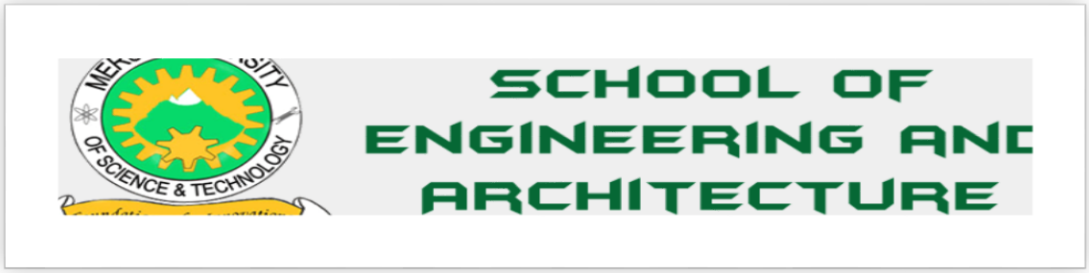
MERU UNIVERSITY OF SCIENCE AND TECHNOLOGY



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BACHELOR OF TECHNOLOGY IN ELECTRICTRAL AND ELECTRONIS ENGINEERING

TITLE: LAB REPORT: Traffic Control System Simulation Using Intel 8085 Microprocessor (Assembly Language)

NAME: JOSHUA MUTHENYA WAMBUA

REG NO: EG209/109705/22

UNIT CODE: 3400

DESCRIPTION: MICROPROCESSOR ARCHITECTURE AND INTERFACING

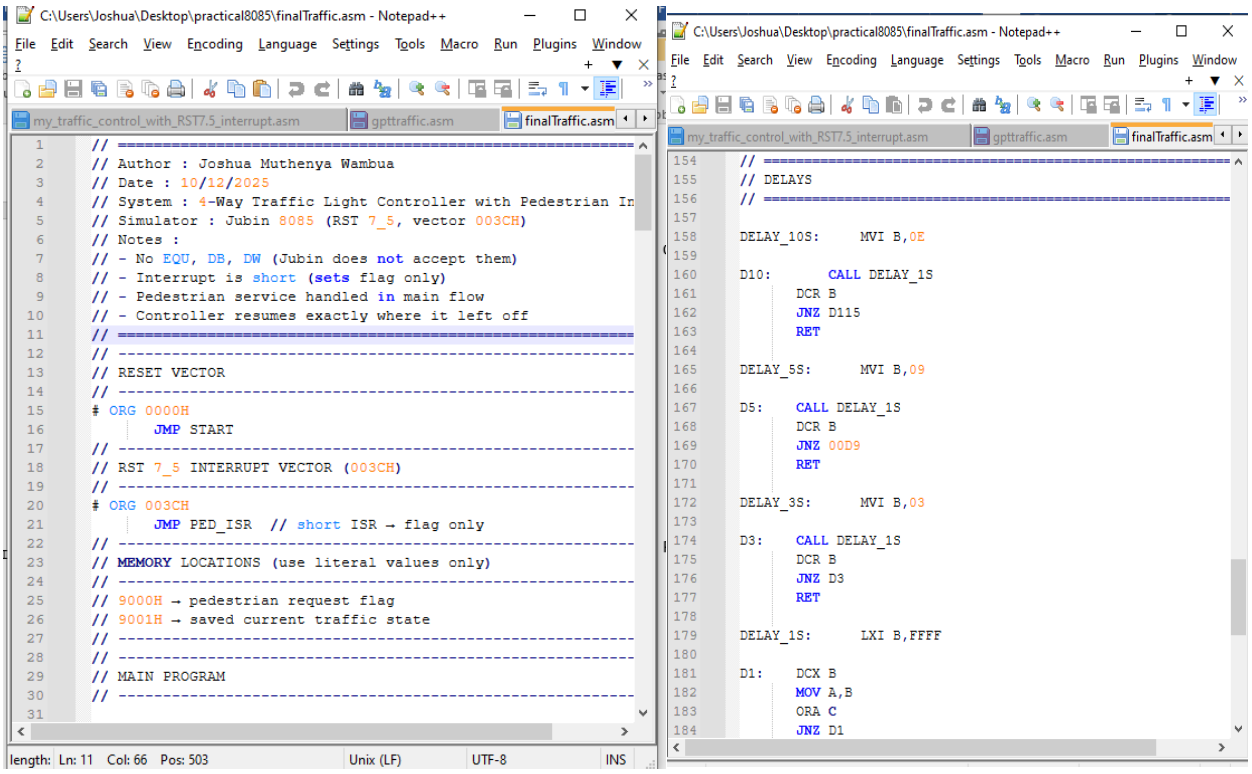
DATE:05/12/2025

Introduction

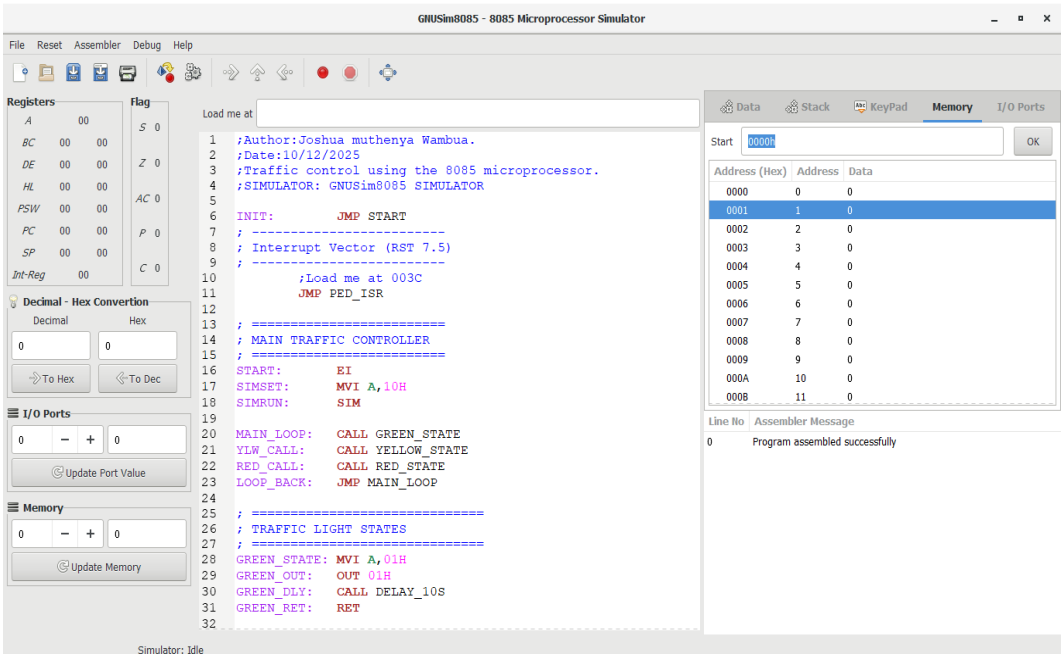
Objectives

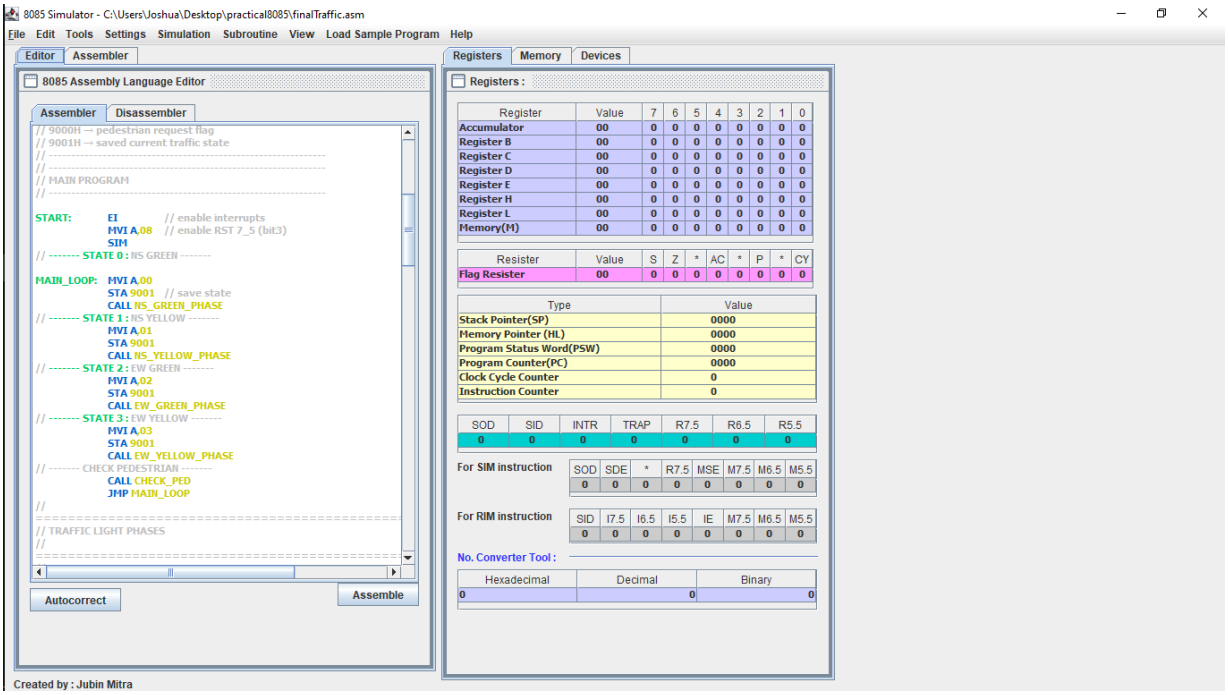
Softwares/Kits used.

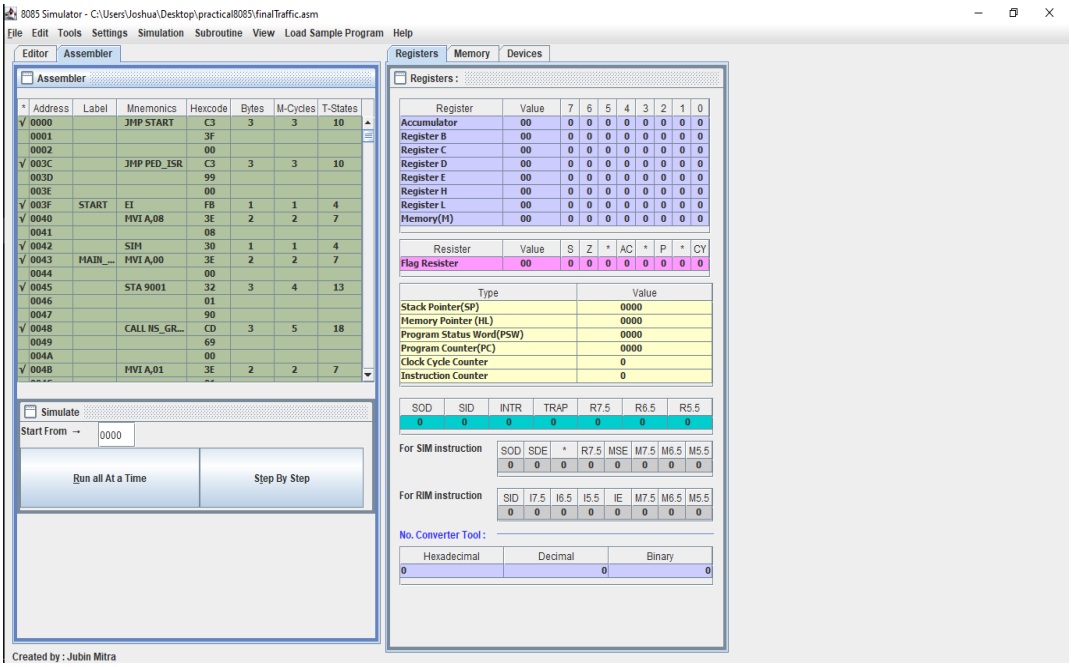
Procedure

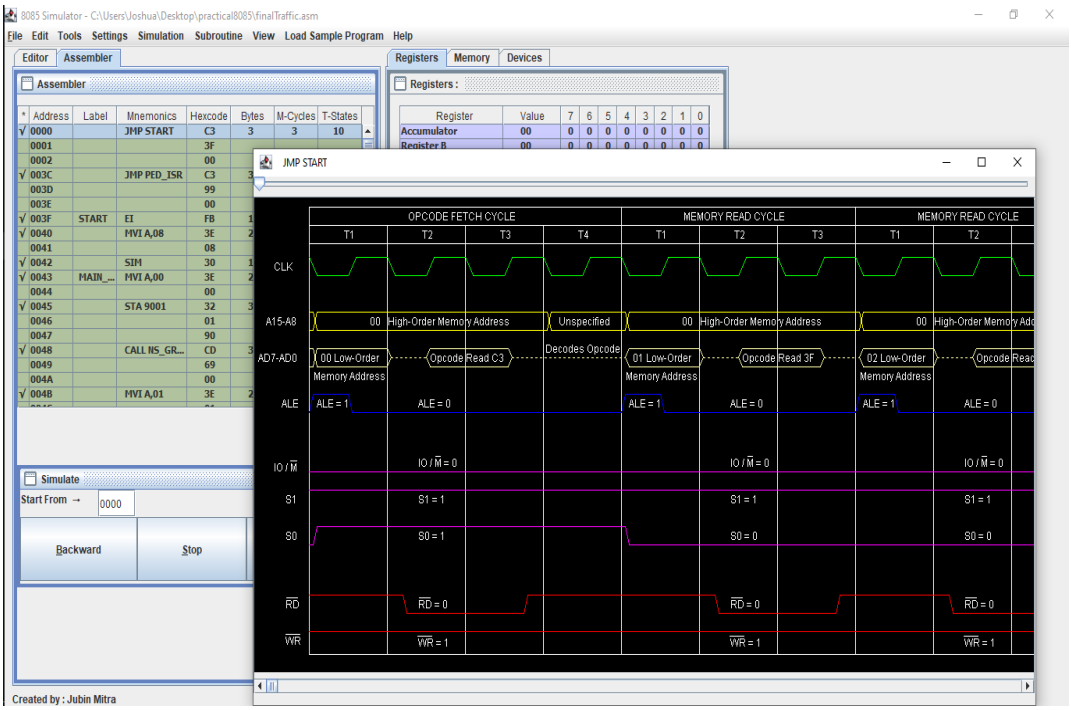

C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
my_traffic_control_with_RST7.5_interrupt.asm
gpttraffic.asm
finalTraffic.asm
// ==========
// Author : Joshua Muthenya Wambua
// Date : 10/12/2025
// System : 4-Way Traffic Light Controller with Pedestrian In
// Simulator : Jubin 8085 (RST 7_5, vector 003CH)
// Notes :
// - No EQU, DB, DW (Jubin does not accept them)
// - Interrupt is short (sets flag only)
// - Pedestrian service handled in main flow
// - Controller resumes exactly where it left off
// ==========
// RESET VECTOR
ORG 0000H
JMP START
// RST 7_5 INTERRUPT VECTOR (003CH)
ORG 003CH
JMP PED_ISR // short ISR → flag only
// MEMORY LOCATIONS (use literal values only)
// 9000H → pedestrian request flag
// 9001H → saved current traffic state
// MAIN PROGRAM
length: Ln: 11 Col: 66 Pos: 503
Unix (LF)
UTF-8
INS
C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
my_traffic_control_with_RST7.5_interrupt.asm
gpttraffic.asm
finalTraffic.asm
// DELAYS
DELAY_10S: MVI B,0E
D10: CALL DELAY_1S
DCR B
JNZ D115
RET
DELAY_5S: MVI B,09
D5: CALL DELAY_1S
DCR B
JNZ 00D9
RET
DELAY_3S: MVI B,03
D3: CALL DELAY_1S
DCR B
JNZ D3
RET
DELAY_1S: LXI B,FFFF
D1: DCX B
MOV A,B
ORA C
JNZ D1

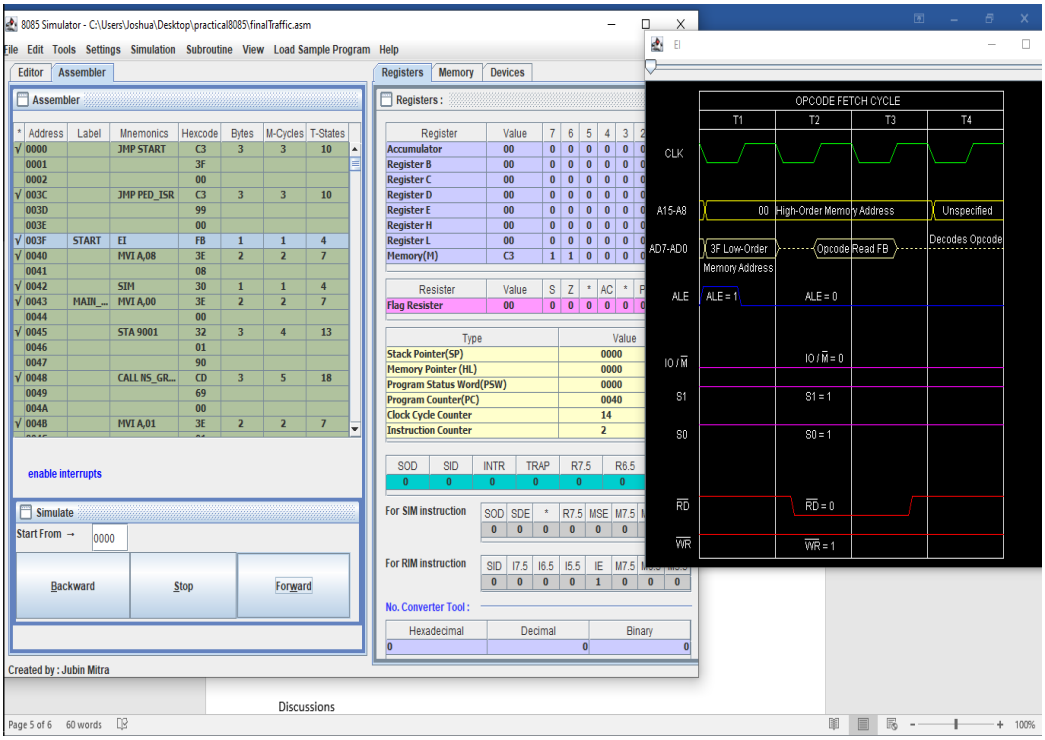
Results

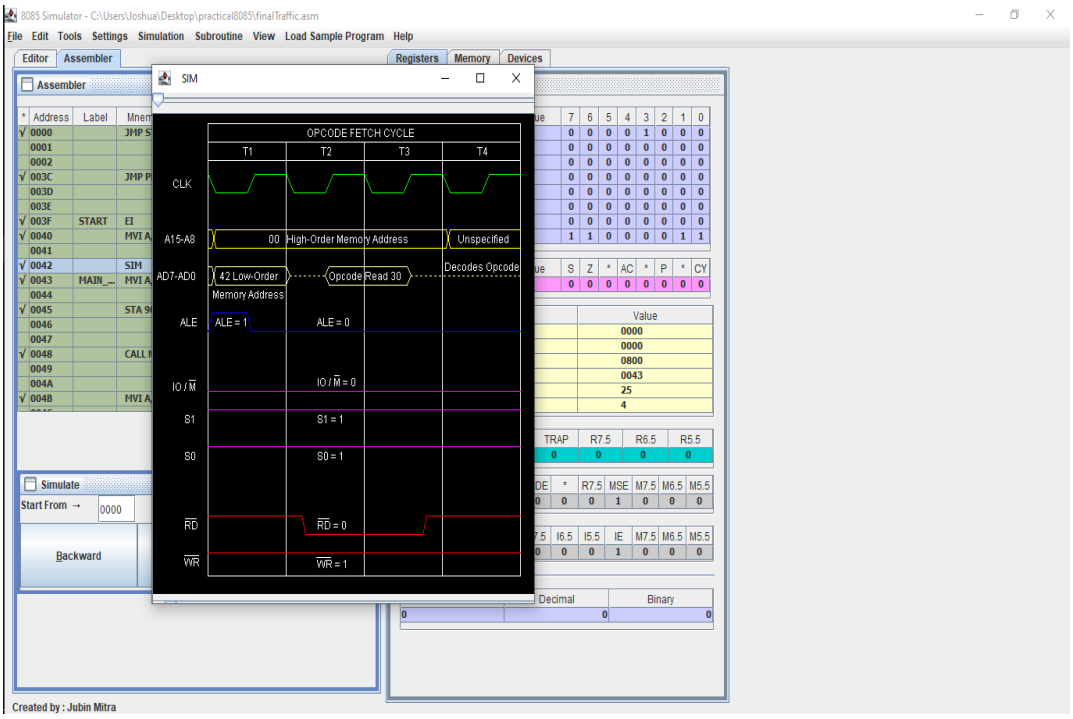

GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Registers
Flag
Load me at
;Author:Joshua muthenya Wambua.
;Date:10/12/2025
;Traffic control using the 8085 microprocessor.
;SIMULATOR: GNUSim8085 SIMULATOR
INIT: JMP START
; Interrupt Vector (RST 7.5)
;Load me at 003C
JMP PED_ISR
; MAIN TRAFFIC CONTROLLER
START: EI
SIMSET: MVI A,10H
SIMRUN: SIM
MAIN_LOOP: CALL GREEN_STATE
YLW_CALL: CALL YELLOW_STATE
RED_CALL: CALL RED_STATE
LOOP_BACK: JMP MAIN_LOOP
; TRAFFIC LIGHT STATES
GREEN_STATE: MVI A,01H
GREEN_OUT: OUT 01H
GREEN_DLY: CALL DELAY_10S
GREEN_RET: RET
Data Stack KeyPad Memory I/O Ports
Start 0000h OK
Address (Hex) Address Data
Line No Assembler Message
0 Program assembled successfully
Decimal - Hex Convertion
I/O Ports
Memory
Simulator: Idle

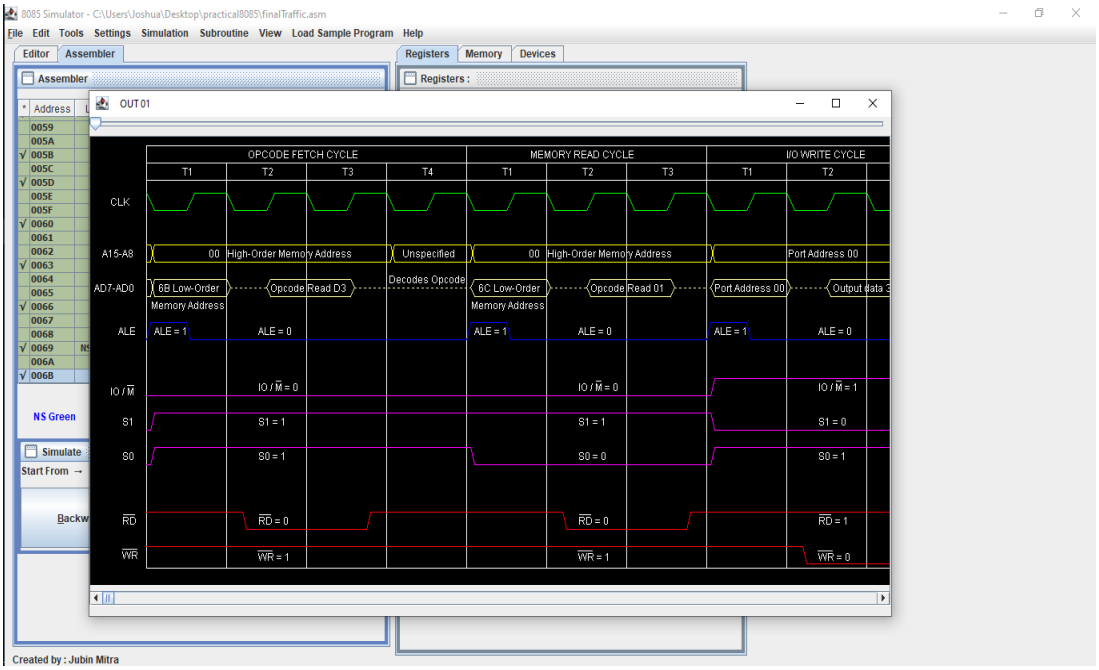

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
8085 Assembly Language Editor
Assembler Disassembler
// 9000H → pedestrian request flag
// 9001H → saved current traffic state
// MAIN PROGRAM
START: EI // enable interrupts
MVI A,08 // enable RST 7_5 (bit3)
SIM
// ------- STATE 0 : NS GREEN -------
MAIN_LOOP: MVI A,00
STA 9001 // save state
CALL NS_GREEN_PHASE
// ------- STATE 1 : NS YELLOW -------
MVI A,01
STA 9001
CALL NS_YELLOW_PHASE
// ------- STATE 2 : EW GREEN -------
MVI A,02
STA 9001
CALL EW_GREEN_PHASE
// ------- STATE 3 : EW YELLOW -------
MVI A,03
STA 9001
CALL EW_YELLOW_PHASE
// ------- CHECK PEDESTRIAN -------
CALL CHECK_PED
JMP MAIN_LOOP
// TRAFFIC LIGHT PHASES
Autocorrect
Assemble
Registers Memory Devices
Registers :
Register Value 7 6 5 4 3 2 1 0
Accumulator 00
Register B 00
Register C 00
Register D 00
Register E 00
Register H 00
Register L 00
Memory(M) 00
Resister Value S Z * AC * P * CY
Flag Resister 00
Type Value
Stack Pointer(SP) 0000
Memory Pointer (HL) 0000
Program Status Word(PSW) 0000
Program Counter(PC) 0000
Clock Cycle Counter 0
Instruction Counter 0
SOD SID INTR TRAP R7.5 R6.5 R5.5
For SIM instruction SOD SDE * R7.5 MSE M7.5 M6.5 M5.5
For RIM instruction SID I7.5 I6.5 I5.5 IE M7.5 M6.5 M5.5
No. Converter Tool :
Hexadecimal Decimal Binary
Created by : Jubin Mitra

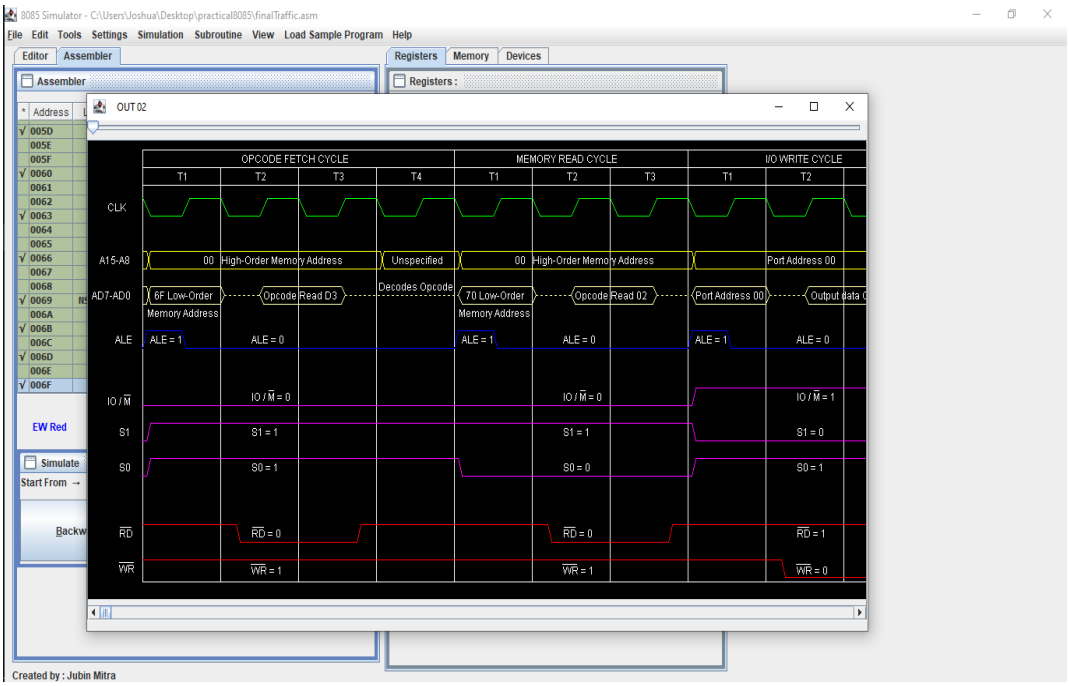

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Assembler
Registers Memory Devices
Registers :
Simulate
Start From →
0000
Run all At a Time
Step By Step
For SIM instruction
For RIM instruction
No. Converter Tool :
Hexadecimal
Decimal
Binary
Created by : Jubin Mitra

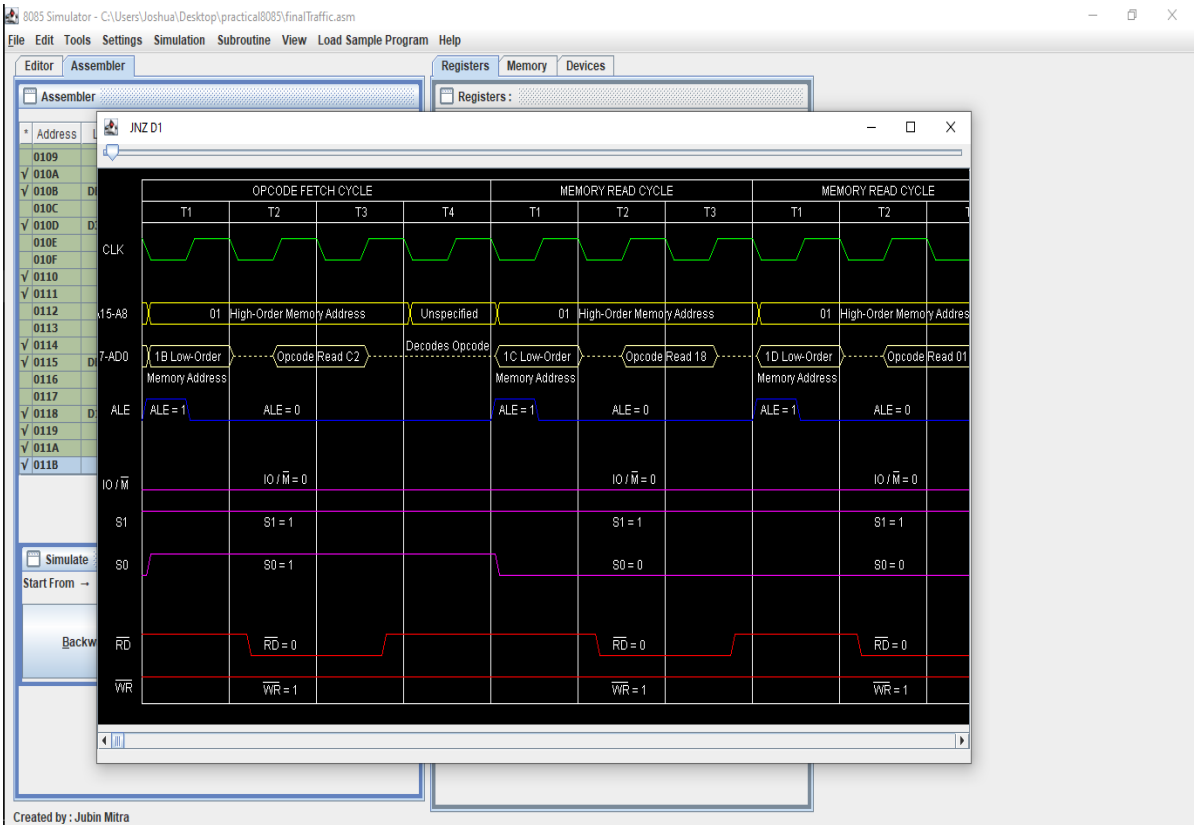

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Assembler
Registers Memory Devices
Registers :
Simulate
Start From →
0000
Backward
Stop
Created by : Jubin Mitra
JMP START
OPCODE FETCH CYCLE
MEMORY READ CYCLE
MEMORY READ CYCLE
T1 T2 T3 T4 T1 T2 T3 T1 T2
CLK
A15-A8
AD7-AD0
ALE
IO / M
S1
S0
RD
WR
00 High-Order Memory Address
Unspecified
Decodes Opcode
00 Low-Order Memory Address
Opcode Read C3
01 Low-Order Memory Address
Opcode Read 3F
02 Low-Order Memory Address
ALE = 1
ALE = 0
IO / M = 0
S1 = 1
S0 = 1
S0 = 0
RD = 0
WR = 1

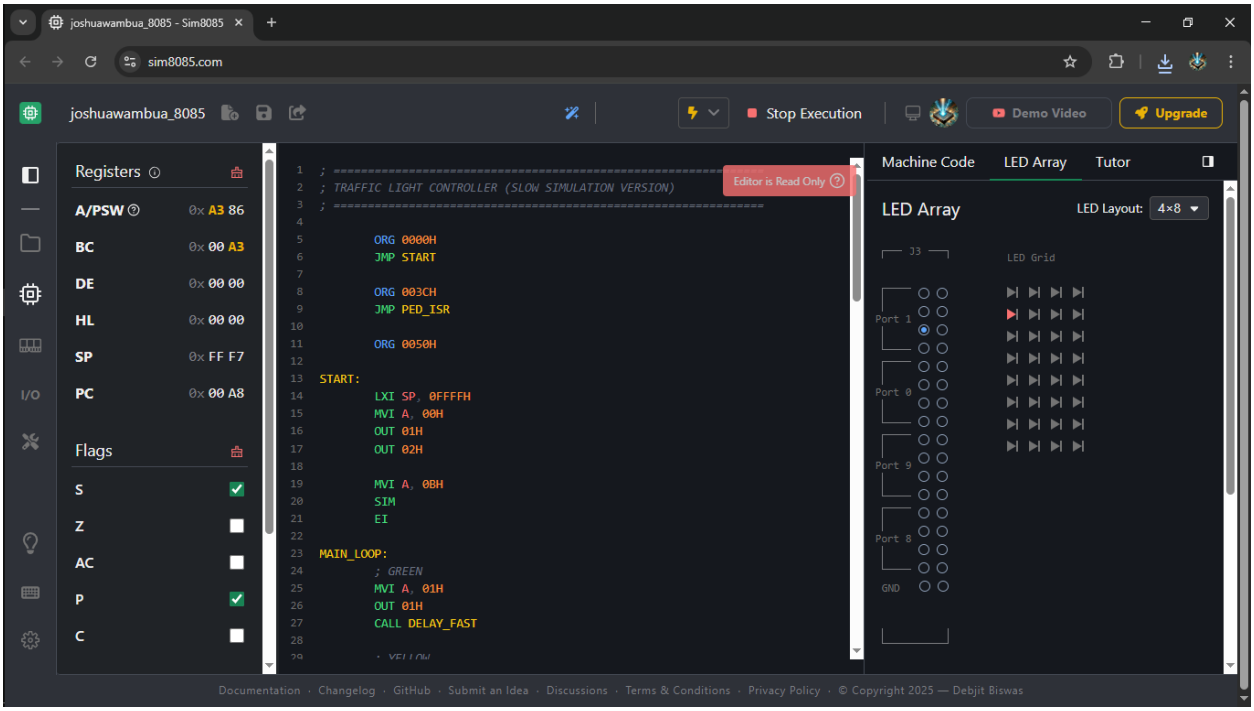

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices
Assembler
Registers :
enable interrupts
Simulate
Start From →
0000
Backward
Stop
Forward
No. Converter Tool :
Created by : Jubin Mitra
Discussions
OPCODE FETCH CYCLE
T1 T2 T3 T4
CLK
A15-A8
00 High-Order Memory Address
Unspecified
AD7-AD0
3F Low-Order Memory Address
Opcode Read FB
Decodes Opcode
ALE
ALE = 1
ALE = 0
IO / M
IO / M = 0
S1
S1 = 1
S0
S0 = 1
RD
RD = 0
WR
WR = 1

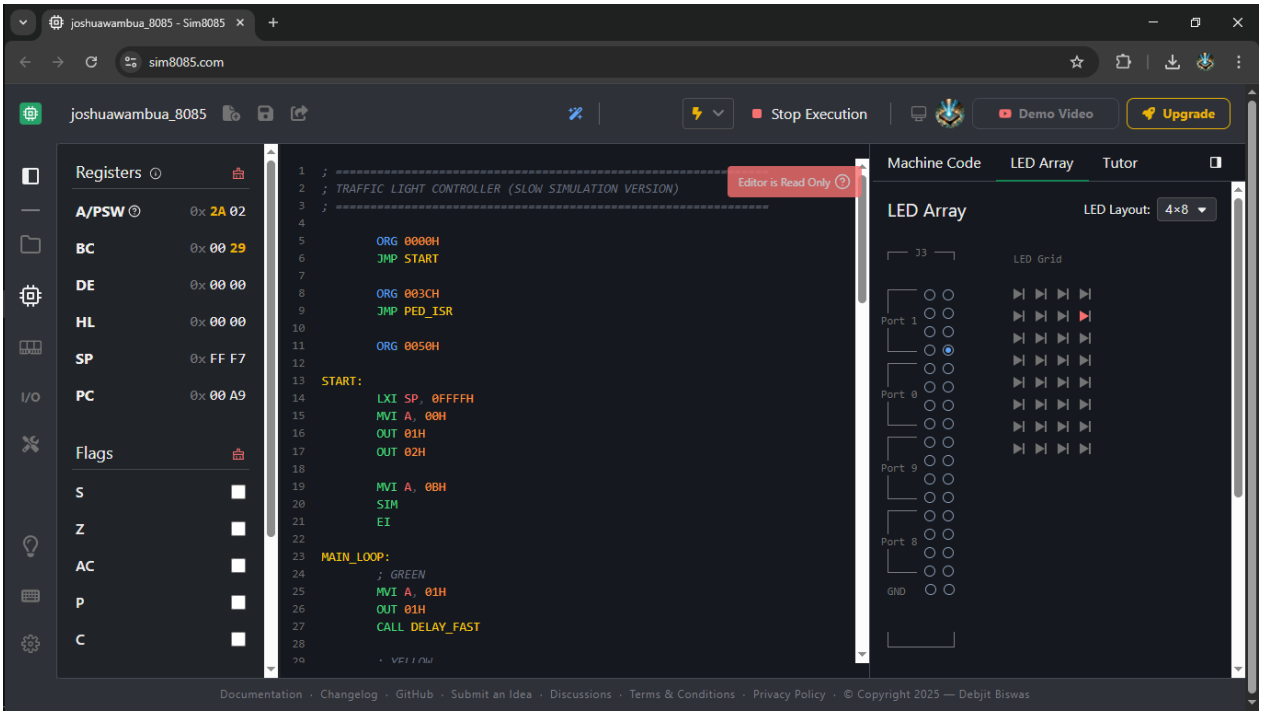

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices
Assembler
SIM
OPCODE FETCH CYCLE
T1 T2 T3 T4
CLK
A15-A8
00 High-Order Memory Address
Unspecified
AD7-AD0
42 Low-Order Memory Address
Opcode Read 30
Decodes Opcode
ALE
ALE = 1
ALE = 0
IO / M
IO / M = 0
S1
S1 = 1
S0
S0 = 1
RD
RD = 0
WR
WR = 1
Simulate
Start From →
0000
Backward
Created by : Jubin Mitra

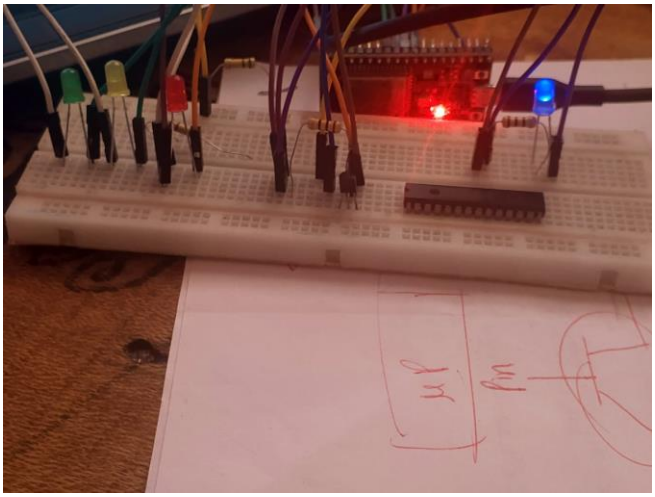

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices
Assembler
Registers :
OUT 01
OPCODE FETCH CYCLE
MEMORY READ CYCLE
I/O WRITE CYCLE
T1 T2 T3 T4 T1 T2 T3 T1 T2
CLK
A15-A8
00 High-Order Memory Address
Unspecified
00 High-Order Memory Address
Port Address 00
AD7-AD0
6B Low-Order Memory Address
Opcode Read D3
Decodes Opcode
6C Low-Order Memory Address
Opcode Read 01
Port Address 00
Output data
ALE
ALE = 1
ALE = 0
IO / M
IO / M = 0
IO / M = 1
S1
S1 = 1
S1 = 0
S0
S0 = 1
S0 = 0
RD
RD = 0
RD = 1
WR
WR = 1
WR = 0
NS Green
Simulate
Start From
Created by : Jubin Mitra

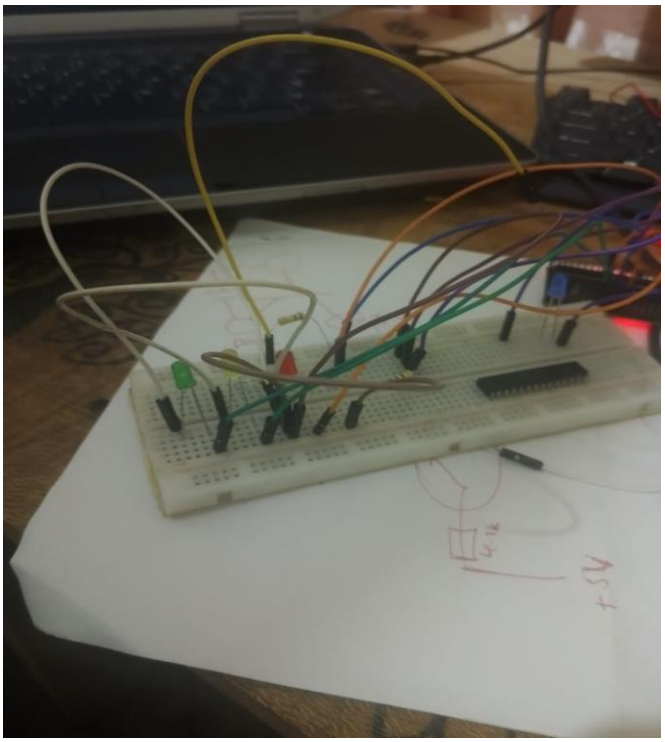

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices
Assembler
Registers :
OUT 02
OPCODE FETCH CYCLE
MEMORY READ CYCLE
I/O WRITE CYCLE
T1 T2 T3 T4 T1 T2 T3 T1 T2
CLK
A15-A8
00 High-Order Memory Address
Unspecified
00 High-Order Memory Address
Port Address 00
AD7-AD0
6F Low-Order Memory Address
Opcode Read D3
Decodes Opcode
70 Low-Order Memory Address
Opcode Read 02
Port Address 00
Output data
ALE
ALE = 1
ALE = 0
IO / M
IO / M = 0
IO / M = 1
S1
S1 = 1
S1 = 0
S0
S0 = 1
S0 = 0
RD
RD = 0
RD = 1
WR
WR = 1
WR = 0
EW Red
Simulate
Start From
Created by : Jubin Mitra

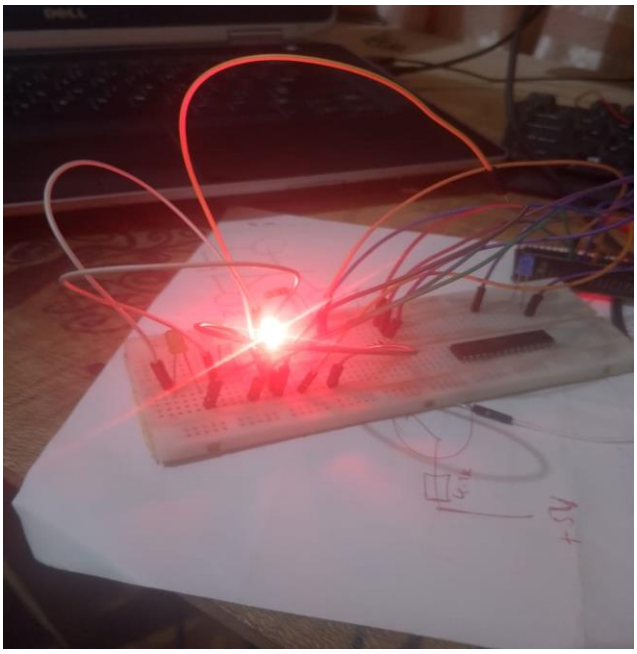

8085 Simulator - C:\Users\Joshua\Desktop\practical8085\finalTraffic.asm
File Edit Tools Settings Simulation Subroutine View Load Sample Program Help
Editor Assembler
Registers Memory Devices
Assembler
Registers :
JNZ D1
OPCODE FETCH CYCLE
MEMORY READ CYCLE
MEMORY READ CYCLE
T1 T2 T3 T4 T1 T2 T3 T1 T2
CLK
A15-A8
AD7-AD0
ALE
IO / M
S1
S0
RD
WR
01 High-Order Memory Address
Unspecified
Decodes Opcode
1B Low-Order Memory Address
Opcode Read C2
1C Low-Order Memory Address
Opcode Read 18
1D Low-Order Memory Address
Opcode Read 01
ALE = 1
ALE = 0
IO / M = 0
S1 = 1
S0 = 1
S0 = 0
RD = 0
WR = 1
Simulate
Start From
Created by : Jubin Mitra

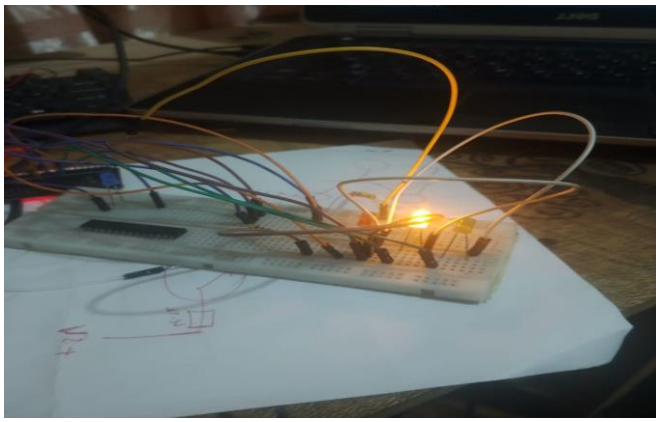

joshuawambua_8085 - Sim8085
sim8085.com
joshuawambua_8085
Stop Execution
Demo Video
Upgrade
Registers
A/PSW 0x A3 86
BC 0x 00 A3
DE 0x 00 00
HL 0x 00 00
SP 0x FF F7
PC 0x 00 A8
Flags
S
Z
AC
P
C
; TRAFFIC LIGHT CONTROLLER (SLOW SIMULATION VERSION)
Editor is Read Only
ORG 0000H
JMP START
ORG 003CH
JMP PED_ISR
ORG 0050H
START:
LXI SP, 0FFFFH
MVI A, 00H
OUT 01H
OUT 02H
MVI A, 0BH
SIM
EI
MAIN_LOOP:
; GREEN
MVI A, 01H
OUT 01H
CALL DELAY_FAST
Machine Code
LED Array
Tutor
LED Array
LED Layout: 4×8
J3
LED Grid
Port 1
Port 0
Port 9
Port 8
GND
Documentation · Changelog · GitHub · Submit an Idea · Discussions · Terms & Conditions · Privacy Policy · © Copyright 2025 — Debjit Biswas


joshuawambua_8085 - Sim8085
sim8085.com
joshuawambua_8085
Stop Execution
Demo Video
Upgrade
Registers
A/PSW 0x 2A 02
BC 0x 00 29
DE 0x 00 00
HL 0x 00 00
SP 0x FF F7
PC 0x 00 A9
Flags
S
Z
AC
P
C
; TRAFFIC LIGHT CONTROLLER (SLOW SIMULATION VERSION)
Editor is Read Only
ORG 0000H
JMP START
ORG 003CH
JMP PED_ISR
ORG 0050H
START:
LXI SP, 0FFFFH
MVI A, 00H
OUT 01H
OUT 02H
MVI A, 0BH
SIM
EI
MAIN_LOOP:
; GREEN
MVI A, 01H
OUT 01H
CALL DELAY_FAST
Machine Code
LED Array
Tutor
LED Array
LED Layout: 4×8
J3
LED Grid
Port 1
Port 0
Port 9
Port 8
GND
Documentation · Changelog · GitHub · Submit an Idea · Discussions · Terms & Conditions · Privacy Policy · © Copyright 2025 — Debjit Biswas









Discussions

Conclusion

References