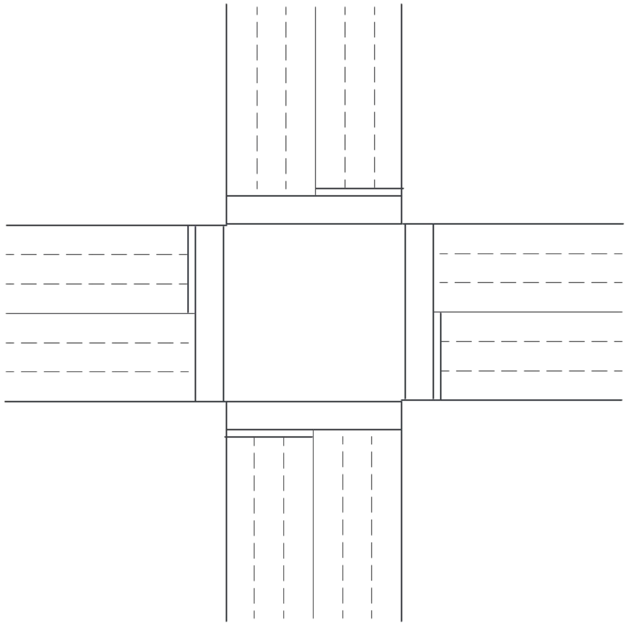
CO221 Assignment

Traffic Light System

System Description

1. The junction is a four-way intersection, with each direction having three separate lanes for left turns, right turns, and forward movement.
2. There is no roundabout at the center of the junction.
3. Traffic congestion is evenly distributed in all four directions, and pedestrian crossings are significant.
4. The junction prioritizes left-turning vehicles.
5. Only pedestrian crossings in opposite directions can have green signals simultaneously; otherwise, only one pedestrian crossing can be green at a time.
6. No traffic signal will be green for consecutive periods to prevent excessive waiting times and to ensure efficient traffic flow.



System Constraints and Assumptions

First, we created a table of possible traffic signal combinations based on our constraints and assumptions. We selected the three best combinations, ensuring that in each case, at least two lanes receive green signals per cycle, along with pedestrian crossings.

N				E				W				S			
P	L	F	R	P	L	F	R	P	L	F	R	P	L	F	R
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0
0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0

Optimal Signal Combinations

We evaluated different traffic signal arrangements to determine the most efficient sequence. The selected three combinations provide:

- 2 Left (L) signals
- 2 Right (R) signals
- 3 Forward (F) signals
- 4 Pedestrian (P) signals

From these combinations, we generated 12 different signal patterns ($3 \times 4 = 12$). Using a Python script, we applied permutations ($12!$) to optimize the signal order while adhering to the given constraints.

Best Choice for Signal Combination

The optimal traffic signal sequence was determined as follows:
(Table representation of N, E, W, S lanes with signal states)

				4	2	3	2	4	2	3	2	4	2	3	2	4	2	3	2
Timer				N				E				W				S			
A	B	C	D	P	L	F	R	P	L	F	R	P	L	F	R	P	L	F	R
0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0
0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
1	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0

Karnaugh Map for Simplification

A Karnaugh map was used to simplify the combinational logic circuit. "Don't care" conditions were strategically placed in the middle to enable XOR gate-based minimization, reducing the overall number of logic gates.

				C		
	AA	00	01	11	10	
	00	0	0	0	0	
	01	0	X	X	0	B
A	11	1	X	X	0	
	10	0	1	0	0	
			D			

North = {Z, AA, AB, AC} | East = {AD, AE, AF, AG} | South = {AH, AI, AJ, AK} | West = {AL, AM, AN, AO}

Traffic Light Control Logic (Boolean Expressions)

1. North Lane
    a. Pedestrian

    $$F = A \cdot D + \neg A \cdot \neg D \cdot (\neg(B \oplus C))$$

    b. Left Side

    $$F = (A \oplus C) \cdot \neg(B \cdot D)$$

    c. Forward

    $$F = \neg A \cdot D + A \cdot C \cdot \neg(B \cdot D)$$

    d. Right Side

    $$F = B \cdot C \cdot (A \oplus D) + \neg A \cdot D \cdot (B \oplus C)$$

2. East Lane
    a. Pedestrian

    $$F = \neg A \cdot D + A \cdot \neg D \cdot (B \oplus C)$$

    b. Left Side

    $$F = A \cdot D \cdot (B \oplus C) + B \cdot \neg C \cdot \neg(A \oplus D)$$

    c. Forward

    $$F = \neg A \cdot \neg D \cdot (\neg(B \oplus C)) + \neg A \cdot C \cdot \neg(B \cdot D)$$

    d. Right Side

    $$F = (A \oplus C) \cdot \neg(B \cdot D)$$

3. South Lane
    a. Pedestrian

    $$F = B \cdot C + \neg A \cdot \neg(B \cdot D)$$

    b. Left Side

    $$F = A \cdot \neg C \cdot (B \cdot D)$$

    c. Forward

    $$F = \neg A \cdot \neg C \cdot (B \cdot D) + A \cdot C \cdot \neg(B \cdot D)$$

    d. Right Side

    $$F = B \cdot \neg C \cdot \neg(A \oplus D) + A \cdot D \cdot (B \oplus C)$$

4. West Lane
    a. Pedestrian

    $$F = \neg A \cdot \neg C \cdot (B \cdot D) + A \cdot \neg(B \cdot D)$$

    b. Left Side

    $$F = \neg A \cdot D \cdot (B \oplus C) + B \cdot C \cdot (A \oplus D)$$

    c. Forward

    $$F = \neg A \cdot \neg D \cdot (\neg(B \oplus C)) + B \cdot \neg C \cdot \neg(A \oplus D)$$

    d. Right Side

    $$F = A \cdot \neg C \cdot (B \cdot D)$$

Gate Count Calculation

We calculated the number of logic gates required to implement the combinational circuit based on the derived Boolean expressions.

(A⊕C) - 2

———————————-

¬(A⊕D) - 3

(A⊕D) - 2

———————————-

¬(B⊕C) - 3

(B⊕C) - 5

———————————-

¬(B⊕D) - 7

(B⊕D) - 4

———————————-

A - 1

¬ A - 1

AD - 3

¬A¬D - 3

¬AD - 4

A¬D - 1

———————————-

AC - 2

A¬C - 2

¬A¬C - 2

¬AC - 1

———————————-

BC - 3

B¬C - 2

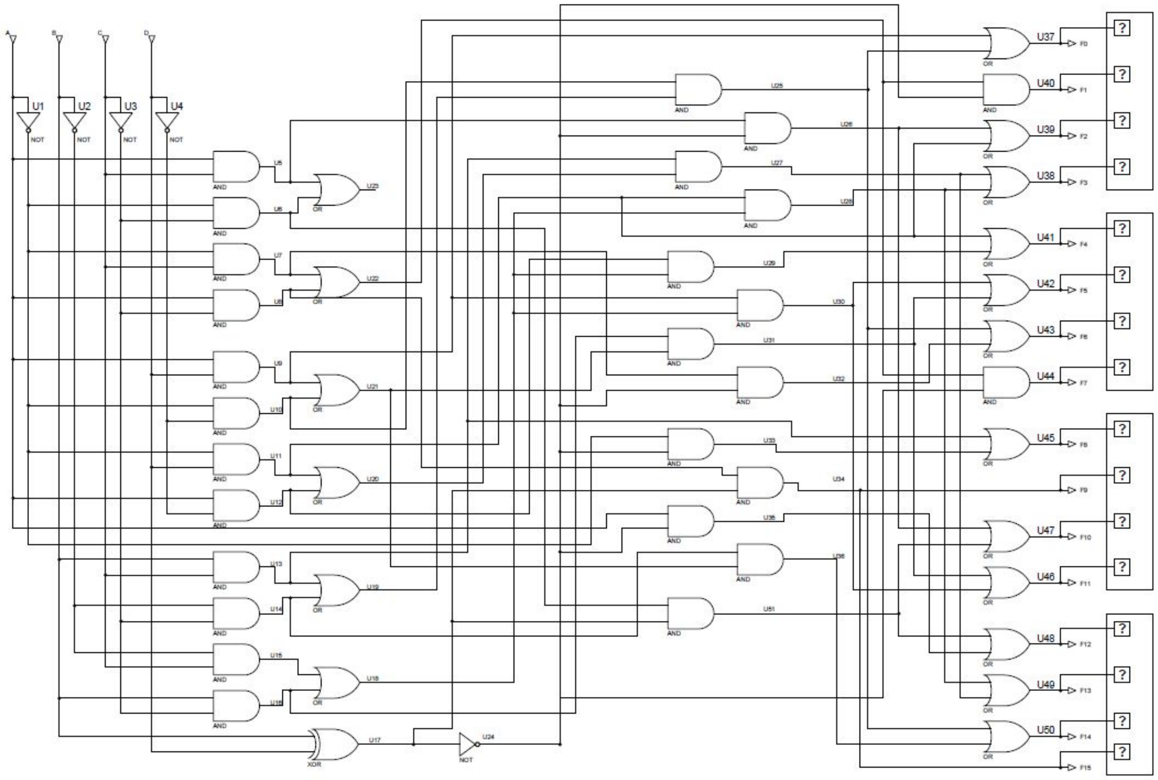
¬B¬C - 1

¬BC - 1

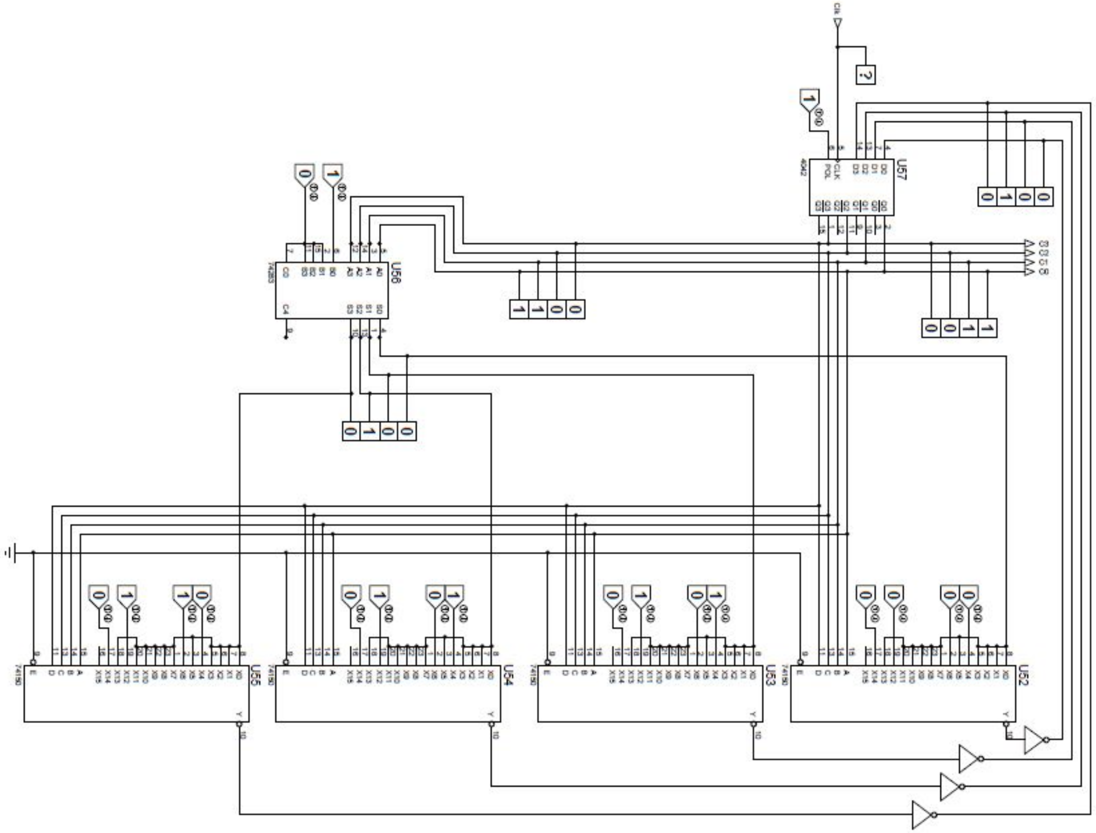
Simulation and Implementation

1. Traffic Signal Simulation (Proteus) – A module was created to simulate the traffic signal logic.
2. Timer Module – A counter was implemented, ranging from 0 to 14, skipping 5, 7, and 13.
3. Final Simulation – The complete traffic signal system was simulated by integrating all modules.

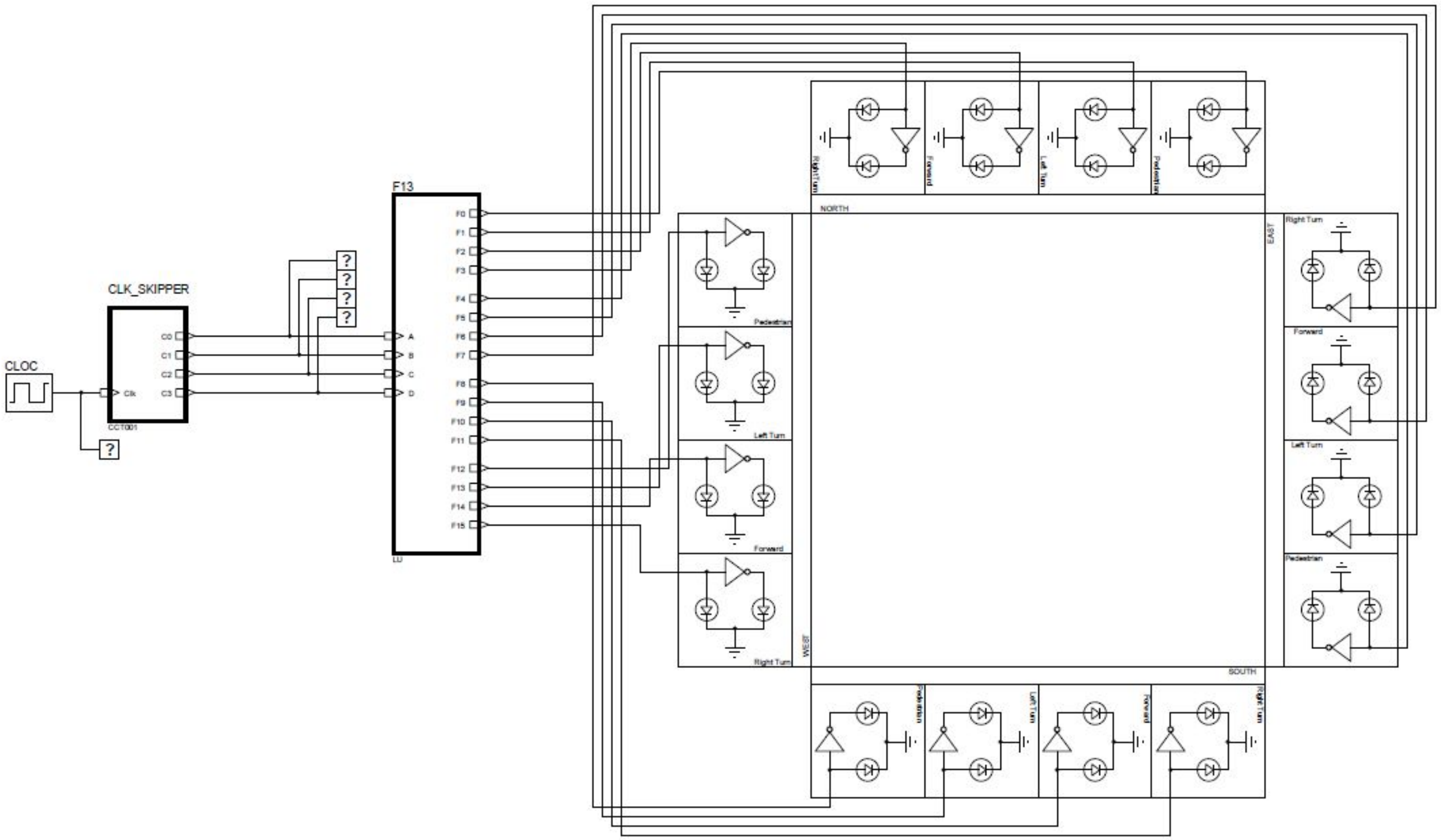
Traffic Signal Module



Timer Signal Module



Final Model


CLOC
CLK_SKIPPER
CCT001
F13
LU
NORTH
EAST
SOUTH
WEST
Right Turn
Forward
Left Turn
Pedestrian