

UCS1512 – Microprocessors Lab

Case Conversion

Exp no : 1 2

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: <pre> mov r0 ,#00 mov a ,r1 add a ,r2 jnc label inc r0 label : mov r4 ,a mov 03,r0 here : sjmp here </pre>	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:

EdSim51DI - Version 2.1.21

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O W/O TH0 TL0 R7 0x00 B 0x00

0x00 0x00 0x00 0x00 R6 0x00 ACC 0xB5

RXD TXD 1 1 TMOD 0x00 R5 0x00 PSW 0x45

SCON 0x00 TCON 0x00 R4 0xB5 IP 0x00

pins bits TH1 TL1 R3 0x00 IE 0x00

0xFF 0xFF P3 0x00 0x00 R2 0x3B PCON 0x00

0xFF 0xFF P2 0x00 0x00 R1 0x7A DPH 0x00

0xFF 0xFF P1 0x00 0x00 R0 0x00 DPL 0x00

0xFF 0xFF P0 0x00 0x00 PC 8051 SP 0x07

PSW 0 1 0 0 0 1 0 1

Modify RAM

Data Memory

addr	0x00	0x00	value
0	00	00	00
1	00	00	00
2	00	00	00
3	00	00	00
4	00	00	00
5	00	00	00
6	00	00	00
7	00	00	00
8	00	00	00
9	00	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00
00	00	00	00
01	00	00	00
02	00	00	00
03	00	00	00
04	00	00	00
05	00	00	00
06	00	00	00
07	00	00	00
08	00	00	00
09	00	00	00
0A	00	00	00
0B	00	00	00
0C	00	00	00
0D	00	00	00
0E	00	00	00
0F	00	00	00
10	00	00	00
11	00	00	00
12	00	00	00
13	00	00	00
14	00	00	00
15	00	00	00
16	00	00	00
17	00	00	00
18	00	00	00
19	00	00	00
1A	00	00	00
1B	00	00	00
1C	00	00	00
1D	00	00	00
1E	00	00	00
1F	00	00	00
20	00	00	00
21	00	00	00
22	00	00	00
23	00	00	00
24	00	00	00
25	00	00	00
26	00	00	00
27	00	00	00
28	00	00	00
29	00	00	00
2A	00	00	00
2B	00	00	00
2C	00	00	00
2D	00	00	00
2E	00	00	00
2F	00	00	00
30	00	00	00
31	00	00	00
32	00	00	00
33	00	00	00
34	00	00	00
35	00	00	00
36	00	00	00
37	00	00	00
38	00	00	00
39	00	00	00
3A	00	00	00
3B	00	00	00
3C	00	00	00
3D	00	00	00
3E	00	00	00
3F	00	00	00
40	00	00	00
41	00	00	00
42	00	00	00
43	00	00	00
44	00	00	00
45	00	00	00
46	00	00	00
47	00	00	00
48	00	00	00
49	00	00	00
4A	00	00	00
4B	00	00	00
4C	00	00	00
4D	00	00	00
4E	00	00	00
4F	00	00	00
50	00	00	00
51	00	00	00
52	00	00	00
53	00	00	00
54	00	00	00
55	00	00	00
56	00	00	00
57	00	00	00
58	00	00	00
59	00	00	00
5A	00	00	00
5B	00	00	00
5C	00	00	00
5D	00	00	00
5E	00	00	00
5F	00	00	00
60	00	00	00
61	00	00	00
62	00	00	00
63	00	00	00
64	00	00	00
65	00	00	00
66	00	00	00
67	00	00	00
68	00	00	00
69	00	00	00
6A	00	00	00
6B	00	00	00
6C	00	00	00
6D	00	00	00
6E	00	00	00
6F	00	00	00
70	00	00	00
71	00	00	00
72	00	00	00
73	00	00	00
74	00	00	00
75	00	00	00
76	00	00	00
77	00	00	00
78	00	00	00
79	00	00	00
7A	00	00	00
7B	00	00	00
7C	00	00	00
7D	00	00	00
7E	00	00	00
7F	00	00	00

Copyright ©2005-2016 James Rogers Remove All Breakpoints

Time: 34us - Instructions: 19

```

0000| mov r0 ,#00
0002| mov a ,r1
0003| add a ,r2
0004| jnc label
0006| inc r0
0007| label : mov r4 ,a
0008| mov 03,r0
000A| here : sjmp here
  
```

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:	
<pre>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</pre>	<p>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</p> <p>Move the result to R4 Move the carry to R3 Terminate the program</p>

Snapshot of sample input and output:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUS

R/O	W/O	TH0	TL0	R7	0x00	B	0x00	
0x00	0x00	0x00	0x00	R6	0x00	ACC	0x03	
RXD	TXD			R5	0x00	PSW	0x00	
1	1	TMOD	0x00	R4	0x03	IP	0x00	
SCON	0x00	TCON	0x00	R3	0x01	IE	0x00	
				R2	0x08	PCON	0x00	
pins	bits	TH1	TL1	R1	0x05	DPH	0x00	
0xFF	0xFF	P3	0x00	0x00	R0	0x01	DPL	0x00
0xFF	0xFF	P2				SP	0x07	
0xFF	0xFF	P1						
0xFF	0xFF	P0						

8051

PC 0x000E PSW 0 0 0 0 0 0 0 0

Modify RAM

Data Memory

addr	0x00	0x00	value												
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	01	05	08	01	03	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2016 James Rogers Remove All Breakpoints

RST Step Run New Load Save Copy Paste

Time: 28us - Instructions: 18

```
0000| clr c
0001| mov r0 ,#00
0003| mov a ,r1
0004| subb a ,r2
0005| jnc label
0007| inc r0
0008| cpl a
0009| add a ,#01
000B| label : mov r4 ,a
000C| mov 03,r0
000E| here : sjmp here
```

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:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O W/O TH0 TL0 R7 0x00 B 0x00

0x00 0x00 0x00 0x00 R6 0x00 ACC 0x28

RXD TXD R5 0x00 PSW 0x00

1 1 TMOD 0x00 R4 0x28 IP 0x00

SCON 0x00 TCON 0x00 R3 0x00 IE 0x00

pins bits TH1 TL1 R2 0x08 PCON 0x00

0xFF 0xFF P3 0x00 0x00 R1 0x05 DPH 0x00

0xFF 0xFF P2 PC 8051 SP 0x07

0xFF 0xFF P1 0x0007 PSW 0 0 0 0 0 0 0 0

0xFF 0xFF P0

Data Memory

addr 0x00 0x00 value

0 1 2 3 4 5 6 7 8 9 A B C D E F

00 01 05 08 00 28 00 00 00 00 00 00 00 00 00 00

10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Copyright ©2005-2016 James Rogers Remove All Breakpoints

RST Step Run New Load Save Copy Paste

Time: 46us - Instructions: 23

0000 | mov a ,r1

0001 | mov b ,r2

0003 | mul ab

0004 | mov r3 ,b

0006 | mov r4 ,a

0007 | here : sjmp here

8-bit Division:

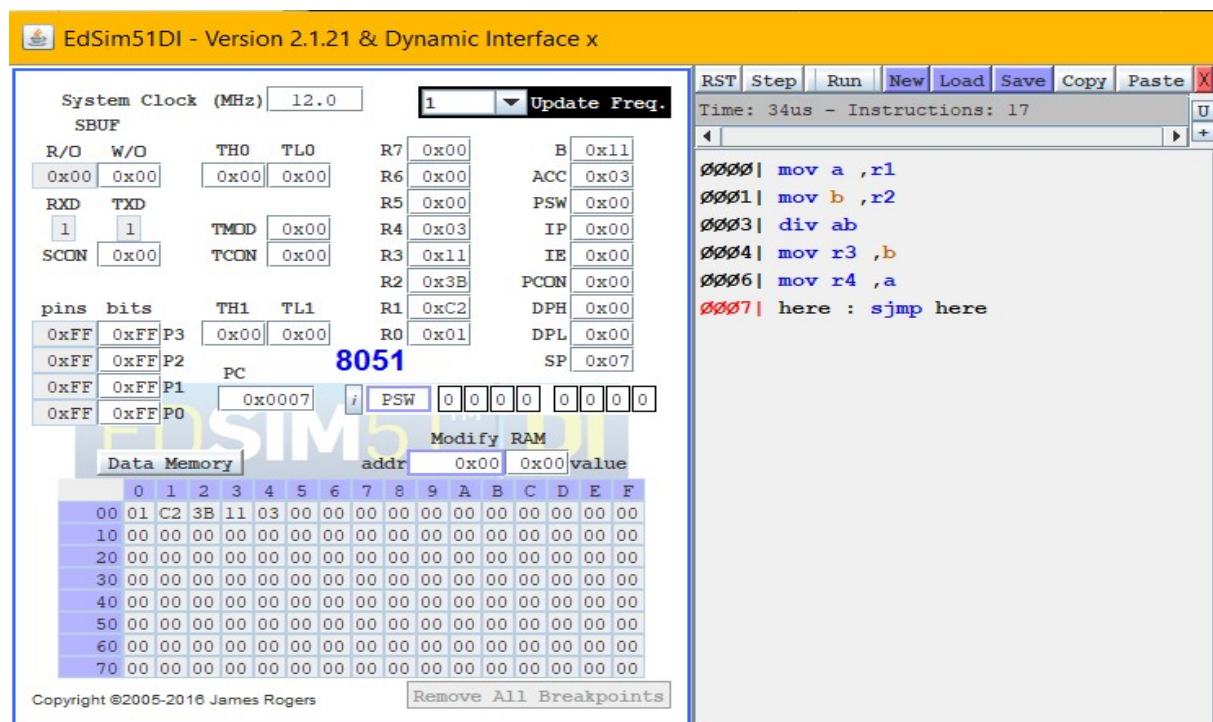
Algorithm:

- 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: <pre> 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 </pre>	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:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O W/O TH0 TL0 R7 0x00 B 0x03

0x00 0x00 0x00 0x00 R6 0x00 ACC 0xE8

RXD TXD 1 1 TMOD 0x00 R5 0x00 PSW 0x04

SCON 0x00 TCON 0x00 R4 0xE8 IP 0x00

R3 0x03 IE 0x00

R2 0x00 PCON 0x00

pins bits TH1 TL1 R1 0x0A DPH 0x00

0xFF 0xFF P3 0x00 0x00 R0 0x0A DPL 0x00

0xFF 0xFF P2 PC 8051 SP 0x07

0xFF 0xFF P1 0x000B PSW 0 0 0 0 0 1 0 0

0xFF 0xFF P0

Modify RAM

Data Memory addr 0x00 0x00 value

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0A	0A	00	03	E8	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2016 James Rogers Remove All Breakpoints

RST Step Run New Load Save Copy Paste X

Time: 57us - Instructions: 28

```

0000 | mov a ,r1
0001 | mov b ,r1
0003 | mov r0 ,a
0004 | mul ab
0005 | mov b ,r0
0007 | mul ab
0008 | mov r3 , b
000A | mov r4 , a
000B | here : sjmp here

```

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: <pre> 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 </pre>	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:

EdSim51DI - Version 2.1.21 & Dynamic Interface x

System Clock (MHz) 12.0 1 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	B
0x00	0x00	0x00	0x00	0x00	0x00

RXD TXD

1	1	TMOD	0x00	R4	0x38	PSW	0x01
---	---	------	------	----	------	-----	------

SCON 0x00 TCON 0x00 R3 0x34 IE 0x00

pins bits TH1 TL1 R2 0x00 PCON 0x00

0xFF	0xFF	P3	0x00	0x00	R1	0x48	DPH	0x00
------	------	----	------	------	----	------	-----	------

0xFF 0xFF P2 PC 8051

0xFF	0xFF	P1	0x000D	PSW	0	0	0	0	0	0	0	1
------	------	----	--------	-----	---	---	---	---	---	---	---	---

0xFF 0xFF P0

Data Memory

addr	0x00	0x00	value
------	------	------	-------

0 1 2 3 4 5 6 7 8 9 A B C D E F

00	0A	48	00	34	38	00	00	00	00	00	00	00	00	00	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Copyright ©2005-2016 James Rogers Remove All Breakpoints

RST Step Run New Load Save Copy Paste

Time: 43us - Instructions: 26

```

0000| mov a, r1
0001| anl a, #0f0h
0003| swap a
0004| add a, #30h
0006| mov r3, a
0007| mov a, r1
0008| anl a, #00fh
000A| add a, #30h
000C| mov r4, a
000D| here : sjmp here
  
```

Result:

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