National University of Computer and Emerging Sciences

SOLVED 20L-0921

Lab Manual 03
Computer Organization and Assembly Language Lab

Course Instructor	Ms. Aleena
Lab Instructor (s)	Maham Saleem
Section	BCS 3E
Semester	Fall 2021

Department of Computer Science

FAST-NU, Lahore, Pakistan

Chapter 4 - Bit Manipulations
Lab Manual 03

Activity 1: Write a program to swap every pair of bits in the AX register i.e. swap bit no 0 with bit no 1, bit no 2 with bit no 3 and so on.

Sample Run:

AX before Swap	**10 11 00 10 01 01 11 01**
AX after Swap	**1 11 00 01 10 10** **11 10**

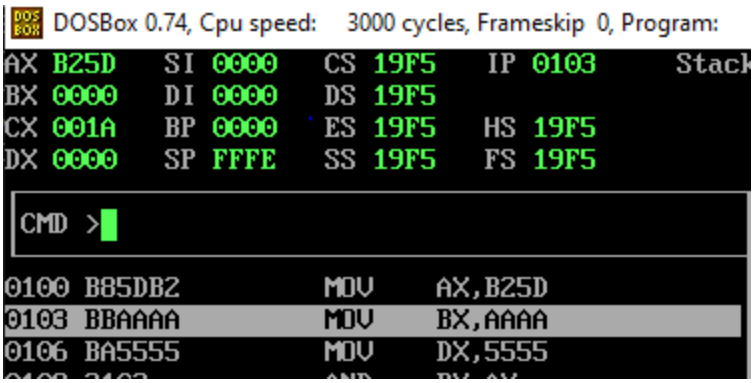
SOLUTION:

Activity 1 20L-0921 - Notepad

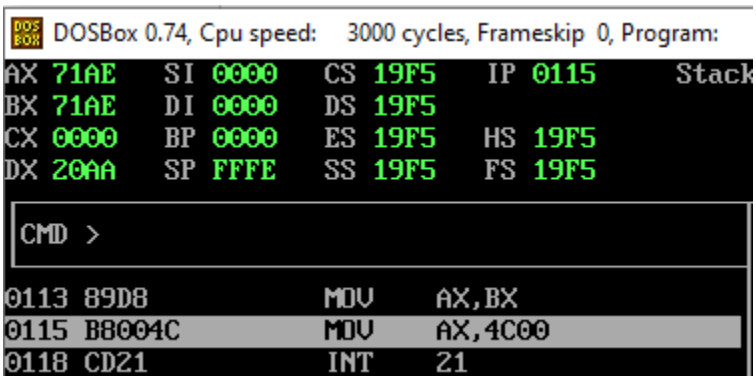
File Edit Format View Help

```
[org 0x100]
mov ax,1011001001011101b
mov bx,1010101010101010b
mov dx,0101010101010101b

and bx,ax
and dx,ax

shr bx,1
shl dx,1

or bx,dx

mov ax,bx

mov ax,0x4c00
int 0x21
```

Before run:



After run:



Activity 2: [Bit Manipulation] Calculate the number of one bits in BX and complement an equal number of least significant bits in AX. HINT: Use the XOR instruction.

Sample Run:

Initial value of BX	Total No of 1 Bits in BX	Initial value of AX	AX after Complementing 7 least significant bits
1011 0001 1000 1001	7	1010 1011 **1010 0101**	1010 1 1**101 1010**

SOLUTION:

*Activity 2 20L-0921 - Notepad

File Edit Format View Help

```
[org 0x0100]
mov bx,1011000110001001b
mov cx,0
mov dx,0
start:
shr bx,1
jnc noc
add cx,1
noc:
add dx,1
cmp dx,16
jnz start
mov ax,1010101110100101b
mov bx,0
start2:
xor ax,1
ror ax,1
add bx,1
cmp bx,cx
jnz start2
mov dx,0
start3:
rol ax,1
add dx,1
cmp dx,cx
jnz start3
mov ax,0x4c00
int 0x21
```

Before run:

```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:
AX 0000   SI 0000   CS 19F5   IP 0103     Stack
BX B189   DI 0000   DS 19F5
CX 0040   BP 0000   ES 19F5   HS 19F5
DX 0000   SP FFFE   SS 19F5   FS 19F5

CMD >

0100 BB89B1         MOV    BX,B189
0103 B90000         MOV    CX,0000
0106 BA0000         MOV    DX,0000
0109 D1EB           SHR    BX,1
010B 7304           JNC    0111
```

After run:

```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX ABDA   SI 0000   CS 19F5   IP 013B     Stack +0
BX 0007   DI 0000   DS 19F5                     +2
CX 0007   BP 0000   ES 19F5   HS 19F5           +4
DX 0007   SP FFFE   SS 19F5   FS 19F5           +6

CMD >

0139 75F6           JNZ    0131
013B B8004C         MOV    AX,4C00
013E CD21           INT    21
0140 0000           ADD    [BX+SI],AL
0142 0000           ADD    [BX+SI],AL
0144 0000           ADD    [BX+SI],AL
```

Activity 3: AX contains a number between 0-15. Write code to complement the corresponding bit in BX. For example if AX contains 6; complement the 6th bit of BX.

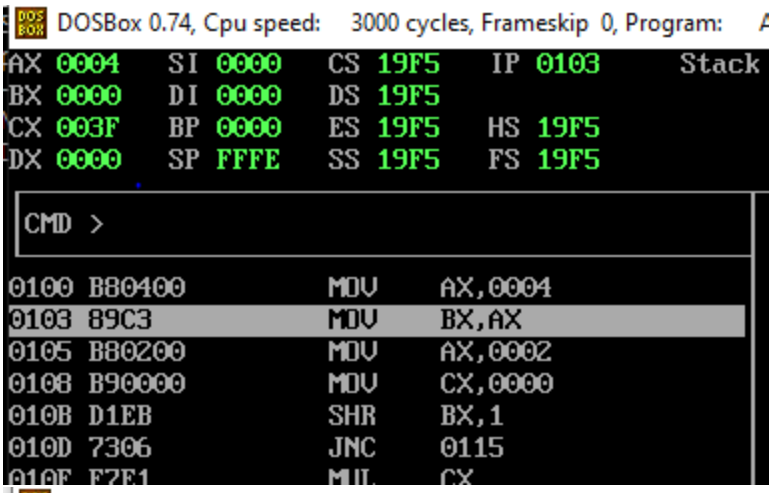
SOLUTION:

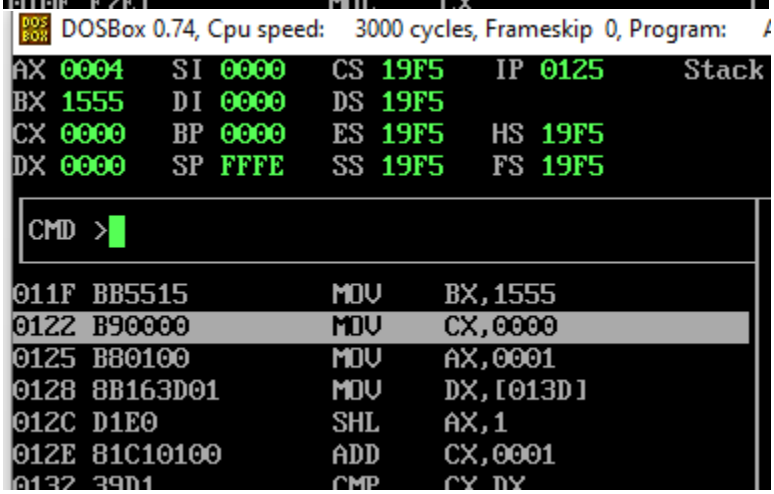
*Activity 3 20L-0921 - Notepad

File Edit Format View Help

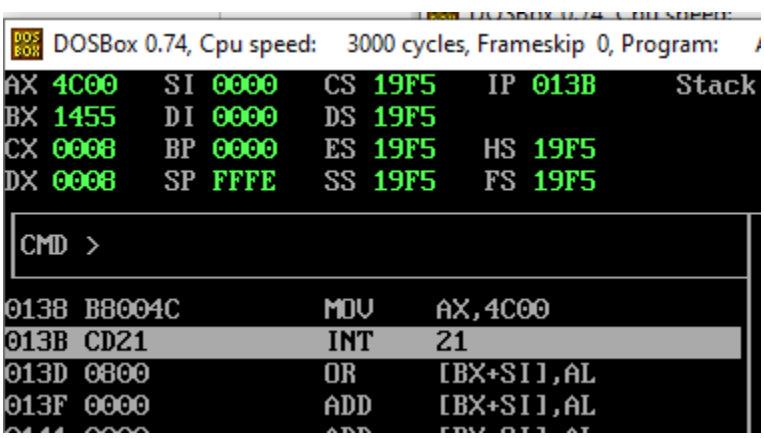
```
[org 0x0100]
mov ax,0000000000000100b
mov bx,ax
mov ax,0000000000000010b
mov cx,0
start:
shr bx,1
jnc con
mul cx
add [result],ax
con:
add cx,1b
cmp cx,0000000000010000b
jnz start
mov bx,0001010101010101b
mov cx,0
mov ax,1b
mov dx,[result]
start2:
shl ax,1
add cx,1
cmp cx,dx
jnz start2
xor bx,ax
mov ax,0x4c00
int 21h
result : dw 0
```

Before run:





After run:



Activity 4: Write a program to search a particular element from an array using binary search. If the element is found set AX to one and otherwise to zero. Binary Search searches a number from a sorted array. Shifting a number to right divides it by 2. Do not use division instruction use shifting for division.

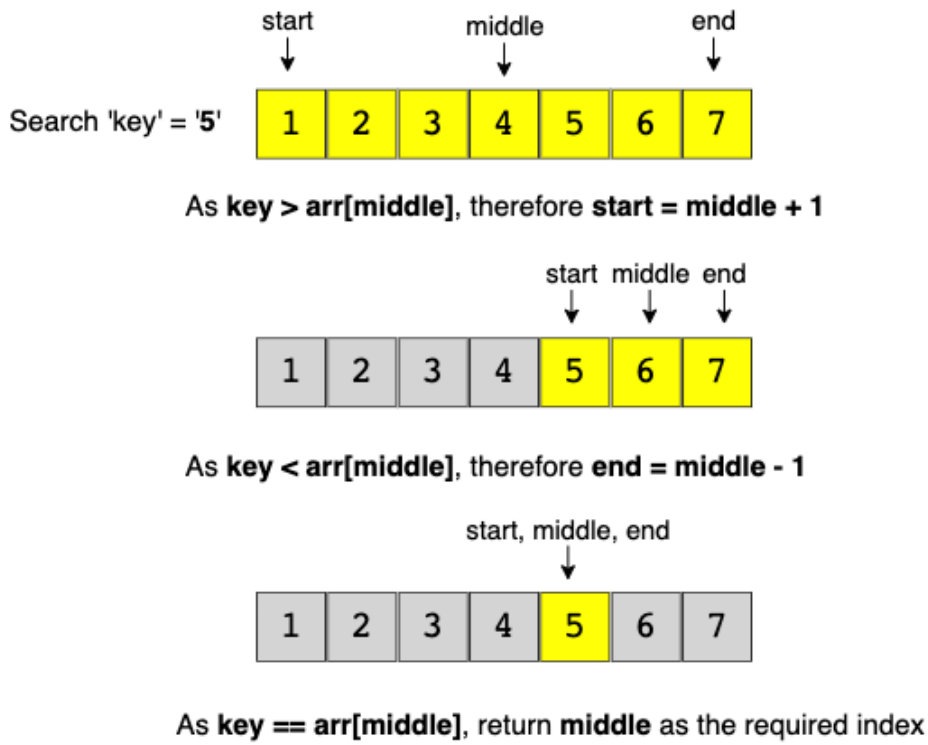


Figure 1: Binary Search Procedure

SOLUTION:

*Activity 4 20L-0921 - Notepad

File Edit Format View Help

```
[org 0x0100]
jmp start
array: dw 1h,2h,3h,4h,5h,6h,7h
start:
mov ax,5h ;key (ax used as a key in the program but in the end as result)
mov bx,0 ;mid
mov cx,12 ;last (6*2=12 multiply by 2 because word used)
mov dx,0 ;first
mov bx,cx
add bx,dx
shr bx,1 ; equivalent to mid=(last+first)/2
start2:
cmp[array+bx],ax
jl less
cmp [array+bx],ax
jne great
mov ax,1 ; ax=1 indicates key found
jmp end
great:
mov cx,bx
sub cx,2 ; subtract two instead of one becuase word used
jmp check
less:
mov dx,bx
add dx,2 ; add two instead of one becuase word used
check:
mov bx,cx
add bx,dx
shr bx,1 ; shr 1 equivalent to dividing by two
cmp dx,cx
jbe start2
cmp dx,cx ; last check to verify if key found
jbe end
mov ax,0 ; ax = 0 if key not found
end:
mov ax,0x4c00
int 21h
```

Before run:

```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0005   SI 0000   CS 19F5   IP 0114     Stack +0 0000  Flags 7200
BX 0000   DI 0000   DS 19F5                    +2 20CD
CX 005A   BP 0000   ES 19F5   HS 19F5          +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5          +6 EA00   0  0  1  0  0  0  0  0

CMD >

0111 B80500        MOV    AX,0005
0114 BB0000        MOV    BX,0000
0117 B90C00        MOV    CX,000C
011A BA0000        MOV    DX,0000
011D 89CB          MOV    BX,CX
011F 01D3          ADD    BX,DX
0121 D1EB          SHR    BX,1
0123 39870301      CMP    [0103+BX],AX
0127 7C15          JL     013E

1          0  1  2  3  4  5  6  7
DS:0000   CD 20 FF 9F 00 EA F0 FE
DS:0008   AD DE 1B 05 C5 06 00 00
DS:0010   18 01 10 01 18 01 92 01
DS:0018   01 01 01 00 02 FF FF FF
DS:0020   FF FF FF FF FF FF FF FF
DS:0028   FF FF FF FF EB 19 C0 11
DS:0030   A2 01 14 00 18 00 F5 19
DS:0038   FF FF FF FF 00 00 00 00
DS:0040   05 00 00 00 00 00 00 00
DS:0048   00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA F0 FE   AD DE 1B 05 C5 06 00 00   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......ª.  .....
DS:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 C0 11               δ.└.
DS:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....J.     ....
DS:0040   05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........

1  Step   2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```

After run:

```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0001   SI 0000   CS 19F5   IP 0155     Stack +0 0000  Flags 7244
BX 0008   DI 0000   DS 19F5                    +2 20CD
CX 0008   BP 0000   ES 19F5   HS 19F5          +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0008   SP FFFE   SS 19F5   FS 19F5          +6 EA00   0  0  1  0  1  0  1  0

CMD >

0132 E92000        JMP    0155
0155 B8004C        MOV    AX,4C00
0158 CD21          INT    21
015A 0000          ADD    [BX+SI],AL
015C 0000          ADD    [BX+SI],AL
015E 0000          ADD    [BX+SI],AL
0160 0000          ADD    [BX+SI],AL
0162 0000          ADD    [BX+SI],AL
0164 0000          ADD    [BX+SI],AL

1          0  1  2  3  4  5  6  7
DS:0000   CD 20 FF 9F 00 EA FF FF
DS:0008   AD DE 1B 05 C5 06 00 00
DS:0010   18 01 10 01 18 01 92 01
DS:0018   01 01 01 00 02 FF FF FF
DS:0020   FF FF FF FF FF FF FF FF
DS:0028   FF FF FF FF EB 19 E6 11
DS:0030   A2 01 14 00 18 00 F5 19
DS:0038   FF FF FF FF 00 00 00 00
DS:0040   05 00 00 00 00 00 00 00
DS:0048   00 00 00 00 00 00 00 00

2          0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
DS:0000   CD 20 FF 9F 00 EA FF FF   AD DE 1B 05 C5 06 00 00   = ƒ.Ω    ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......ª.  .....
DS:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 E6 11               δ.µ.
DS:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....J.     ....
DS:0040   05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........

1  Step   2ProcStep 3Retrieve 4Help ON  5BRK Menu 6      7 up  8 dn  9 le 10 ri
```