Laboratory Report | 2022

COMPUTER ORGANISATION AND ARCHITECTURE (EET 2211)

Name: ……..

Registration No: ……..……...

Branch:……..…..Section:……..

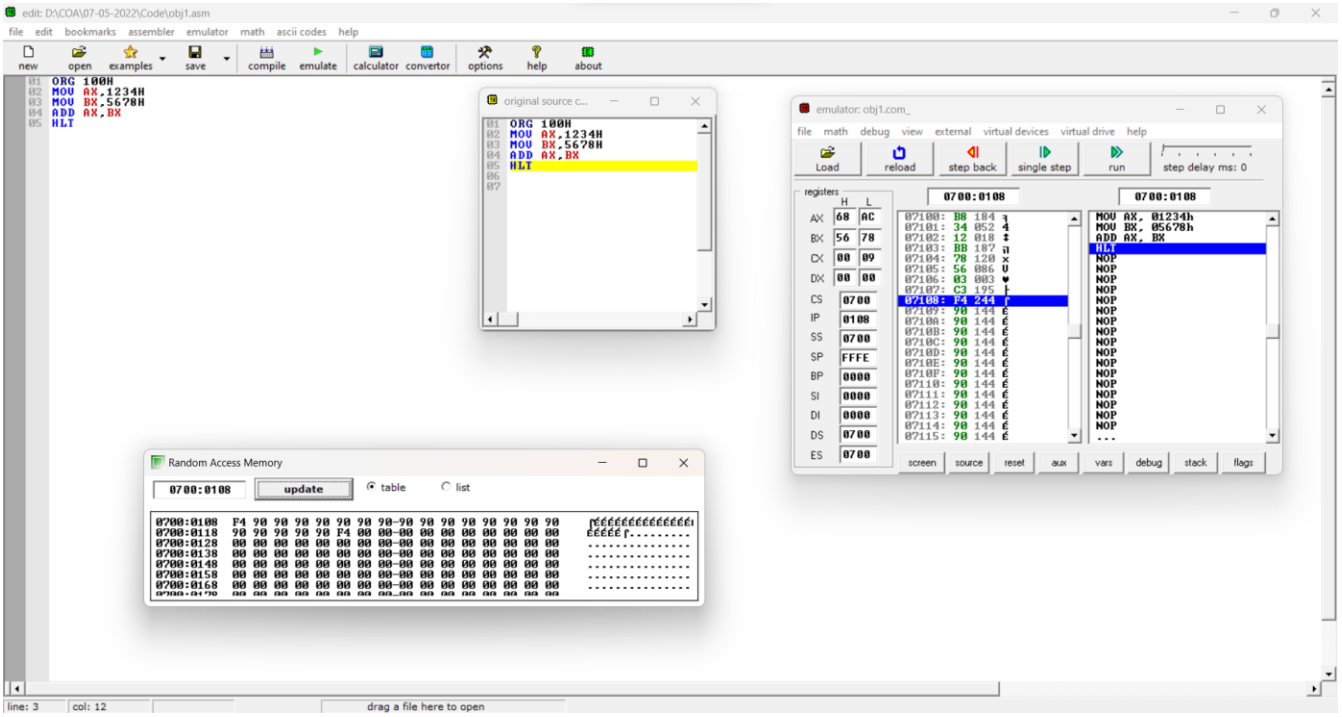
sInstitute Of Technical Education & Research (Faculty Of Engineering)
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha

CONTENTS

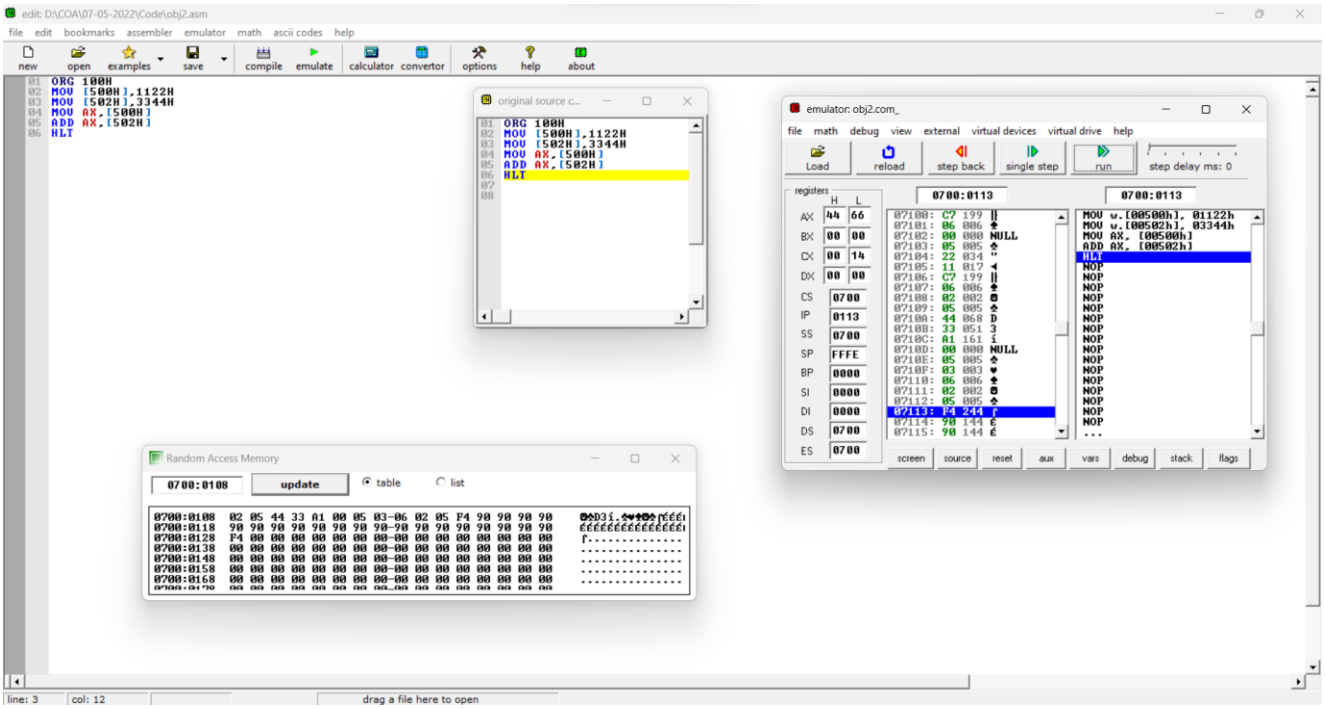
CONTENTS

Observations (with screen shots):

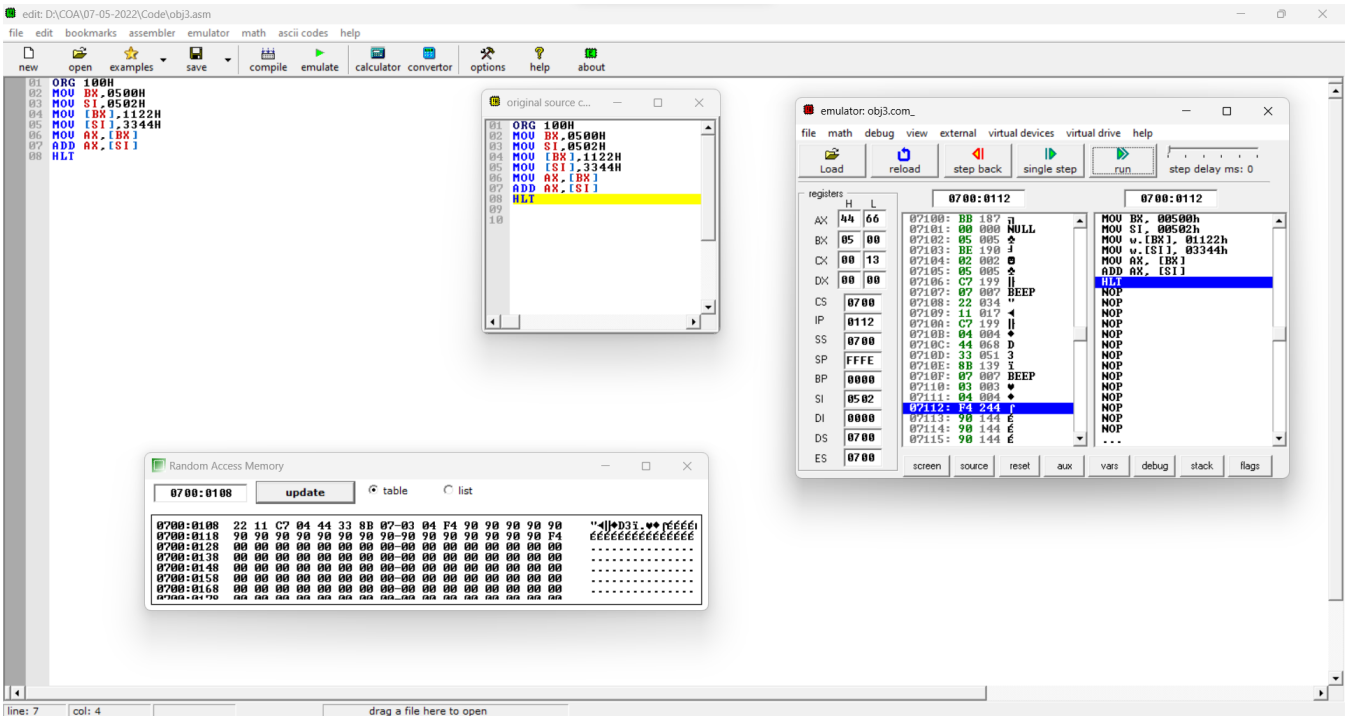
Objective 1:



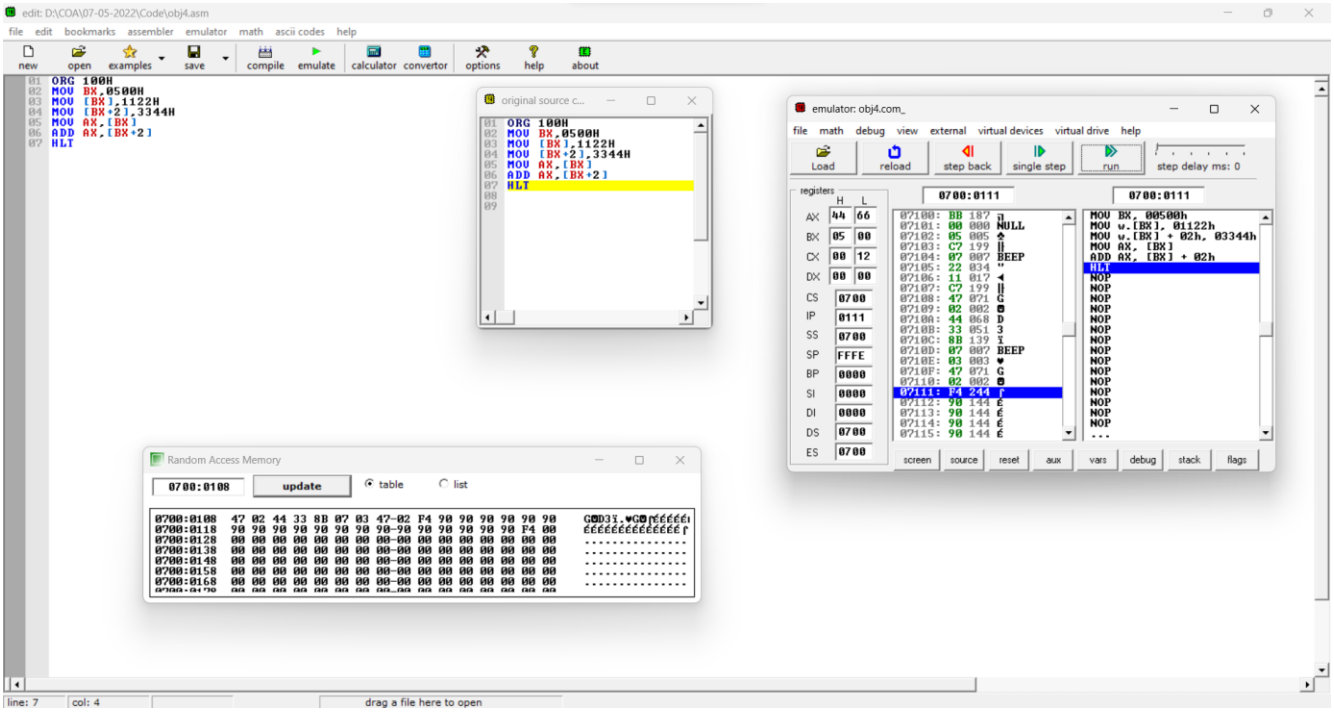
Objective 2:



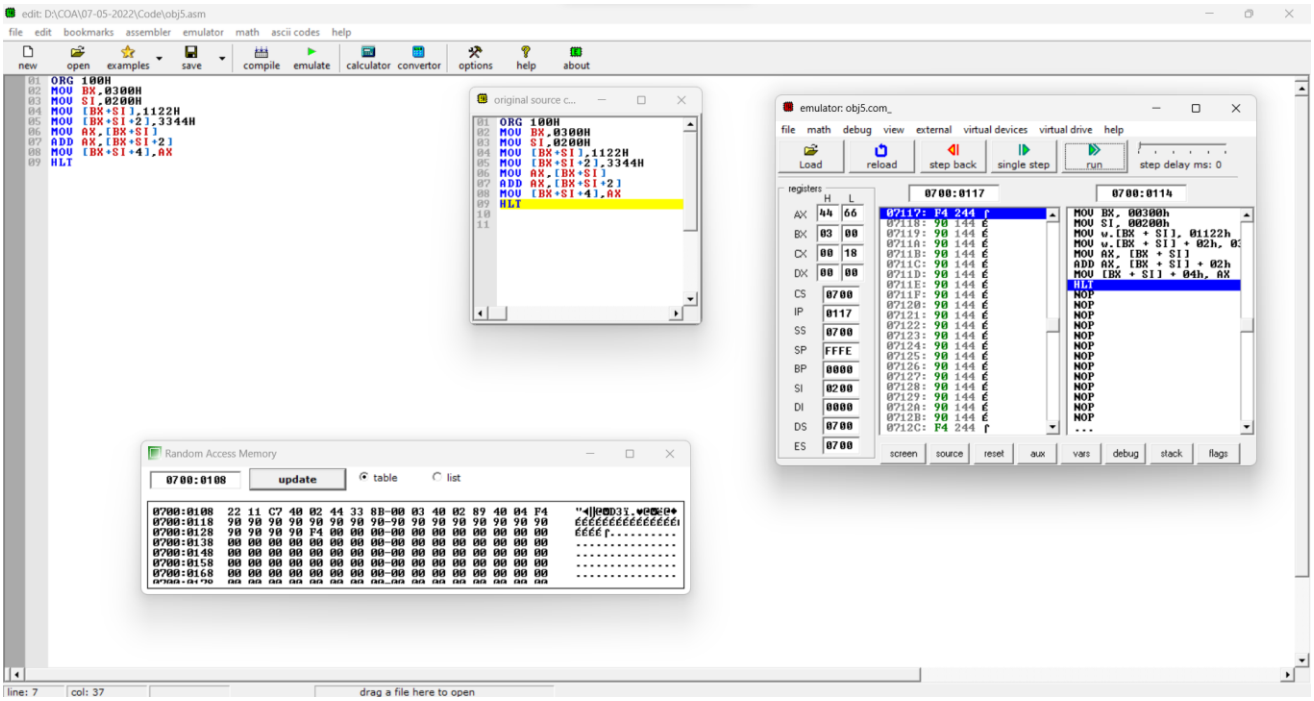
Objective 3:



Objective 4:

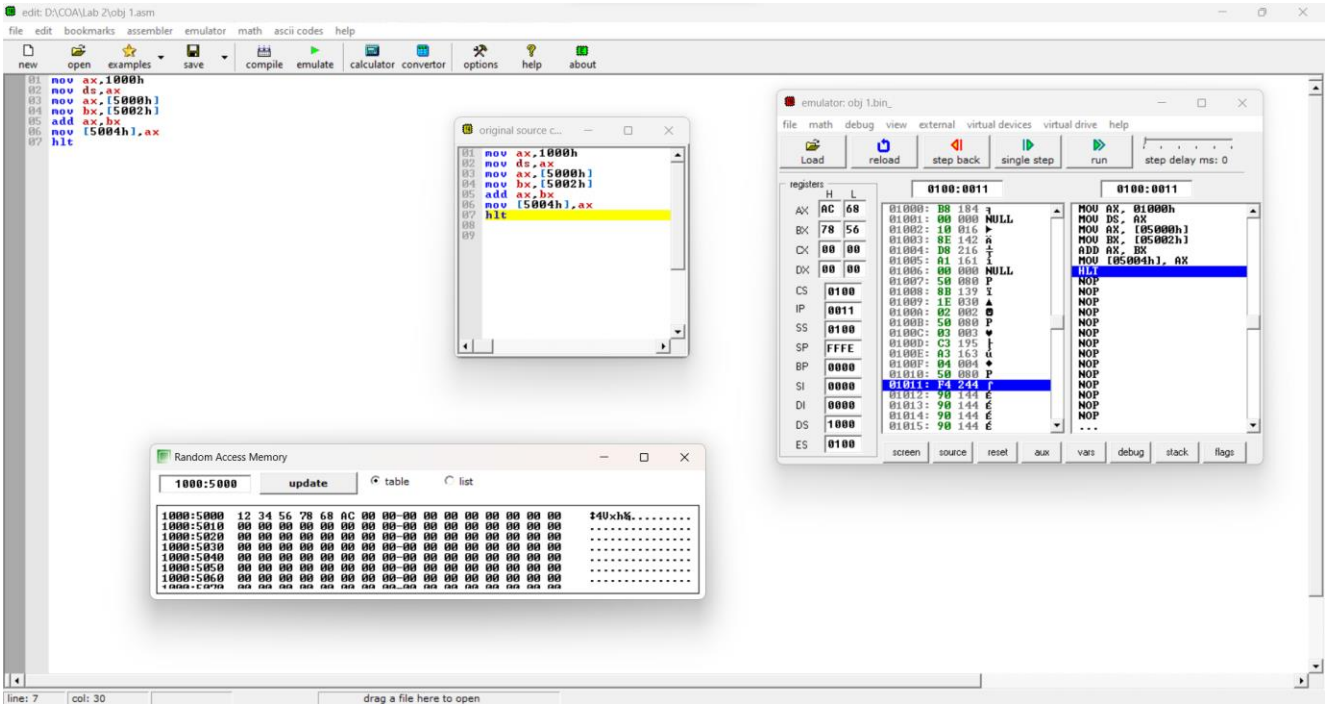


Objective 5:

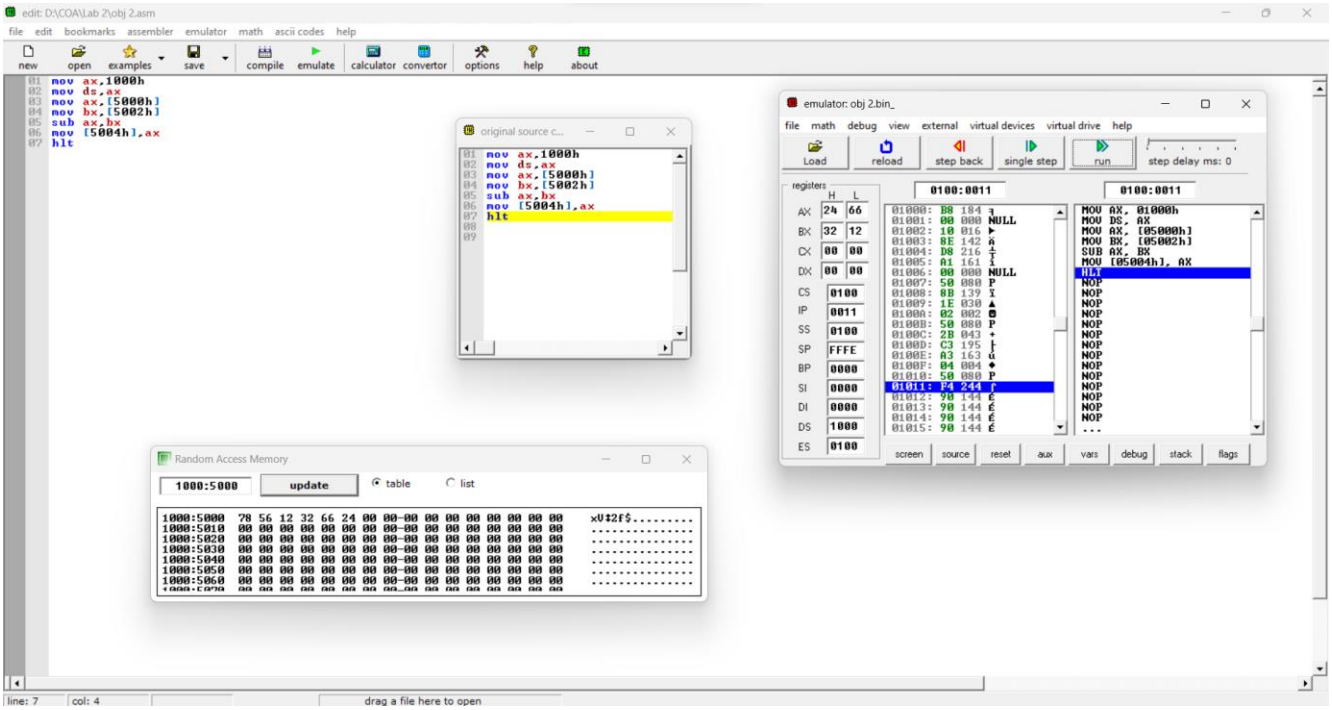

edit: D:\COA\07-05-2022\Code\obj5.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
ORG 100H
MOV BX,0300H
MOV SI,0200H
MOV [BX+SI],1122H
MOV [BX+SI+2],3344H
MOV AX,[BX+SI]
ADD AX,[BX+SI+2]
MOV [BX+SI+4],AX
HLT
original source c...
emulator: obj5.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
AX 44 66
BX 03 00
CX 00 18
DX 00 00
CS 0700
IP 0117
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:0117
0700:0114
screen source reset aux vars debug stack flags
Random Access Memory
0700:0108
update
table
list
line: 7
col: 37
drag a file here to open

Observations (with screen shots):

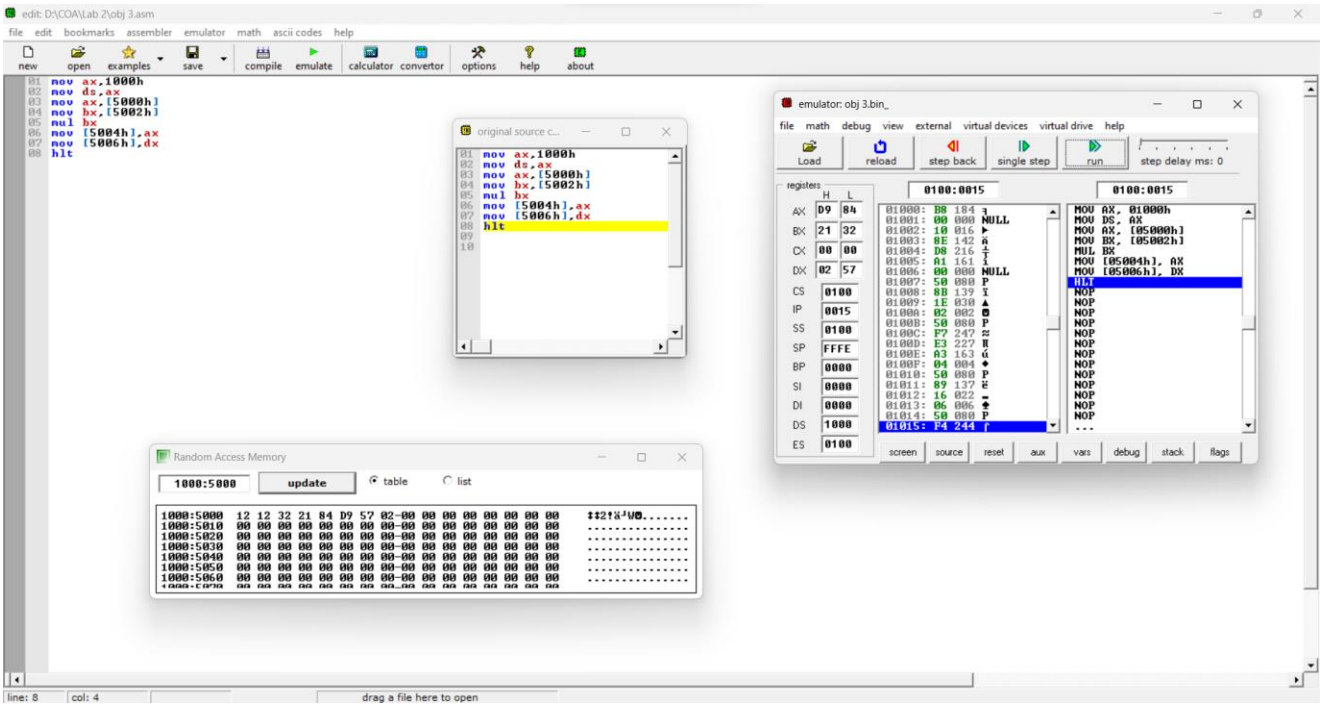
For Obj. 1



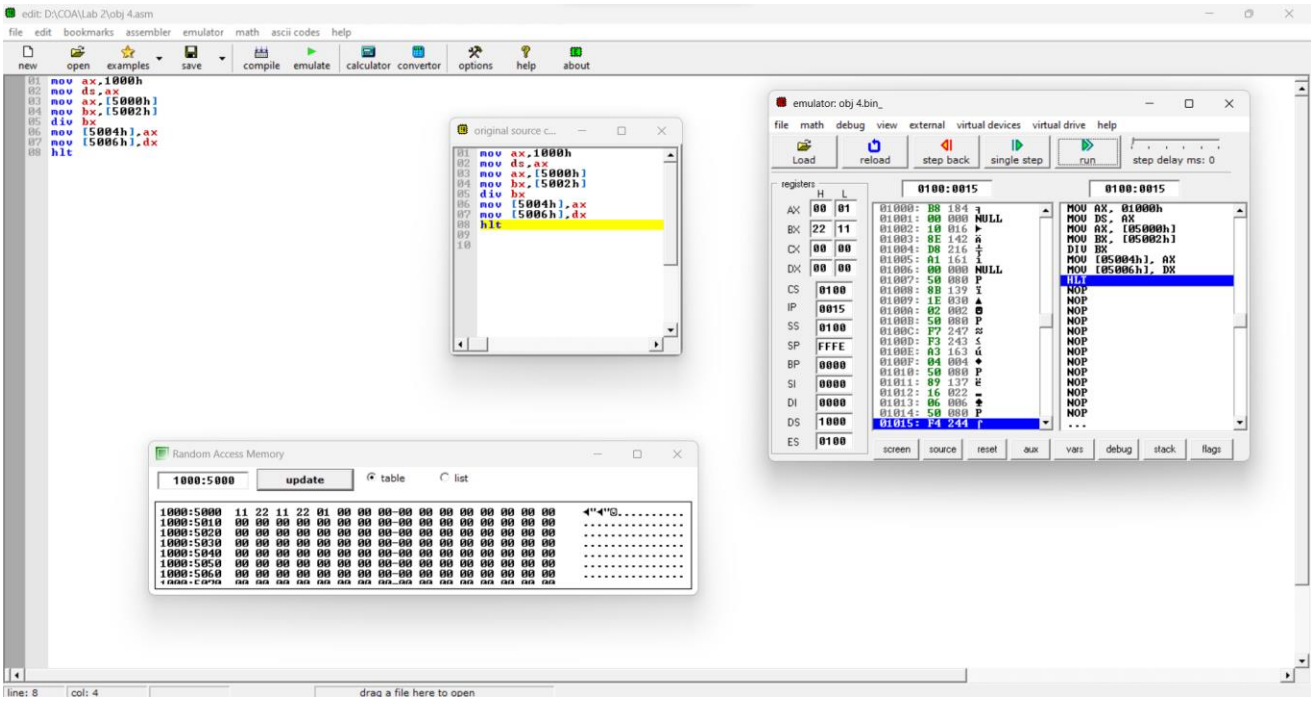
For Obj. 2



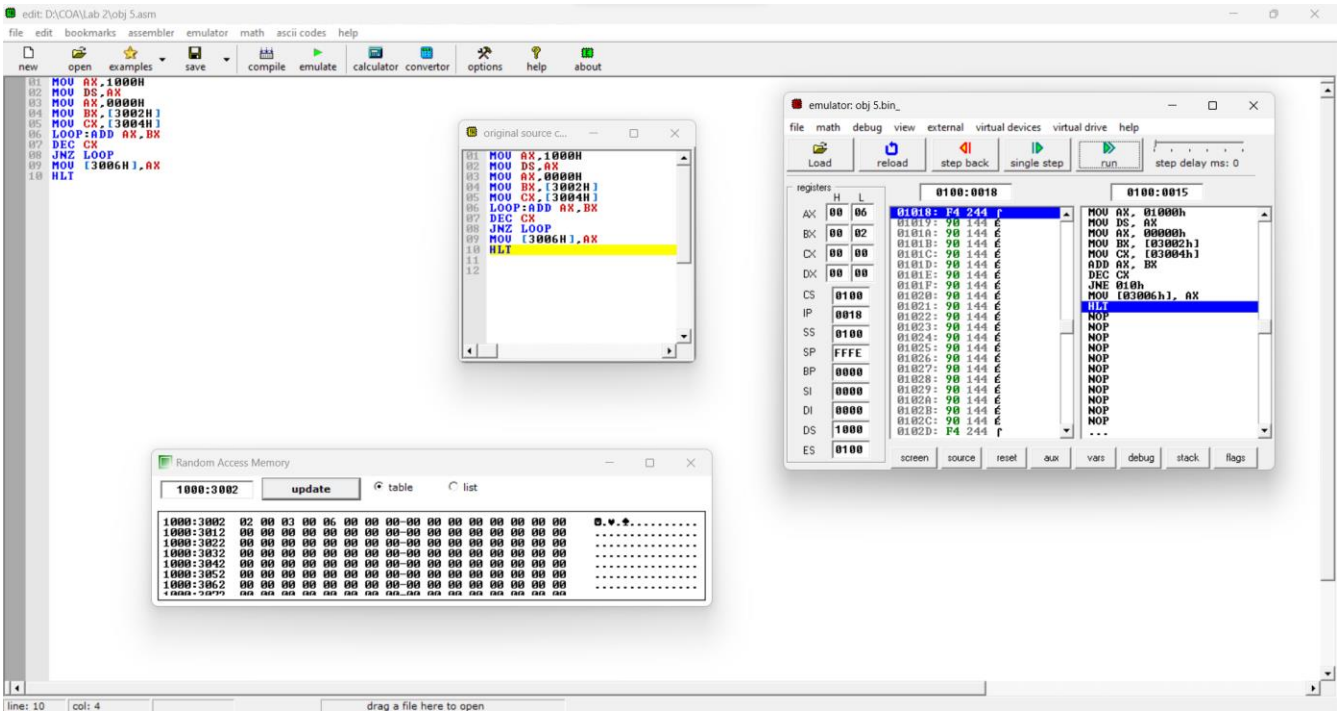
For Obj. 3



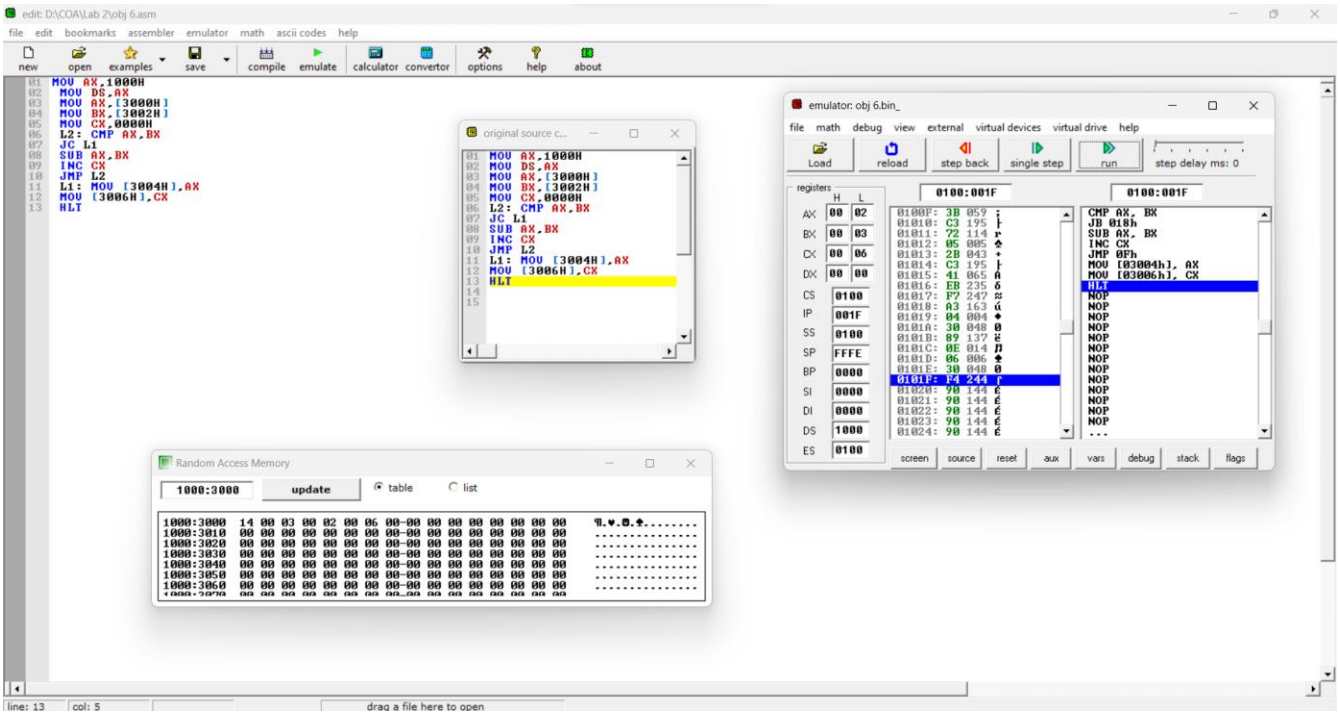
For Obj. 4



For Obj. 5



For Obj. 6



```
;NAME - Yash Rout
;REG- 2041018131
ORG 100H
MOV AX, 2000H
MOV BX, 6200H
MOV DX, AX
MOV SI, BX
MOV CX, AX
MOV [1000H], AX
OR AX, BX
AND [1000H], BX
XOR CX, BX
NOT DX
NOT SI
HLT
```

original source code

```
;NAME - Yash Rout
;REG- 2041018131
ORG 100H
MOV AX, 2000H
MOV BX, 6200H
MOV DX, AX
MOV SI, BX
MOV CX, AX
MOV [1000H], AX
OR AX, BX
AND [1000H], BX
XOR CX, BX
NOT DX
NOT SI
HLT
```

emulator: mycode1.com_

file math debug view external virtual devices virtual drive help

Load | reload | step back | single step | run | step delay ms: 0

registers

	H	L
AX	62	00
BX	62	00
CX	42	00
DX	DF	FF
CS	0700	
IP	011B	
SS	0700	
SP	FFFE	
BP	0000	
SI	9DFF	
DI	0000	
DS	0700	
ES	0700	

0700:011B

```
07117: F7 247 ≈
07118: D2 210 π
07119: F7 247 ≈
0711A: D6 214 π
0711B: F4 244 ⌠
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: 90 144 É
07129: 90 144 É
0712A: 90 144 É
0712B: 90 144 É
0712C: 90 144 É
```

0700:011B

```
MOV AX, 02000h
MOV BX, 06200h
MOV DX, AX
MOV SI, BX
MOV CX, AX
MOV [01000h], AX
OR AX, BX
AND [01000h], BX
XOR CX, BX
NOT DX
NOT SI
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen | source | reset | aux | vars | debug | stack | flags

```
;NAME - Yash Rout
;REG- 2041018131
ORG 100H
MOV AL,05H
NOT AL
ADD AL,01H
MOV CL,AL
HLT
```

original source code

```
;NAME - Yash Rout
;REG- 2041018131
ORG 100H
MOV AL,05H
NOT AL
ADD AL,01H
MOV CL,AL
HLT
```

emulator: mycode1.com_

file math debug view external virtual devices virtual drive help

Load | reload | step back | single step | run | step delay ms: 0

registers

	H	L
AX	00	FB
BX	00	00
CX	00	FB
DX	00	00
CS	0700	
IP	0108	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0108

```
07100: B0 176 ▒
07101: 05 005 ♣
07102: F6 246 ÷
07103: D0 208 ╨
07104: 04 004 ♦
07105: 01 001 ☺
07106: 8A 138 è
07107: C8 200 ╚
07108: F4 244 ⌠
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

0700:0108

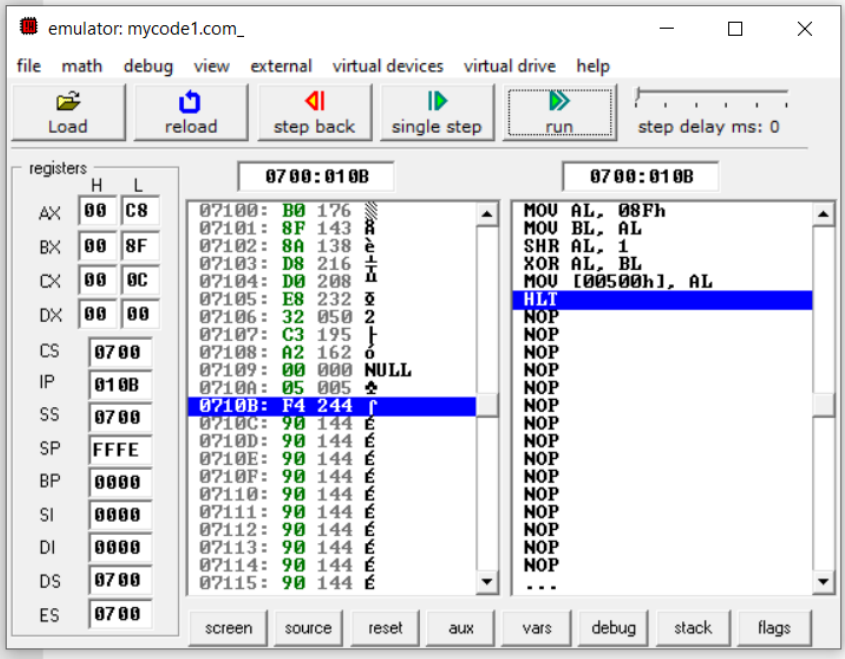
```
MOV AL, 05h
NOT AL
ADD AL, 01h
MOV CL, AL
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen | source | reset | aux | vars | debug | stack | flags

```
;NAME - Yash Rout
;REG- 2041018131
ORG 100H
MOV AL,8FH
MOV BL,AL
SHR AL,1
XOR AL,BL
MOV [500H],AL
HLT
```

original source code

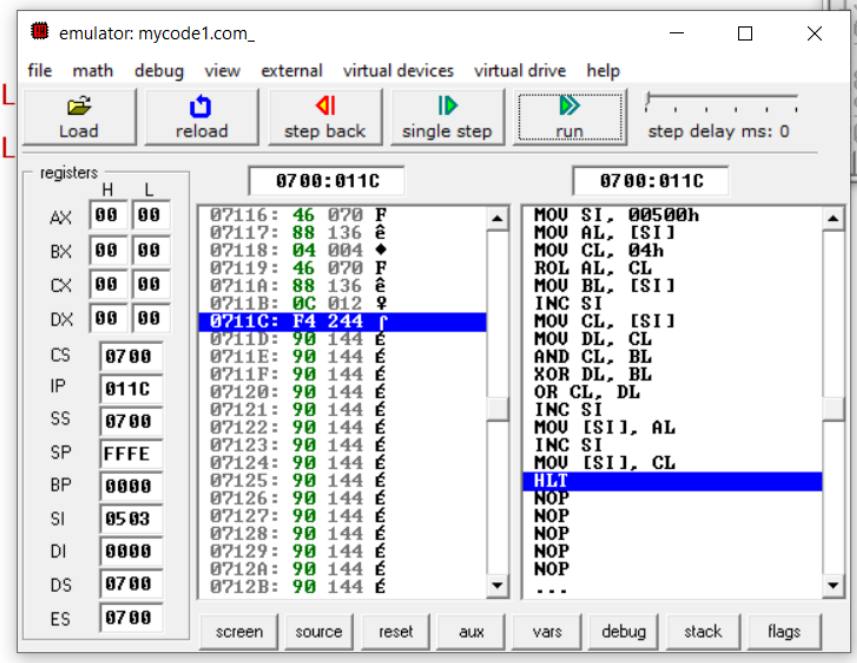
```
;NAME - Yash Rout
;REG- 2041018131
ORG 100H
MOV AL,8FH
MOV BL,AL
SHR AL,1
XOR AL,BL
MOV [500H],AL
HLT
```

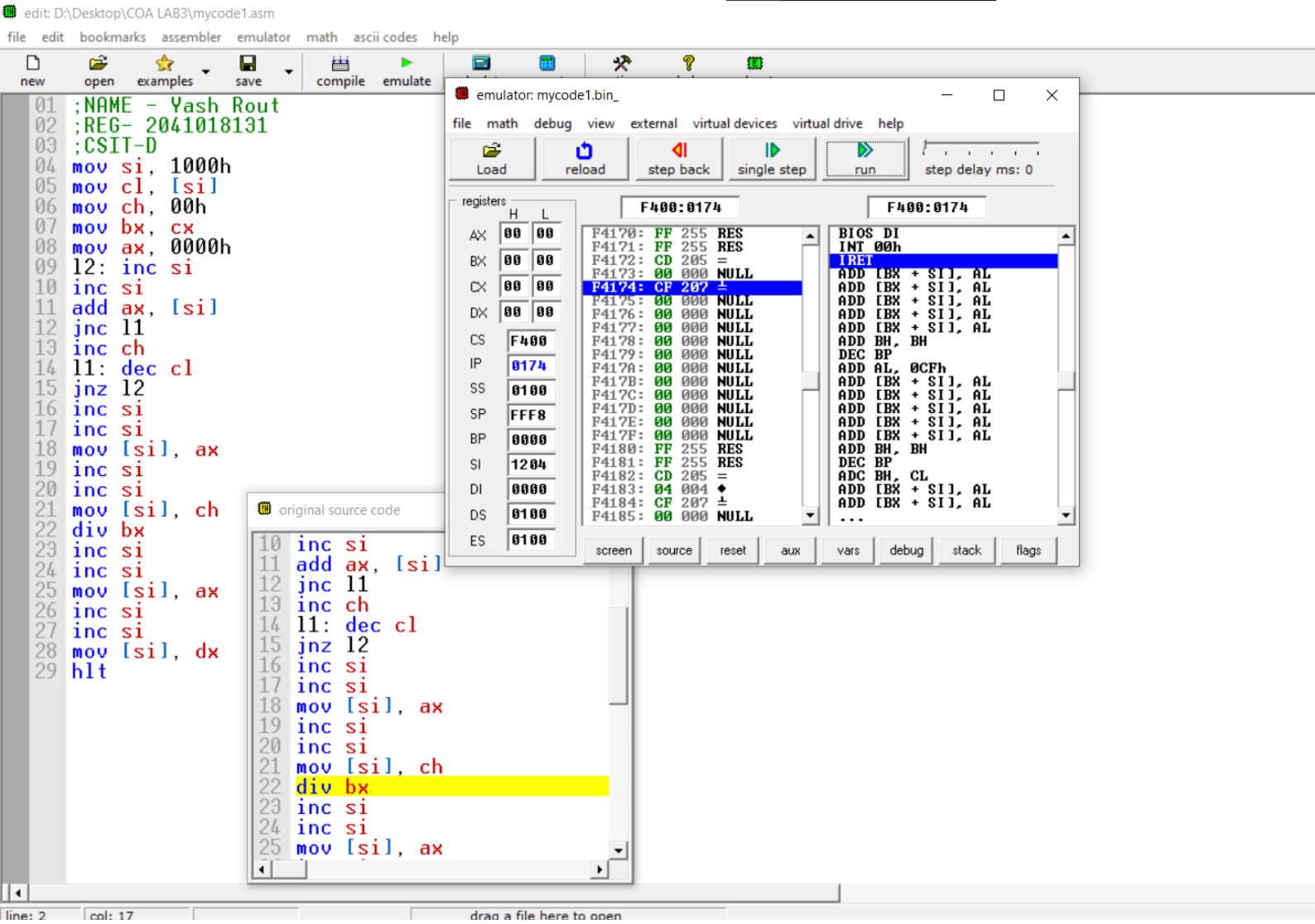


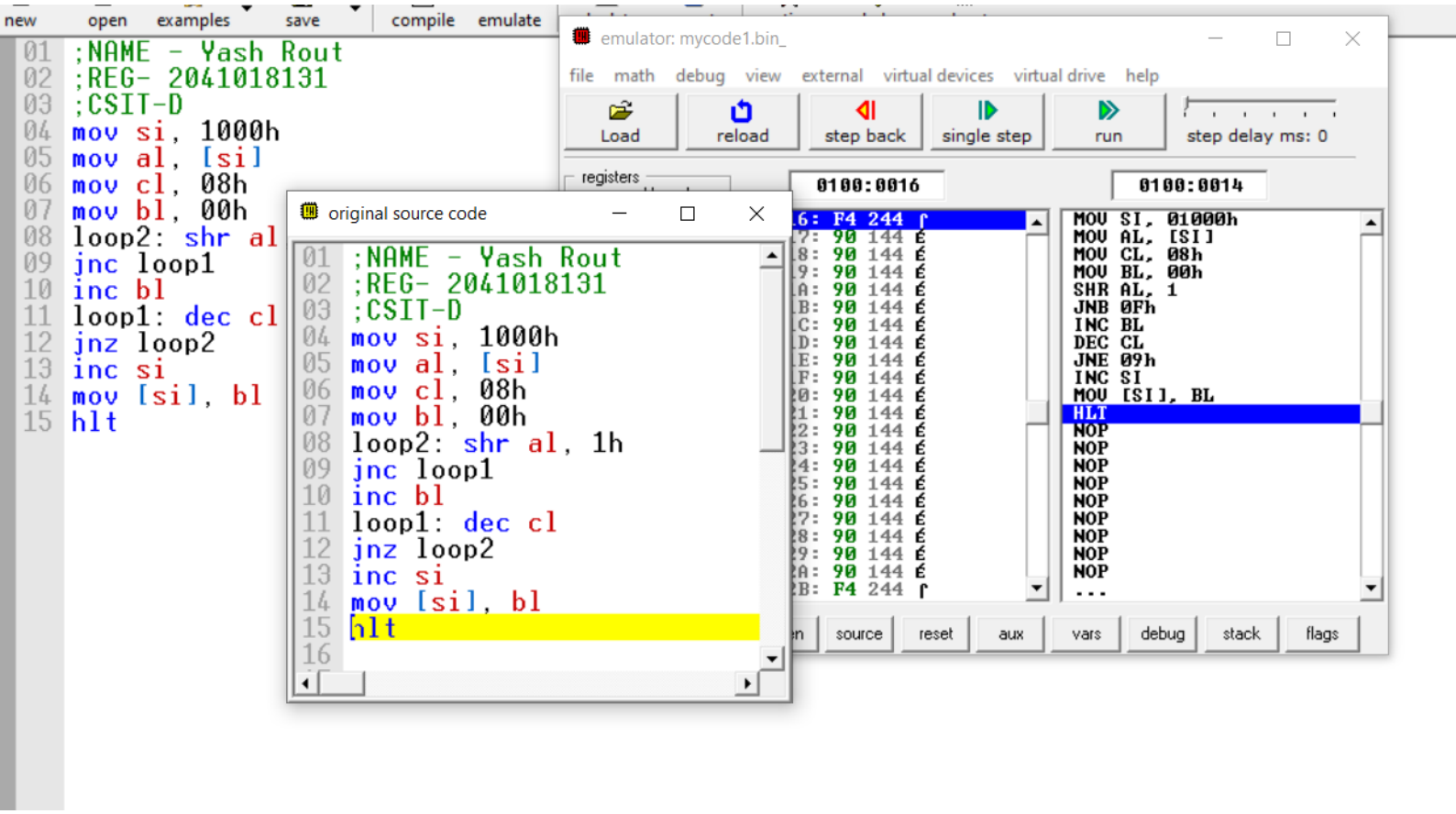

```
;NAME - Yash Rout
;REG- 2041018131

ORG 100H
MOV SI, 0500H
MOV AL, [SI]
MOV CL, 04H
ROL AL, CL
MOV BL, [SI]
INC SI
MOV CL, [SI]
MOV DL, CL
AND CL, BL
XOR DL, BL
OR CL, DL
INC SI
MOV [SI], AL
INC SI
MOV [SI], CL
HLT
```

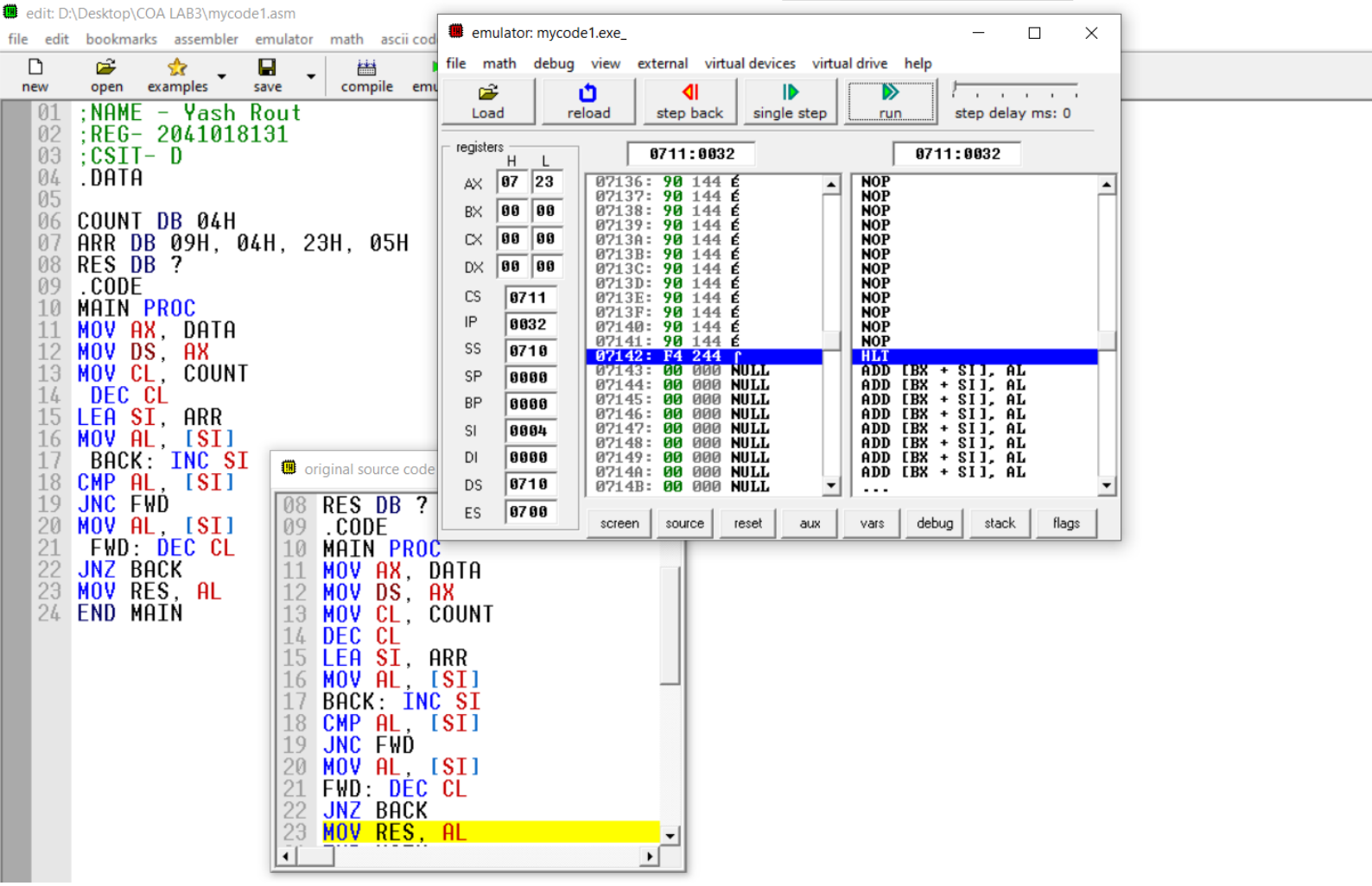
original source code

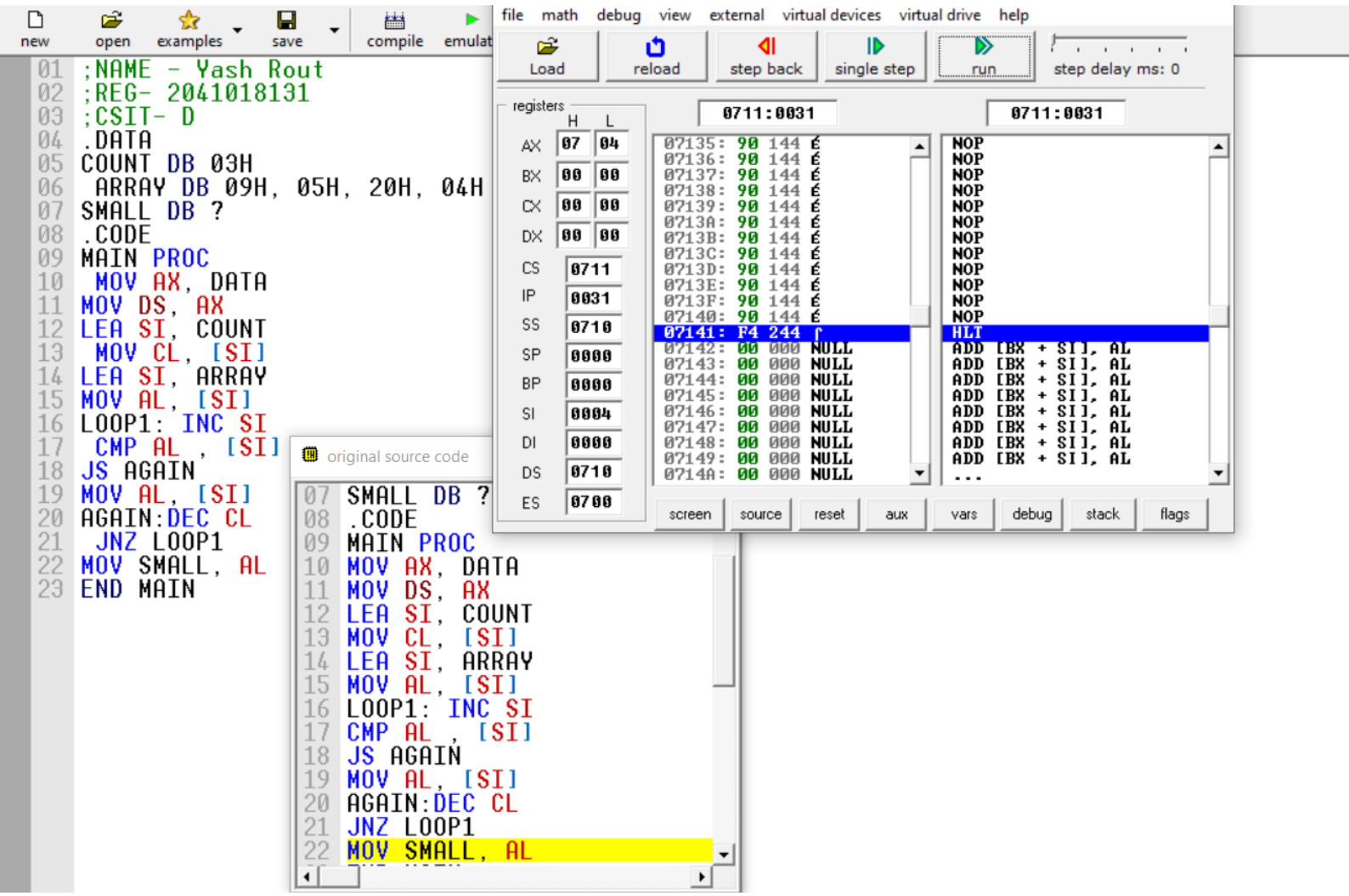
```
MOV SI, 0500H
MOV AL, [SI]
MOV CL, 04H
ROL AL, CL
MOV BL, [SI]
INC SI
MOV CL, [SI]
MOV DL, CL
AND CL, BL
XOR DL, BL
OR CL, DL
INC SI
MOV [SI], AL
INC SI
MOV [SI], CL
HLT
```




edit: D:\Desktop\COA LAB3\mycode1.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate
;NAME - Yash Rout
;REG- 2041018131
;CSIT-D
mov si, 1000h
mov cl, [si]
mov ch, 00h
mov bx, cx
mov ax, 0000h
l2: inc si
inc si
add ax, [si]
jnc l1
inc ch
l1: dec cl
jnz l2
inc si
inc si
mov [si], ax
inc si
inc si
mov [si], ch
div bx
inc si
inc si
mov [si], ax
inc si
inc si
mov [si], dx
hlt
emulator: mycode1.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 00
BX 00 00
CX 00 00
DX 00 00
CS F400
IP 0174
SS 0100
SP FFF8
BP 0000
SI 1204
DI 0000
DS 0100
ES 0100
F400:0174
F4170: FF 255 RES
F4171: FF 255 RES
F4172: CD 205 =
F4173: 00 000 NULL
F4174: CF 207 ±
F4175: 00 000 NULL
F4176: 00 000 NULL
F4177: 00 000 NULL
F4178: 00 000 NULL
F4179: 00 000 NULL
F417A: 00 000 NULL
F417B: 00 000 NULL
F417C: 00 000 NULL
F417D: 00 000 NULL
F417E: 00 000 NULL
F417F: 00 000 NULL
F4180: FF 255 RES
F4181: FF 255 RES
F4182: CD 205 =
F4183: 04 004 ♦
F4184: CF 207 ±
F4185: 00 000 NULL
F400:0174
BIOS DI
INT 00h
IRET
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADD AL, 0CFh
ADC BH, CL
screen source reset aux vars debug stack flags
original source code
inc si
add ax, [si]
jnc l1
inc ch
l1: dec cl
jnz l2
inc si
inc si
mov [si], ax
inc si
inc si
mov [si], ch
div bx
inc si
inc si
mov [si], ax
line: 2 col: 17
drag a file here to open


new open examples save compile emulate
;NAME - Yash Rout
;REG- 2041018131
;CSIT-D
mov si, 1000h
mov al, [si]
mov cl, 08h
mov bl, 00h
loop2: shr al
jnc loop1
inc bl
loop1: dec cl
jnz loop2
inc si
mov [si], bl
hlt
emulator: mycode1.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
0100:0016
0100:0014
MOV SI, 01000h
MOV AL, [SI]
MOV CL, 08h
MOV BL, 00h
SHR AL, 1
JNB 0Fh
INC BL
DEC CL
JNE 09h
INC SI
MOV [SI], BL
HLT
NOP
source reset aux vars debug stack flags
original source code
;NAME - Yash Rout
;REG- 2041018131
;CSIT-D
mov si, 1000h
mov al, [si]
mov cl, 08h
mov bl, 00h
loop2: shr al, 1h
jnc loop1
inc bl
loop1: dec cl
jnz loop2
inc si
mov [si], bl
hlt


edit: D:\Desktop\COA LAB3\mycode1.asm
file edit bookmarks assembler emulator math ascii cod
new open examples save compile emu
;NAME - Yash Rout
;REG- 2041018131
;CSIT- D
.DATA
COUNT DB 04H
ARR DB 09H, 04H, 23H, 05H
RES DB ?
.CODE
MAIN PROC
MOV AX, DATA
MOV DS, AX
MOV CL, COUNT
DEC CL
LEA SI, ARR
MOV AL, [SI]
BACK: INC SI
CMP AL, [SI]
JNC FWD
MOV AL, [SI]
FWD: DEC CL
JNZ BACK
MOV RES, AL
END MAIN
emulator: mycode1.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 07 23
BX 00 00
CX 00 00
DX 00 00
CS 0711
IP 0032
SS 0710
SP 0000
BP 0000
SI 0004
DI 0000
DS 0710
ES 0700
0711:0032
0711:0032
07136: 90 144 É NOP
07137: 90 144 É NOP
07138: 90 144 É NOP
07139: 90 144 É NOP
0713A: 90 144 É NOP
0713B: 90 144 É NOP
0713C: 90 144 É NOP
0713D: 90 144 É NOP
0713E: 90 144 É NOP
0713F: 90 144 É NOP
07140: 90 144 É NOP
07141: 90 144 É NOP
07142: F4 244 ⌠ HLT
07143: 00 000 NULL ADD [BX + SI], AL
07144: 00 000 NULL ADD [BX + SI], AL
07145: 00 000 NULL ADD [BX + SI], AL
07146: 00 000 NULL ADD [BX + SI], AL
07147: 00 000 NULL ADD [BX + SI], AL
07148: 00 000 NULL ADD [BX + SI], AL
07149: 00 000 NULL ADD [BX + SI], AL
0714A: 00 000 NULL ADD [BX + SI], AL
0714B: 00 000 NULL ...
screen source reset aux vars debug stack flags
original source code
08 RES DB ?
09 .CODE
10 MAIN PROC
11 MOV AX, DATA
12 MOV DS, AX
13 MOV CL, COUNT
14 DEC CL
15 LEA SI, ARR
16 MOV AL, [SI]
17 BACK: INC SI
18 CMP AL, [SI]
19 JNC FWD
20 MOV AL, [SI]
21 FWD: DEC CL
22 JNZ BACK
23 MOV RES, AL


new open examples save compile emulat
;NAME - Yash Rout
;REG- 2041018131
;CSIT- D
.DATA
COUNT DB 03H
ARRAY DB 09H, 05H, 20H, 04H
SMALL DB ?
.CODE
MAIN PROC
MOV AX, DATA
MOV DS, AX
LEA SI, COUNT
MOV CL, [SI]
LEA SI, ARRAY
MOV AL, [SI]
LOOP1: INC SI
CMP AL , [SI]
JS AGAIN
MOV AL, [SI]
AGAIN:DEC CL
JNZ LOOP1
MOV SMALL, AL
END MAIN
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 07 04
BX 00 00
CX 00 00
DX 00 00
CS 0711
IP 0031
SS 0710
SP 0000
BP 0000
SI 0004
DI 0000
DS 0710
ES 0700
0711:0031
0711:0031
07135: 90 144 É NOP
07136: 90 144 É NOP
07137: 90 144 É NOP
07138: 90 144 É NOP
07139: 90 144 É NOP
0713A: 90 144 É NOP
0713B: 90 144 É NOP
0713C: 90 144 É NOP
0713D: 90 144 É NOP
0713E: 90 144 É NOP
0713F: 90 144 É NOP
07140: 90 144 É NOP
07141: F4 244 ⌠ HLT
07142: 00 000 NULL ADD [BX + SI], AL
07143: 00 000 NULL ADD [BX + SI], AL
07144: 00 000 NULL ADD [BX + SI], AL
07145: 00 000 NULL ADD [BX + SI], AL
07146: 00 000 NULL ADD [BX + SI], AL
07147: 00 000 NULL ADD [BX + SI], AL
07148: 00 000 NULL ADD [BX + SI], AL
07149: 00 000 NULL ADD [BX + SI], AL
0714A: 00 000 NULL ...
screen source reset aux vars debug stack flags
original source code
07 SMALL DB ?
08 .CODE
09 MAIN PROC
10 MOV AX, DATA
11 MOV DS, AX
12 LEA SI, COUNT
13 MOV CL, [SI]
14 LEA SI, ARRAY
15 MOV AL, [SI]
16 LOOP1: INC SI
17 CMP AL , [SI]
18 JS AGAIN
19 MOV AL, [SI]
20 AGAIN:DEC CL
21 JNZ LOOP1
22 MOV SMALL, AL

040

```
;NAME - Yash Rout
;REG- 2041018131
;CSIT- D
;EXP 5 OBJ 3

.DATA
COUNT DB 04H
 ARR DB  0F4H, 32H, 05H, 04H
.CODE
MAIN PROC
MOV AX, DATA
MOV DS, AX
MOV CH, COUNT
DEC CH
UP2: MOV CL, CH
LEA SI, ARR
UP1: MOV AL, [SI]
CMP AL, [SI+1]
JC DOWN
MOV DL, [SI+1]
XCHG [SI], DL
MOV [SI+1],DL
DOWN: INC SI
DEC CL
JNZ UP1
DEC CH
JNZ UP2

END MAIN
```

original source code

```
MOV DS, AX
MOV CH, COU
DEC CH
UP2: MOV CL, CH
LEA SI, ARR
UP1: MOV AL, [SI]
CMP AL, [SI+1]
JC DOWN
MOV DL, [SI+1]
XCHG [SI], DL
MOV [SI+1],DL
DOWN: INC SI
DEC CL
JNZ UP1
DEC CH
JNZ UP2
```

file math debug view external virtual devices virtual drive help

Load | reload | step back | single step | run | step delay ms: 0

registers

	H	L
AX	07	05
BX	00	00
CX	00	00
DX	00	05
CS	0711	
IP	003C	
SS	0710	
SP	0000	
BP	0000	
SI	0002	
DI	0000	
DS	0710	
ES	0700	

0711:003C

```
07148: 90 144 É
07149: 90 144 É
0714A: 90 144 É
0714B: 90 144 É
0714C: F4 244 ⌠
0714D: 00 000 NULL
0714E: 00 000 NULL
0714F: 00 000 NULL
07150: 00 000 NULL
07151: 00 000 NULL
07152: 00 000 NULL
07153: 00 000 NULL
07154: 00 000 NULL
07155: 00 000 NULL
07156: 00 000 NULL
07157: 00 000 NULL
07158: 00 000 NULL
07159: 00 000 NULL
0715A: 00 000 NULL
0715B: 00 000 NULL
0715C: 00 000 NULL
0715D: 00 000 NULL
```

0711:003C

```
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

screen | source | reset | aux | vars | debug | stack | flags

```
;NAME - Yash Rout
;REG- 2041018131
;CSIT- D
;LAB 5 OBJ 4
.DATA

COUNT DB 04H
ARRAY DB OF 4H, 32H, 05H, 04H
.CODE
MAIN PROC
MOV AX, DATA
MOV DS, AX
MOV CH, COUNT
 DEC CH
UP2: MOV CL, CH
LEA SI, ARRAY
UP1: MOV AL, [SI]
CMP AL, [SI+1]
JNC DOWN
MOV DL, [SI+1]
DOWN: INC SI
DEC CL
JNZ UP1
DEC CH
JNZ UP2
END MAIN
```

original source code

```
MAIN PROC
MOV AX, DATA
MOV DS, AX
MOV CH, COUNT
DEC CH
UP2: MOV CL, CH
LEA SI, ARRAY
UP1: MOV AL, [SI]
CMP AL, [SI+1]
JNC DOWN
MOV DL, [SI+1]
DOWN: INC SI
DEC CL
JNZ UP1
DEC CH
JNZ UP2
```

file math debug view external virtual devices virtual drive help

Load | reload | step back | single step | run | step delay ms: 0

registers

	H	L
AX	07	04
BX	00	00
CX	00	00
DX	00	32
CS	0711	
IP	0037	
SS	0710	
SP	0000	
BP	0000	
SI	0002	
DI	0000	
DS	0710	
ES	0700	

0711:0037

```
07147: F4 244 ⌠
07148: 00 000 NULL
07149: 00 000 NULL
0714A: 00 000 NULL
0714B: 00 000 NULL
0714C: 00 000 NULL
0714D: 00 000 NULL
0714E: 00 000 NULL
0714F: 00 000 NULL
07150: 00 000 NULL
07151: 00 000 NULL
07152: 00 000 NULL
07153: 00 000 NULL
07154: 00 000 NULL
07155: 00 000 NULL
07156: 00 000 NULL
07157: 00 000 NULL
07158: 00 000 NULL
07159: 00 000 NULL
0715A: 00 000 NULL
0715B: 00 000 NULL
0715C: 00 000 NULL
```

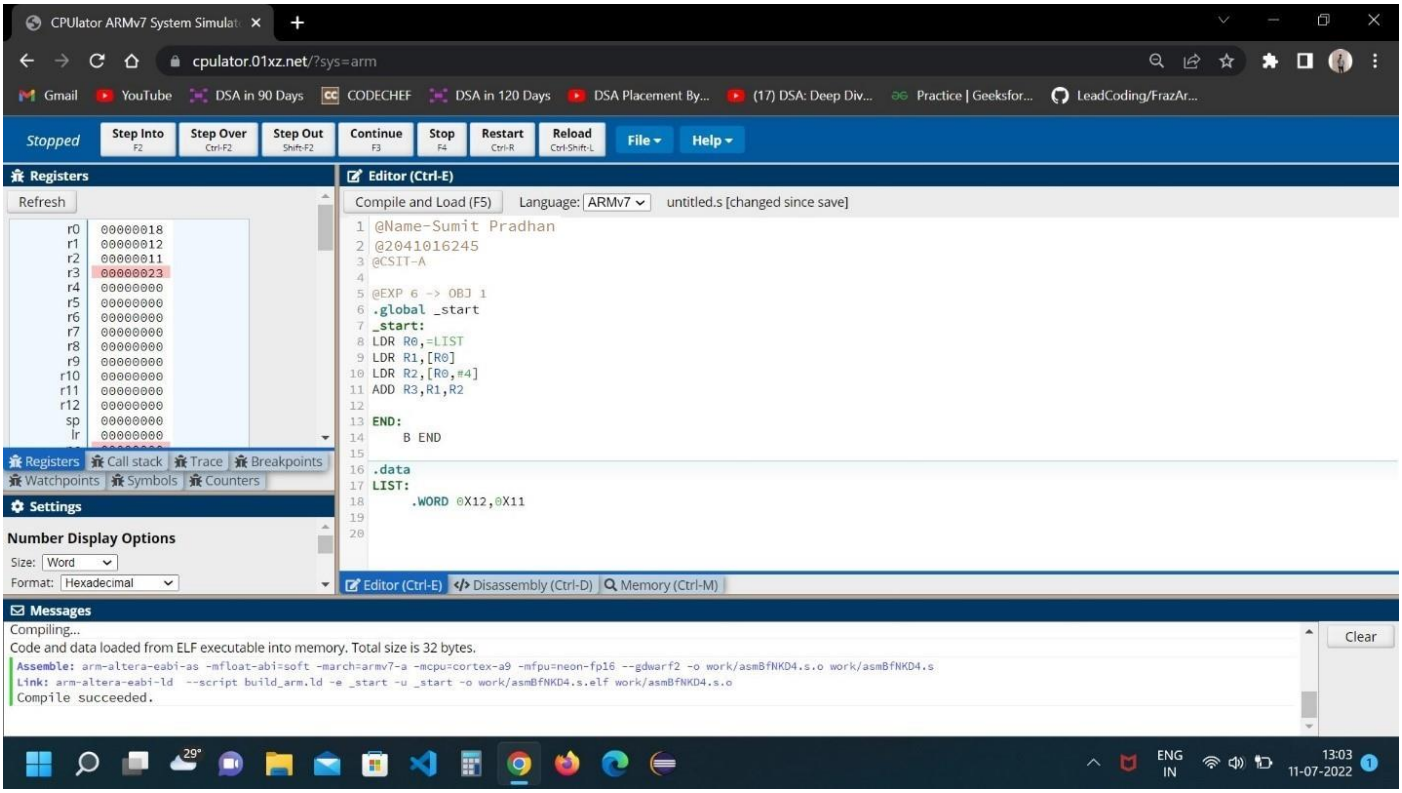
0711:0036

```
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

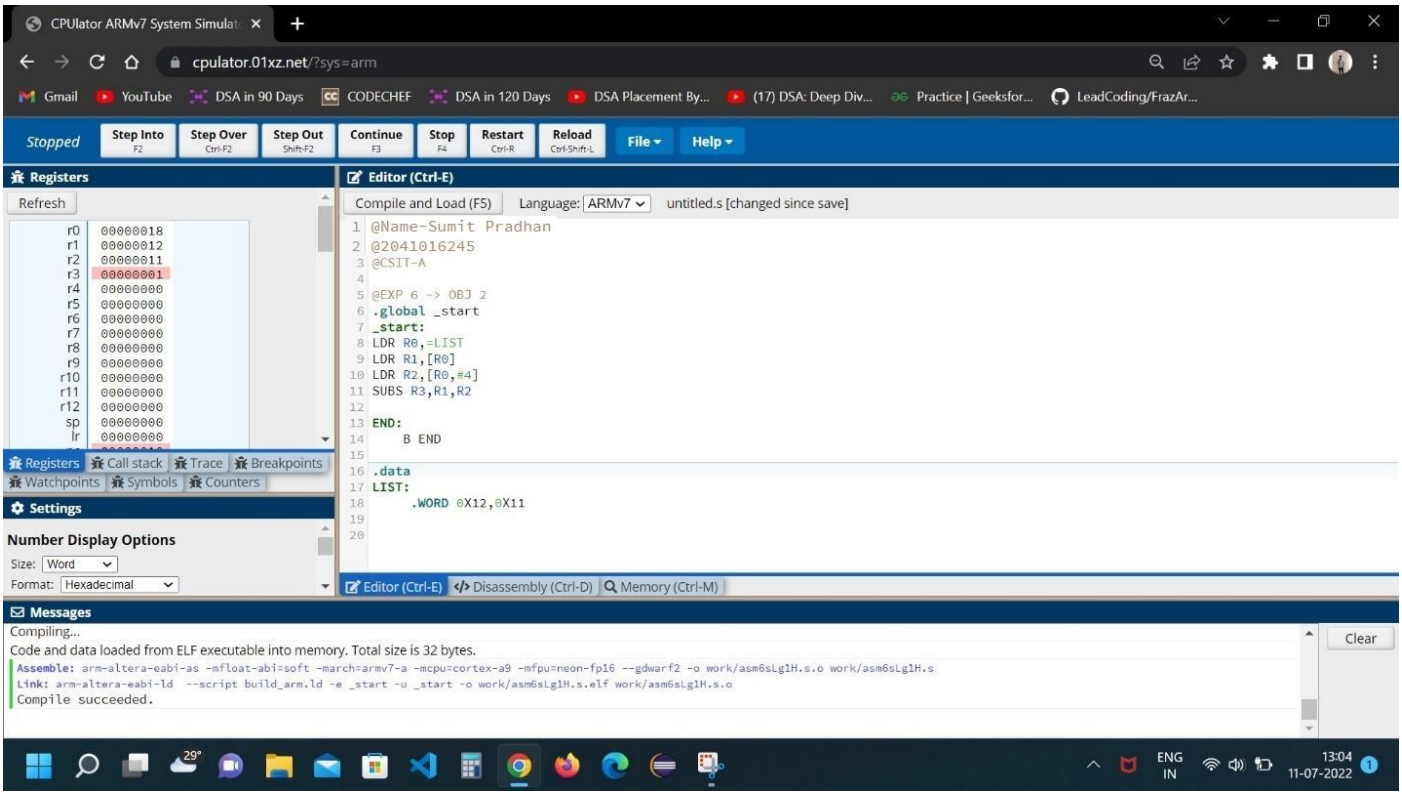
screen | source | reset | aux | vars | debug | stack | flags

Observation (with screenshots)

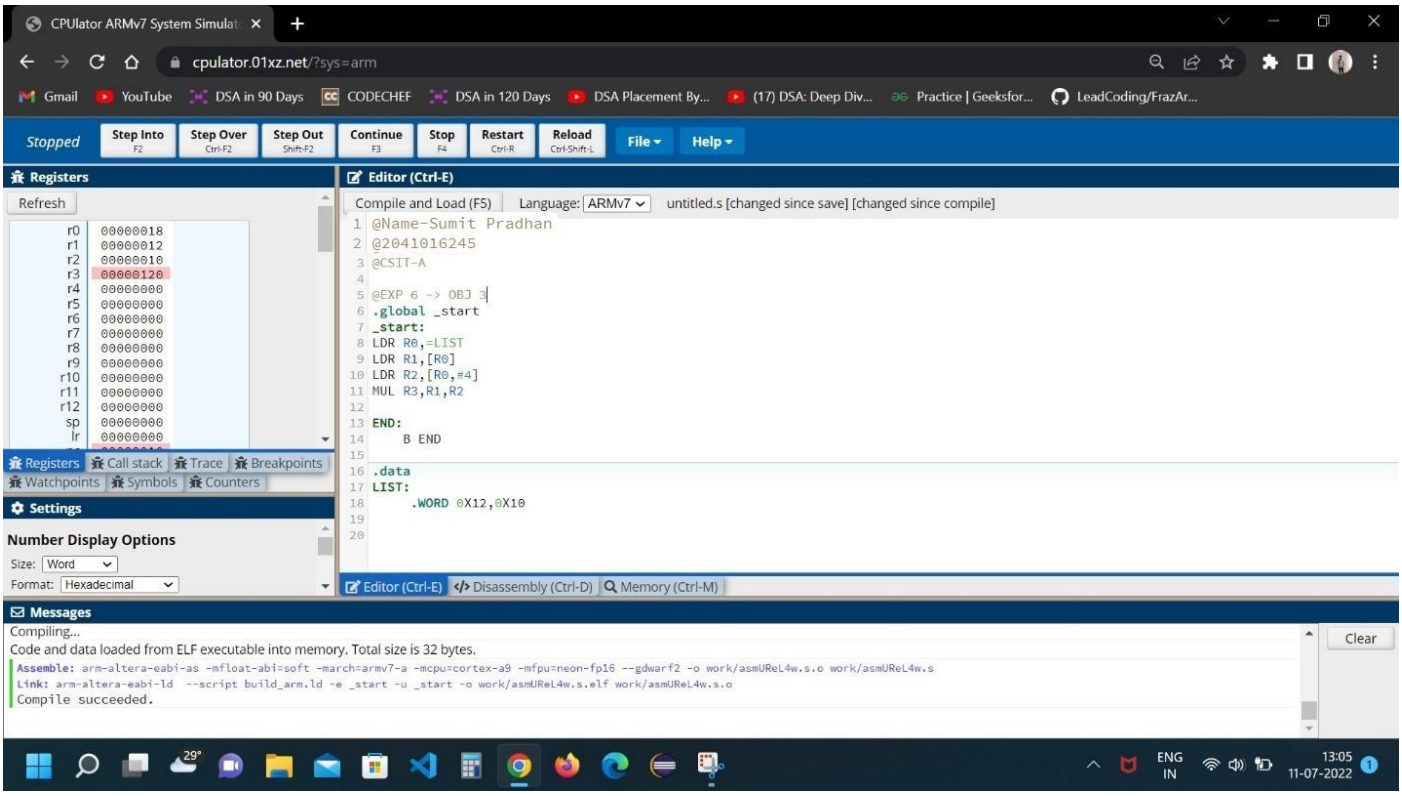
Objective 1



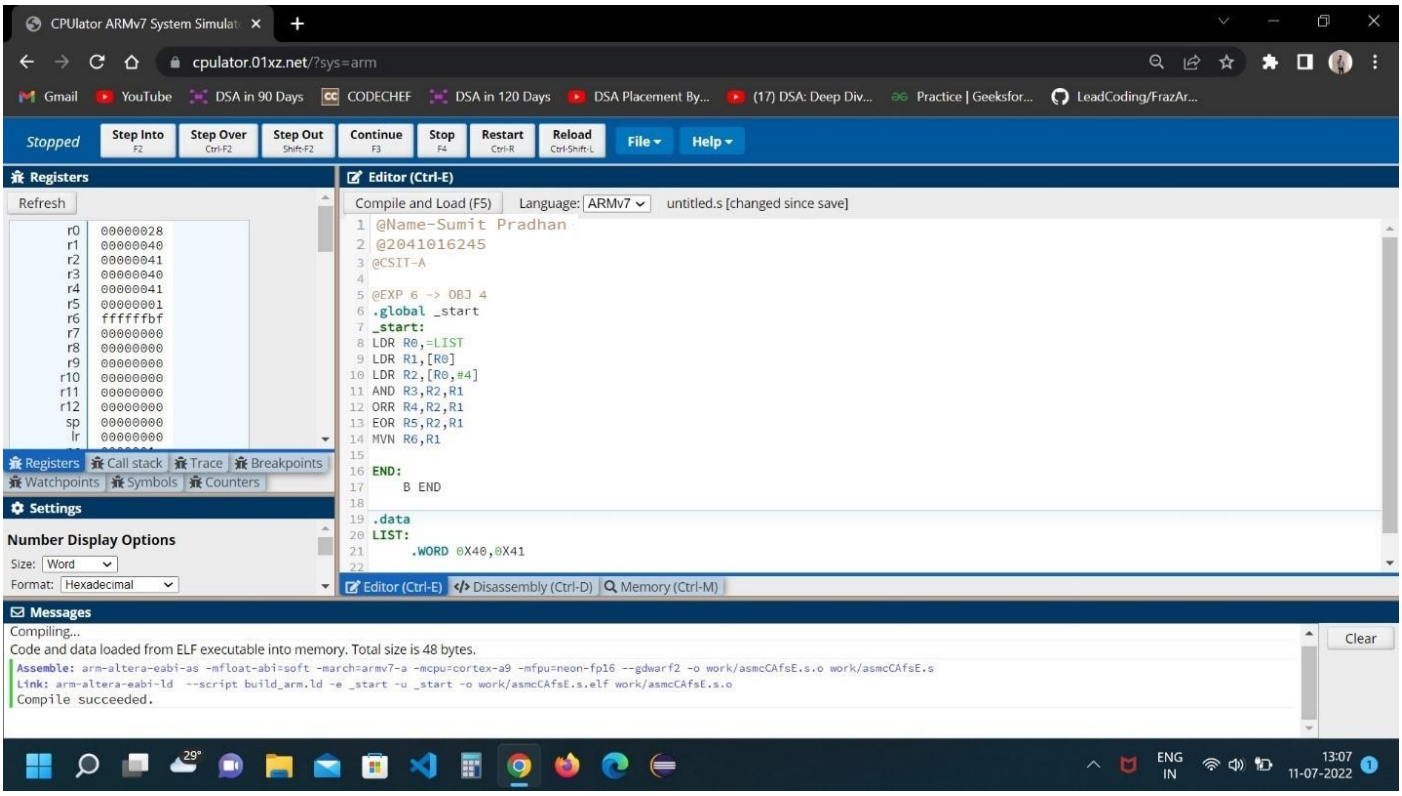
Objective 2



Objective 3

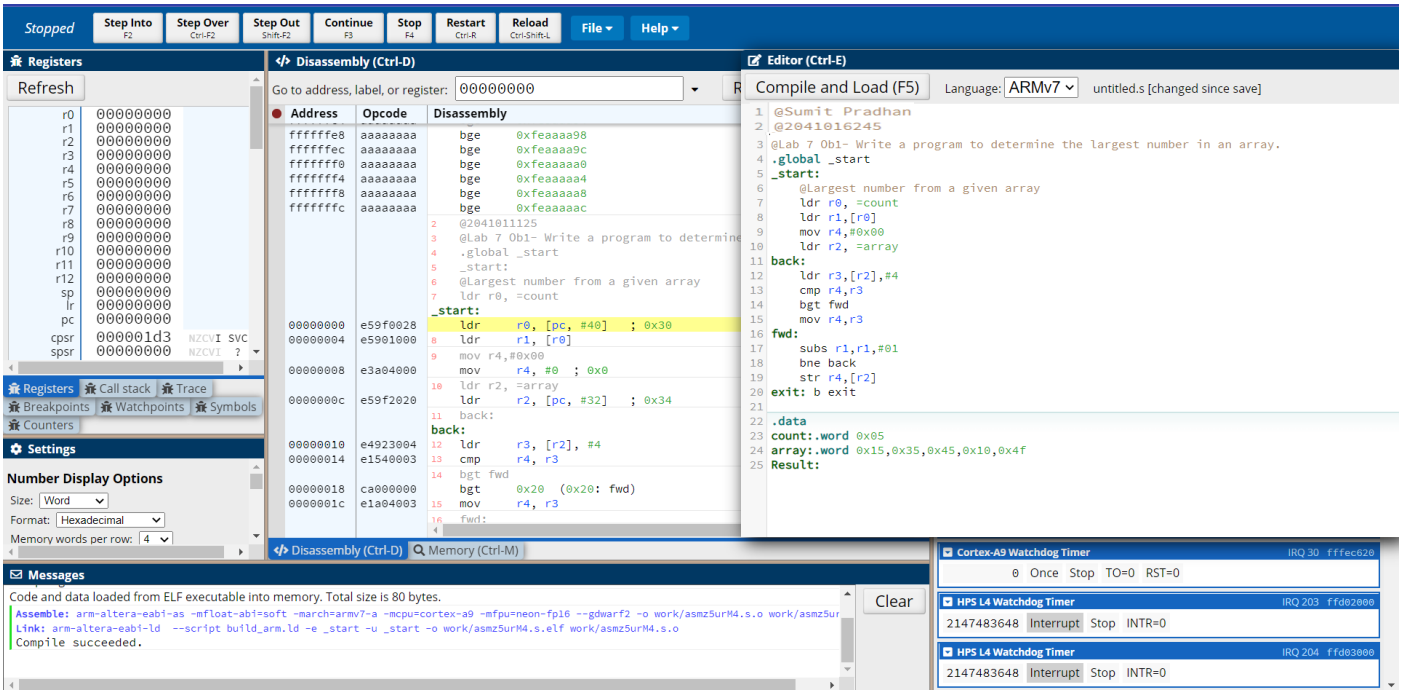


Objective 4

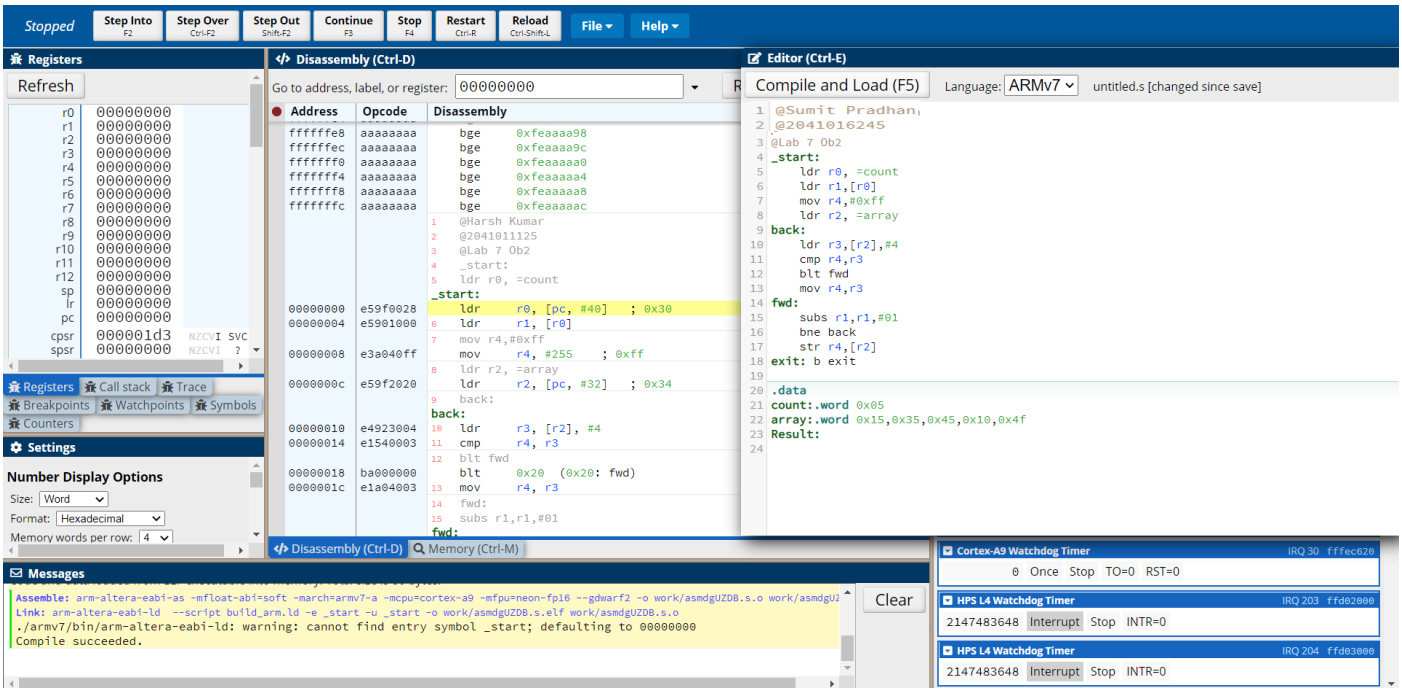


Observation (with screenshots)

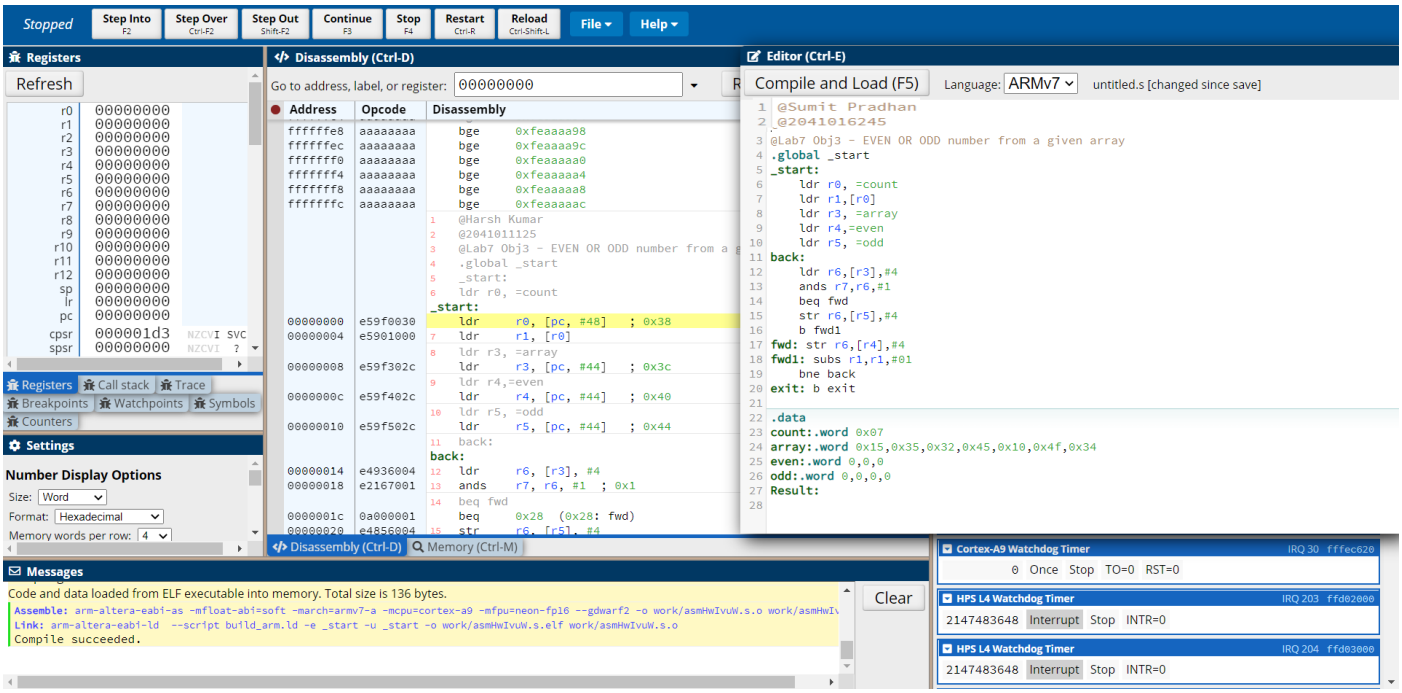
Objective 1



Objective 2



Objective 3


Stopped
Step Into F2
Step Over Ctrl-F2
Step Out Shift-F2
Continue F3
Stop F4
Restart Ctrl-R
Reload Ctrl-Shift-L
File
Help
Registers
Refresh
r0 00000000
r1 00000000
r2 00000000
r3 00000000
r4 00000000
r5 00000000
r6 00000000
r7 00000000
r8 00000000
r9 00000000
r10 00000000
r11 00000000
r12 00000000
sp 00000000
lr 00000000
pc 00000000
cpsr 000001d3 NZCVI SVC
spsr 00000000 NZCVI ?
Registers
Call stack
Trace
Breakpoints
Watchpoints
Symbols
Counters
Settings
Number Display Options
Size: Word
Format: Hexadecimal
Memory words per row: 4
Disassembly (Ctrl-D)
Go to address, label, or register: 00000000
Address
Opcode
Disassembly
fffffff8 aaaaaaaa bge 0xfeaaaa98
fffffffc aaaaaaaa bge 0xfeaaaa9c
00000000 aaaaaaaa bge 0xfeaaaaa0
00000004 aaaaaaaa bge 0xfeaaaaa4
00000008 aaaaaaaa bge 0xfeaaaaa8
0000000c aaaaaaaa bge 0xfeaaaaac
1 @Harsh Kumar
2 @2041011125
3 @Lab7 Obj3 - EVEN OR ODD number from a
4 .global _start
5 _start:
6 ldr r0, =count
_start:
00000000 e59f0030 ldr r0, [pc, #48] ; 0x38
00000004 e5901000 7 ldr r1, [r0]
8 ldr r3, =array
00000008 e59f302c ldr r3, [pc, #44] ; 0x3c
9 ldr r4,=even
0000000c e59f402c ldr r4, [pc, #44] ; 0x40
10 ldr r5, =odd
00000010 e59f502c ldr r5, [pc, #44] ; 0x44
11 back:
back:
00000014 e4936004 12 ldr r6, [r3], #4
00000018 e2167001 13 ands r7, r6, #1 ; 0x1
14 beq fwd
0000001c 0a000001 beq 0x28 (0x28: fwd)
00000020 e4856004 15 str r6, [r5], #4
Disassembly (Ctrl-D)
Memory (Ctrl-M)
Editor (Ctrl-E)
Compile and Load (F5)
Language: ARMv7
untitled.s [changed since save]
1 @Sumit Pradhan
2 @2041016245
3 @Lab7 Obj3 - EVEN OR ODD number from a given array
4 .global _start
5 _start:
6 ldr r0, =count
7 ldr r1,[r0]
8 ldr r3, =array
9 ldr r4,=even
10 ldr r5, =odd
11 back:
12 ldr r6,[r3],#4
13 ands r7,r6,#1
14 beq fwd
15 str r6,[r5],#4
16 b fwd1
17 fwd: str r6,[r4],#4
18 fwd1: subs r1,r1,#01
19 bne back
20 exit: b exit
21
22 .data
23 count:.word 0x07
24 array:.word 0x15,0x35,0x32,0x45,0x10,0x4f,0x34
25 even:.word 0,0,0
26 odd:.word 0,0,0,0
27 Result:
28
Messages
Code and data loaded from ELF executable into memory. Total size is 136 bytes.
Assemble: arm-altera-eabi-as -mfloat-abi=soft -march=armv7-a -mcpu=cortex-a9 -mfpu=neon-fp16 --gdwarf2 -o work/asmHwIvuW.s.o work/asmHwIv
Link: arm-altera-eabi-ld --script build_arm.ld -e _start -u _start -o work/asmHwIvuW.s.elf work/asmHwIvuW.s.o
Compile succeeded.
Clear
Cortex-A9 Watchdog Timer IRQ 30 fffec620
0 Once Stop TO=0 RST=0
HPS L4 Watchdog Timer IRQ 203 ffd02000
2147483648 Interrupt Stop INTR=0
HPS L4 Watchdog Timer IRQ 204 ffd03000
2147483648 Interrupt Stop INTR=0