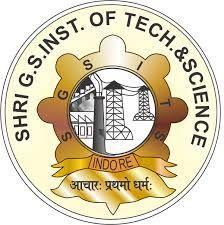
**Shri G. S. Institute of Technology & Science, Indore**

**Department of Computer Engineering**



**Subject: Computer Interfacing & IoT [CO44003]**

**Session: July – December 2020**

**Lab Assignment**

**Submitted To: Submitted By:**

**Mr. Veerendra Shrivastava Shikhar Mahajan**

**Assistant Professor 0801CS171077**

**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Assignment | Page No. | Submission Date | Signature |
| 1. | Lab Assignment 1   1. features of EdSim51 2. Addressing modes 3. Assembly language program 4. Copy 5 byte data block 5. Sum of first 10 natural numbers | 2  2  3  5  14  15 | 15-12-2020 |  |

**Lab Assignment 1**

**Q1. Download and install the EdSim51 simulator for 8051. Study and write the basic features of EdSim51.**

**Answer:**

1. The bottom panel shows all the peripherals that are connected to the 8051.
2. Syntax Highlighting : assembly code written in EdSim51 is automatically syntax highlighted
3. Originally, the simulator ran with a system clock frequency of 12 MHz. Now the user can enter a value for the system clock frequency in MHz.
4. A list of 8-bit numbers (written in HEX) can be transmitted. To do so, the user encloses the list in curly braces, each number separated by a comma.
5. A simulation of the popular Hitachi HD44780 LCD module has been implemented for the EdSim51 Simulator. And now CGRAM has also been implemented.
6. For high resolution monitors, click on the zoom button. The zoom button is located below the red Exit button.
7. A breakpoint can be set by double-clicking the instruction's address.
8. We can now save our source code in Intel HEX format and also we can write C code for the 8051 using one of many available 8051 C compilers, then import the HEX code into the EdSim51 Simulator.
9. When running a program, the rate at which the screen updates is determined by the setting in the Update Freq. menu
10. Load and Save - The user can write code directly into the text box when it is in edit mode, or an existing program can be loaded from a file using the Load button. Similarly, code in the text box can be saved to file using the Save button
11. Keypad Modes - User can select from three modes of operation-standard,pulse ,radio

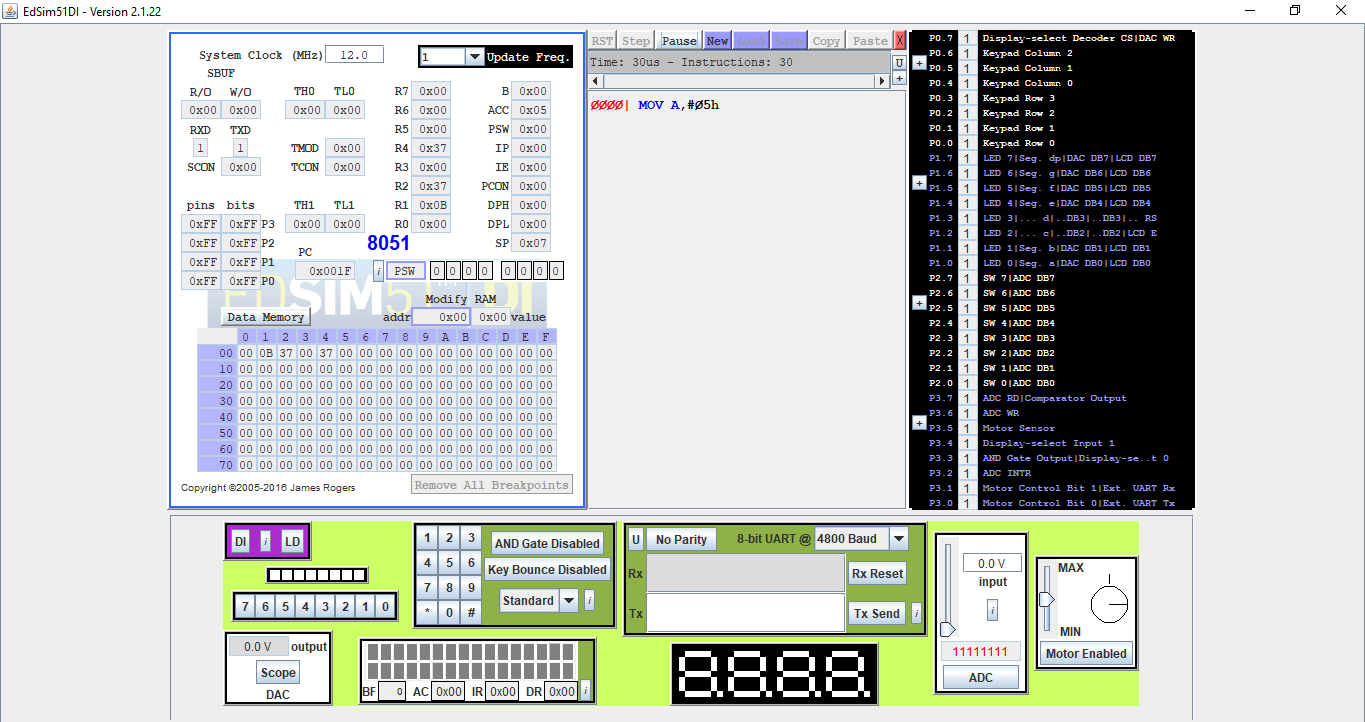
**Q2. Study the 8051 addressing modes. Write and execute/simulate the assembly language program in EdSim51 based on following addressing modes:**

**Answer:**

Addressing modes are an aspect of the instruction set architecture in most central processing unit designs. The various addressing modes that are defined in a given instruction set architecture define how the machine language instructions in that architecture identify the operand of each instruction.

1. **Immediate Addressing Mode**: In this mode data is present in the address field of instruction .

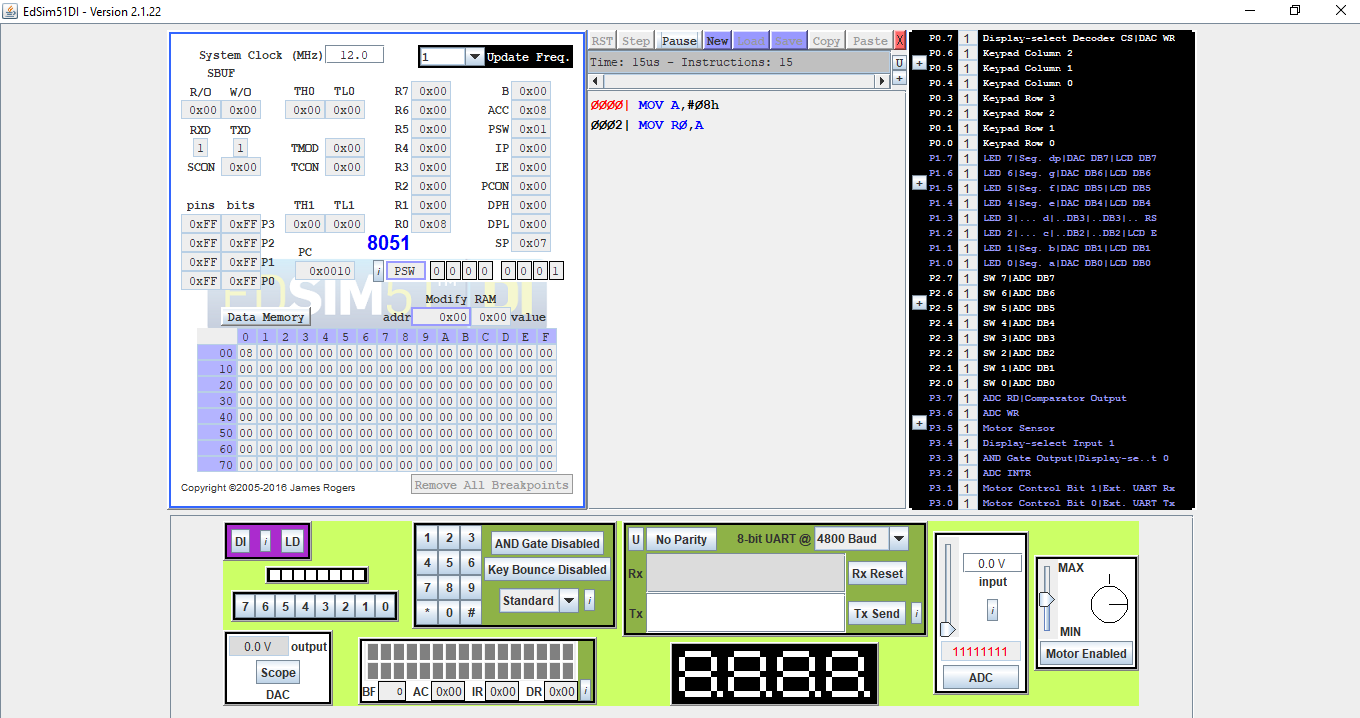
MOV A,#05h



1. **Register Addressing Mode**: In register addressing the operand is placed in one of 8 bit or 16 bit general purpose registers.

MOV A,#08h

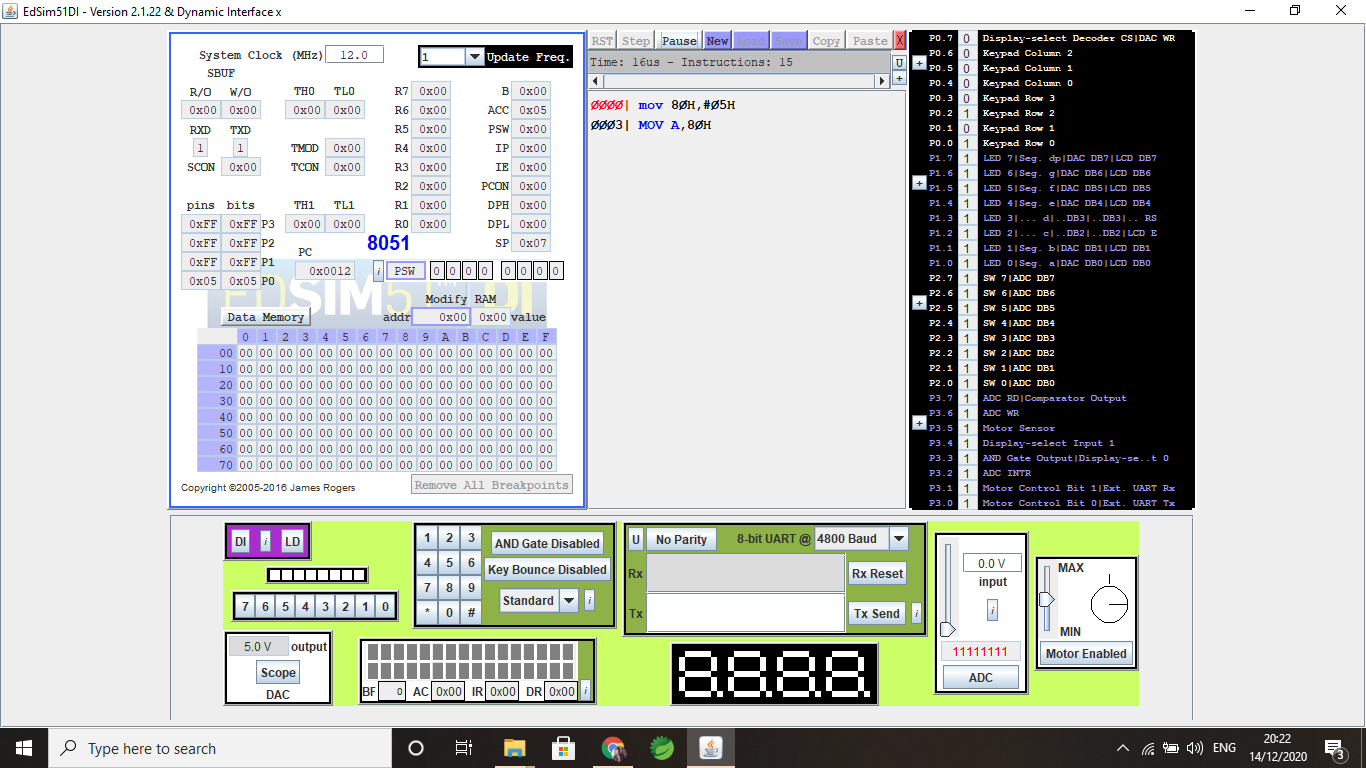
MOV R0,A



1. **Direct Addressing Mode**

mov 80H,#05H // location 80H has data 5

MOV A,80H // 5 moved to Accumulator



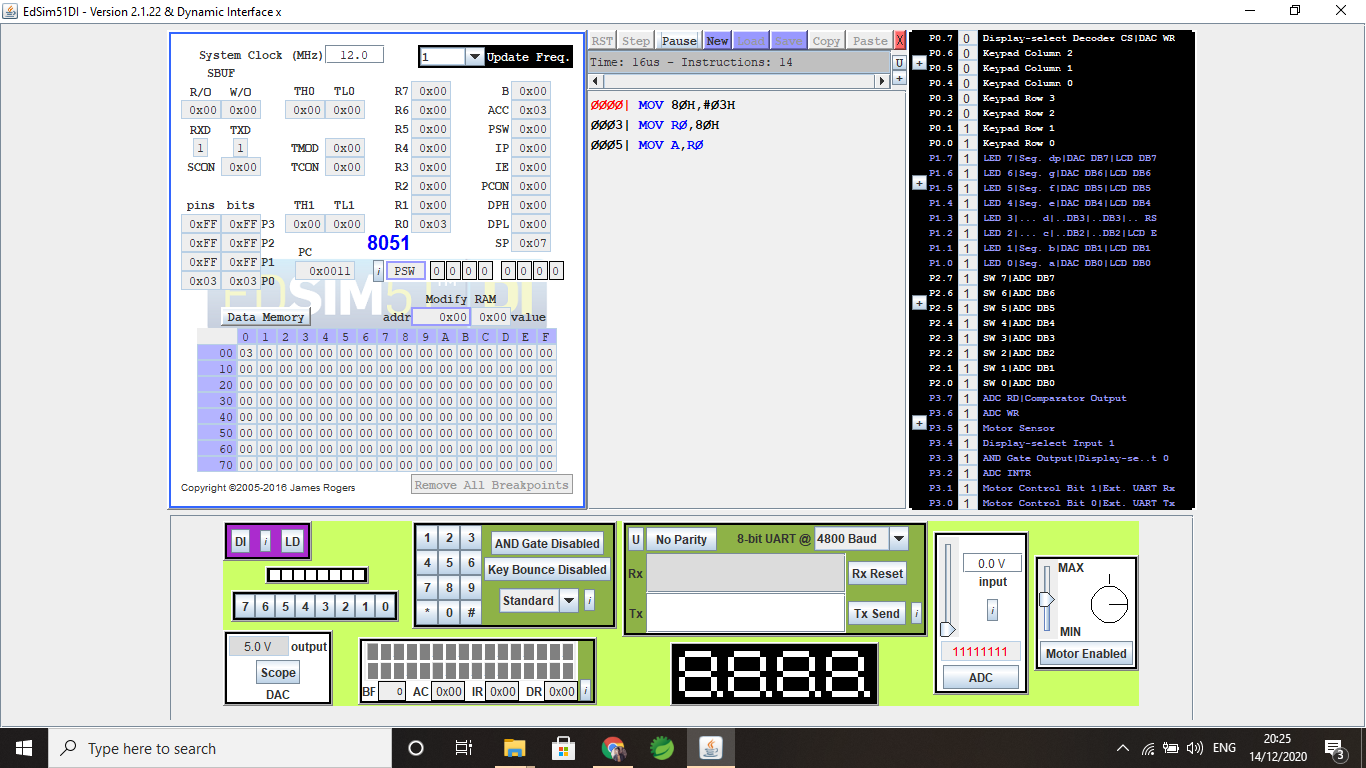
1. **Indirect Addressing Mode**

Location 80H has data 3, whose address is saved in Ro, whose value is moved to A

MOV 80H,#03H

MOV R0,80H

MOV A,R0



**Q3. Write and execute/simulate the assembly language program in EdSim51 based on:**

**Answer:**

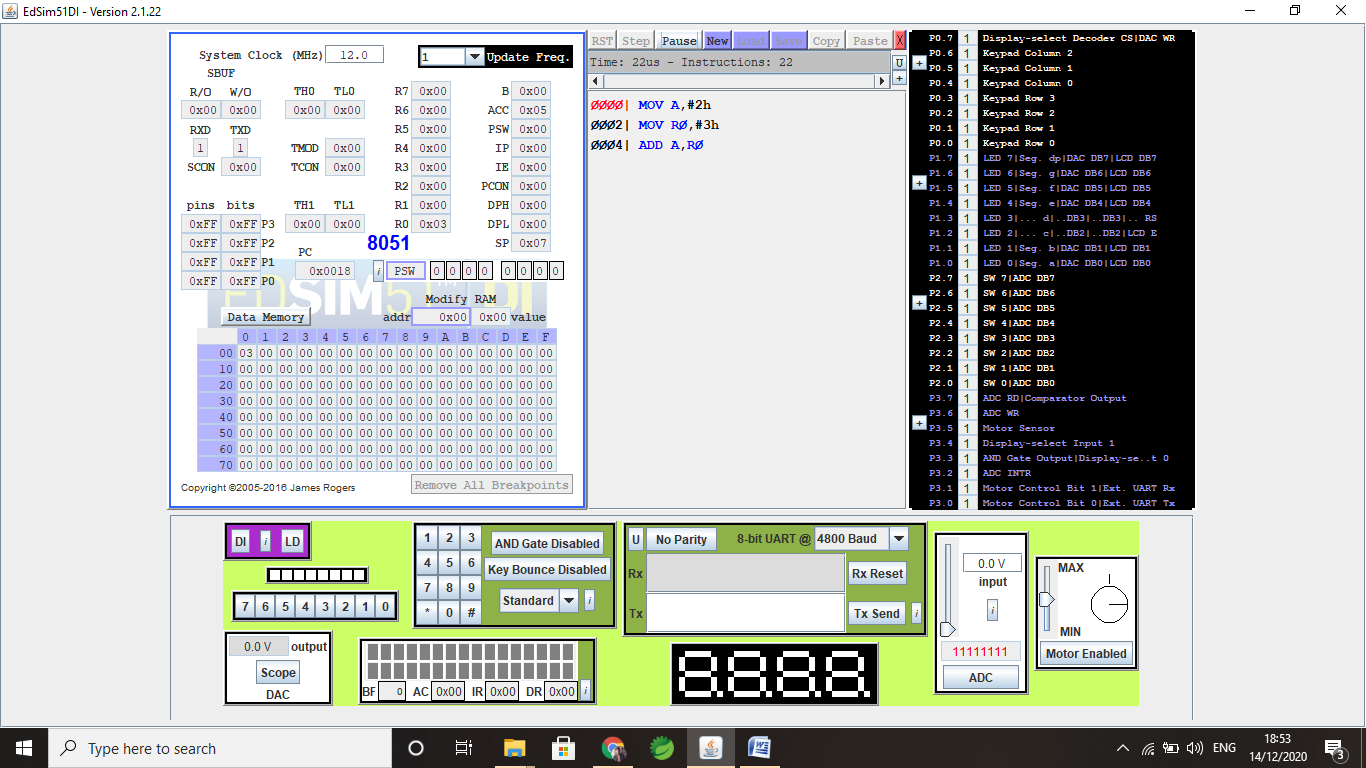
1. **Arithmetic Operations**

* **Addition**

MOV A,#2h

MOV R0,#3h

ADD A,R0

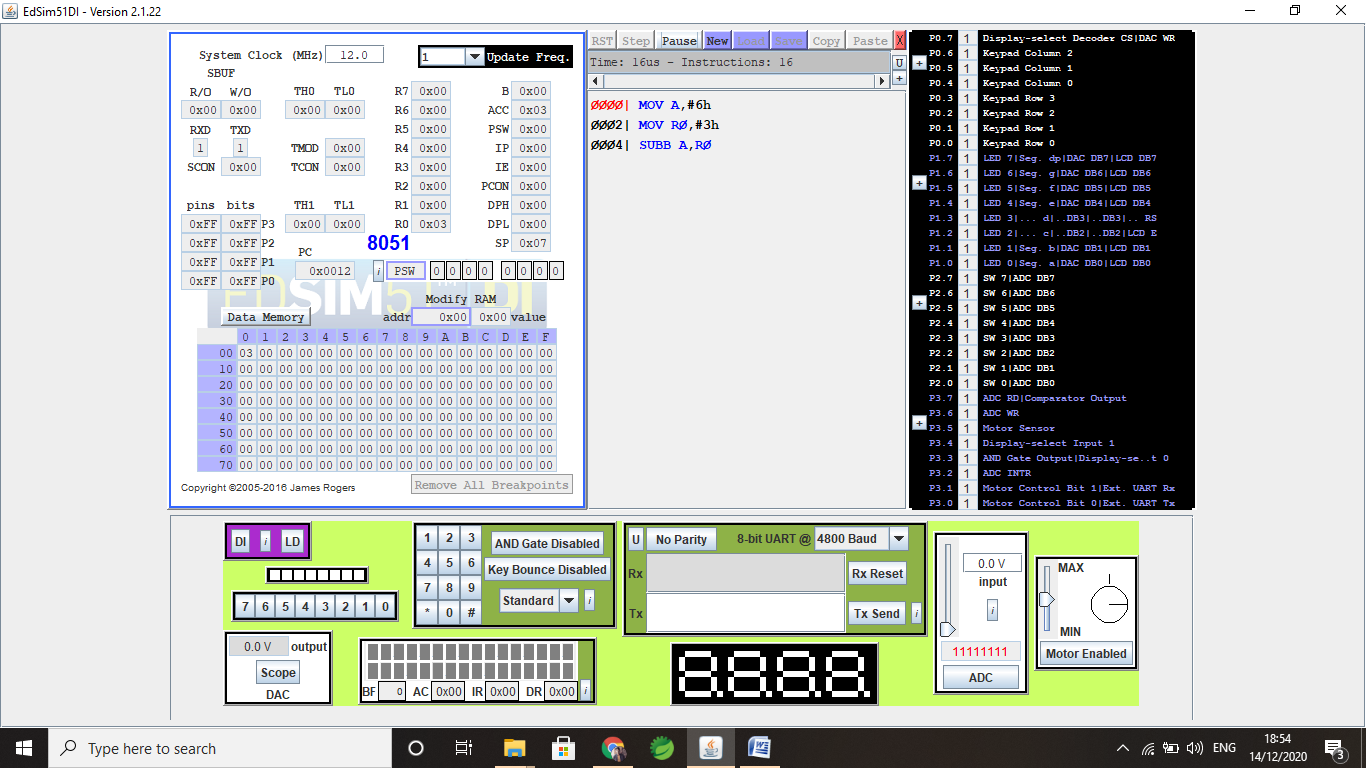


* **Subtraction**

MOV A,#6h

MOV R0,#3h

SUBB A,R0



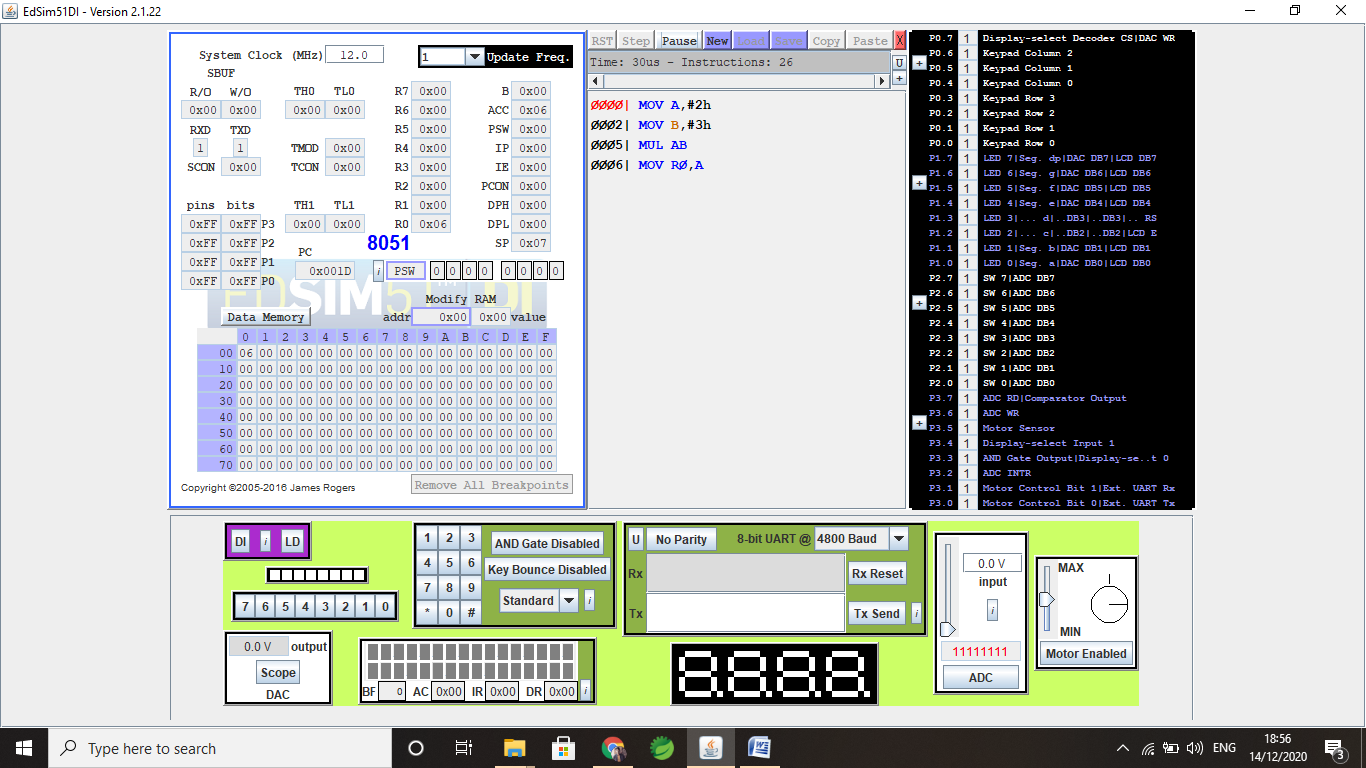
* **Multiplication**

MOV A,#2h

MOV B,#3h

MUL AB

MOV R0,A



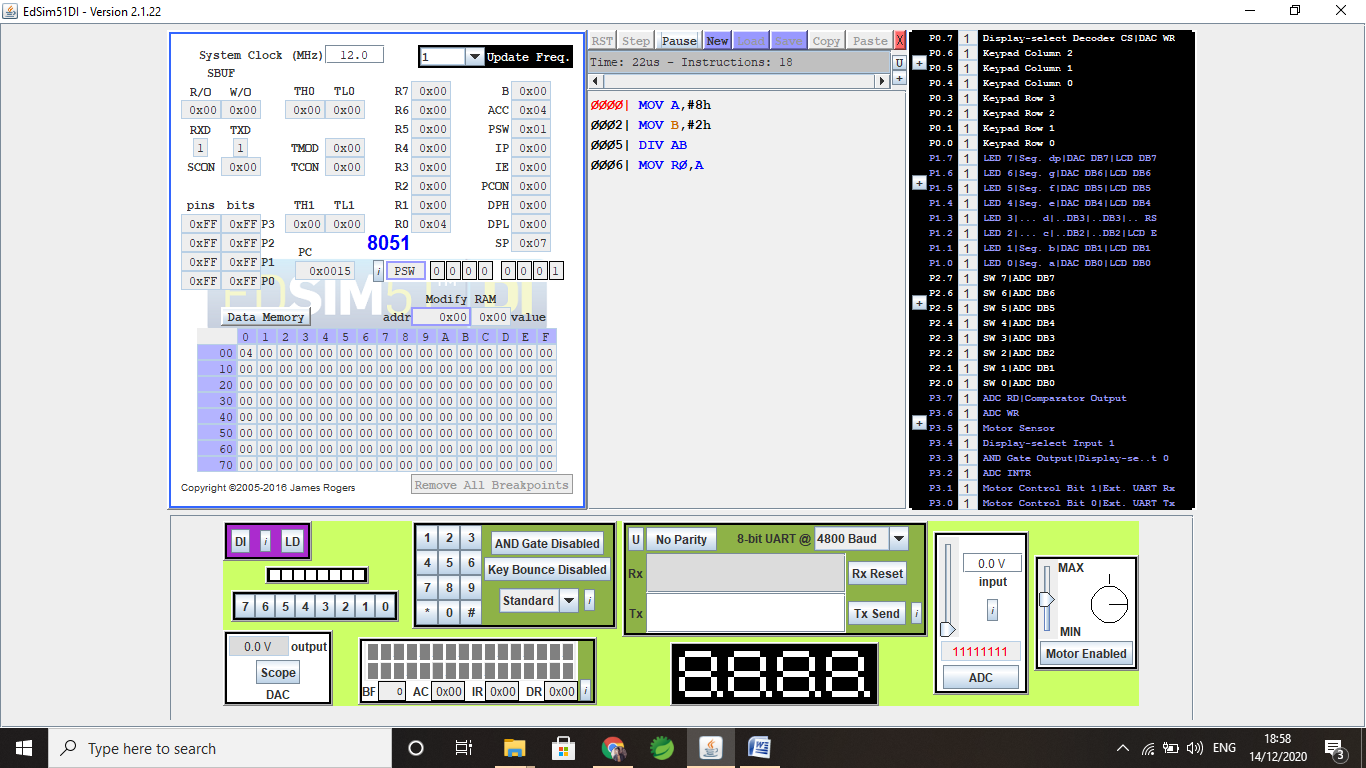
* **Division**

MOV A,#8h

MOV B,#2h

MUL AB

DIV R0,A



1. **Conditional Instruction**

* **JZ**

Initialize A with 5 unit after 3 unit of counter

MOV R0,#3h

loop:

MOV A,R0

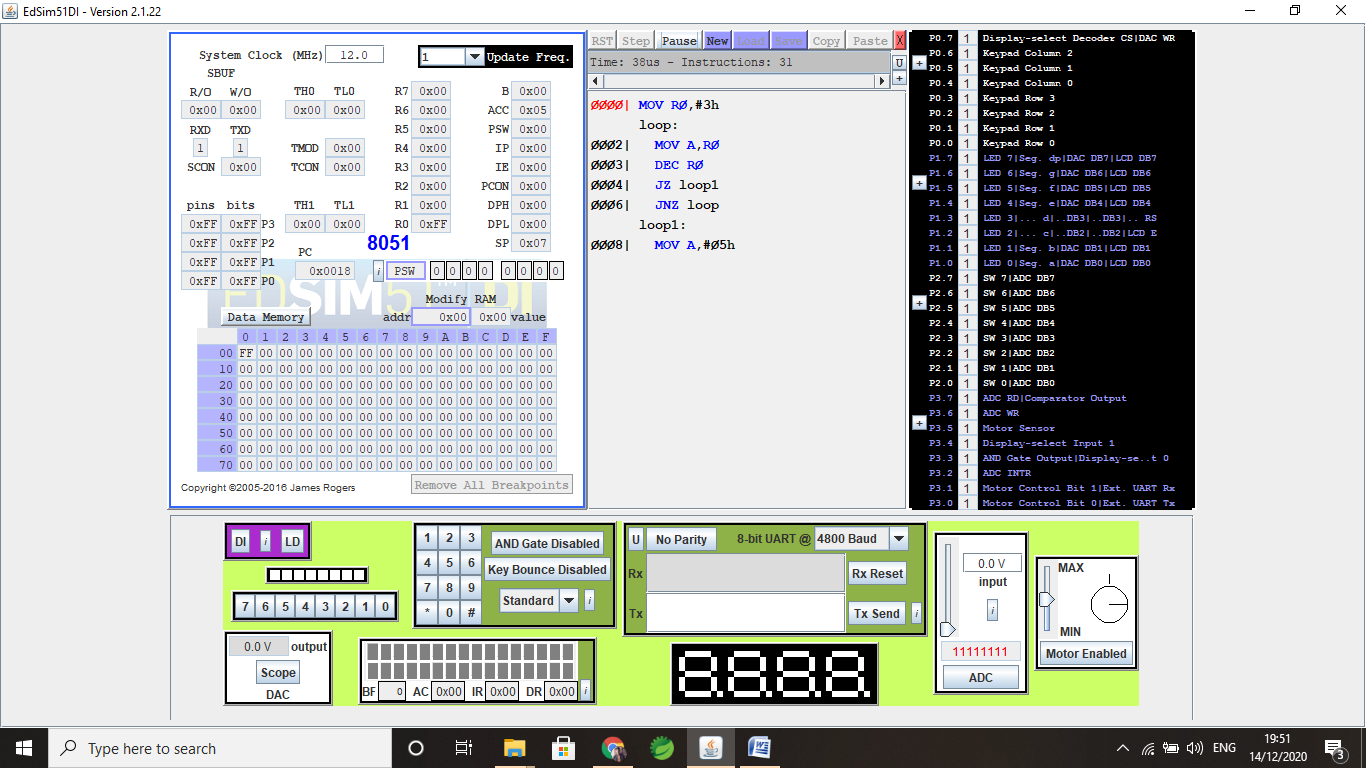
DEC R0

JZ loop1

JNZ loop

loop1:

MOV A,#05h



* **JNZ**

Timer/Counter for 3 unit

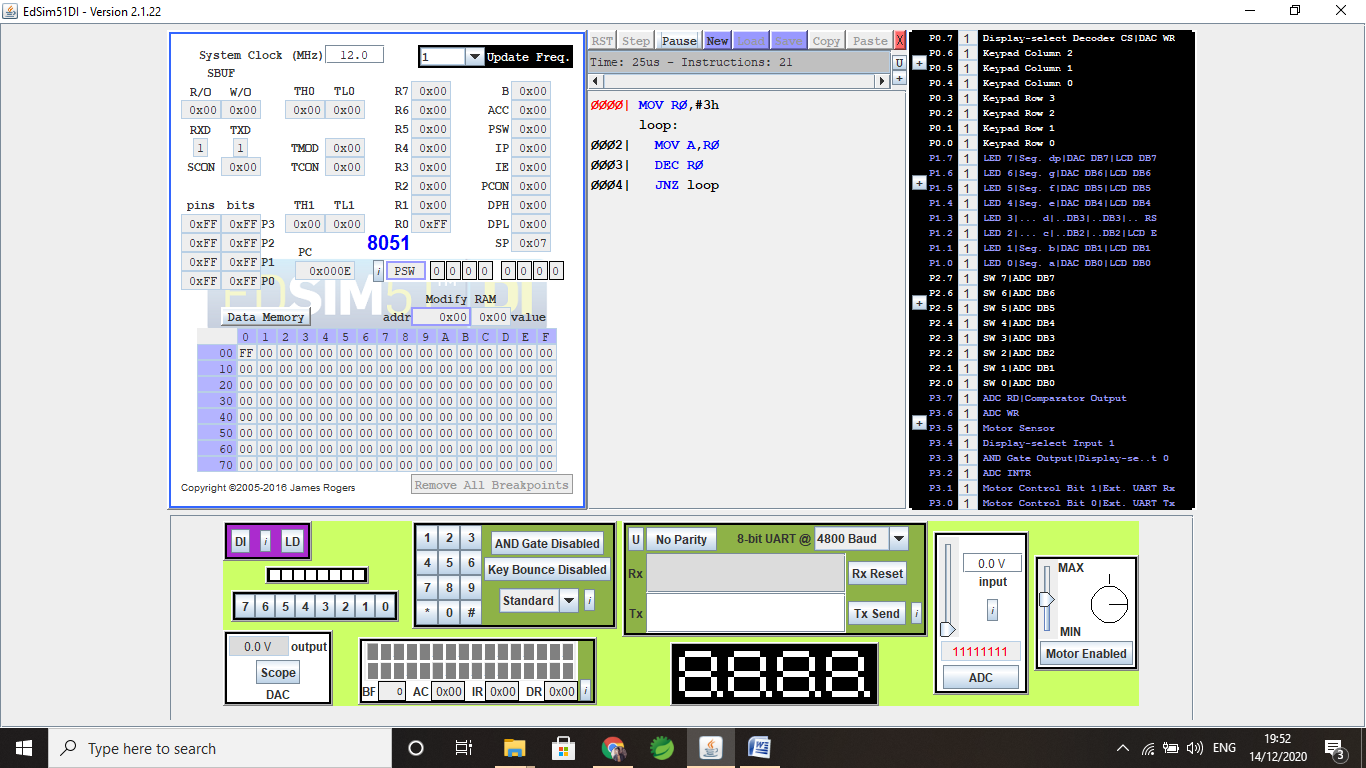
MOV R0,#3h

loop:

MOV A,R0

DEC R0

JNZ loop



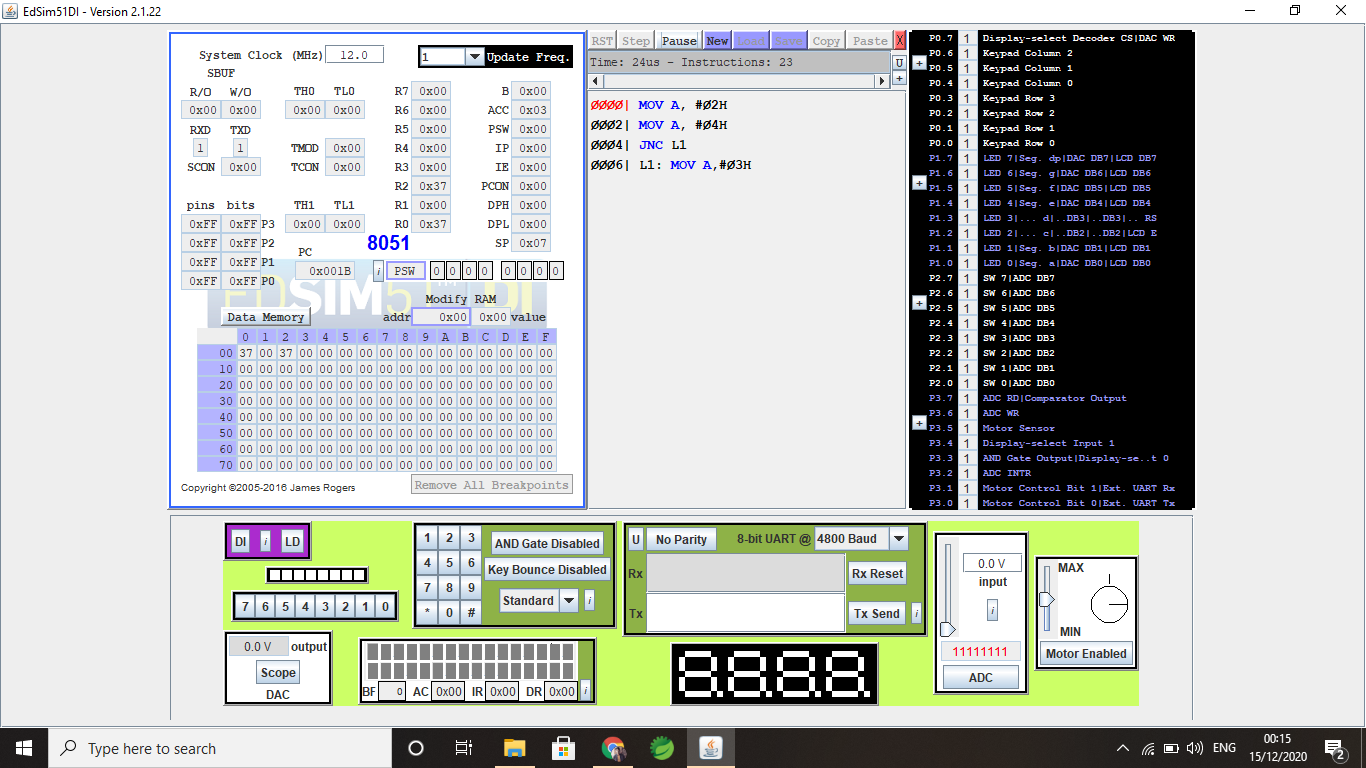
* **JNC**

MOV A, #02H

MOV A, #04H

JNC L1

L1: MOV A,#03H



1. **Loop Instructions**

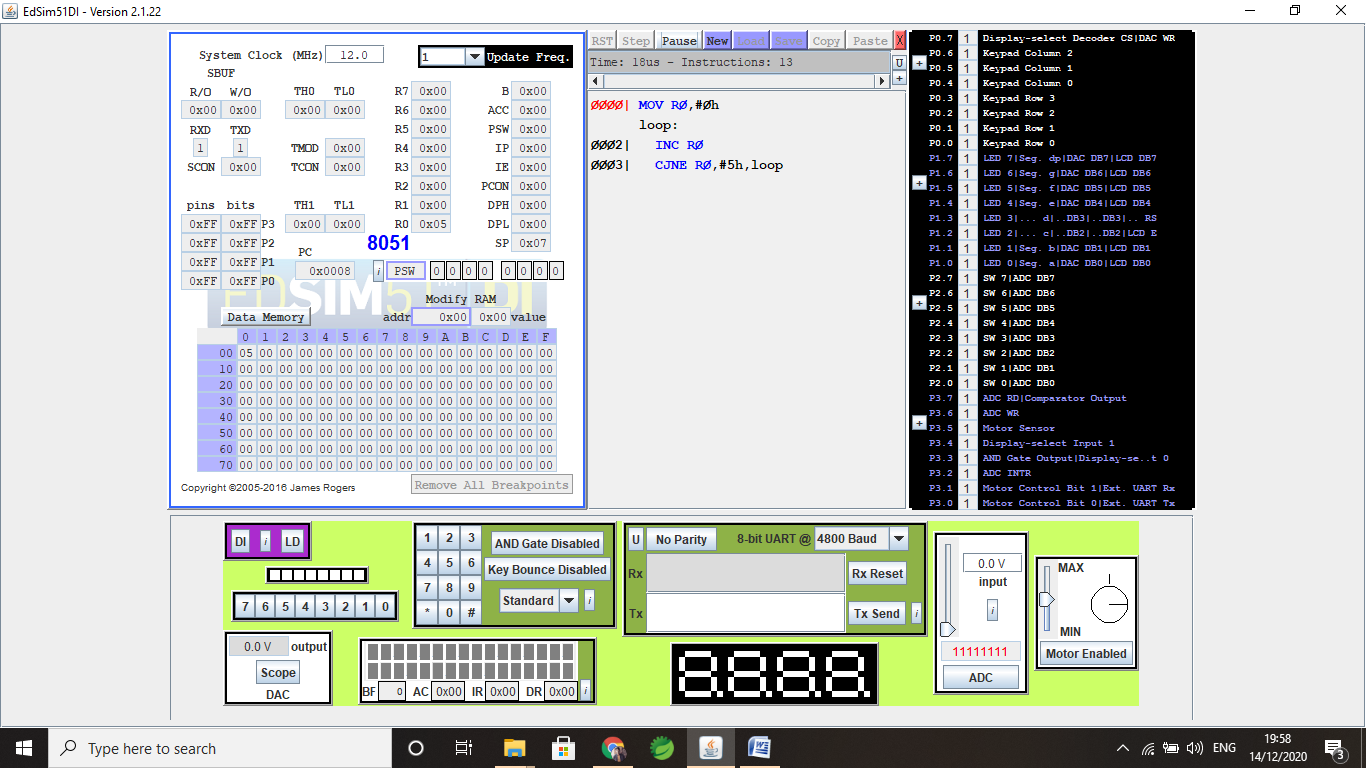
* **CJNE - Counter of 5 unit**

MOV R0,#0h

loop:

INC R0

CJNE R0,#5h,loop

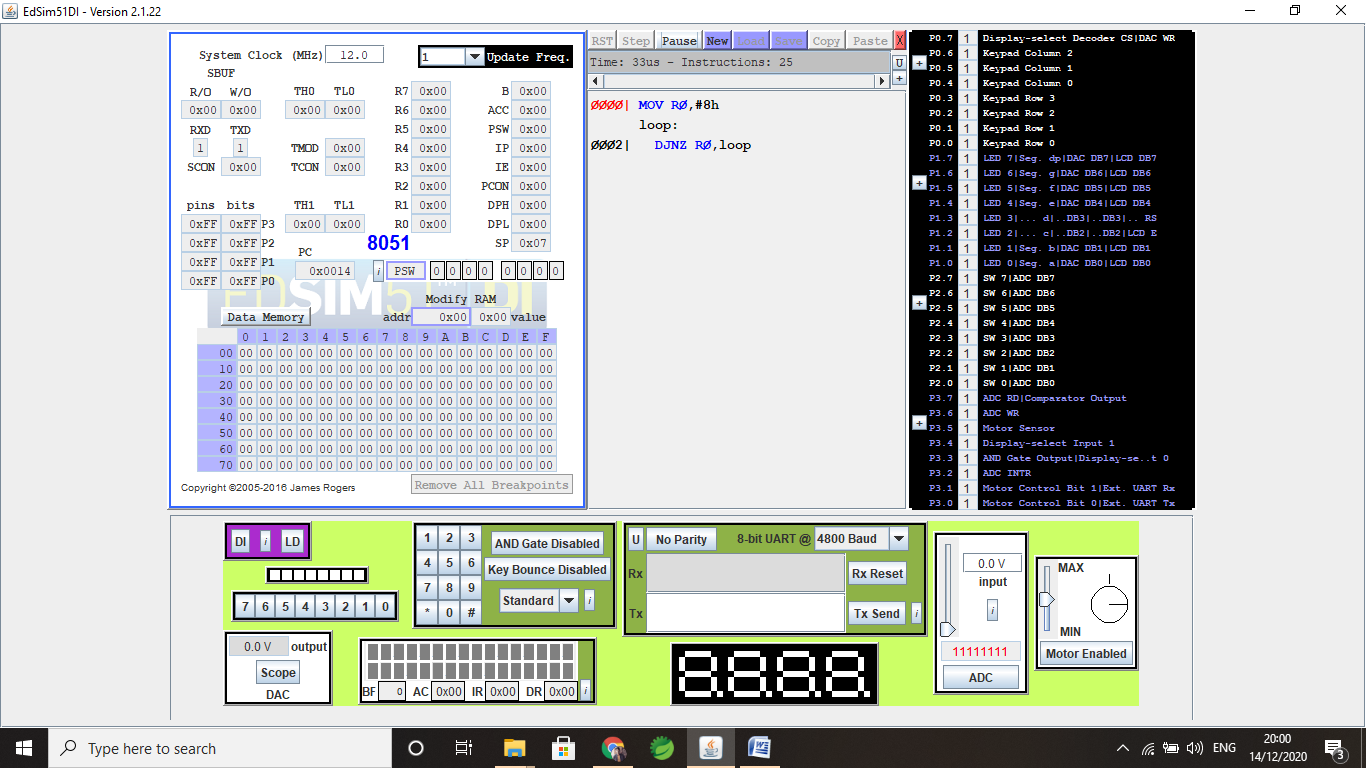
****

* **DJNZ-Counter of 8 unit**

MOV R0,#8h

loop:

DJNZ R0,loop

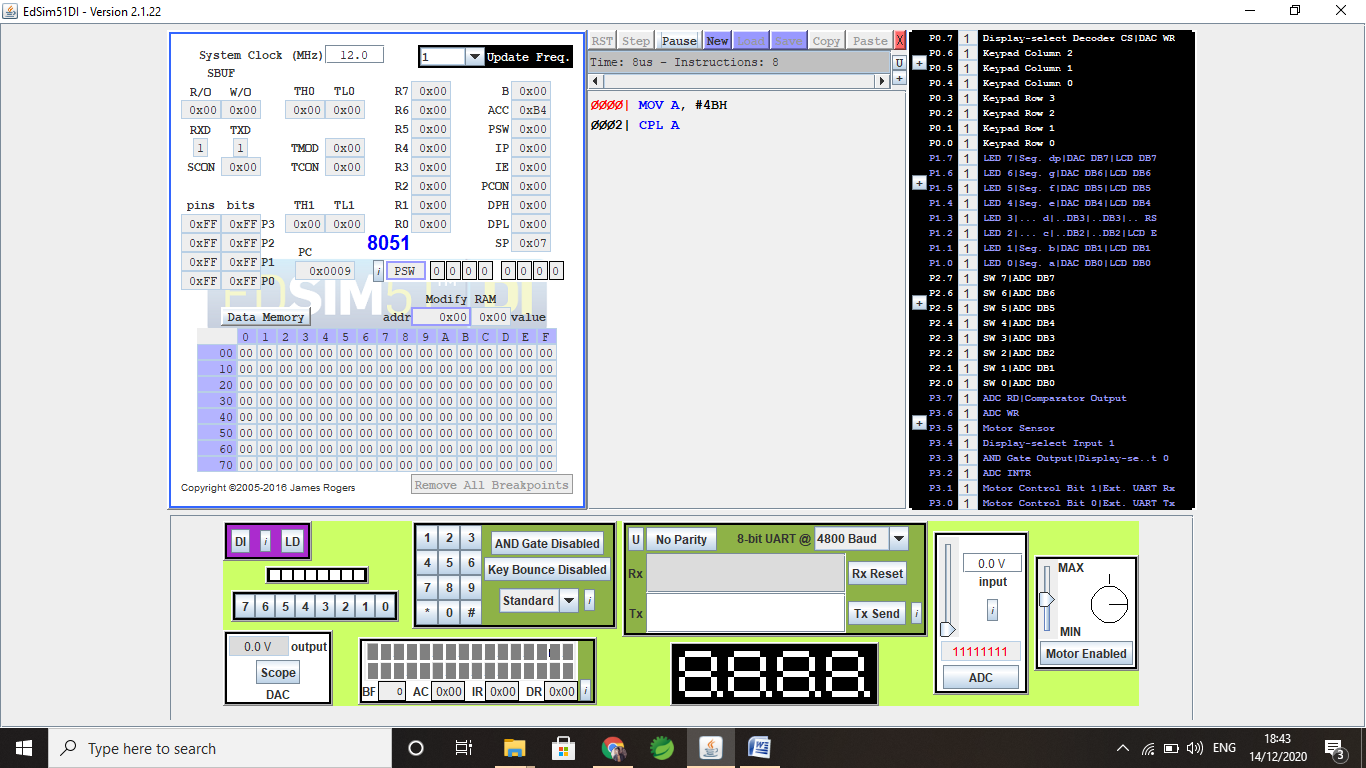
****

1. **Logical Instructions**

* **NOT:**

MOV A, #5BH

CPL A

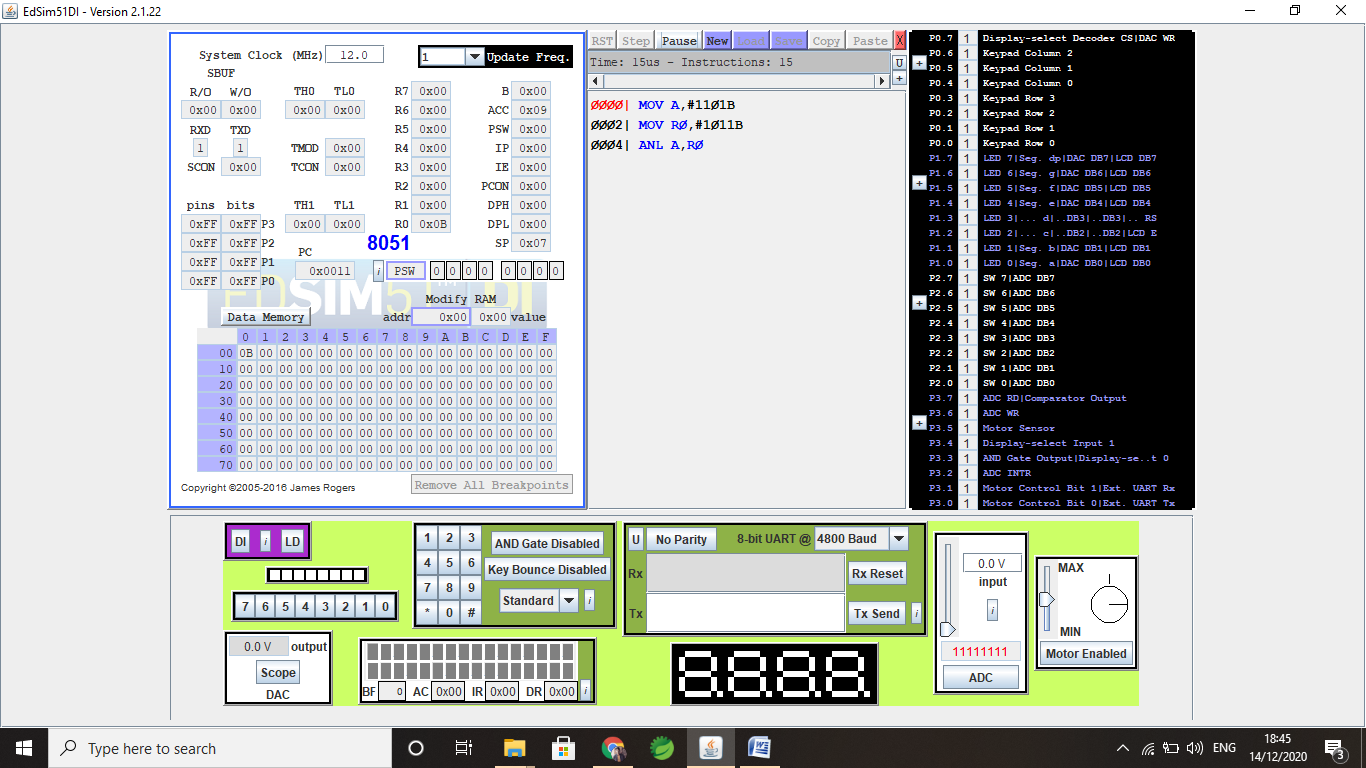


* **AND:**

MOV A,#1101B

MOV R0,#1011B

ANL A,R0

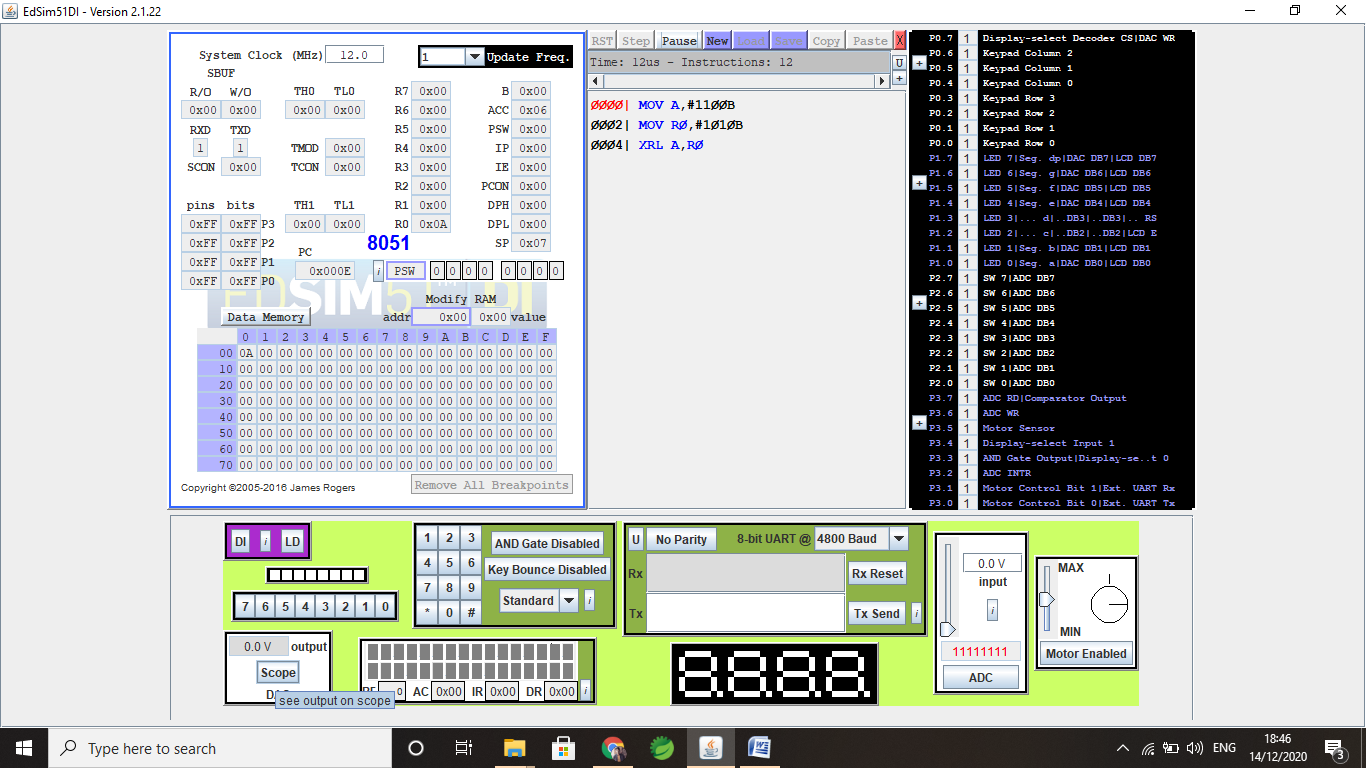


* **XOR:**

MOV A,#1100B

MOV R0,#1010B

XRL A,R0

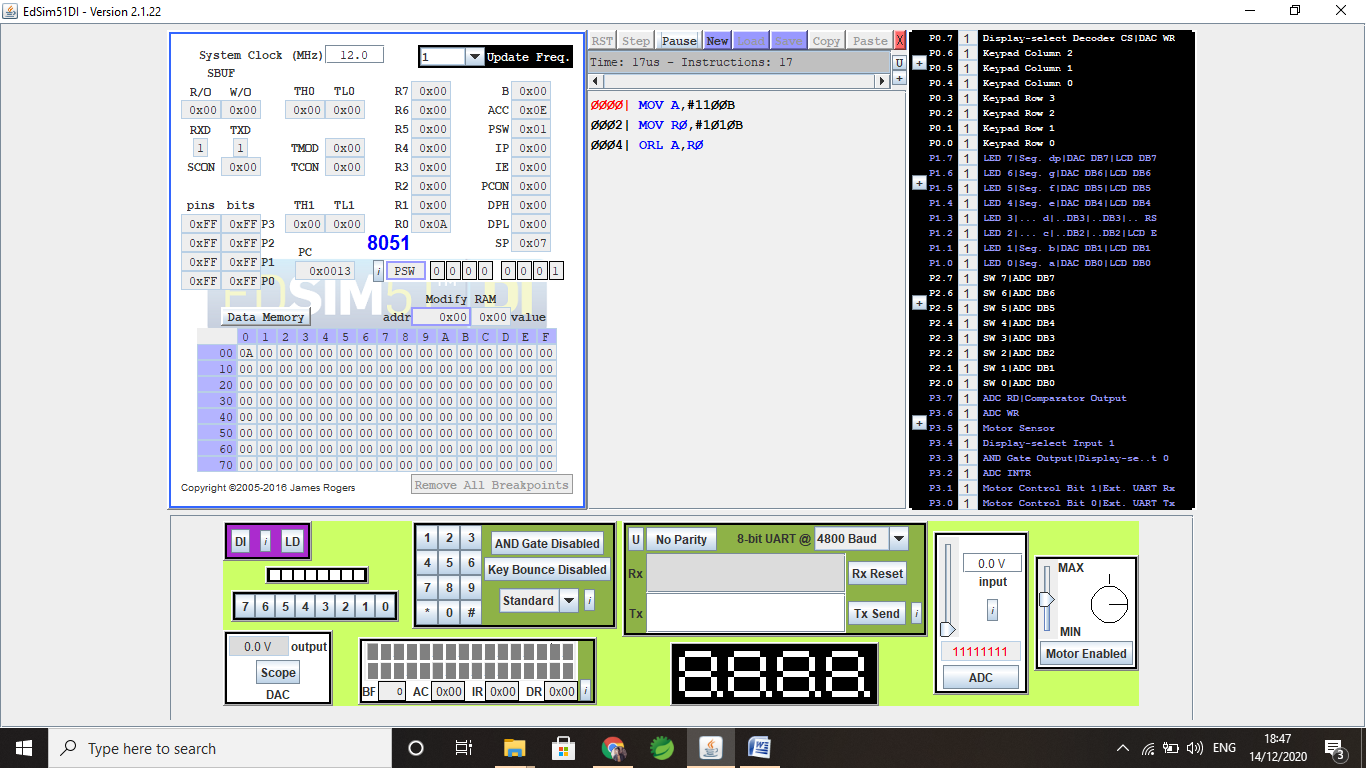


* **OR:**

MOV A,#1100B

MOV R0,#1010B

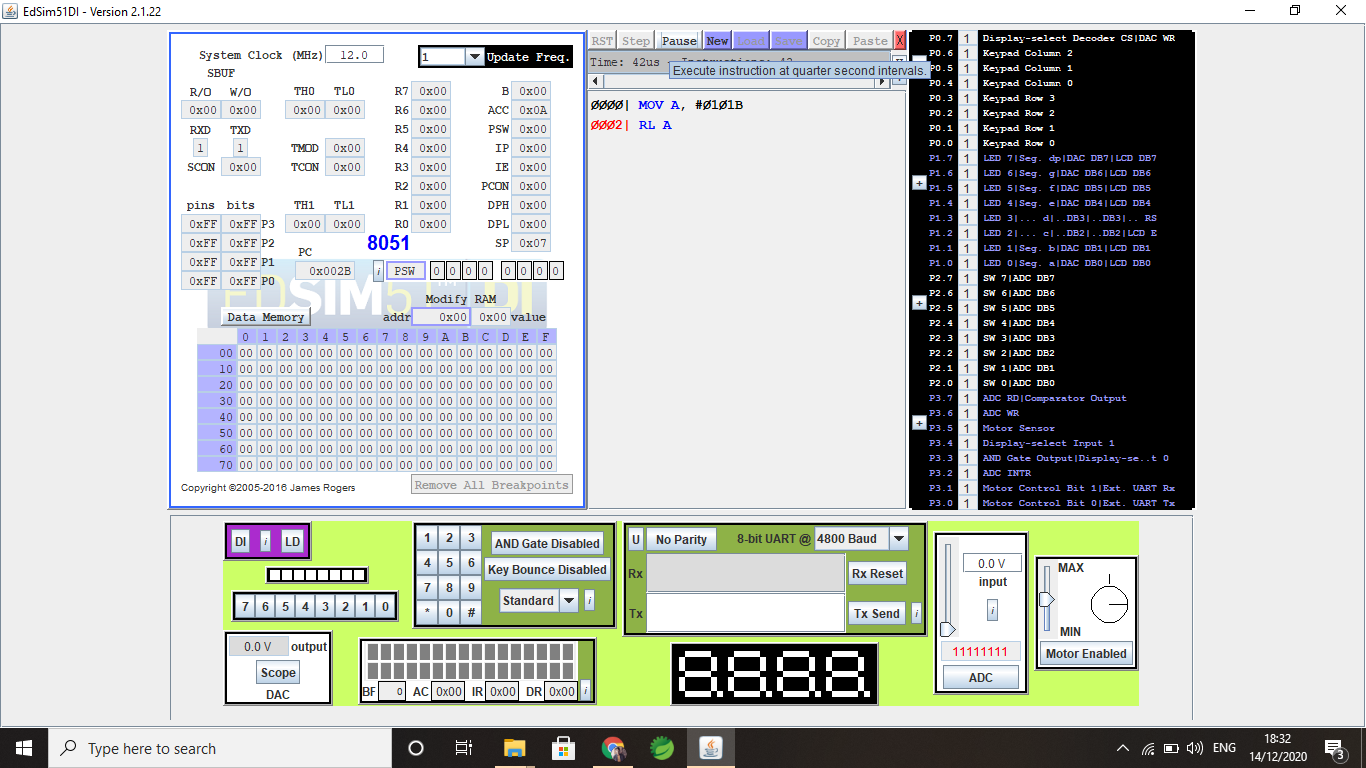
ORL A,R0



1. **Shifting Operations**

* **Rotate Left:**

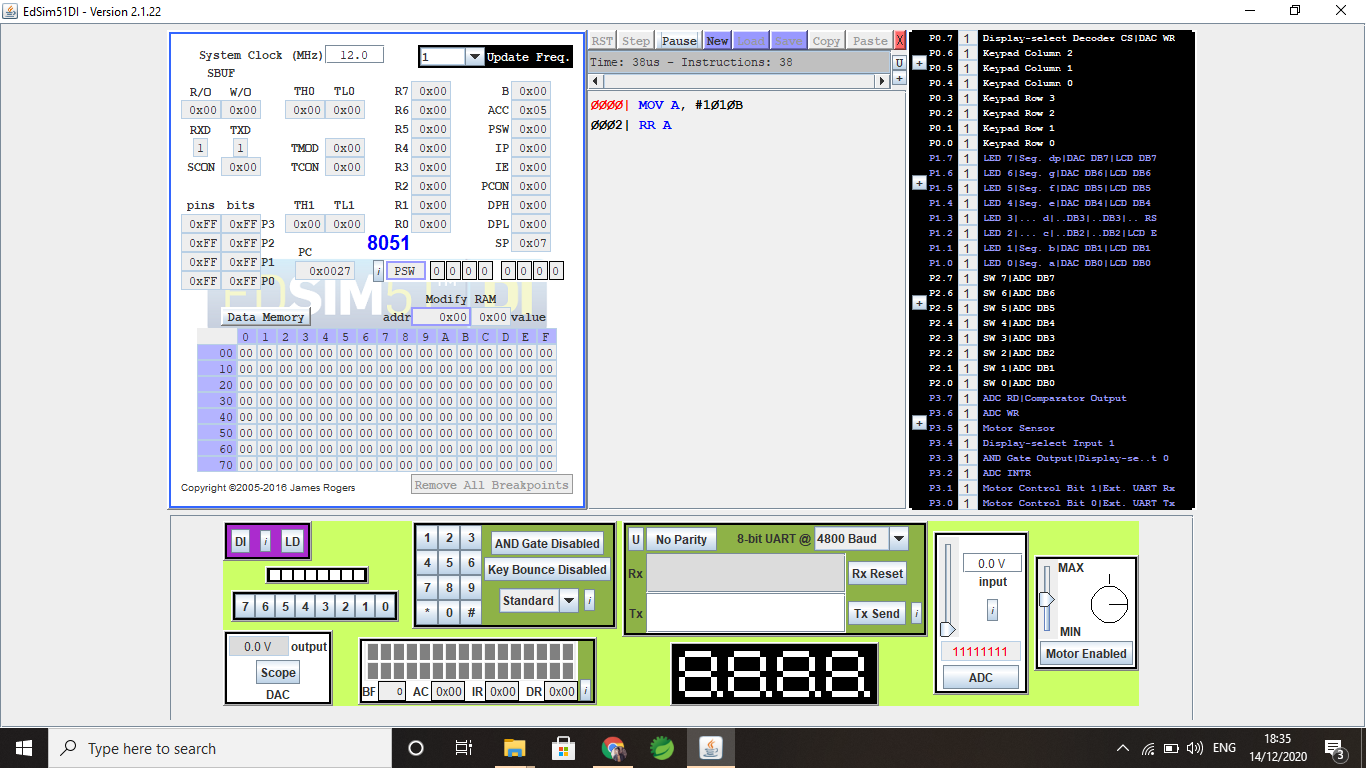
MOV A, #1010B

RL A 

* **Rotate Right:**

MOV A, #1010B

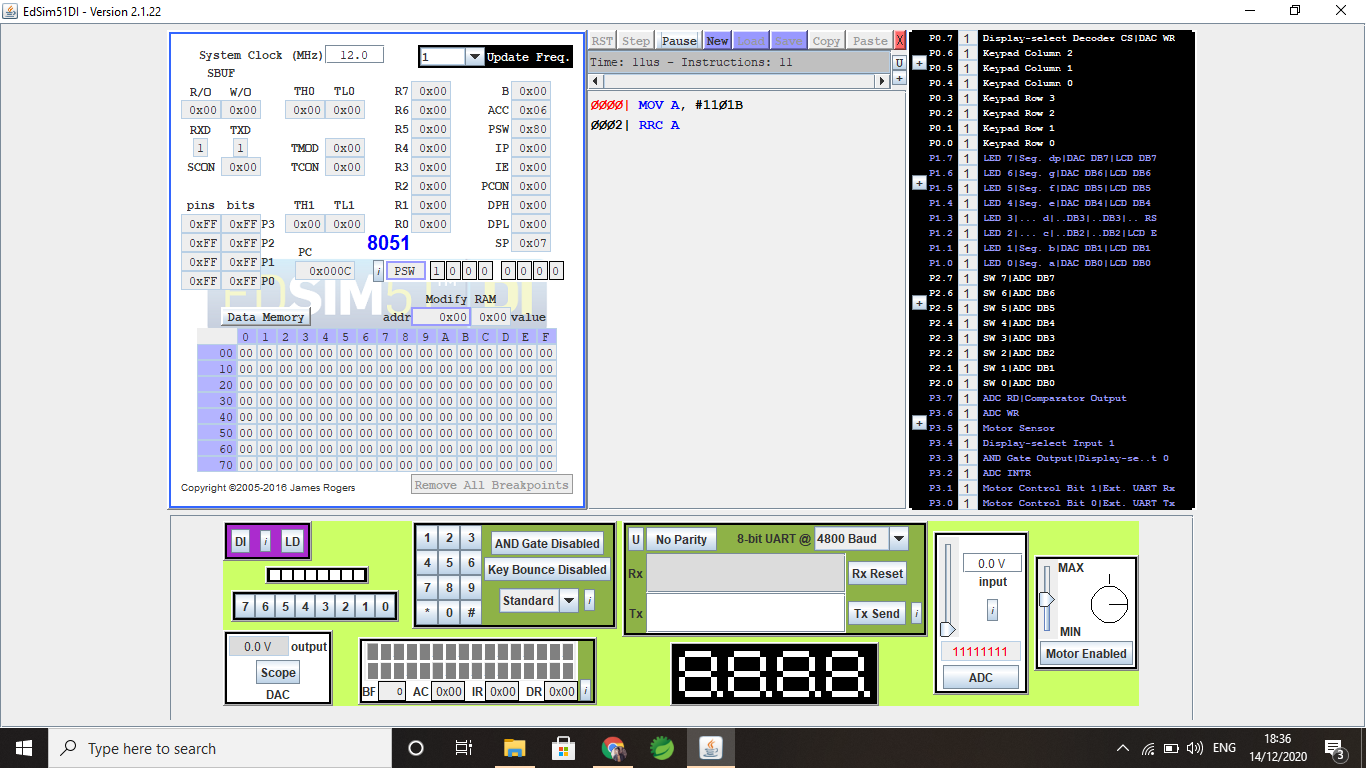
RR A



* **Rotate Right with Carry**:

MOV A, #1101B

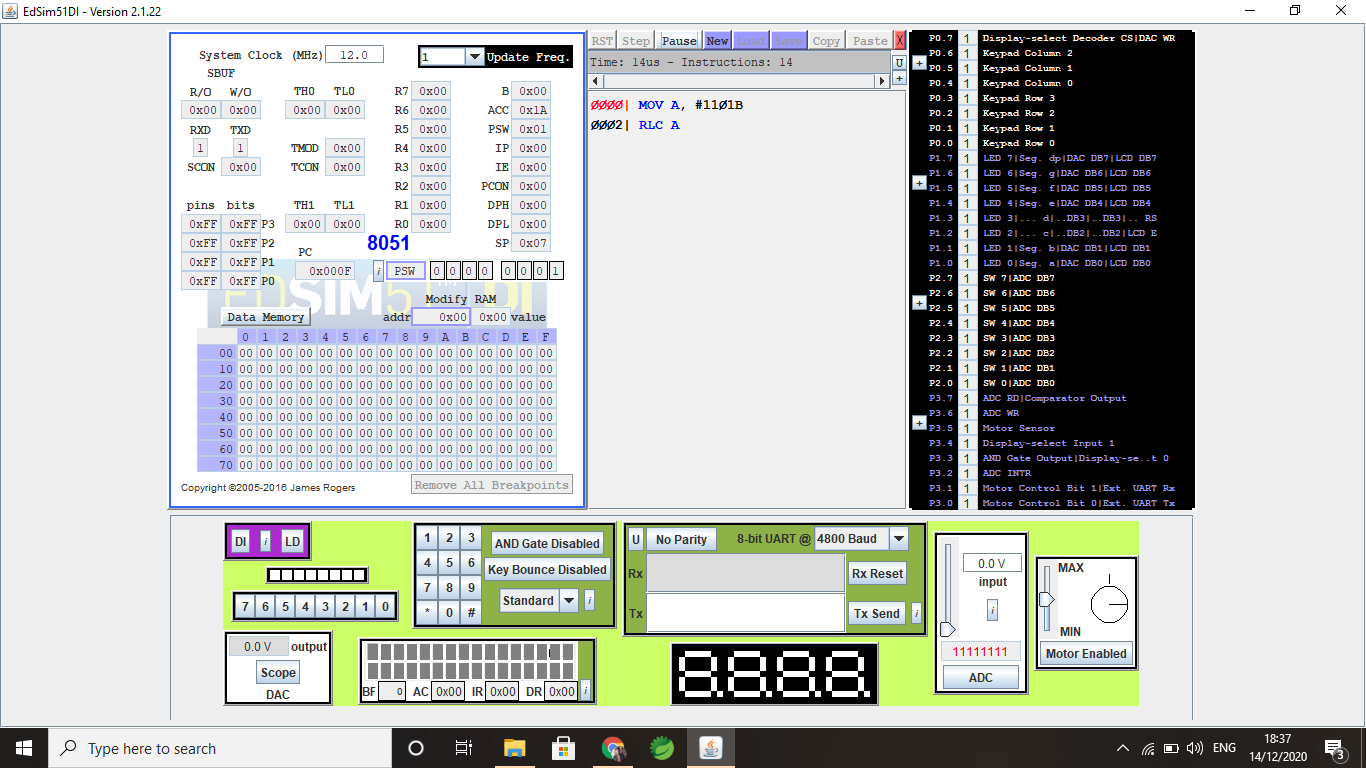
RRC A



* **Rotate Left with Carry:**

MOV A, #1101B

RLC A



**Q4. Write and execute/simulate the assembly language program in EdSim51 to copy a 5Byte data block from one memory location to another memory location.**

Start:

MOV R0, #30H

MOV A, #00H

MOV R7, #07H

MOV R2, #0200H

BACK : MOV A, R0

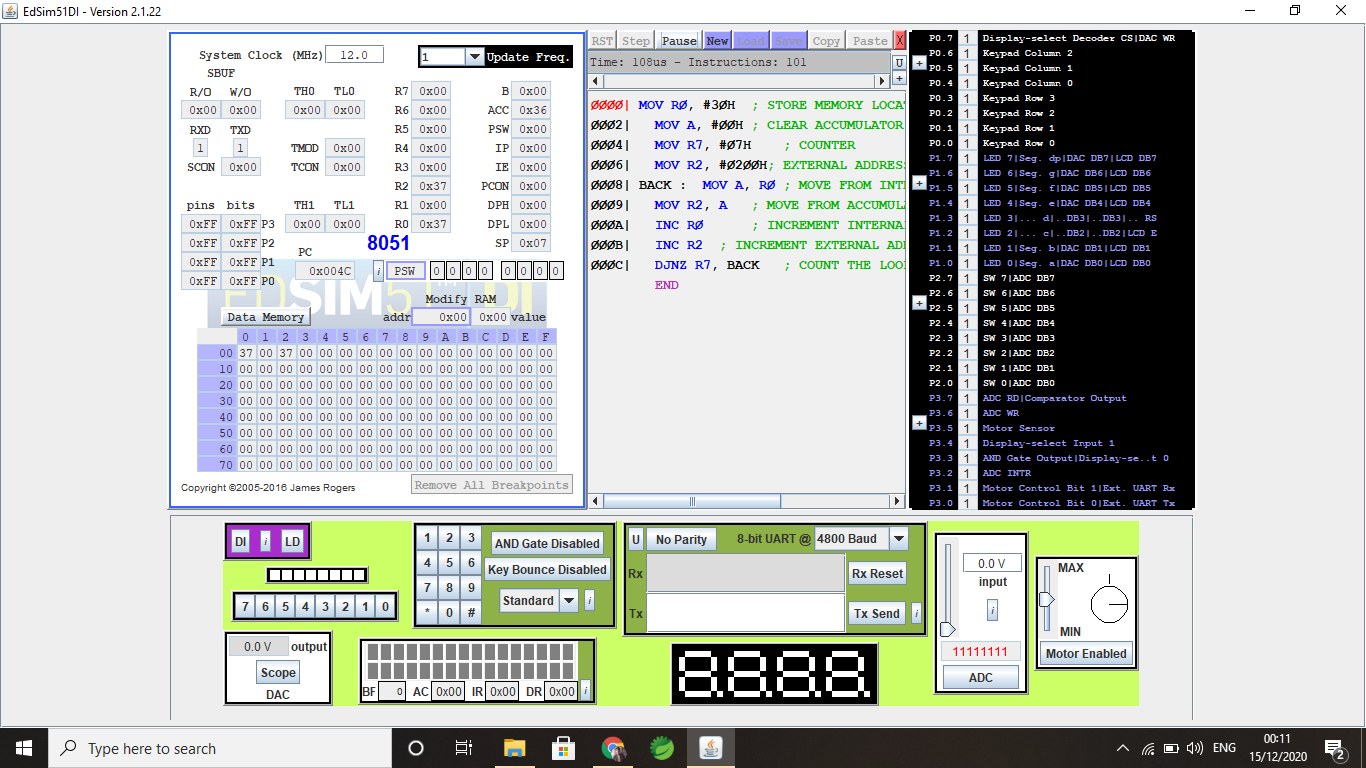
MOV R2, A

INC R0

INC R2

DJNZ R7, BACK

END

****

**Q5. Write and execute/simulate the assembly language program in EdSim51 to sum of first 10 natural numbers. Store the sum in some register.**

**Answer:**

ORG 0000h

MOV R0,#0Ah

MOV R1,#01h

loop:

ADD A,R1

INC R1

DJNZ R0, loop

MOV R2,A

End

