CO Lab Assignment-6

--- Report--

Roll number: CSB19057

Name: PRASANJIT DUTTA

Objectives:

1. To learn the basic functions of a debugger-
   a. accessing registers and memory,
   b. editing program and data,
   c. single-stepping, executing program;
2. Observe the following-
   a. the location of the data elements, the location of the code;
   b. the program execution, one instruction at-a-time;
   c. the content of the registers and the flags, the change in them with execution of the instructions;
   d. Observing the execution of branch instructions;
   e. Observing the execution of program loop.

Exercises:

```
CR EQU 0DH

LF EQU 0AH

MAX_SIZE EQU 20

.MODEL SMALL

.STACK 100H

.DATA

array DW MAX_SIZE DUP(?)

sum DW 0

prompt_msg DB 'Enter some non-zero single digit positive integers.', CR, LF, '$'

output_msg DB CR,LF, 'Sum of the integers is: $'

ten DB 10
```

```
.CODE
main PROC
.STARTUP
lea dx, prompt_msg
mov ah, 09h
int 21h
mov bx, OFFSET array
mov cx, MAX_SIZE
mov dx, 0
read:
   mov ax, 0
   mov ah, 01h ; to read a character from the keyboard, 1 byte char gets stored in AL
   int 21h
   sub al, 30h ; to convert from character to number
   cmp al, 0
   je calc_sum
   mov ah, 0 ; AH = 0, AL = [num] : AX = [num]
   mov [bx], ax
   add bx, 2
   dec cx
   inc  dx
   cmp cx, 0
   je calc_sum
   jmp read

calc_sum:
mov bx, OFFSET array
mov cx, dx

mov ax, 0
;summation of array elements using conditional jump instruction
summa:
 add ax, [bx]
add bx, 2
sub cx, 1
cmp cx, 0
jnz summa

mov sum, ax
mov dx,OFFSET output_msg
mov ah, 09h
int 21h
mov  dx, 0
mov ax, sum
div ten ; al = ax/10, ah = ax%10
mov dx, ax
add dh, 30h ; converting number to character
add dl, 30h
mov ah, 02h
int 21h
mov dl, bh
mov ah, 02h
int 21h
mov ah ,4ch
mov al ,0

int 21h

main  ENDP

END main
```

Observations:

Initial Register Contents:

AL	AH	BL	BH	CL	CH	DL	DH	IP	SP	BP	SI	DI	CS	DS	SS	ES
00	00	00	00	F5	01	00	00	0000	0100	0000	0000	0000	0728			

Variables:

Name	**Array**	**Num**	**output_msg**	**PROMPT_MSG**
Location	000h	000h	0Dh	45h
Bytes occupied	40	2	25	51

Execution:

Step	Instruction Executed		IP	Operand 1		Operand 2		Result		Flags					Remarks
	Assembly	Machine Code		Reg	Value	Reg	Value	Reg	Value	OF	SF	ZF	AF	CF	
1	lea	BA186	0005	dx	0002Ah	-	-	dx	0002Ah	0	0	0	0	0	
2	mov	B4180	0008	ah	09h	-	-	ah	09h	0	0	0	0	0	
3	INT	CD205	000A												
4	mov	BB187	000C	bx	00000h	-	-								
5	mov	B9185	000F	cx	00014h	-	-	cx	00014h	0	0	0	0	0	
6	sub	2B043	0012	dx	0002Ah	dx	0002Ah	dx	0	0	0	0	0	0	
7	mov	2B043	0014	ax	09h	ax	09h	ax	0	0	0	1	0	0	
8	mov	B4180	0016	ah	01h	-	-	ah	01h	0	0	1	0	0	
9	INT	CD205	0018	-	-	-	-	-	-	0	0	1	0	0	
10	SUB	2C044	001	al	1		30h	al	1-30h	0	0	1	0	0	

			A											
11	CMP	3C060	001C	al	1		00h	al	1-00h	0	0	0	0	0
12	mov	B4180	0020	ah	0			ah	01	0	0	0	0	0
13	mov	89137	0020	[bx]	-	ax	1	[bx]	ax	0	0	0	0	0
14	add	83131	0024	bx	-		02h	bx	bx+02h	0	0	0	0	0
15	dec	83131	0027	cx	20	-	-	cx	cx-01h	0	0	0	0	0
16	inc	83131	002A	dx	0	-	01h	dx	dx+01h	0	0	0	0	0
17	CMP	83131	002D	cx	19	-	00h	cx	19	0	0	0	0	0
18	jnz	75117	0030	-	-	-	-	-	-	0	0	0	0	0
20	mov	B4180	0016	ah	-	-	01h	ah	01h	0	0	0	0	0
21	INT	CD205	0018	-	-	-	-	-	-	0	0	0	0	0
22	sub	2C044	001A	al	1		30h	al	1-30h	0	0	0	0	0
23	CMP	3C060	001C	al	-		00h	al	1	0	0	0	0	0
24	mov	2B043	0014	ax	-	ax	-	ax	0	0	0	0	0	0
25	mov	B4180	0016	ah	-	-	01h	ah	01h	0	0	0	0	0
26	INT	CD205	0018	-	-	-	-	-	-	0	0	0	0	0
27	sub	2C044	001A	al	0		30h	al	0-30h	0	0	1	0	0
28	CMP	3C060	001C	al	-		00h	al	0	0	0	1	0	0
33	add	03003	003A	ax	-	-	[bx]	ax	ax+[bx]	0	0	1	0	0
34	add	83131	003C	bx	-	-	02h	bx	bx+02h	0	0	1	0	0
35	sub	83131	003F	cx		-	01h	cx	cx-01h	0	0	1	0	0
36	CMP	83131	0042	cx	-	-	00h	cx	1	0	0	1	0	0
29	je	74116	001E	-	-	-	-	-	-	0	0	1	0	0
30	mov	BB187	0032	bx	-	-	00000h	bx	00000h	0	0	1	0	0
31	mov	8B139	0035	cx	-	dx	-	cx	dx	0	0	1	0	0

32	mov	B8184	0037	ax	-		00000h	ax	00000h	0	0	1	0	0
37	jne	75117	0045	-	-	-	-	-	-	0	0	1	0	0
38	add	03003	003A	ax	-	-	[bx]	ax	ax+[bx]	0	0	1	0	0
39	add	83131	003C	bx	-	-	02h	bx	bx+02h	0	0	1	0	0
40	sub	83131	003F	cx		-	01h	cx	cx-01h	0	0	1	0	0
41	CMP	83131	0042	cx	-	-	00h	cx	0	0	0	1	0	0
42	mov	A3163	0047	sum	-	ax	2	sum	ax	0	0	1	0	0
43	mov	BA186	004A	dx	-		00060h	dx	00060h	0	0	1	0	0
44	mov	B4180	0008	ah	09h	-	-	ah	09h	0	0	1	0	0
45	INT	CD205	0018	-	-	-	-	-	-	0	0	1	0	0
46	sub	2B043	0051	dx	-	dx	-	dx	0	0	0	1	0	0
47	mov	A1161	0053	ax	-		[00028h]	ax	2	0	0	1	0	0
48	mov	8B139	005A	bx	-	ax	-	bx	ax	0	0	1	0	0
49	add	80128	005C	bh	-		30h	bh	bh+30h	0	0	1	0	0
50	add	80128	005F	bl	-	-	30h	bl	bl+30h	0	0	0	0	0
51	mov	8A138	0062	dl	-	bl	-	dl	dl+bl	0	0	0	0	0
52	mov	B4180	0064	ah	02h	-	-	ah	02h	0	0	0	0	0
53	INT	CD205	0018	-	-	-	-	-	-	0	0	0	0	0

Machine code:

Instruction No.	Starting memory location	Machine Code in Hexadecimal
1	07285	BA186
2	07288	B4180
3	0728A	CD205
4	0728C	BB187
5	0728F	B9185
6	07292	2B043

7	07294	2B043
8	07296	B4180
9	07298	CD205
10	0729A	2C044
11	0729c	3C060
12	072A0	B4180
13	072A2	89137
14	072A4	83131
15	072A7	83131
16	072AA	83131
17	072AD	83131
18	072B0	75117
19	07294	2B043
20	07296	B4180
21	07298	CD205
22	0729A	2C044
23	0729C	3C060
24	07294	2B043
25	07296	B4180
26	07298	CD205
27	0729A	2C044
28	0729C	3C060
29	0729E	74116
30	072B2	BB187
31	072B5	8B139
32	072B7	B8184
33	072BA	03003
34	072BC	83131
35	072BF	83131
36	072C2	83131
37	072C5	75117
38	072BA	03003
39	072BC	83131
40	072BF	83131
41	072C2	83131
42	072C7	A3163
43	072CA	BA186
44	072CD	B4180
45	072CF	CD205
46	072D1	2B043
47	072D3	A1161
48	072DA	8B139
49	072DC	80128
50	072DF	80128
51	072E2	8A138
52	072E4	B4180
53	072E6	CD205

Learning Outcome:

1. Debugger functions and their use: It allows us to run the target program under controlled conditions

2. Location of data and code in a program: All the code and data are stored in the memory in the code segment and the data segment respectively.

3. Observing the machine instructions and their execution: The machine instructions include basic fetch and operations instructions with some prefined macros and microprograms. Their execution involves fetch ,decode and the perform of instructions.