# UCS1512 – Microprocessors Lab

# Case Conversion

Exp no : 12 Name: Sreedhar V

Date : 02-11-2020 Reg no: 185001161

# AIM:

To program and execute the program for 8-bit arithmetic operations using 8051 microcontroller using EDSim.

# 8-bit addition:

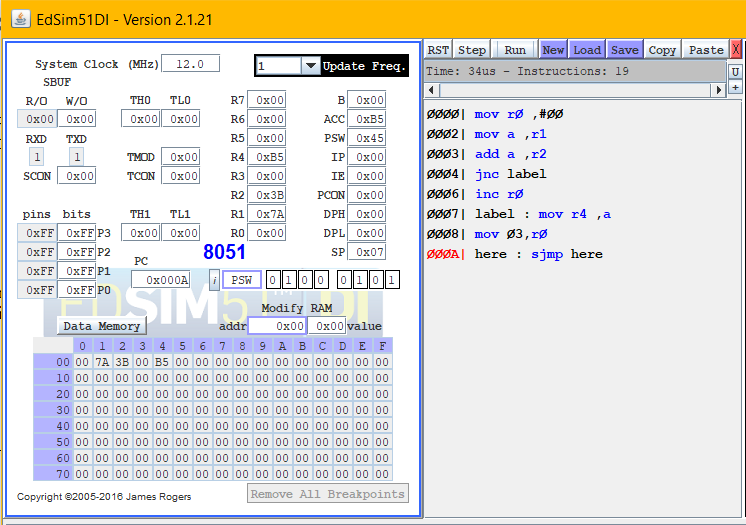
## Algorithm:

* + Move the input1 to A.
  + Move 00h to R0 to make store the carry.
  + Add the input2 to A and store the result in A.
  + Jump if no carry to label.
  + Increment R0 to make carry.
  + Label: move the result to register 4 and carry to register 3.
  + Terminate the program.

## Program:

|  |  |
| --- | --- |
| CODE | COMMENT |
| Program for 8-bit Addition:  mov r0 ,#00  mov a ,r1  add a ,r2  jnc label  inc r0  label : mov r4 ,a  mov 03,r0  here : sjmp here | Move value 00 to R0  Move value1 to R1  Add R2 and the value stored in A  Jump to label if no carry is generated  Increment R0 for carry  Move the result to R4  Move the carry to R3  Terminate the program |

Snapshot of sample input and output:



# 8-bit subtraction:

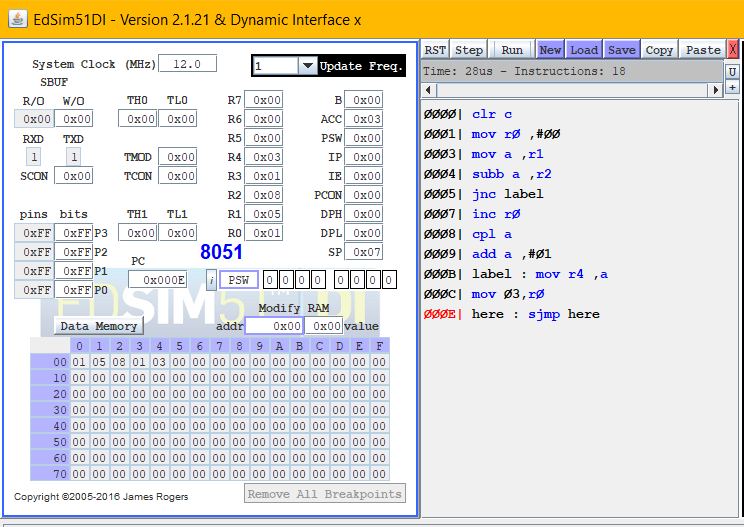
## Algorithm:

* + Move input1 to A.
  + Move 00h to R0 to make store the carry
  + Move the Subtract the second input from A and store the result in A.
  + Jump if no carry to label.
  + Increment R0 to make carry.
  + Complement A if there is a carry
  + Label: move the result to register 4 and carry to register 3.
  + Terminate the program.

## Program:

|  |  |
| --- | --- |
| CODE | COMMENT |
| Program for 8-bit Subtraction:  clr c  mov r0 ,#00  mov a ,r1  subb a ,r2  jnc label  inc r0  cpl a  add a ,#01  label : mov r4 ,a  mov 03,r0  here : sjmp here | Clearing the carry flag  Move value 00 to R0  Move value1 to R1  Subtract R2 from the value stored in A  Jump to label if no carry is generated  Increment R0 for carry  Complement A  Add 01 to A // To get 2’s Complement    Move the result to R4  Move the carry to R3  Terminate the program |

Snapshot of sample input and output:



# 8-bit multiplication:

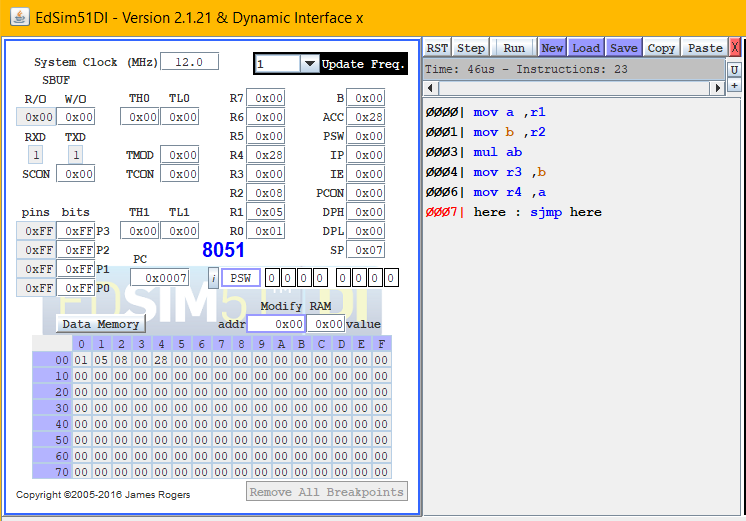
## Algorithm:

* + Move the input1 to A.
  + Move the input2 to B .
  + Multiply A and B.(produces 16-bit output ,stored in BA)
  + Move the higher order byte from B to R3.
  + Move the lower order byte from A to R4.
  + Terminate the program

## Program:

|  |  |
| --- | --- |
| CODE | COMMENT |
| Program for 8-bit Multiplication:  mov a ,r1  mov b ,r2  mul ab  mov r3 ,b  mov r4 ,a  here : sjmp here | Move the input 1 from R1 to A  Move the input 2 from R2 to B  Multiply A and B using MUL  Move the higher order byte from B to R3  Move the lower order byte from A to R4  Terminate the program. |

Snapshot of sample input and output:



# 8-bit Division:

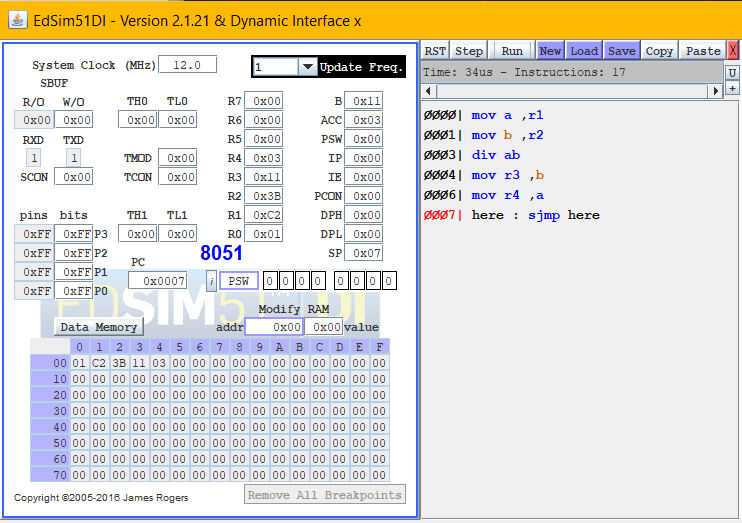
## Algorithm:

* + Move input1 to A.
  + Move input2 to B .
  + Division A and B.
  + Move the remainder from B to R3.
  + Move the quotient from A to R4.
  + Terminate the program.

## Program:

|  |  |
| --- | --- |
| CODE | COMMENT |
| Program for 8-bit Division:  mov a ,r1  mov b ,r2  div ab  mov r3 ,a  mov r4 ,b  here : sjmp here | Move the input 1 from R1 to A  Move the input 2 from R2 to B  Divide A and B using DIV  Move the remainder from B to R3  Move the quotient from A to R4  Terminate the program. |

Snapshot of sample input and output:



Result:

8-bit arithmetic operations is executed and verified using 8051 microcontroller using EDSim.

# UCS1512 – Microprocessors Lab

# Cube of a number

Exp no : 13 Name: Sreedhar V

Date : 02-11-2020 Reg no: 185001161

# AIM:

To program and execute the program for finding cube of a number using 8051 microcontroller using EDSim.

# Cube of a number:

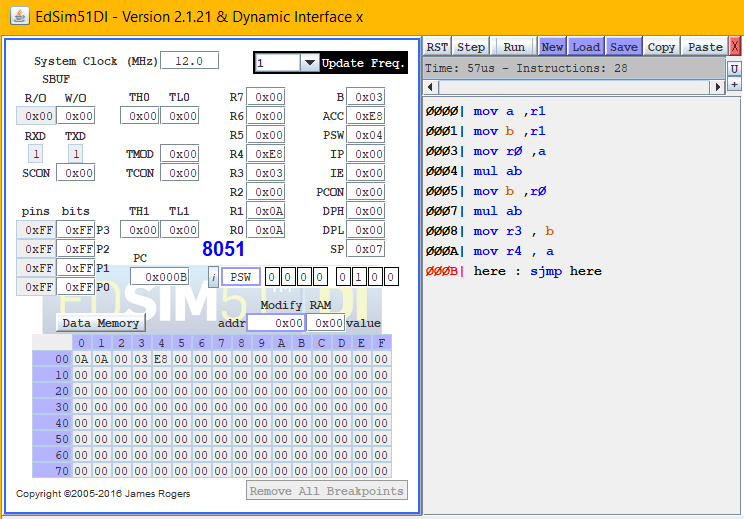
## Algorithm:

* + Move the input to A.
  + Move the input to B.
  + Move the input to R0 to save a copy of input.
  + Multiply A and B.
  + Move the input from R0 to B.
  + Multiply A and B.
  + Move the higher order byte from B to R3.
  + Move the lower order byte from A to R4.
  + Terminate the program.

## Program:

|  |  |
| --- | --- |
| CODE | COMMENT |
| Program for Cube of a number:  mov a ,r1  mov b ,r1  mov r0 ,a  mul ab  mov b ,r0  mul ab  mov r3 , b  mov r4 , a  here : sjmp here | Move the input from R1 to A  Move the input from R1 to B  Move the input from A to R0.  Multiply A and B using MUL  Move the input from R0 to B.  Multiply A and B using MUL  Move the higher order byte from B to R3  Move the lower order byte from A to R4  Terminate the program |

Snapshot of sample input and output:



Result:

Cube of a number is executed and verified using 8051 microcontroller using EDSim.

# UCS1512 – Microprocessors Lab

# Conversion of BCD to ASCII

Exp no : 14 Name: Sreedhar V

Date : 02-11-2020 Reg no: 185001161

# AIM:

To program and execute the program for finding cube of a number using 8051 microcontroller using EDSim.

# Cube of a number:

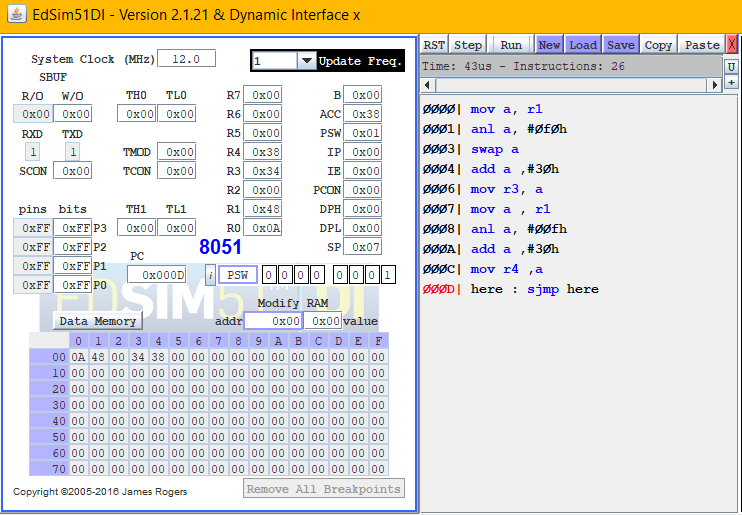
## Algorithm:

* + Move the input to A.
  + Get the higher order byte of A by performing logical AND over A and F0h
  + Swap A to get the higher order byte.
  + Add A with 30h to get the ASCII value
  + Move the output in A to R3.
  + Move the input to A.
  + Get the lower order byte of A by performing logical AND over A and 0Fh.
  + Add A with 30h to get the ASCII value
  + Move the output in A to R4.
  + Terminate the program.

## Program:

|  |  |
| --- | --- |
| CODE | COMMENT |
| Program for Cube of a number:  mov a, r1  anl a, #0f0h  swap a  add a ,#30h  mov r3, a  mov a , r1  anl a, #00fh  add a ,#30h  mov r4 ,a  here : sjmp here | Move the input from R1 to A  AND A with 0Fh to mask the lower order byte  Swap A to get the higher order byte.  Add A with 30h to get the ASCII value  Move A to R3  Move the input from R1 to A  AND A with F0h to mask the higher order byte  Add A with 30h to get the ASCII value  Move A to R4  Terminate the program |

Snapshot of sample input and output:



Result:

Convertion of BCD to ASCII is executed and verified using 8051 microcontroller using EDSim