

Experiment No. 7

Aim: Arrange the given numbers of series in order of increasing and decreasing magnitude of numbers.

Algorithm:

- 1) Assign source index a value to point at an address in RAM (let's say '0100:1000' or 1000h).
- 2) Stores the value at address 2000h in RAM to the CL register (number of terms) decrements value in CL register by 1 (n-1 terms).
- 3) Assigns source index a value to point at an address '0100:1000' in RAM stores the value at address '0100:1000' in RAM to the CL register (number of terms), decrements value in CH register by 1 (n-1 terms).
- 4) Assigns source index a value to point at an address '0100:1010' in RAM.
- 5) Stores the value at address pointed by source index in RAM to the AL register, increments the value of source index. Then compares the value in AL register and at RAM address pointed by source index.
- 6) Jumps to 'end' section of code depending on the carry flag status ("jc" for increasing, "jnc" for decreasing).
- 7) Exchanges the value in the AL register and at RAM address pointed by source index, exchanges the value in the AL register and at RAM address pointed by 'si-1'.
- 8) Decrements value in CH register by 1, jumps of 'inner' section of code (inner loop, L2) as long as the value in CH register is not 0, decrements value in CL register by 1.
- 9) Jumps of 'outer' section of code (outer loop, L1) as long as the value in CL register is not 0.
- 10) The program is Halted.

Code:

```
;Increasing
mov si,1000h
mov cl,[si]
dec cl
L1:
mov si,1000h
mov ch,[si]
dec ch
mov si,1010h
L2:
mov al,[si]
inc si
cmp al,[si]
jc Chk
xchg al,[si]
xchg al,[si-1]
Chk:
dec ch
jnz L2
dec cl
jnz L1
hlt

ret
```

Output:

Before execution

Random Access Memory

0100:1000 update table list

0100:1000	05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1010	06 0C 07 0B 03 00 00 00 00 00 00 00 00 00 00 00
0100:1020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

emulator: noname.bin_ original source co...

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	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000 0100:0000

Address	Hex	Asm
01000: BE 190	BE 190	MOV SI, 01000h
01001: 00 000	00 000	MOV CL, [SI]
01002: 10 016	10 016	DEC CL
01003: 8A 138	8A 138	MOV SI, 01000h
01004: 0C 012	0C 012	MOV CH, [SI]
01005: FE 254	FE 254	DEC CH
01006: C9 201	C9 201	MOV SI, 01010h
01007: BE 190	BE 190	MOV AL, [SI]
01008: 00 000	00 000	INC SI
01009: 10 016	10 016	CMF AL, [SI]
0100A: 8A 138	8A 138	JB 01Dh
0100B: 2C 044	2C 044	XCHG [SI], AL
0100C: FE 254	FE 254	XCHG [SI] - 01h, AL
0100D: CD 205	CD 205	DEC CH
0100E: BE 190	BE 190	JNE 011h
0100F: 10 016	10 016	DEC CL
01010: 10 016	10 016	JNE 07h
01011: 8A 138	8A 138	HIT
01012: 04 004	04 004	NOP
01013: 46 070	46 070	NOP
01014: 3A 058	3A 058	NOP
01015: 04 004	04 004	...

01 mov si, 1000h
02 mov cl, [si]
03 dec cl
04
05 L1:
06 mov si, 1000h
07 mov ch, [si]
08 dec ch
09 mov si, 1010h
10
11 L2:
12 mov al, [si]
13 inc si
14 cmp al, [si]
15 jc Chk
16 xchg al, [si]
17 xchg al, [si-1]
18
19 Chk:
20 dec ch
21 jnz L2
22 dec cl
23 jnz L1
24
25 hlt
26
27

flags

CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

screen source reset aux vars debug stack flags analyse

After execution

Random Access Memory

0100:1000 update table list

0100:1000	05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1010	03 06 07 0B 0C 00 00 00 00 00 00 00 00 00 00 00
0100:1020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100:1060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

emulator: noname.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	0B
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0025	
SS	0100	
SP	FFFE	
BP	0000	
SI	1014	
DI	0000	
DS	0100	
ES	0100	

0100:0025 0100:0025

Address	Hex	Asm	Comment
01011:	8A 138	MOV SI, 01000h	
01012:	04 004	MOV CL, [SI]	
01013:	46 070	DEC CL	
01014:	3A 058	MOV SI, 01000h	
01015:	04 004	MOV CH, [SI]	
01016:	72 114	DEC CH	
01017:	05 005	MOV SI, 01010h	
01018:	86 134	MOV AL, [SI]	
01019:	04 004	INC SI	
0101A:	86 134	MOV AL, [SI]	
0101B:	44 068	MOV AL, [SI]	
0101C:	FF 255	RES	
0101D:	FE 254	XCHG [SI], AL	
0101E:	CD 205	XCHG [SI] - 01h, AL	
0101F:	75 117	DEC CH	
01020:	F0 240	JNE 011h	
01021:	FE 254	DEC CL	
01022:	C9 201	JNE 07h	
01023:	75 117	HLL	
01024:	E2 226	NOP	
01025:	F4 244	NOP	
01026:	90 144	NOP	

original source co...

```

01 mov si,1000h
02 mov cl,[si]
03 dec cl
04
05 L1:
06 mov si,1000h
07 mov ch,[si]
08 dec ch
09 mov si,1010h
10
11 L2:
12 mov al,[si]
13 inc si
14 cmp al,[si]
15 jc Chk
16 xchg al,[si]
17 xchg al,[si-1]
18
19 Chk:
20 dec ch
21 jnz L2
22 dec cl
23 jnz L1
24
25 hlt
26
27

```

flags

CF	1
ZF	1
SF	0
OF	0
PF	1
AF	0
IF	1
DF	0

analyse

Decreasing:

```

;Decreasing
mov si,1000h
mov cl,[si]
dec cl
L1:
mov si,1000h
mov ch,[si]
dec ch
mov si,1010h
L2:
mov al,[si]
inc si
cmp al,[si]
jnc Chk
xchg al,[si]
xchg al,[si-1]
Chk:
dec ch
jnz L2
dec cl
jnz L1
hlt

```

Output:

The screenshot displays the Immunity Debugger interface with the following components:

- Memory View (Top):** Shows a memory dump starting at address 0100:1000. The data is displayed in hexadecimal and ASCII. The address 0100:1010 is highlighted.
- Emulator Window (Middle):** Displays the loaded file 'noname.bin'. It includes a toolbar with buttons for Load, reload, step back, single step, run, and step delay (ms: 0). Below the toolbar are tabs for registers, memory, and disassembly.
- Registers Window (Bottom Left):** Shows the state of various registers. The EAX register is highlighted with a value of 00000000.
- Disassembly Window (Bottom Right):** Shows the assembly code for the selected memory address (0100:1010). The code includes instructions like MOV SI, 01000h, DEC CL, MOV SI, 01000h, MOV CH, [SI], DEC CH, MOV SI, 01010h, MOV AL, [SI], INC SI, CMP AL, [SI], JNB 0101h, XCHG [SI], AL, XCHG [SI] - 01h, AL, DEC CH, JNE 011h, DEC CL, JNE 07h, HLT, and NOP.
- Flags Window (Bottom Right):** Shows the state of various flags. The CF (Carry Flag) is highlighted with a value of 0.

After execution

The screenshot displays a debugger interface with three main panels:

- Random Access Memory:** A table showing memory addresses from 0100:1000 to 0100:1060. The data is mostly zeros, with some non-zero values at 0100:1010 (0F 0A 07 03) and 0100:1011 (00 00 00 00).
- Registers:** A list of registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) with their current values. For example, AX is 00 03, BX is 00 00, CX is 00 00, DX is 00 00, CS is 0100, IP is 0025, SS is 0100, SP is FFFE, BP is 0000, SI is 1014, DI is 0000, DS is 0100, and ES is 0100.
- Source Code:** A list of assembly instructions. The instruction at address 0100:1025 is highlighted in blue: `HLT`. The instruction at address 0100:1026 is highlighted in yellow: `NOP`.

The flags panel on the right shows the status of various flags: CF (0), ZF (1), SF (0), OF (0), PF (1), AF (0), IF (1), and DF (0).

(ARKAJYOTI CHAKRABORTY 2K19/EP/022)