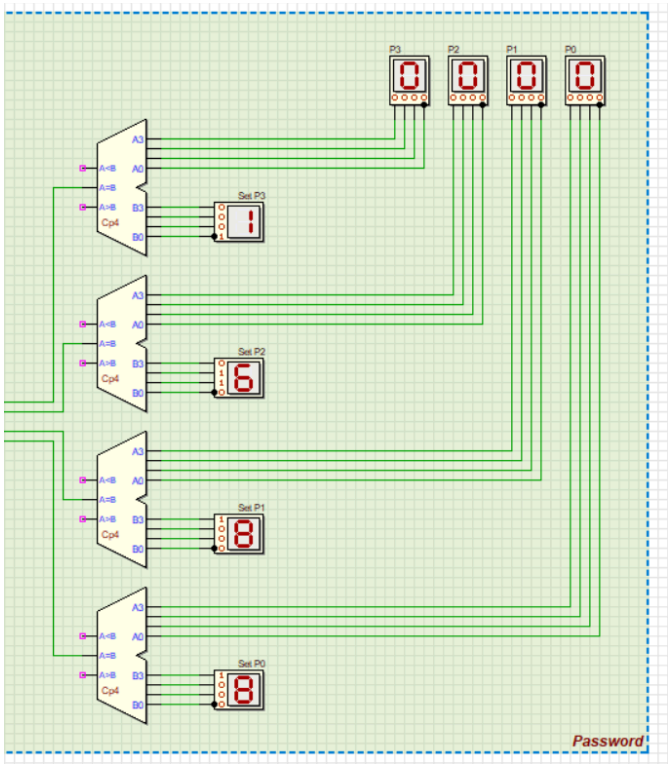


Figure 5.2.4 : Four 4-Bit Comparators

- P3 refers to the fourth digit of the password.
- There are three additional comparators to compare the third (P2), second (P1), and first (P0) digits of the user-entered password.
- Each comparator compares one digit of the user-input password with the corresponding digit of the system-stored password.
- The system determines if the entire password is correct only when all four comparators output A = B (1) simultaneously.

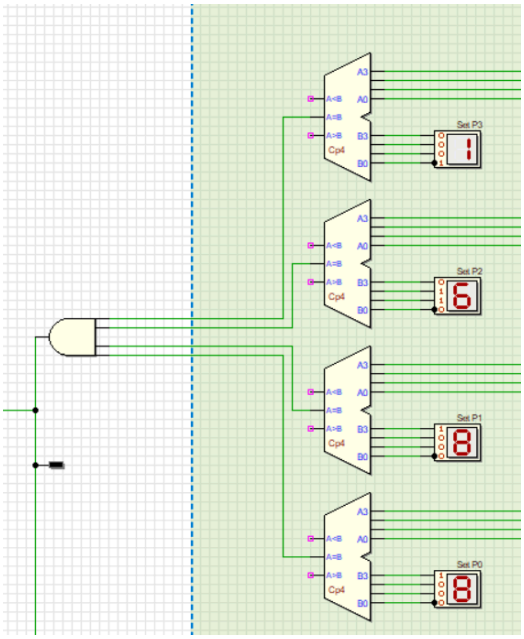


Figure 5.2.5 : AND Gate & Four 4-Bit Comparators

- The A = B outputs from each digit comparison are sent to a 4-input AND gate on the left.
- The AND gate outputs 1 only when all four A = B outputs are 1, meaning the entire password is correct.
- If any digit does not match, the AND gate outputs 0, indicating an incorrect password.
- The AND gate's output is used in subsequent system functions.

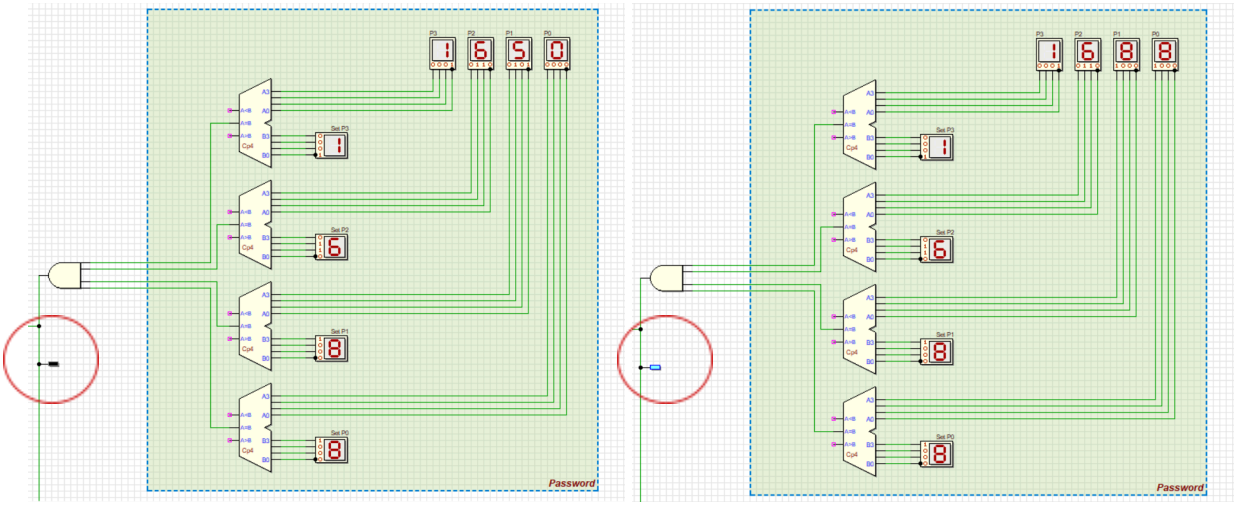


Figure 5.2.6 : Status Test LED (Password Correct / Password Incorrect)

- The AND gate outputs 1 when all digits match (password correct).
- The test LED turns blue (on) when receiving 1 (password correct).
- If any digit is incorrect, the AND gate outputs 0 and the LED stays off (black).

2. MUX & DEMUX :

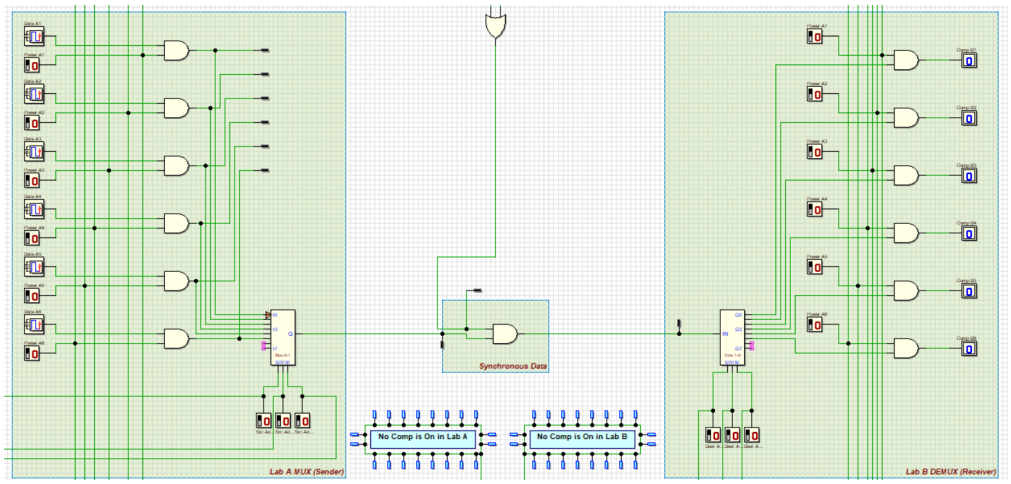


Figure 5.2.7 : Lab A MUX (Sender) & Lab B DEMUX (Receiver)

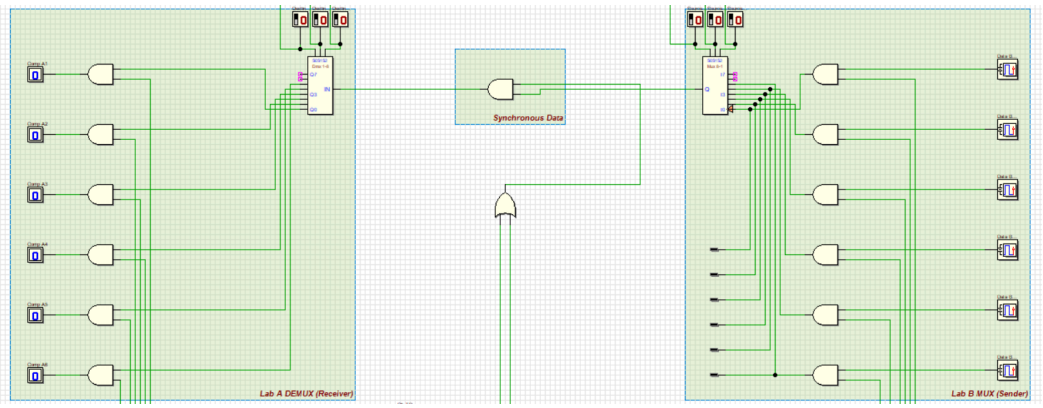


Figure 5.2.8 : Lab A DEMUX (Receiver) & Lab B MUX (Sender)

- MUX and DEMUX play important roles in the transmission of the data between two devices in different labs. Our system consists of two MUX and two DEMUX which is the MUX from Lab A and Lab B and DEMUX from Lab A and Lab B. This allows the data not only to transmit from Lab A to Lab B only but it also allows the data to be transmitted from Lab B to Lab A.
- The power-on button for the Lab A computer is located inside the Lab A MUX.

- The power-on button for the Lab B computer is located inside the Lab B DEMUX.
- Apart from this difference, the circuit structure of:
    - Lab A MUX and Lab B MUX, and
    - Lab A DEMUX and Lab B DEMUX

    is almost identical.

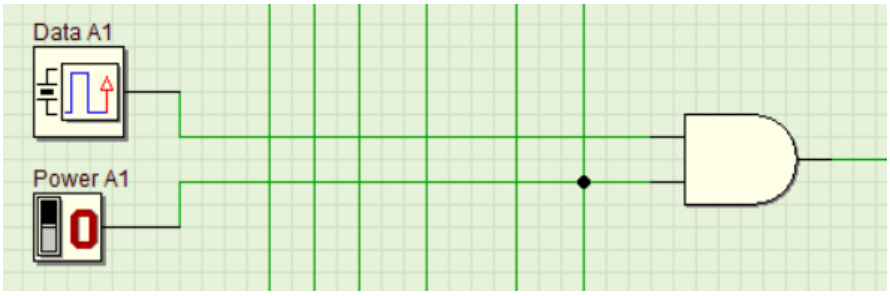


Figure 5.2.9 : Power On Button & Clock Data Transmission Lab A MUX

- The MUX section in Lab A consists of several smaller components, as shown in Figure 5.2.9.
- Data A1 represents a clock signal, which serves as the source data from computer 1 in Lab A.
- Power A1 is the power-on button for computer 1 in Lab A. When Power A1 is 1, computer 1 is powered on. When Power A1 is 0, computer 1 is powered off.
- Both Data A1 and Power A1 inputs are connected to a 2-input AND gate on the right side, which generates an output of 1 when data is transmitted from computer 1 in Lab A, and the power of computer 1 is on.

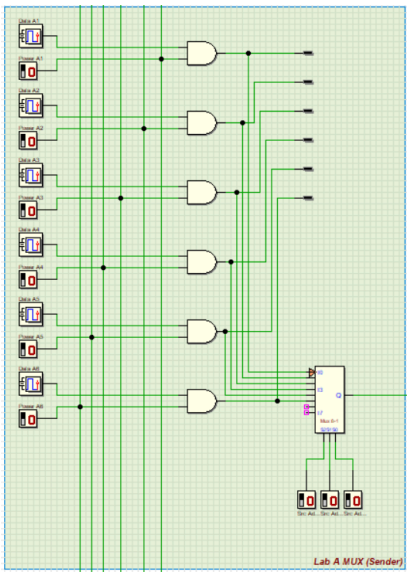


Figure 5.2.10 : Lab A MUX

- Since we have six computers inside Lab A, we will have set six of the same parts as Figure 5.2.10 in the Lab A MUX part.
- All of the output from the AND Gate of each computer is connected to the corresponding input (I0 until I5) at 1 x 8 MUX below.

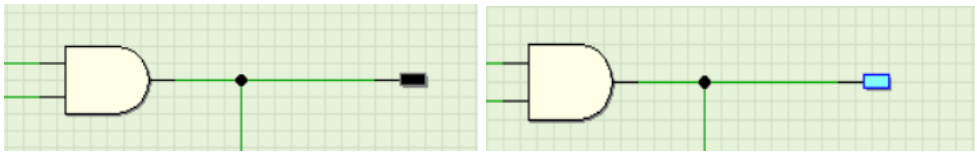


Figure 5.2.11 : Changing of Output of AND Gate

- The output from the AND gate is also connected to a test LED, allowing users to visually observe changes in the output.

- The test LED should flash at a fixed frequency whenever the corresponding Data (Clock) and Power signals are active.
- Each time the test LED flashes, it indicates that a bit of data has been successfully output from its corresponding computer.

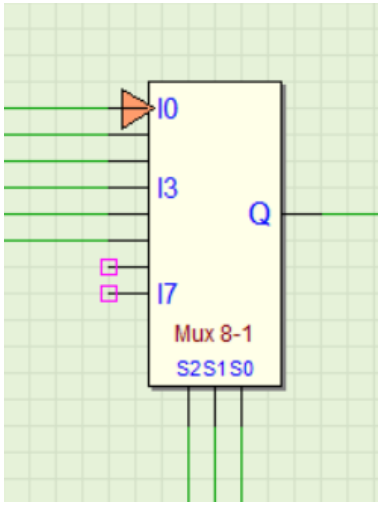


Figure 5.2.12 : Lab A 1 x 8 MUX

- Since there are only six computers in Lab A, the inputs I6 and I7 of the MUX are not connected to any output.
- S0 to S2 are the data select inputs of the MUX. These inputs determine which of the inputs from I0 to I7 will be selected and output to Q.
- The inputs for S0 to S2 in the Lab A 1x8 MUX are provided by the user through manual key-in inputs, using a 3-bit adder.

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	X	X	X
1	1	1	X	X	X	X	X	X	X	X	X

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 (Not Exist / Do Not Care)
1	1	1	I7	8th Computer (Not Exist / Do Not Care)

- Data Input (I0 until I7) will become 1 if its corresponding computer is powered on and ready to transmit the data (Clock of Source Data Active). Otherwise, the Data Input (I0 until I7) of the corresponding computer will become 0.
- When the selected computer in Data Select (S0 until S2) is active and ready to transmit the data (Data Input of Corresponding Computer = 1), the output Q will become 1 which means the data will be transmitted from the selected computer to another computer through MUX.

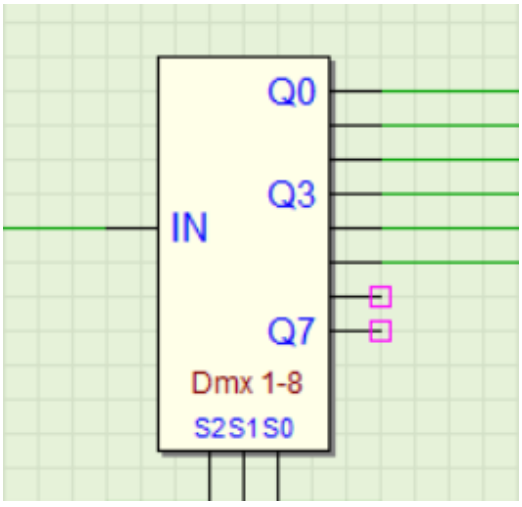


Figure 5.2.13 : Lab B 1 x 8 DEMUX

- The input (IN) of 1 x 8 DEMUX at Lab B will receive the output Q from 1 x 8 MUX at Lab A. Since we also only have six computers at Lab B, the output Q6 and Q7 are not connected to any input.
- S0 until S2 is the data select of the DEMUX. It will determine which output from Q0 until Q7 will be chosen to output the data received at input (IN). Input of S0 until S2 for Lab B 1 x 8 DEMUX are provided by the user through manual key-in inputs, using a 3-bit adder.

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 (Not Exist / Do Not Care)
1	1	1	Q7	8th Computer (Not Exist / Do Not Care)

- When there is an input (IN) which is the data accepted from Lab A MUX, the DEMUX will assign the input (IN) to the corresponding output (Q0 until Q7) based on the selected computer in Data Select (S0 until S2).

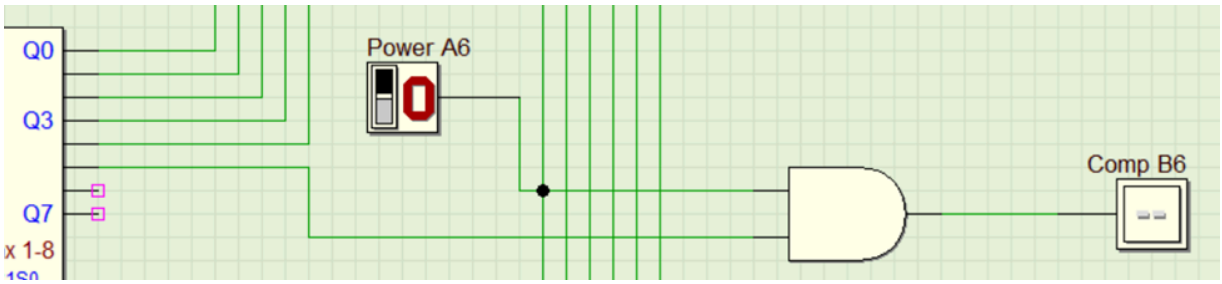


Figure 5.2.14 : Power On Button & Data Output Lab B DEMUX

- The DEMUX section in Lab B consists of several smaller components, as shown in Figure 5.2.14.
- Power A6 is the power-on button for computer 6 in Lab B. When Power A6 is 1, computer 6 is powered on. When Power A6 is 0, computer 6 is powered off.
- Both the Power A6 input and the output Q5 from the DEMUX are connected to a 2-input AND gate on the right side, which generates an output of 1 when data is output from Q5 of the DEMUX, and the power of computer 6 in Lab B is on.

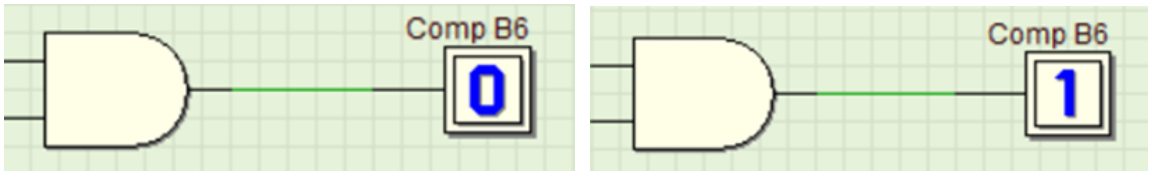


Figure 5.2.15 : Changing of Output AND Gate

- The output from the AND Gate connects with a 1-bit display (Comp B6) which can allow users to see the change of the output visually.

- The 1-bit display should change from 0 to 1 when the computer 6 successfully receives a bit of data and it will change back from 1 to 0 again immediately once the data is received.
- This process will be repeated many times during the data transmission process.

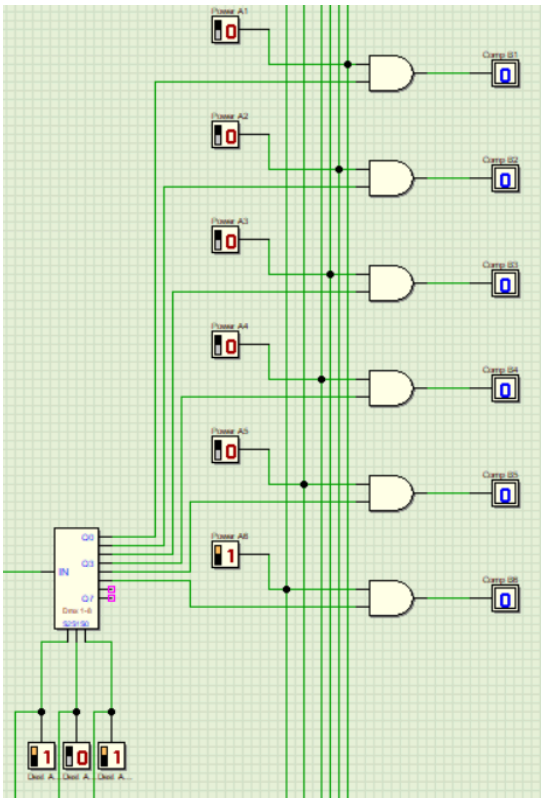


Figure 5.2.16 : Lab B DEMUX

- Since we have six computers also inside Lab B, we will have set six of the same parts as Figure 5.2.16 in the Lab B DEMUX part.
- All of the second input of AND Gate of each computer is connected to the corresponding output (Q0 until Q5) from 1 x 8 DEMUX at the below.

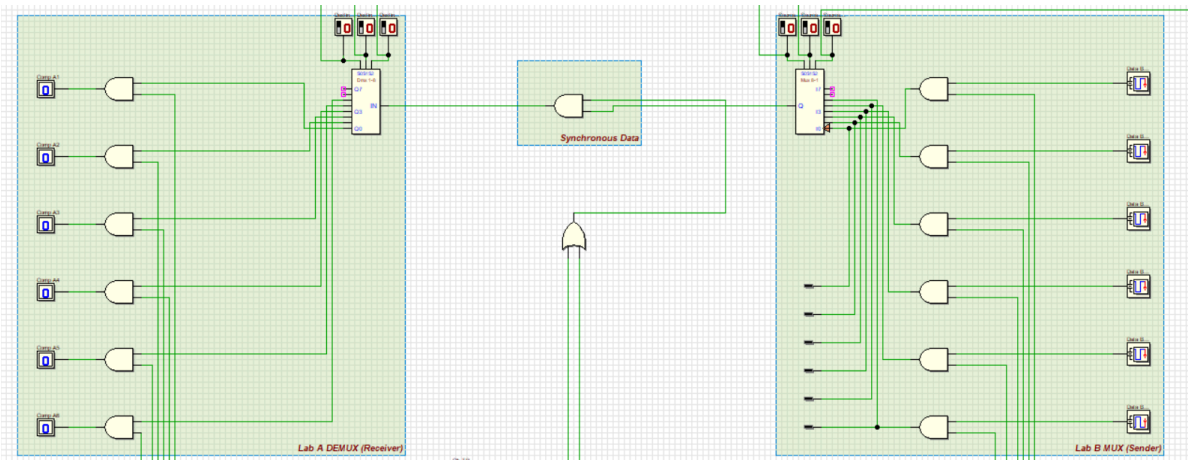


Figure 5.2.17 : Lab A DEMUX (Receiver) & Lab B MUX (Sender)

- Lab A DEMUX and Lab B MUX also apply the same concept as Lab B DEMUX and Lab A MUX. Lab A MUX and Lab B DEMUX allow the system to transmit data from Lab A to Lab B. However, the Lab B MUX and Lab A DEMUX allow the system to transmit data from Lab B to Lab A.

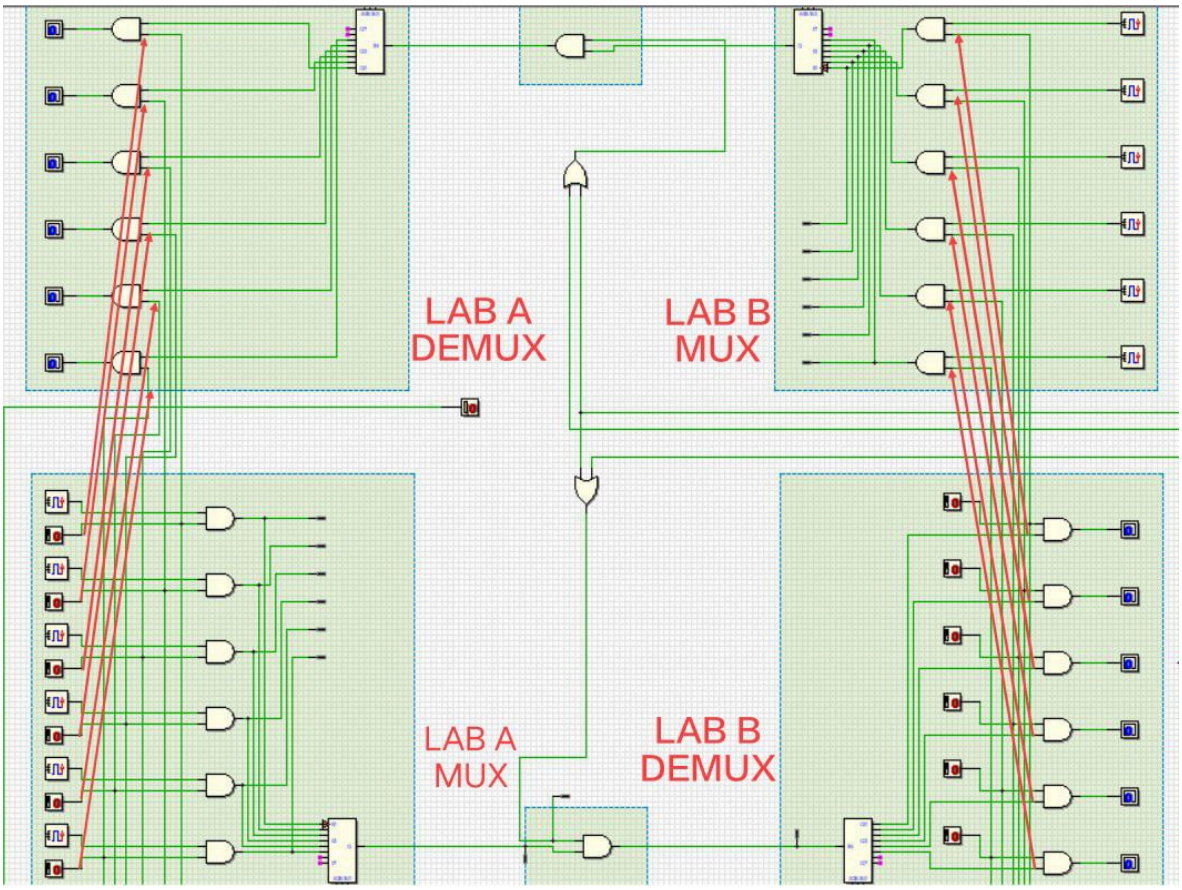


Figure 5.2.18 : Connection of Power On Button

- At the Lab A DEMUX, we did not prepare a power-on button because the power-on button for Lab A's computer is set in the Lab A MUX. We connect the power-on button from the computer at Lab A MUX to the Lab A DEMUX using a wire to ensure that the Lab A DEMUX can also receive the power-on signal for operation.
- At the Lab B MUX, we also did not prepare a power-on button because the power-on button for Lab B's computer is set in the Lab B DEMUX. We connect the power-on button from the computer at Lab B DEMUX to the Lab B MUX using a wire to ensure that the Lab B MUX can also receive the power-on signal for operation.

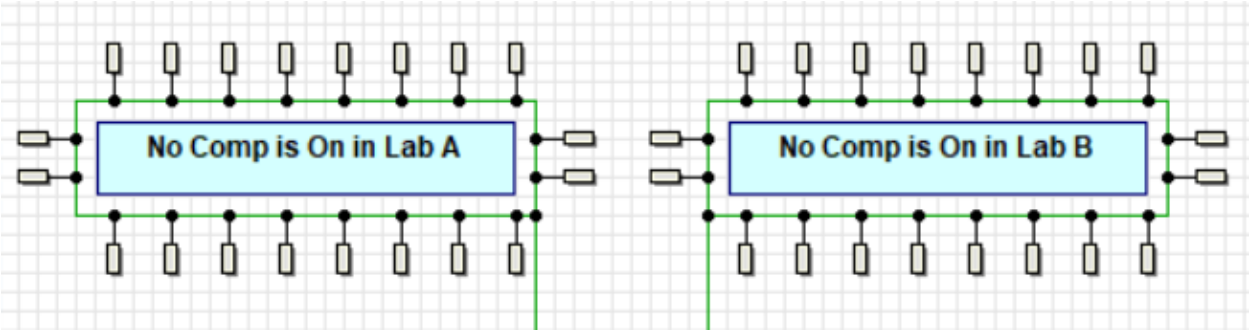


Figure 5.2.19: Fun Display (Detector for Activation of Computer)

- Additionally, all the power-on buttons from Lab A are connected to an 8-input OR gate. The OR gate will output a result of 1 when at least one of its inputs is 1, indicating that at least one computer in Lab A is powered on. If no computers in Lab A are powered on, the OR gate will output 0.
- Since we only have 6 computers in Lab A, we can only connect 6 inputs to the OR gate. However, we have chosen not to use a 6-input OR gate in our circuit, as the system is designed to allow future expansion up to 8 computers. Therefore, we have selected an 8-input OR gate to accommodate potential future growth.
- The remaining 2 inputs of the OR gate are left empty, and we connect both of these inputs to 0. This will not affect the operation of the OR gate.
- The output from the OR gate is connected to a NOT gate positioned above it, which inverts the input it receives.
- When at least one computer in Lab A is powered on, the OR gate will output 1, which is then inverted by the NOT gate to produce 0.
- When no computer in Lab A is powered on, the OR gate will output 0, which is inverted by the NOT gate to produce 1.

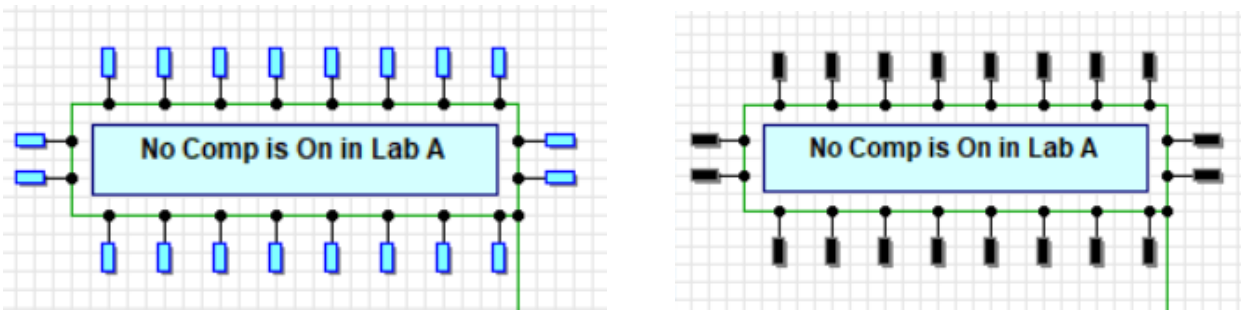


Figure 5.2.20: Status Test LED (At Least One Computer On / No Computer On)

- The output of the NOT gate is connected to the test LED positioned above it.
- The test LED will turn black (off) when receiving input 0, indicating that at least one computer in Lab A is powered on.
- The test LED will turn blue (on) when receiving input 1, indicating that no computers in Lab A are powered on.
- The same concept is applied on the Lab B side to build a similar display, detecting whether any computer in Lab B is powered on or not.

3. <u>Input:</u>

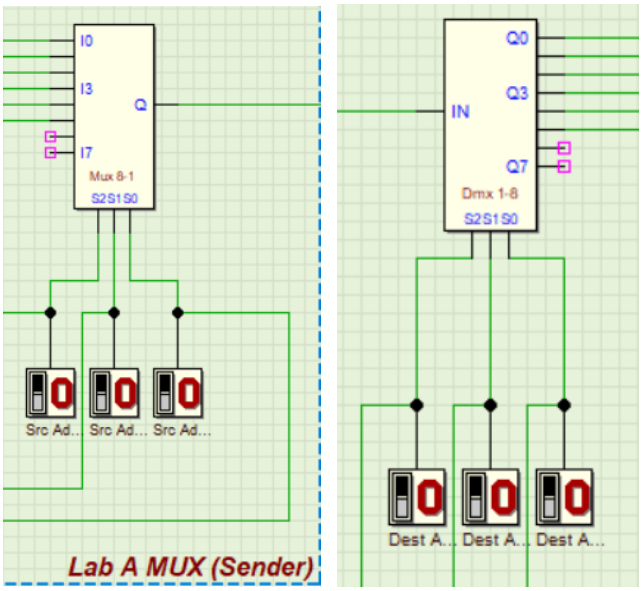


Figure 5.2.21 : Lab A Mux Input & Lab B Demux Input

- The system utilizes a 3-bit input through an adder for communication between Lab A (Sender) and Lab B (Receiver). The multiplexer (MUX) at Lab A is responsible for selecting and transmitting data, while the demultiplexer (DEMUX) at Lab B ensures the correct destination receives it.
- The user inputs a 3-bit value (000 to 101 in binary, equivalent to 0–5 in decimal).
- This value corresponds to one of the six available computers in Lab B.
- If the user selects a value greater than 5 (110 or 111 in binary, equivalent to 6 or 7 in decimal), the data transmission is blocked.
- The blocking mechanism is implemented using an AND gate, ensuring only values within the range (0–5) are processed.
- This setup ensures controlled communication between computers based on specific bit-addressed selection.

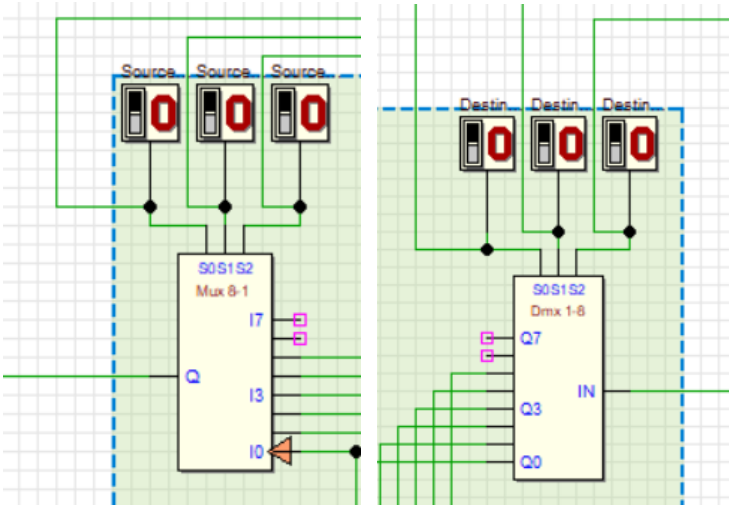


Figure 5.2.22 : Lab B Mux Input & Lab A Demux Input

Lab B → Lab A:

- Lab B also has its own MUX to send data back to Lab A.
- Lab A has a DEMUX to receive and process the data.

- Each Lab has its own 3-bit input that selects which computer will receive the data.
- The adder ensures that data is processed and directed correctly at both ends.
- The system also operates in Full Duplex Mode, meaning that communication between Lab A and Lab B happens simultaneously in both directions. This means that while Lab A (Sender) transmits data to Lab B (Receiver), Lab B (Sender) can also send data back to Lab A (Receiver) at the same time.

4. Max Data Comparator :

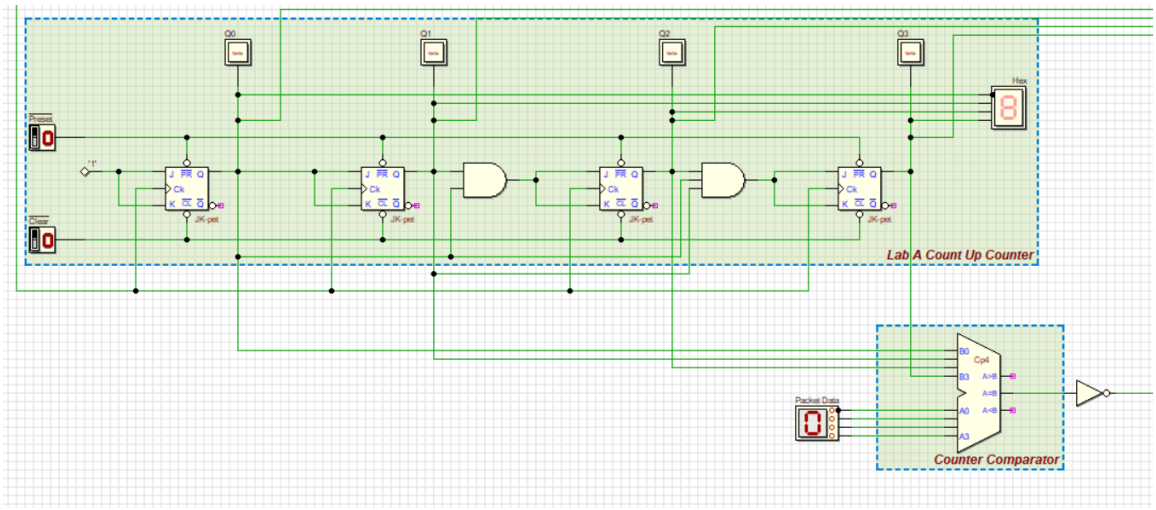


Figure 5.2.23 : Lab A Max Data Comparator

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

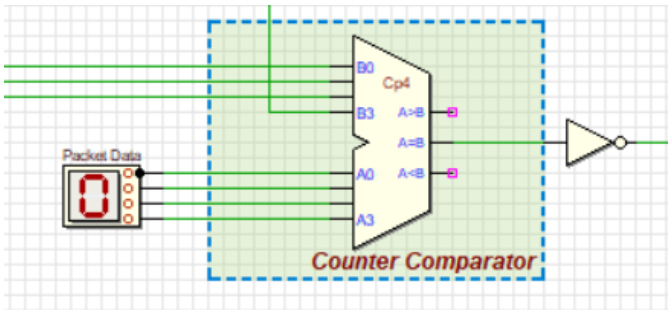


Figure 5.2.24 : Predefined Maximum Data Lab A

- The maximum data that a computer in Lab A can accept is defined by the users through a 4-bit hex digit input (Max Data).
- The input from the 4-bit hex digit (Max Data) is connected to the comparator's input pins A0 to A3.
- The comparator's input pins B0 to B3 are connected to the output of a counter that counts the total data received by the computer in Lab A.
- The comparator compares the actual received data (B0 to B3) with the predefined max data (A0 to A3) to determine which output should be set to 1.
- If the actual received data (B0 to B3) reaches the predefined max data (A0 to A3), the comparator's A = B output will be set to 1, signaling that the system must block further data transmission.
- If the user changes the predefined max data to a lower value after some data has already been transmitted, the comparator's A < B output will be set to 1, indicating an error condition.
- The A > B output is ignored because the system does not need to take any action when the predefined max data is larger than the actual received data.

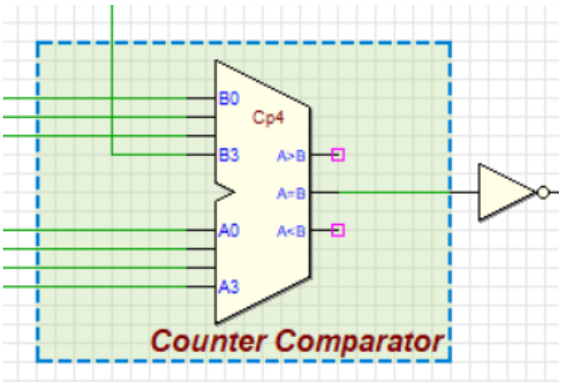


Figure 5.2.25: Clock Enabler Blocker

- The output of A = B and A < B from the comparator is connected to a NOT gate.
- The NOT gate will output 1 when the input is 0, meaning that the actual received data from a computer in Lab A is either not equal to or not greater than the predefined max data.
- If the input to the NOT gate is 1, it will output 0, meaning the data from Lab A is equal to or greater than the predefined max.
- The output of this NOT gate will be used in later functions for further processing.

5. Clock Enabler :

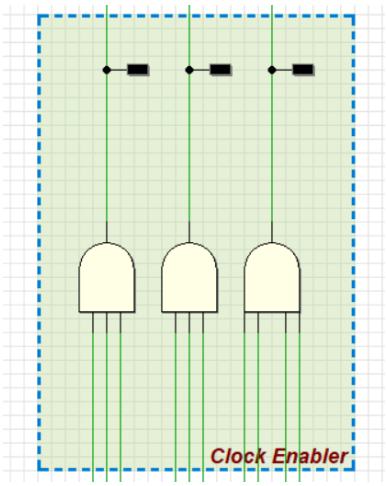


Figure 5.2.26 : Clock Enabler

- There are three AND Gates inside the Clock Enabler.
- **Left :** 3 input AND Gate which allow the system to transmit the data from Lab A to Lab B.
- **Center :** 3 input AND Gate which allow the system to transmit the data from Lab B to Lab A.
- **Right :** 4 input AND Gate which allow the system to transmit the data from Lab A to Lab B and Lab B to Lab A (Full Duplex Mode).

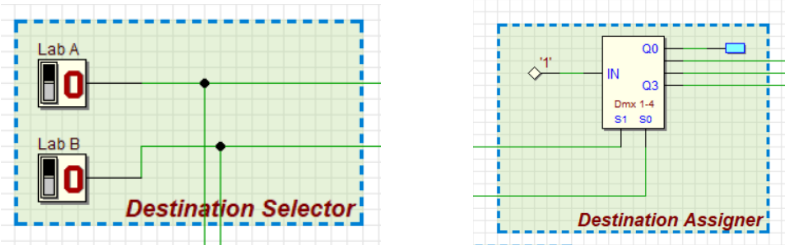


Figure 5.2.27 : Destination Selector & Destination Assigner

- The Destination Selector and Destination Assigner allow the Clock Enabler to assign the destination of the data transmitted.

Table 5.2.5 : Truth Table Destination Assigner

Input			Output				Other
Data Input	Data Select		Data Output				
	Lab A	Lab B					
IN	S1	S0	Q3	Q2	Q1	Q0	Mode
1	0	0	X	X	X	X	X
1	0	1	0	0	1	X	A to B
1	1	0	0	1	0	X	B to A
1	1	1	1	0	0	X	Full Duplex

X = Do Not Care

- When only Lab B input is 1, the DEMUX in the Destination Assigner outputs 1 from the data input (IN) to Q1, which connects to the AND gate on the left of the Clock Enabler.
- When only Lab A input is 1, the DEMUX in the Destination Assigner outputs 1 from the data input (IN) to Q2, which connects to the AND gate at the center of the Clock Enabler.
- When both Lab A and Lab B inputs are 1, the DEMUX does not have a dedicated Q3 output. This case may need to be handled separately or may result in an undefined state.
- When both Lab A and Lab B inputs are 0, the DEMUX outputs 1 from the data input (IN) to Q0, which is an invalid state and does not connect to any gate.

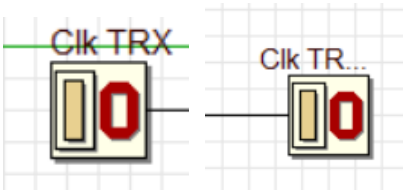
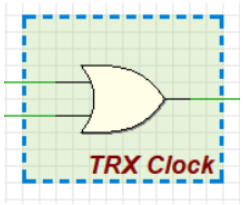


Figure 5.2.28 : TRX_CLK

- We had set two TRX_CLK (Pulse Trigger Button) in our circuit which is with the Start button and beside Lab A DEMUX.
- These two TRX_CLK hold the same function in our system which is pushed to transmit a bit of data to the destination.
- If we only provide one button under Lab A MUX, the user is not able to watch the transmission process of Lab B to Lab A when they push the button. Thus, we also put a button beside Lab A DEMUX to provide a good view to the users when running different modes of transmission.



-

Figure 5.2.29 : Both TRX Clock & OR Gate

- Both TRX_CLK signals are intended to connect to all AND gates inside the Clock Enabler.
- Before doing so, both TRX_CLK signals are first connected to a 2-input OR gate outside the Clock Enabler.
- The OR gate outputs 1 if at least one of its inputs is 1.

- This ensures that regardless of which TRX_CLK is activated in the circuit, the OR gate will always output 1 and send it to the Clock Enabler.

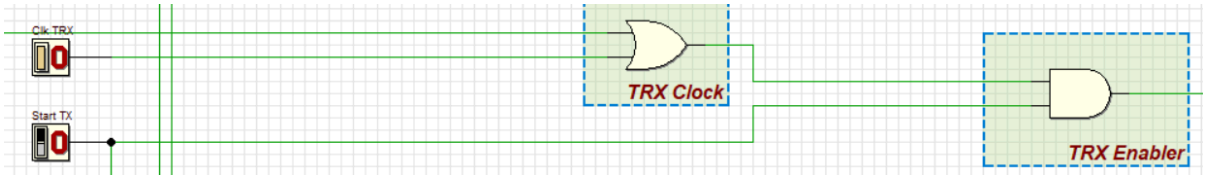


Figure 5.2.30 : TRX Clock & TRX Enabler

- The TRX Enabler is set up with a 2-input AND gate to control whether the TRX Clock is enabled or not.
- When the StartTRX input (on the left) is set to 1 and the TRX_CLK is activated, both inputs of the AND gate in the TRX Enabler become 1.
- This allows the TRX Clock signal to pass through the TRX Enabler and reach the Clock Enabler.
- If the StartTRX input is 0, the TRX Clock signal will be blocked and will not pass to the Clock Enabler.

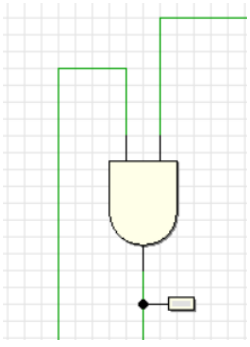


Figure 5.2.31: Password & TRX Enabler & AND Gate

- The output from the Password Comparator and TRX Enabler is connected to a 2-input AND gate.
- The AND gate will output 1 only when both inputs are 1, meaning the password is correct and the TRX Enabler allows the TRX Clock pulse. Otherwise, it will output 0.
- The output of this AND gate is connected to all AND gates inside the Clock Enabler.
- The correctness of the password and whether the TRX Enabler allows the TRX Clock pulse directly impact the operation of the Clock Enabler.

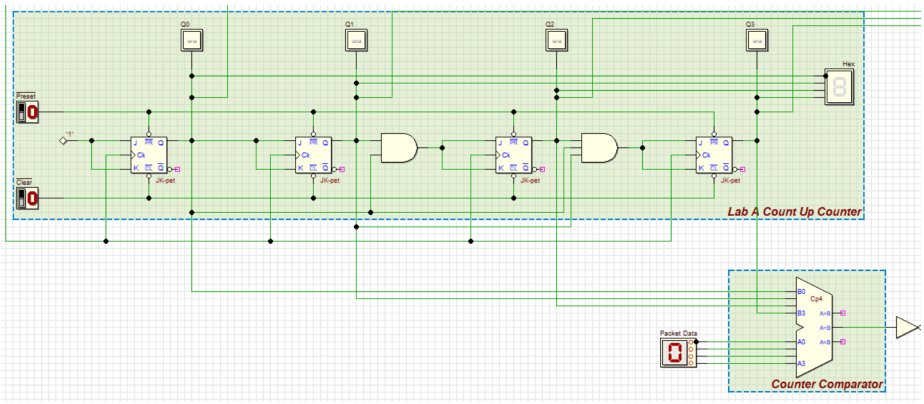


Figure 5.2.32 : Lab A Max Data Comparator & Clock Enabler

- The output of the input NOT Gate in the Lab A Max Data Comparator will connect to the AND Gate at Clock Enabler at center and right side because this is a comparator of the total data received by the computer at Lab A.

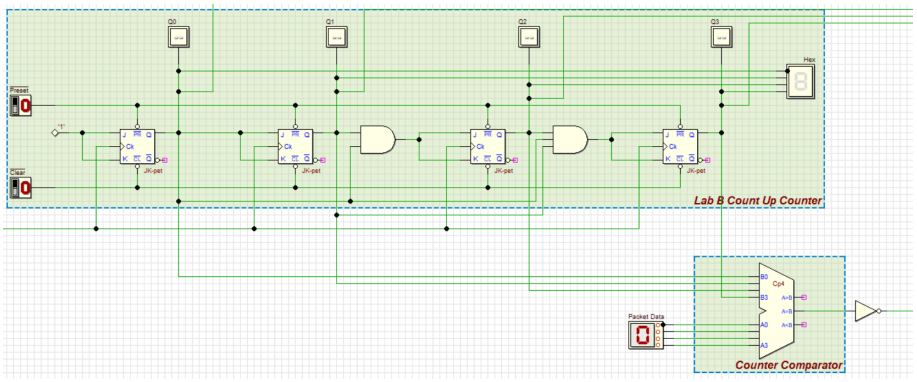


Figure 5.2.33: Lab B Max Data Comparator & Clock Enabler

- The output of the input NOT Gate in the Lab B Max Data Comparator will connect to the AND Gate at Clock Enabler at left side and right side because this is a comparator of the total data received by the computer at Lab B.

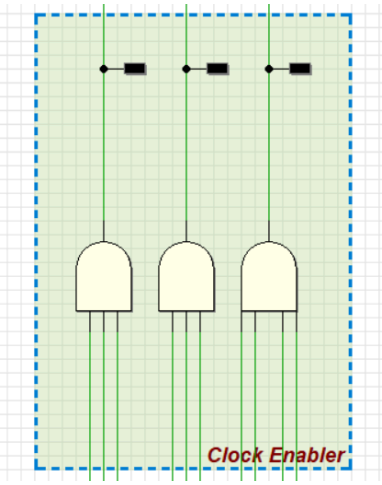


Figure 5.2.34 : Clock Enabler

Table 5.2.6 : Summary of Condition to Active the Each AND Gate in Clock Enabler

Requirement	**AND Gate**		
	Left (Lab A to Lab B)	**Center (Lab B to Lab A)**	**Right (Full Duplex)**
Destination Selector	Lab A (= 0) Lab B (= 1)	Lab A (= 1) Lab B (= 0)	Lab A (= 1) Lab B (= 1)
TRX_CLK	Pushed (= 1)	Pushed (= 1)	Pushed (= 1)
TRX Enabler (ReadyTRX)	Enable (= 1)	Enable (= 1)	Enable (= 1)

Password	Correct	Correct	Correct
Lab A Sender	No invalid input exists. Valid computer is selected. Selected computer is powered on.	X	No invalid input exists. Valid computer is selected. Selected computer is powered on.
Lab B Receiver	No invalid input exists. Valid computer is selected. Selected computer is powered on.	X	No invalid input exists. Valid computer is selected. Selected computer is powered on.
Lab A Receiver	X	No invalid input exists. Valid computer is selected. Selected computer is powered on.	No invalid input exists. Valid computer is selected. Selected computer is powered on.
Lab B Sender	X	No invalid input exists. Valid computer is selected. Selected computer is powered on.	No invalid input exists. Valid computer is selected. Selected computer is powered on.

Lab A Max Data Comparator	X	Actuals receive data at Lab A less or equal than predefined max data.	Actuals receive data at Lab A less or equal than predefined max data.
Lab B Max Data Comparator	Actuals receive data at Lab B less or equal than predefined max data.	X	Actuals receive data at Lab B less or equal than predefined max data.

X = Do Not Care

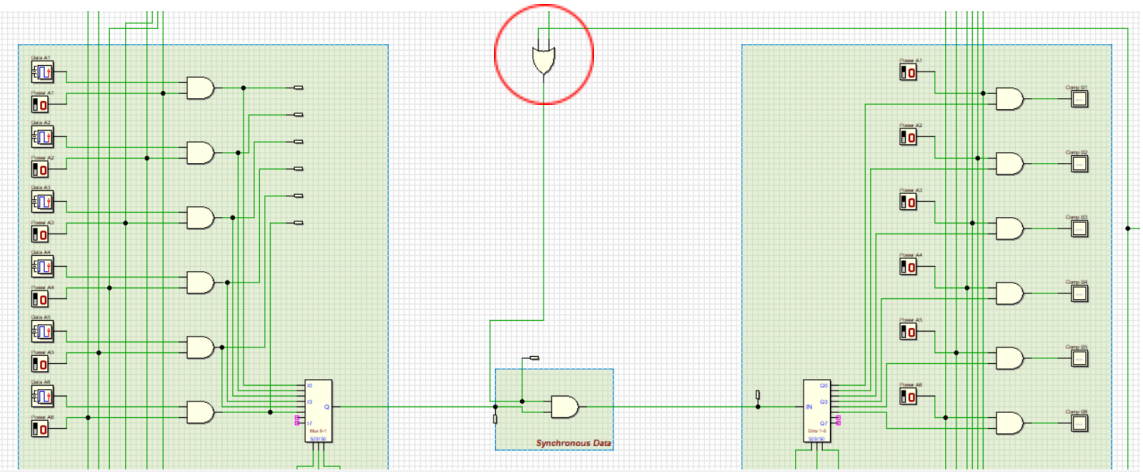


Figure 5.2.35 : Lab A to Lab B Mode & Full Duplex Mode

- The input of the 2-input OR gate above the Lab B DEMUX is connected to the output of the left-side AND gate and the right-side AND gate in the Clock Enabler.
- The OR gate outputs 1 when at least one of its inputs is 1, indicating that the system is operating in either Lab A to Lab B Mode or Full Duplex Mode.
- Otherwise, if both inputs are 0, the OR gate outputs 0.

- The output of the OR gate is connected to the Synchronous Data part between the Lab A MUX and Lab B DEMUX.

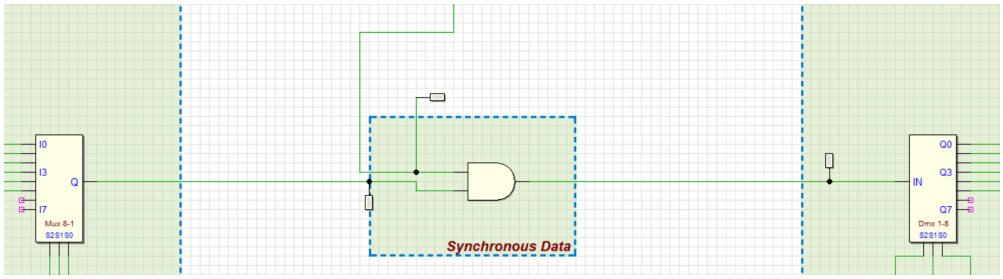


Figure 5.2.36 : Lab A MUX to Lab B DEMUX

- The Lab A MUX can transmit data to the Lab B DEMUX only when the system is in Lab A to Lab B Mode or Full Duplex Mode.
- In these two modes, one of the inputs of the 2-input AND gate inside the Synchronous Data section must be 1, allowing data transmission from Lab A to Lab B.
- If the system is not in these modes, one of the inputs of the 2-input AND gate inside the Synchronous Data section will be 0.
- As a result, the output of this AND gate will always be 0, preventing data transmission.

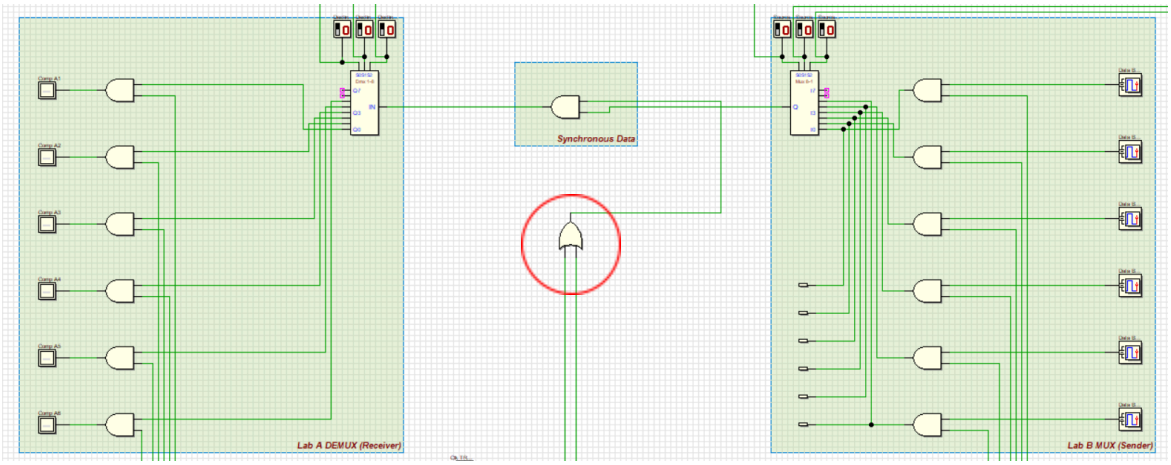


Figure 5.2.37 : Lab B to Lab A Mode & Full Duplex Mode

- The input of the 2-input OR gate above the Lab B MUX is connected to the output of the center-side AND gate and the right-side AND gate in the Clock Enabler.
- The OR gate outputs 1 when at least one of its inputs is 1, indicating that the system is operating in either Lab B to Lab A Mode or Full Duplex Mode.
- Otherwise, if both inputs are 0, the OR gate outputs 0.
- The output of the OR gate is connected to the Synchronous Data part between the Lab A DEMUX and Lab B MUX.

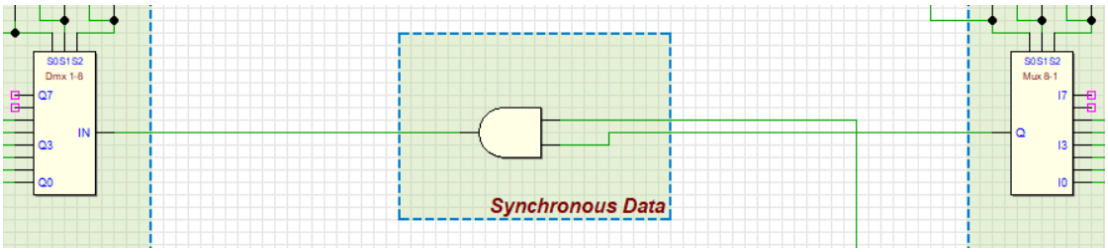


Figure 5.2.38 : Lab B MUX to Lab A DEMUX

- The Lab B MUX can only transmit data to the Lab A DEMUX when the system is in either Lab B to Lab A Mode or Full Duplex Mode.
- In these two modes, one of the inputs of the 2-input AND gate inside the Synchronous Data section must be 1 to allow data transmission from Lab B to Lab A.
- If the system is not in these modes, one of the inputs of the 2-input AND gate inside the Synchronous Data section will be 0.
- As a result, the output of the AND gate will always be 0, preventing data transmission.

6. Counter :

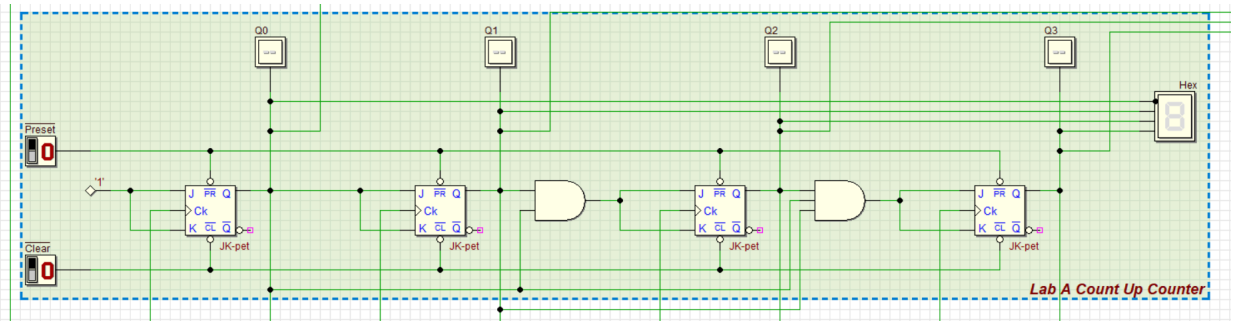


Figure 5.2.39: Lab A Counter

- 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.

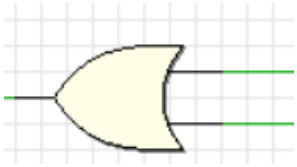


Figure 5.2.40 : Lab B to Lab A Mode & Full Duplex Mode

- The operation of the counter depends on the clock, so the Clock Enable will affect its functioning.
- All of the clock inputs of the JK Flip-Flops inside the counter are connected to the output of a 2-input OR gate.
- The input of this 2-input OR gate is connected to the outputs of the center-side AND gate (Lab B to Lab A Mode) and the right-side AND gate (Full Duplex Mode) at the Clock Enabler.

- This setup means the Lab A Counter will only operate when the system is in either Lab B to Lab A Mode or Full Duplex Mode.

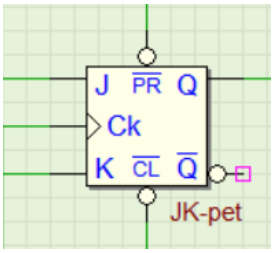


Figure 5.2.41 : Positive Edge Triggered JK Flip-Flop

- The JK Flip-Flop is a type of sequential logic circuit capable of storing binary information in a bitwise manner (GeeksforGeeks, 2024).
- On the left side of the JK Flip-Flop, there are three active-high inputs: Set input (J), Reset input (K), and Clock input (Ck). Inputs J and K can affect both the active-high output Q and Q' on the right side of the JK Flip-Flop.
- In the middle of the JK Flip-Flop, there are two active-low inputs: Preset input (PR') and Clear input (CL'). These inputs control the mode of operation of the JK Flip-Flop, determining whether it operates in synchronous or asynchronous mode.
- In synchronous mode, the output states Q and Q' change in response to the clock signal (Ck).
- In asynchronous mode, the state changes independently of the clock signal (Ck).

Table 5.2.7 : Truth Table JK Flip-Flop (Not Include PR' & CL')

Input			Output		Mode
Clock Input	Set	Reset			
Ck	J	K	Q	Q'	
0	X	X	Q	Q'	No Change
1	0	0	Q	Q'	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	Q'	Q	Toggle

Table 5.2.8 : Truth Table JK Flip-Flop

Input					Output		Mode
Clock Input	Mode Input		Set	Reset			
Ck	PR'	CL'	J	K	Q	Q'	
0	0	0	X	X	X	X	Invalid
0	0	1	X	X	0	0	Asynchronous
0	1	0	X	X	1	1	Asynchronous
0	1	1	X	X	Q	Q'	Synchronous (No Change)

1	0	0	X	X	X	X	Invalid
1	0	1	X	X	0	0	Asynchronous
1	1	0	X	X	1	1	Asynchronous
1	1	1	0	0	Q	Q’	Synchronous (No Change)
1	1	1	0	1	0	1	Synchronous (Reset)
1	1	1	1	0	1	0	Synchronous (Set)
1	1	1	1	1	Q’	Q	Synchronous (Toggle)

X = Do Not Care

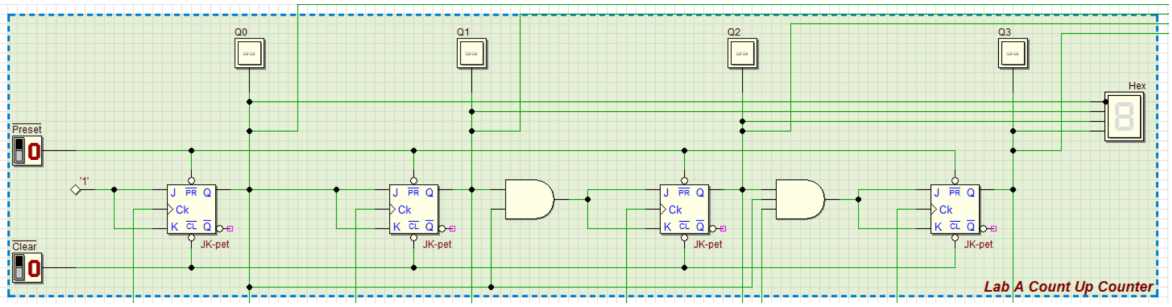


Figure 5.2.42 : JK Flip-Flop & Connection of Clock, Preset and Clear

- Since this is a 4-bit Synchronous Counter, the circuit requires four JK Flip-Flops to store the 4-bit digit.
- All of the Clock Inputs (Ck) of the JK Flip-Flops are connected to the output of the OR gate shown in Figure 5.2.40. The Clock Enabler determines whether the counter can operate based on the conditions outlined in Table 5.2.6.
- All of the **Preset Inputs (PR')** from each JK Flip-Flop are connected to an input labeled **Preset'**.

- All of the **Clear Inputs (CL')** from each JK Flip-Flop are connected to an input labeled **Clear'**.
- The users must set both **Preset Inputs (PR')** and **Clear Inputs (CL')** to 1 before the counter starts operating to change the counter's mode to synchronous mode.
- When the counter reaches its maximum value, the user should set the **Clear Input (CL')** to 0 to clear all values inside the counter.
- After this, the user should set the **Clear Input (CL')** back to 1 to allow the counter to return to synchronous mode, enabling the counter to start a new round of counting.

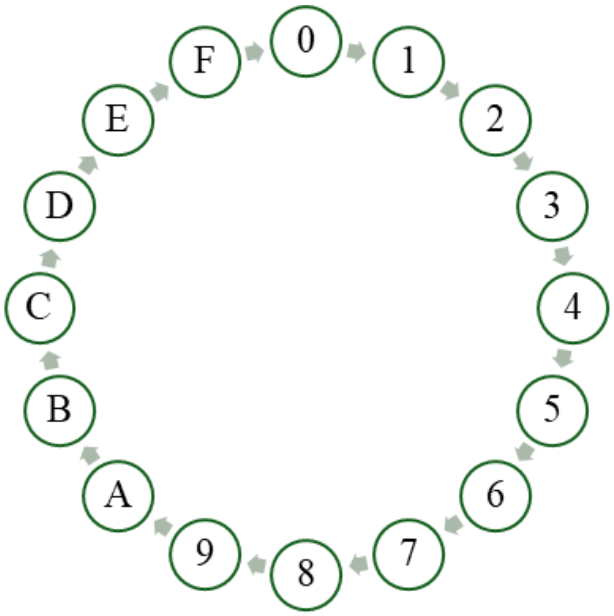


Figure 5.2.43 : State Diagram (Hexadecimal Digit)

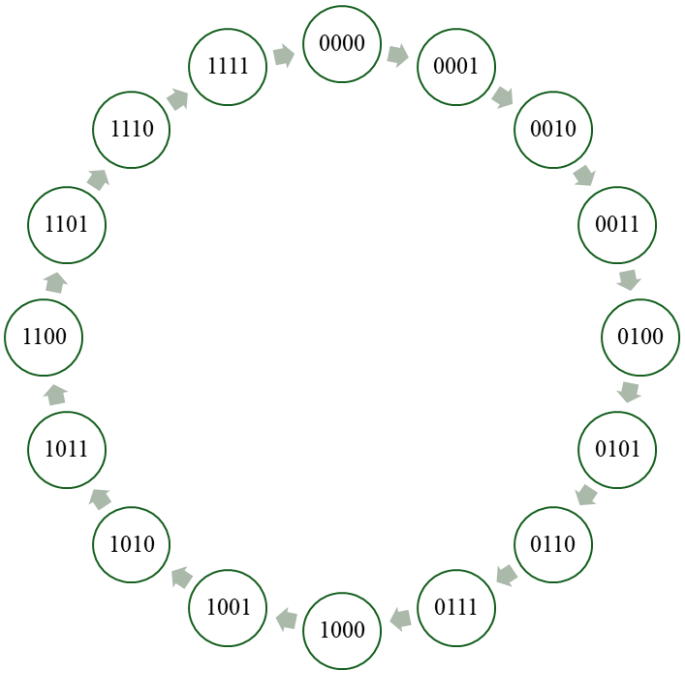


Figure 5.2.44 : State Diagram (Binary Digit)

Table 5.2.9 : Next State Table

Present State				Next State			
Q_3	Q_2	Q_1	Q_0	Q_{3+}	Q_{2+}	Q_{1+}	Q_{0+}
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 5.2.10 : Excitation Table (Original)

Flip-Flop State		**Present State**	**Next State**
J	**K**	**Q_n**	**Q_{n+1}**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0

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

Table 5.2.11 : Excitation Table (Simplified)

Present State	**Next State**	**Flip-Flop State**	
Q_n	**Q_{n+1}**	**J**	**K**
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

X = Do Not Care

Table 5.2.12 : Flip-Flop Transition Table

Present State				**Next State**				**JK Transition**							
Q_3	**Q_2**	**Q_1**	**Q_0**	**Q_{3+}**	**Q_{2+}**	**Q_{1+}**	**Q_{0+}**	**J_3**	**K_3**	**J_2**	**K_2**	**J_1**	**K_1**	**J_0**	**K_0**
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X

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

X = Do Not Care

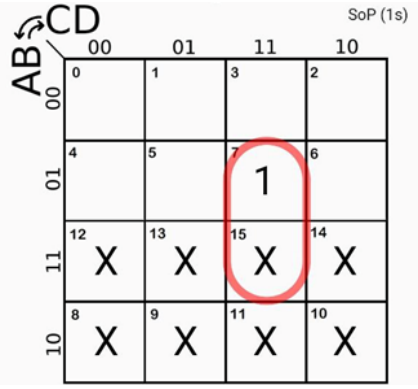
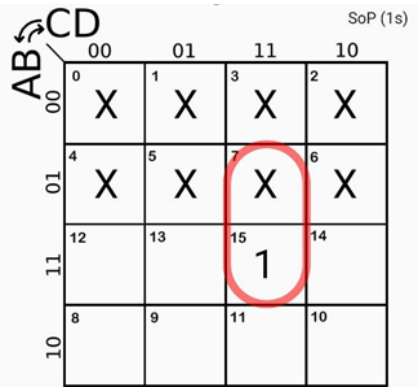
1. Before we start connecting the Set Input (J) and Reset Input (K) of the JK Flip-Flops, we need to follow these steps:

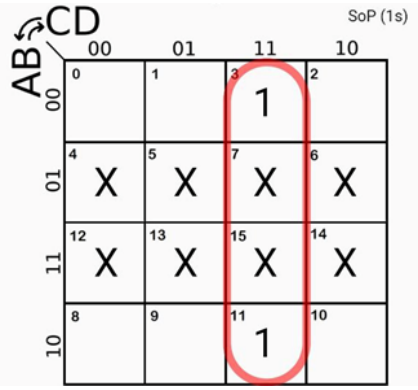
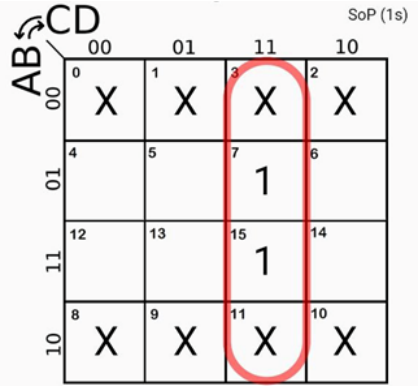
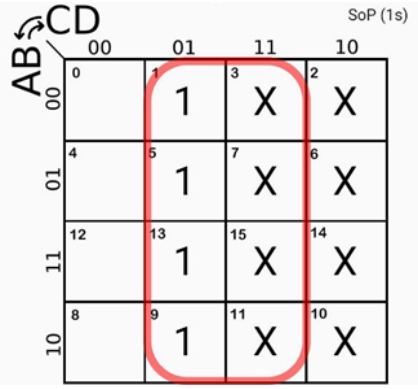
- Determine the State Diagram (Figure 5.2.43 & Figure 5.2.44).

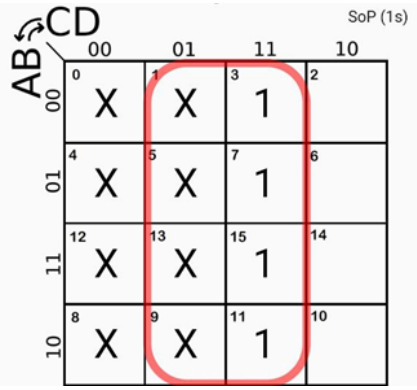
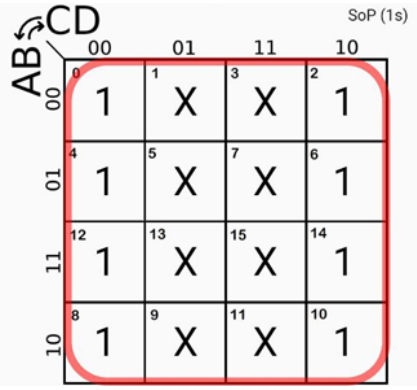
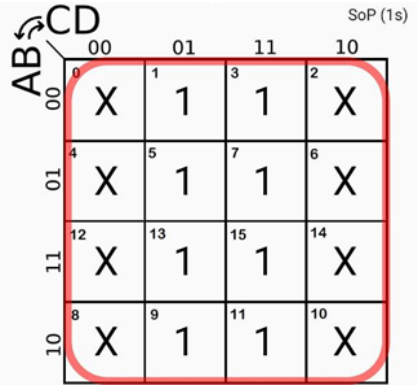
- Determine the Next State Table (Table 5.2.9).
- Determine the Excitation Table of JK Flip-Flops (Table 5.2.10 & Table 5.2.11).
- Determine the Flip-Flop Transition Table (Table 5.2.12).

2. Once we have completed these steps, we need to construct the K-Map using the present state values. This will allow us to obtain the simplified logic equations before proceeding with the circuit connections.

Table 5.2.13 : K-Map

Input	K-Map	Logic Equation
J_3		$J_3 = BCD$ $J_3 = Q_2Q_1Q_0$
K_3		$K_3 = BCD$ $K_3 = Q_2Q_1Q_0$

J_2		$J_2 = CD$ $J_2 = Q_1Q_0$
K_2		$K_2 = CD$ $K_2 = Q_1Q_0$
J_1		$J_1 = D$ $J_1 = Q_0$

K_1		$K_1 = D$ $K_1 = Q_0$
J_0		$J_0 = 1$
K_0		$K_0 = 1$

$Q_3 = A$ / $Q_2 = B$ / $Q_1 = C$ / $Q_0 = D$

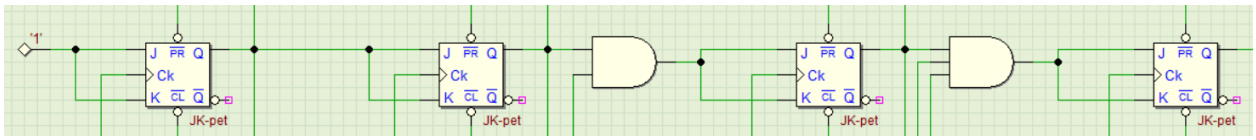


Figure 5.2.45: J & K Input Connection of Lab A Counter

- Since we had constructed the K-Map and logic equation, we can connect all of the Set Input (J) and Preset Input (K) based on the result we get.

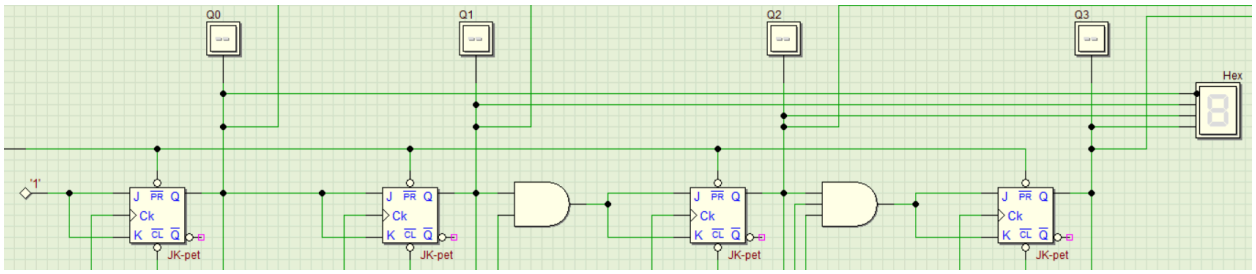


Figure 5.2.46: Q Output & Actual Received Value

- Each Q output from JK Flip-Flop is connected to a 1-bit display.
- Also, 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).

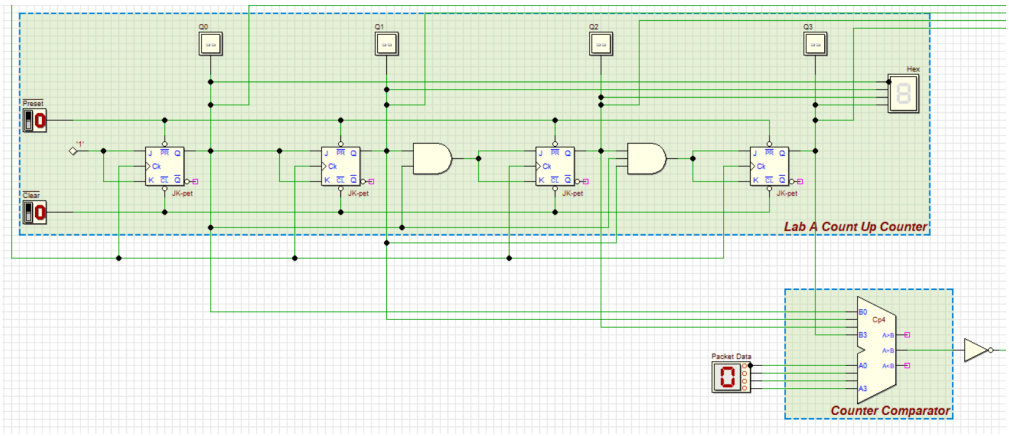


Figure 5.2.47 : Actual Received Data & Predefined Max Data

- All of the output Q from Lab A Counter will also connect to the input of Comparator (B0 until B3) at the Lab A Max Data Comparator.
- The Comparator will compare the value of the data actually received 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 continue or not.

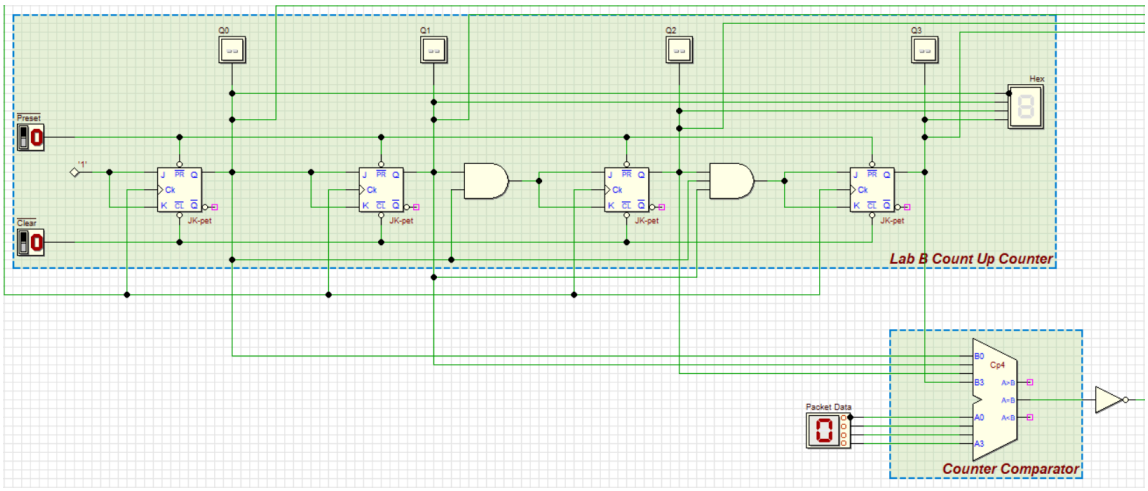


Figure 5.2.48 : Lab B Counter

- All of the concepts that are mentioned in Lab A Counter are the same as the concept 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 JK Flip-Flop is connected to.

7. User Manual :

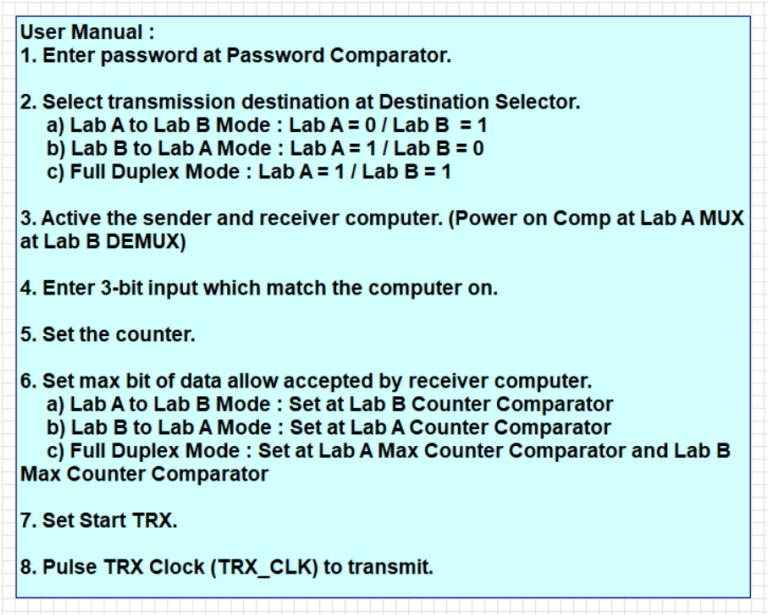


Figure 5.2.49: User Manual

- This user manual serves as a step-by-step guide for users to operate the circuit efficiently. It outlines the essential steps required to establish a secure and successful data transmission between two computers in different modes.
- Password Authentication:

    - Users must enter a password at the Password Comparator to initiate access and ensure secure usage of the circuit.
- Transmission Destination Selection:
    - Users can select their preferred communication mode at the Destination Selector:
        - Lab A to Lab B Mode (One-way communication from A to B)
        - Lab B to Lab A Mode (One-way communication from B to A)
        - Full Duplex Mode (Simultaneous communication between A and B)
- Activating Sender & Receiver Computers:
    - Power must be enabled for both the sender and receiver computers, using the MUX at Lab A and DEMUX at Lab B.
- 3-bit Input Configuration:
    - The user must enter a 3-bit input that matches the operational requirements of the receiver computer.
- Setting the Counter:
    - The counter must be configured to keep track of data transmission.
- Setting Maximum Data Limit:
    - Users must define the maximum data allowance using the Counter Comparator for each mode:
        - Lab A to Lab B Mode → Configure at Lab B Counter Comparator.
        - Lab B to Lab A Mode → Configure at Lab A Counter Comparator.
        - Full Duplex Mode → Configure both Lab A and Lab B Max Counter Comparators to handle bidirectional data flow.
- Starting Transmission (TRX):
    - Once all configurations are complete, the transmission process is initiated by setting Start TRX.
- Pulsing the Transmission Clock (TRX_CLK):
    - The TRX Clock (TRX_CLK) must be pulsed to begin data transmission between the computers.
- This guide ensures that users can properly configure and operate the circuit while minimizing errors and ensuring smooth data communication.

8. Dashboard :

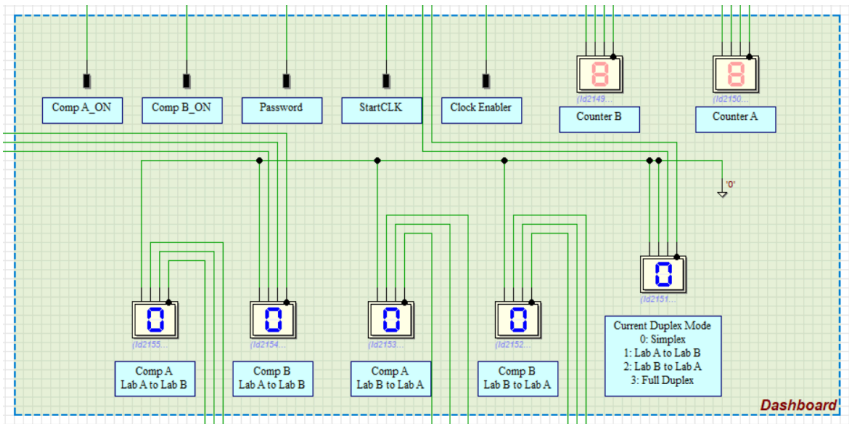


Figure 5.2.50: Dashboard

- The dashboard provides a clear and organized way to monitor the system's status and data flow. Here's a structured breakdown of its functionality:

1. Status Indicators (LEDs)

- Comp A & Comp B ON:
  - LEDs light up when computers A and B are powered on.
- Password Verification LED:
  - Lights up when the correct password is entered, ensuring authentication.
- Start CLK LED:
  - Lights up when the clock signal is active, indicating transmission readiness.
- Clock Enabler LED:
  - Lights up once when the TRX button is pressed, confirming data transmission initiation.

2. Data Monitoring

- Counter Displays (Counter A & Counter B):

- Show bit progress during transmission, indicating which bit is currently being sent/received.

- Base-10 Input Display:
  - Displays input values in decimal format for all computers, improving readability.

3. Transmission Mode Selection

- Current Duplex Mode Display:
  - 0 → Simplex (One-way communication)
  - 1 → Lab A to Lab B (Half-Duplex)
  - 2 → Lab B to Lab A (Half-Duplex)
  - 3 → Full Duplex (Simultaneous bidirectional communication)

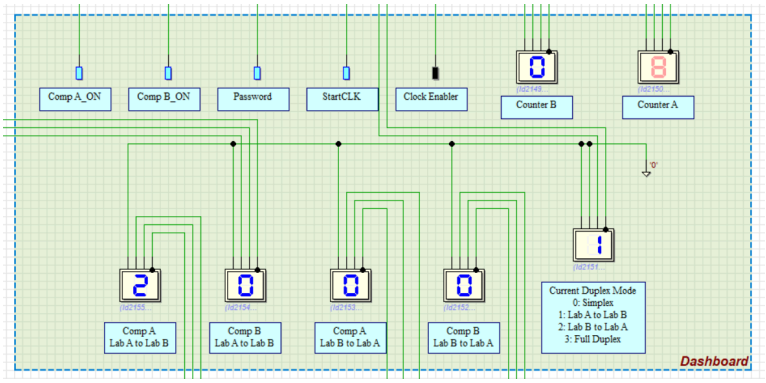


Figure 5.2.51: Example of Dashboard when the Circuit is Functioning

- From the diagram, we can observe that all LEDs are lit, indicating that the corresponding components—Comp A, Comp B, Password, and Start CLK—are active. However, the Clock Enabler LED will only illuminate when the TRX button is pressed, confirming that data transmission has been triggered.
- Additionally, Counter B is initialized and ready to receive data, while Computer A displays "2", showing the data being sent. Computer B shows "0", indicating that it has

not yet received the transmitted data. The Mode Display shows "1", confirming that the system is in Mode 1 (Lab A to Lab B transmission) and is prepared for data transfer.

9. Fun Display:

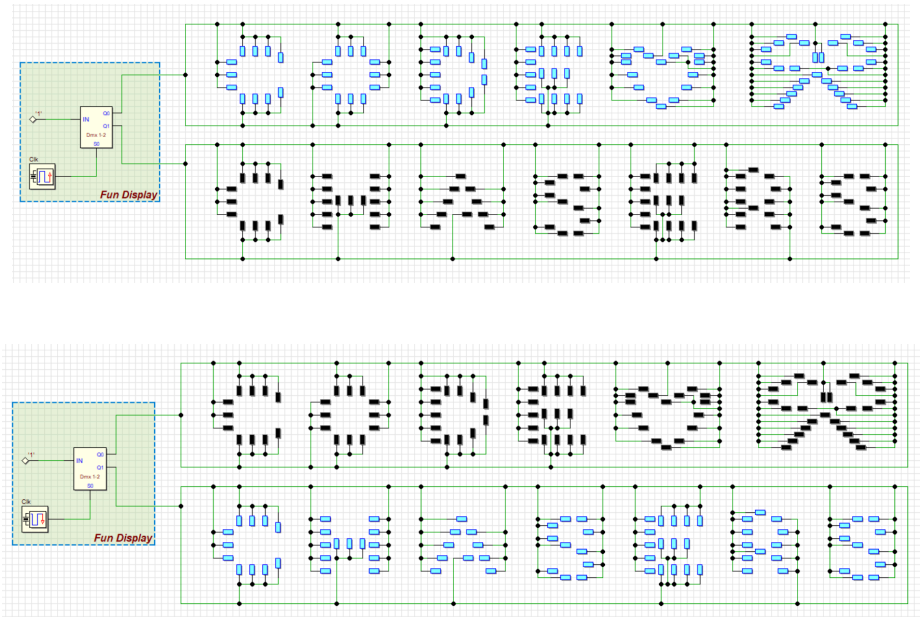


Figure 5.2.52: Fun Display Team Name

- We have used LEDs to display our team name—-Code Chasers with heart and ribbon elements, and the LEDs toggle between ON and OFF (1 and 0) based on clock pulses, creating an animated effect. The circuit includes a clock signal and a demultiplexer (Dmx 1-2) to control which LEDs light up at a given moment.
- This setup makes the circuit more engaging while demonstrating digital logic concepts. It also serves as a visual representation of data flow, reinforcing how bits (0s and 1s) can be used to create meaningful displays.
- When the clock signal is HIGH, the upper row LEDs light up.
- When the clock signal is LOW, the lower row LEDs light up.

6.0 CONCLUSION AND REFLECTION

The development of our 6-computer 4-bit JK flip-flop password system was a valuable learning experience in applying digital logic principles to a real-world scenario. By integrating key components such as the clock signal, TRX enabler, duplex communication modes, and an LED display, we successfully created a system that ensures secure data transmission and user authentication. The clock signal played a crucial role in synchronizing the flip-flops, while the TRX enabler efficiently controlled data flow between computers. Additionally, the LED display provided real-time feedback, improving both functionality and user experience.

One of the project's strengths was the seamless implementation of sequential logic using JK flip-flops to store and manage 4-bit data. The password authentication feature enhanced security, and the dashboard interface enabled users to monitor system operations effectively. Despite encountering challenges such as timing synchronization and data flow management, the system functioned as intended and successfully demonstrated different communication modes, including simplex and full-duplex transmission.

Overall, this project showcased our ability to design, implement, and troubleshoot a complex digital system utilizing MSI circuits and sequential logic. Moving forward, future improvements will focus on enhanced error handling, stronger security mechanisms, and increased scalability to accommodate more computers and data bits. This project serves as a foundation for further advancements in secure digital communication systems.

REFLECTION:

WONG ZI NING

Working on this project was a fantastic learning opportunity that broadened my understanding of digital circuit design and sequential logic. I focused on creating an LED pattern to display a group name using binary logic and demultiplexers. This task helped me improve my skills in controlling outputs and troubleshooting display issues. I explored the practical application of components like multiplexers, JK flip-flops, LEDs, and clock signals, which deepened my knowledge of how they contribute to data transmission. One challenge I faced was aligning the LED pattern by managing logic conditions, which enhanced my problem-solving abilities in circuit design. Through troubleshooting, analyzing circuit behavior, and optimizing designs, I developed critical skills in making informed component choices. This hands-on experience boosted my confidence in digital electronics and sparked my curiosity to explore more advanced circuit design topics.

TAN HUI SHAN

Based on the project, I've improved my skills with the Deeds program, which allows us to construct datasets, cases, and tools while also storing data on a single holistic web platform. The most intriguing aspect of the project was the use of digital logic concepts such as logic gates, flip-flops, and counters, which are required to process network packets for transmission monitoring systems. Overcoming the various challenges faced during the project, such as ensuring synchronization between components and debugging issues related to data capture, was a significant achievement. It deepened my problem-solving skills, particularly in complex systems. One of the weaknesses I faced was time management. While I managed to complete the project, I found that certain tasks, like running the Deeds circuit, took longer than expected. In future projects, I intend to devote more time to testing and troubleshooting to ensure a smoother progression. However, I notice opportunities for improvement, such as increasing my networking expertise and time management skills, which will allow me to approach future tasks with greater efficiency and confidence.

LEE HUI ZHEN

This project successfully created a Network Packet Transmission Monitoring System using digital logic components like flip-flops, multiplexers, and counters to manage and monitor data transfer. Additionally, we implemented advanced features like full-duplex communication, allowing two-way data transfer, a fun display to visually monitor system status, and a password authentication system for security. However, we faced some challenges, including timing issues, complex circuit connections, and limits on system expansion. In the future, we can improve the system by making error detection better, improving timing control, allowing more computers, and adding a simple user interface for easier monitoring. Overall, this project helped us gain valuable skills in designing and troubleshooting digital circuits.

ELEANOR TING PIK EN

This project was a fun and enriching learning experience that enhanced my understanding of digital circuit design and sequential logic. I learned how to build circuits, integrate components, and troubleshoot issues while exploring the role of multiplexers, JK flip-flops, LEDs, and clock signals in data transmission. Additionally, I developed problem-solving skills in managing timing synchronization and signal flow. Beyond the technical aspects, this project helped me develop problem-solving skills as I encountered and resolved challenges related to timing synchronization, data flow, and signal management. I also improved my ability to analyze circuit behavior and make informed decisions about component selection and circuit optimization. Overall, this project strengthened my confidence in digital electronics and inspired me to explore more advanced concepts in circuit design.

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/

GeeksforGeeks. (2024, December 20). *Difference between Multiplexer and Demultiplexer*. GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-multiplexer-and-demultiplexer/

GeeksforGeeks. (2024, September 12). What is JK FlipFlop ? GeeksforGeeks.

https://www.geeksforgeeks.org/what-is-jk-flip-flop/

8.0 APPENDICES

TASK DISTRIBUTION

Name	Task
Eleanor Ting	1. Deeds Circuit Design - Full Duplex Implementation, Dashboard Connection 2. Report - System Implementation, Conclusion,Reflection
Lee Hui Zhen	1. Deeds Circuit Design - Basic Functions, Password, User Manual 2. Report - References, Appendices
Tan Hui Shan	1. Report - Dedication & Acknowledgement, Suggested Solution, Requirement, Reflection
Wong Zi Ning	1. Deeds Circuit Design - Fun Display 2. Report - Introduction, Problem Background, Reflection

GROUP PHOTO

Presentation Slide:

https://docs.google.com/presentation/d/1o45DvrcchVoDPzQiapeQKFaUaEQNvbOYN7KNnl775Jk/edit?usp=sharing

Video recording: