Traffic Light System

Project Report

21/05/2023

—

EE482 – Industrial Process Control

BEE-11A (Spring 2023)

—

Dr Farid Gul

Table of Contents

Team Introduction

Student name	Registration Number
Atika Ishaq	302073
Kainat Khalid	284126
Bushra Azmat Qureshi	285345
Muhammad Sohaib Ikram	283756
Chaudhary Muhammad Shaheer Yasir	286021

Objective

Objectives of this project inlcuded

- Implement a traffic light ladder program using counter, move and equal instructions.
- Create a second traffic light ladder program using sequencer instructions.

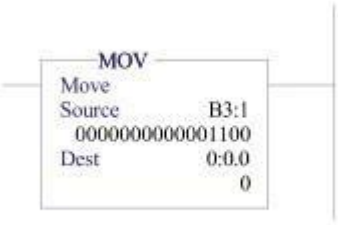
Introduction

1. Count Up (CTU) Instruction

This instruction counts up when the preceding conditions make a false-to-true transition. The done bit is enabled when the accumulated value reaches the preset value. Figure 2-1 shows counter C5:0 which is triggered by a timer done bit.

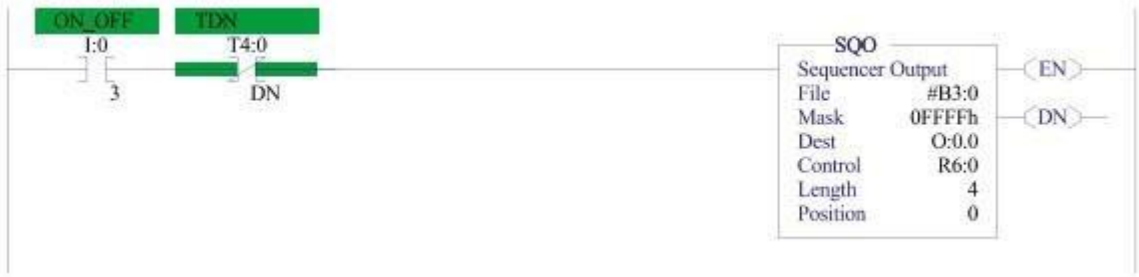
2. Move (MOV) Instruction

The move (MOV) instruction is used to transfer a word of data from one file to another. Figure 2-2 shows how the word B3:1 is transferred to the outputs of slot 0.



3. Sequencer Output (SQO) Instruction

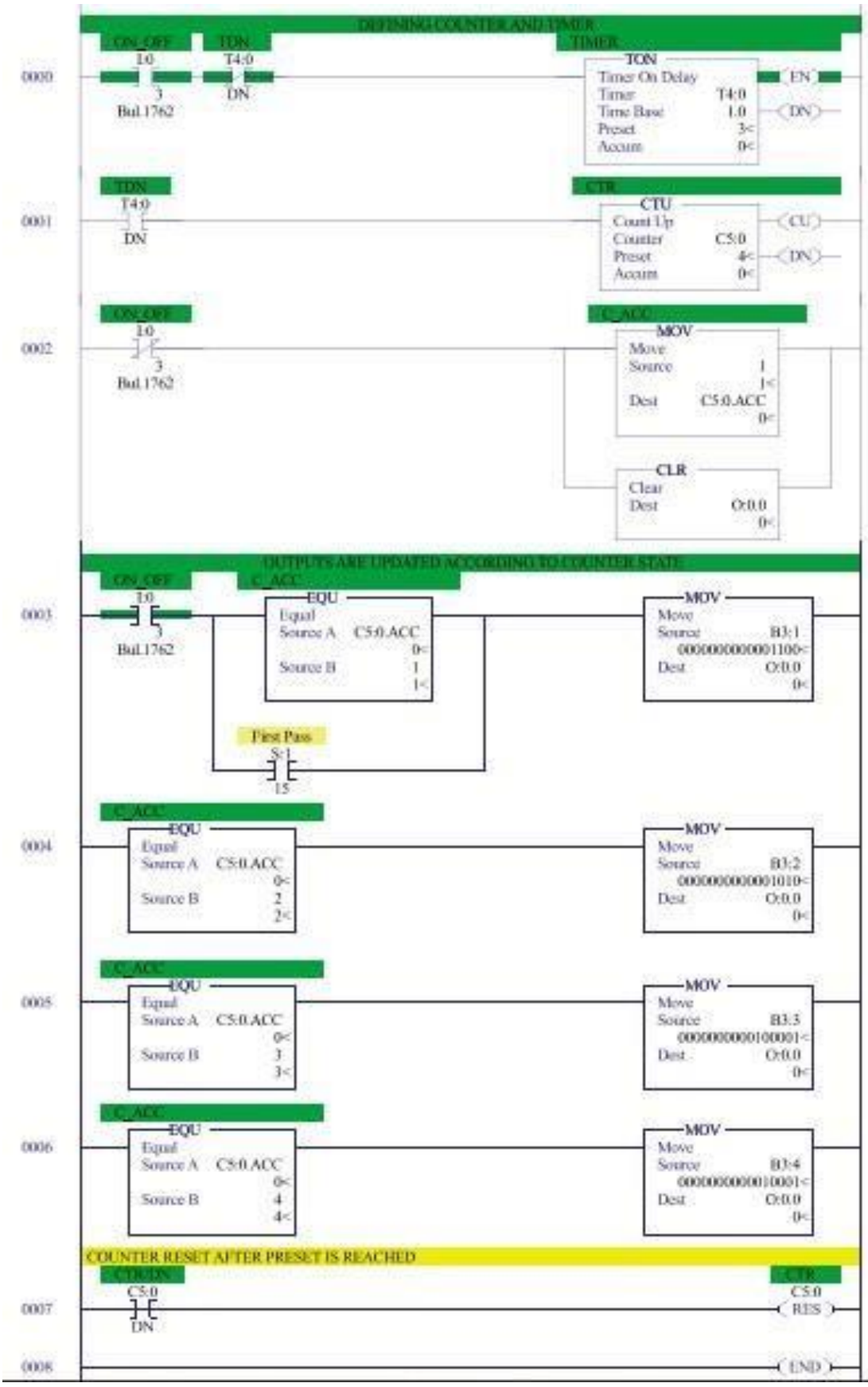
The sequencer output (SQO) instruction is meant to transfer a word of data from a file, through a mask, to a destination file. This process happens every time the same-rung conditions go from false to true. Mask bits that are of logic state 0 will mask (block) same-position bits of the word transferred. The SQO instruction is used for sequential control of various devices. Since traffic lights operate in sequential mode (green - yellow - red - green), the SQO instruction is well suited for this application.

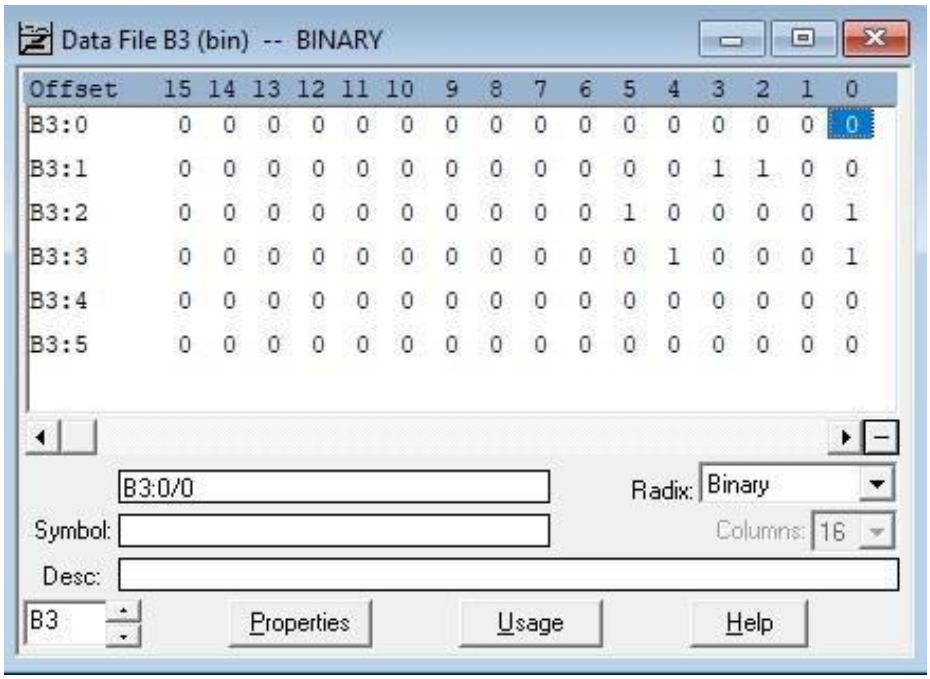


Design and Methodology

Ladder Program with MOV, CTU Commands

Required ladder program to make Traffic Management System using MOV, and CTU command.


DEFINING COUNTER AND TIMER
ON_OFF
I:0
3
Bul.1762
TDN
T4:0
DN
TIMER
TON
Timer On Delay
Timer T4:0
Time Base 1.0
Preset 3<
Accum 0<
EN
DN
TDN
T4:0
DN
CTR
CTU
Count Up
Counter C5:0
Preset 4<
Accum 0<
CU
DN
ON_OFF
I:0
3
Bul.1762
C_ACC
MOV
Move
Source 1
1<
Dest C5:0.ACC
0<
CLR
Clear
Dest O:0.0
0<
OUTPUTS ARE UPDATED ACCORDING TO COUNTER STATE
ON_OFF
I:0
3
Bul.1762
C_ACC
EQU
Equal
Source A C5:0.ACC
0<
Source B 1
1<
MOV
Move
Source B3:1
0000000000001100<
Dest O:0.0
0<
First Pass
S:1
15
C_ACC
EQU
Equal
Source A C5:0.ACC
0<
Source B 2
2<
MOV
Move
Source B3:2
0000000000001010<
Dest O:0.0
0<
C_ACC
EQU
Equal
Source A C5:0.ACC
0<
Source B 3
3<
MOV
Move
Source B3:3
0000000000010001<
Dest O:0.0
0<
C_ACC
EQU
Equal
Source A C5:0.ACC
0<
Source B 4
4<
MOV
Move
Source B3:4
0000000000011001<
Dest O:0.0
0<
COUNTER RESET AFTER PRESET IS REACHED
C5:0.DN
C5:0
DN
CTR
C5:0
RES
END



Input Output Connections

Figure below shows the input output connections.

Address	Input/Output
I:0/3	On/Off Toggle Switch
O:0/0	Red Light 1 (R1)
O:0/1	Yellow Light 1 (Y1)
O:0/2	Green Light 1 (G1)
O:0/3	Red Light 2 (R2)
O:0/4	Yellow Light 2 (Y2)
O:0/5	Green Light 2 (G2)

Rung Details

Rung 0: It includes the timer for 3 seconds.

Rung 1: It consists of count up instruction whose preset value is set to be less than 4.

Rung 2: It has move instructions which move the information back to the counter.

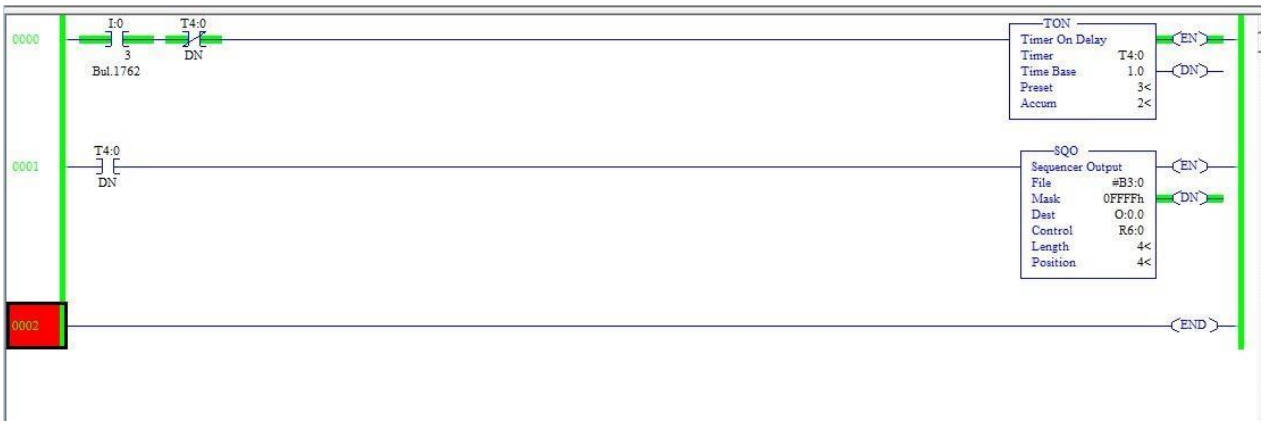
Rung 3: This rung operates the red and green light simultaneously.

Rung 4: This rung is used for operating the green light of the traffic system.

Rung 5: This rung operated the red light of the traffic system.

Rung 6: It controls the operation of the yellow light of the traffic system.
Rung 7: This rung resets the counter when the preset value of 4< is reached.
Rung 8: The ladder programs end in this rung.

Ladder Program with SQU Command



Conclusion

This project involved designing a traffic light control system. We first operated the traffic light system by using instructions like move (MOV), counter (CTU) and equal (EQU). We set the binary values in these instructions in order to operate the system according to the requirement. This ladder program consists of eight rungs. Then in order to simplify the program we used only one instruction named sequencer operator (SQO) which can perform the operations of all these instructions. Then by using the state table, we were able to develop a set of Boolean equations that could be used to program the system and the ladder program now consists of only three rungs.