8 Bit Arithmetic Operations using 8051

Expt No: 12 **Name:** Mahesh Bharadwaj K
Date : 22/10/2020 **Reg No:** 185001089

Aim:

To write and execute 8051 ALP for

- Addition
- Subtraction
- Multiplication
- Division

Of 8 bit numbers

Algorithm:

1. Addition

* Initialise R0 with 00h.
* Move the value at R1 into A using MOV.
* Add value at R2 to A using ADD.
* IF carry flag is set, increment R0
* Move R0 to R3(carry) and A to R4(sum)

2. Subtraction

* Initialise R0 with 00h.
* Move the value at R1 into A using MOV.
* Clear Carry Flag using CLR.
* Subtract value at R2 from A using SUBB.
* IF borrow(carry) flag is set, increment R0, and Complement A register using CPL.
* Move R0 to R3(sum) and A to R4(difference).

3. Multiplication

* Initialise R0 with 00h.
* Move the value at R1 into A using MOV.
* Move the value at R2 into B using MOV.
* Multiply A & B using MUL AB.
* Move lower order bytes into R4 from A.
* Move higher order bytes into R5 from B.

4. Division

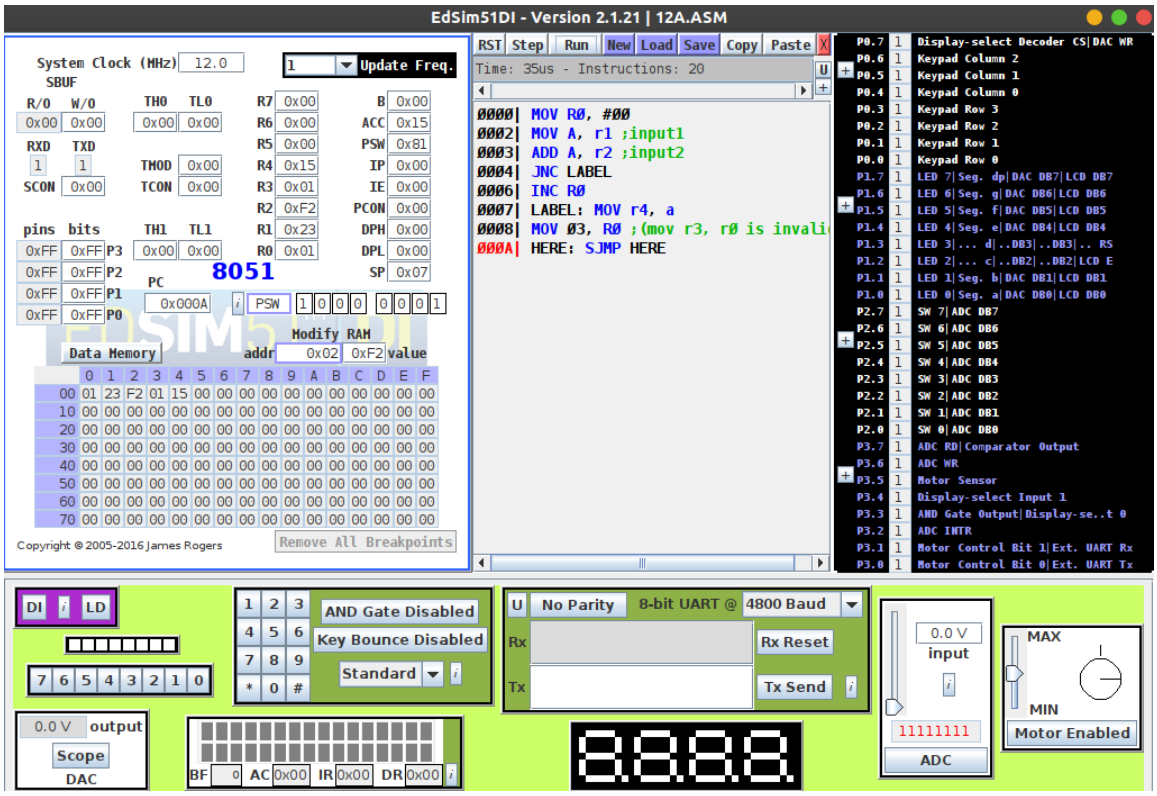
* Initialise R0 with 00h.
* Move the value at R1 into A using MOV.
* Move the value at R2 into B using MOV.
* Divide A & B using DIV AB.
* Move quotient into R4 from A.
* Move remainder into R5 from B.

Program:

1. Addition

Program		Comments
	MOV R0, 00	Initialise R0 with 00
	MOV A, r1	input1
	ADD A, r2	add input1 and input2
	JNC LABEL	jump if no carry
	INC R0	Increment R0 if carry
LABEL:	MOV A4, A	Storing sum
	MOV 03, R0	(mov r3, r0 is invalid)
HERE:	SJMP **HERE**	End program

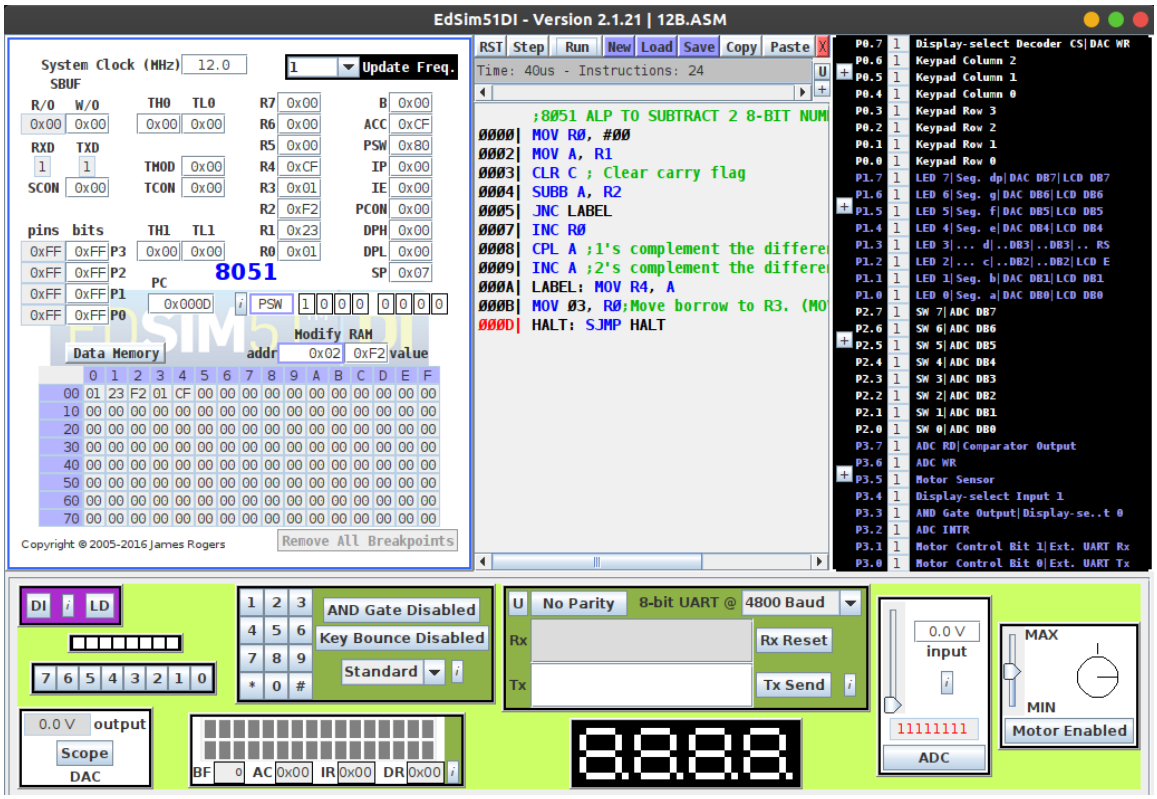
Sample Output:



2. Subtraction

Program		Comments
	MOV R0, 00	
	MOV A, R1	input 1
	CLR C	Clear carry flag
	SUBB A, R2	subtract input2 from input1
	JNC **LABEL**	Jump if no borrow
	INC R0	Set sign flag
	CPL A	1's complement the difference
	INC A	2's complement the difference
LABEL:	MOV R4, A	Moving difference to R4
	MOV 03, R0	Move borrow to R3. (MOV R3, R0) is invalid
HALT:	SJMP **HALT**	Terminate Program

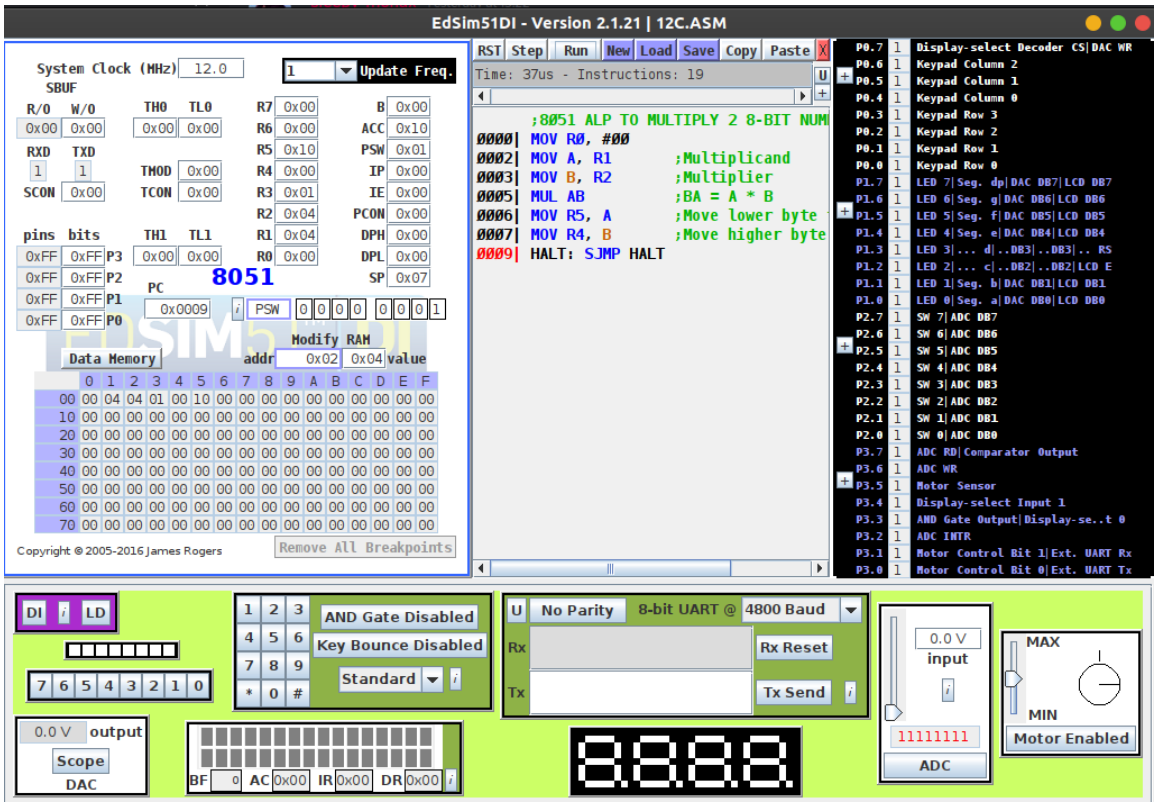
Sample Output



3. Multiplication

Program	Comments
MOV R0, 00	Initialise R0 with 00
MOV A, r1	Multiplicand
MOV B, R2	Multiplier
MUL AB	BA = A * B
MOV R5, A	Move lower byte to R5 from A
MOV R4, B	Move higher byte to R4 from B
HERE: SJMP **HERE**	End program

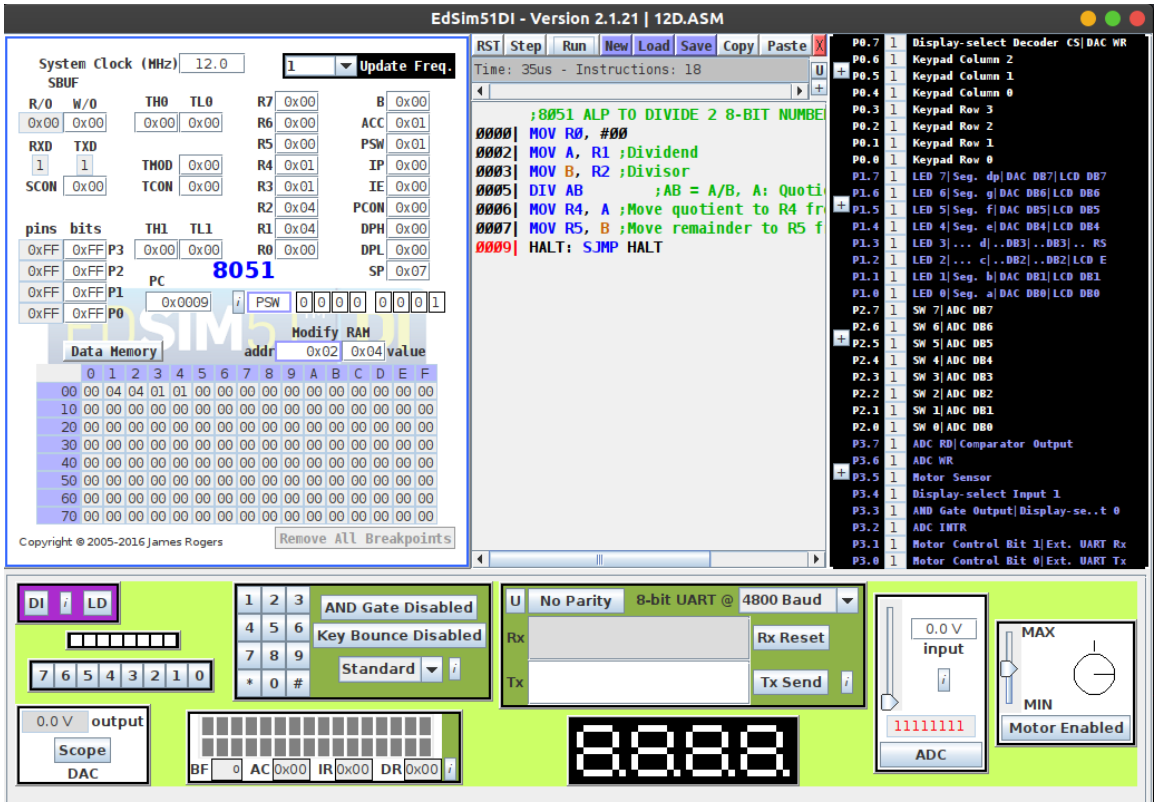
Sample Output:



4. Division

Program	Comments
MOV R0, 00	Initialise R0 with 00
MOV A, r1	Dividend
MOV B, R2	Divisor
DIV AB	AB = A / B, A - quotient, B - remainder
MOV R4, A	Move quotient to R4 from A
MOV R5, B	Move remainder to R5 from B
HERE: SJMP **HERE**	End program

Sample Output:



Result:

8051 ASL programs for 8 bit arithmetic operations has been executed successfully.