Computer Organization and Architecture Lab

LAB ASSIGNMENT – 5

E.Likhith

AP22110010386

CSE-F

1. Write an assembly language program to perform division of 8-bit data.

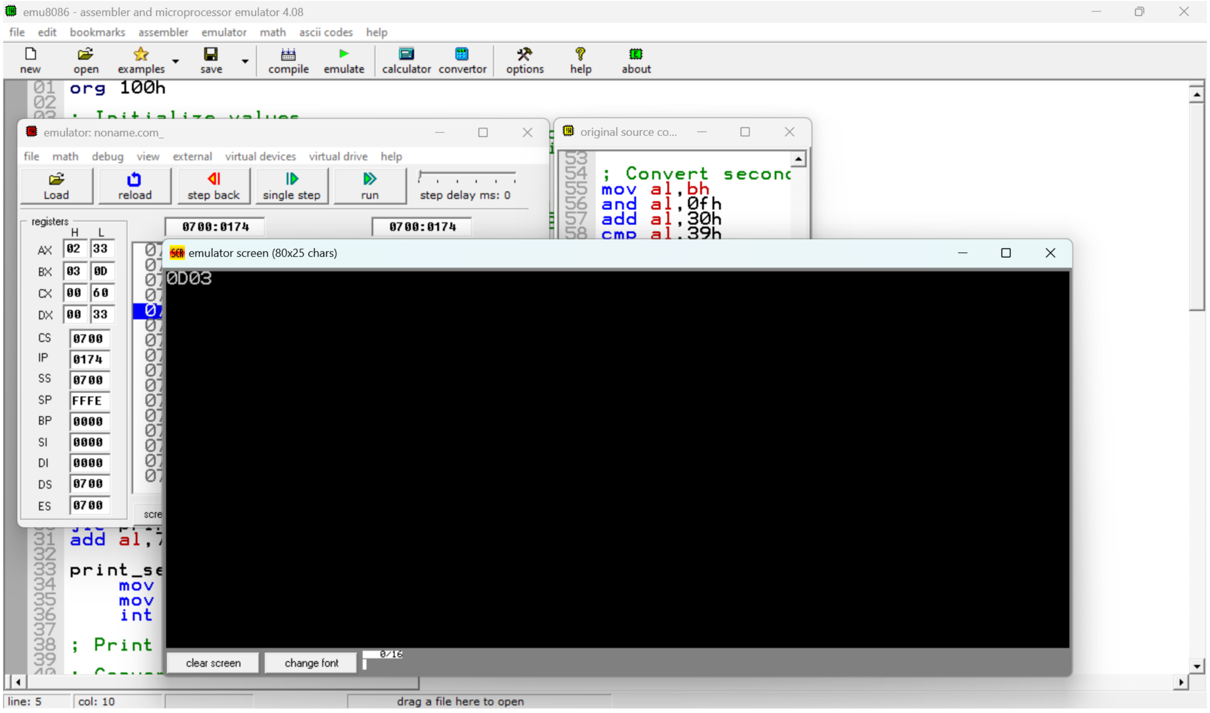
Code:

```
org 100h
mov al,96h
mov bl,10h

; Perform division (AL / BL)
idiv bl          ; AL = quotient, AH = remainder
mov bl,al        ; Store quotient in BL
mov bh,ah        ; Store remainder in BH

; Convert first digit (quotient) to ASCII
and al,0f0h      ; Mask higher nibble of AL
shr al,4         ; Shift right 4 bits to get the first hex digit
add al,30h       ; Convert to ASCII (0-9)
cmp al,39h       ; Check if it's a number or letter (0-9)
jle print_first_digit1
add al,7         ; Convert to ASCII (A-F)

print_first_digit1:
   mov dl,al     ; Move the result to DL (for printing)
   mov ah,02h    ; Print function
```

```
    int 21h       ; Interrupt to print the character

; Convert second digit (quotient) to ASCII
mov al,bl         ; Move the quotient back into AL
and al,0fh        ; Mask the lower nibble of AL
add al,30h        ; Convert to ASCII (0-9)
cmp al,39h        ; Check if it's a number or letter (0-9)
jle print_second_digit1
add al,7          ; Convert to ASCII (A-F)

print_second_digit1:
    mov dl,al     ; Move the result to DL (for printing)
    mov ah,02h    ; Print function
    int 21h       ; Interrupt to print the character
; Print remainder (remainder is in BH)

; Convert first digit (upper nibble of remainder) to ASCII
mov al,bh         ; Move remainder into AL
and al,0f0h       ; Mask the higher nibble
shr al,4          ; Shift right 4 bits to get the first hex digit
add al,30h        ; Convert to ASCII (0-9)
cmp al,39h        ; Check if it's a number or letter (0-9)
jle print_first_rem_digit
add al,7          ; Convert to ASCII (A-F)

print_first_rem_digit:
    mov dl,al     ; Move the result to DL (for printing)
    mov ah,02h    ; Print function
    int 21h       ; Interrupt to print the character
```

```
; Convert second digit (lower nibble of remainder) to ASCII
mov al,bh          ; Move remainder back into AL
and al,0fh         ; Mask the lower nibble
add al,30h         ; Convert to ASCII (0-9)
cmp al,39h         ; Check if it's a number or letter (0-9)
jle print_second_rem_digit
add al,7           ; Convert to ASCII (A-F)

print_second_rem_digit:
    mov dl,al      ; Move the result to DL (for printing)
    mov ah,02h     ; Print function
    int 21h        ; Interrupt to print the character
```

<u>OUTPUT:</u>



emulator screen (80x25 chars)

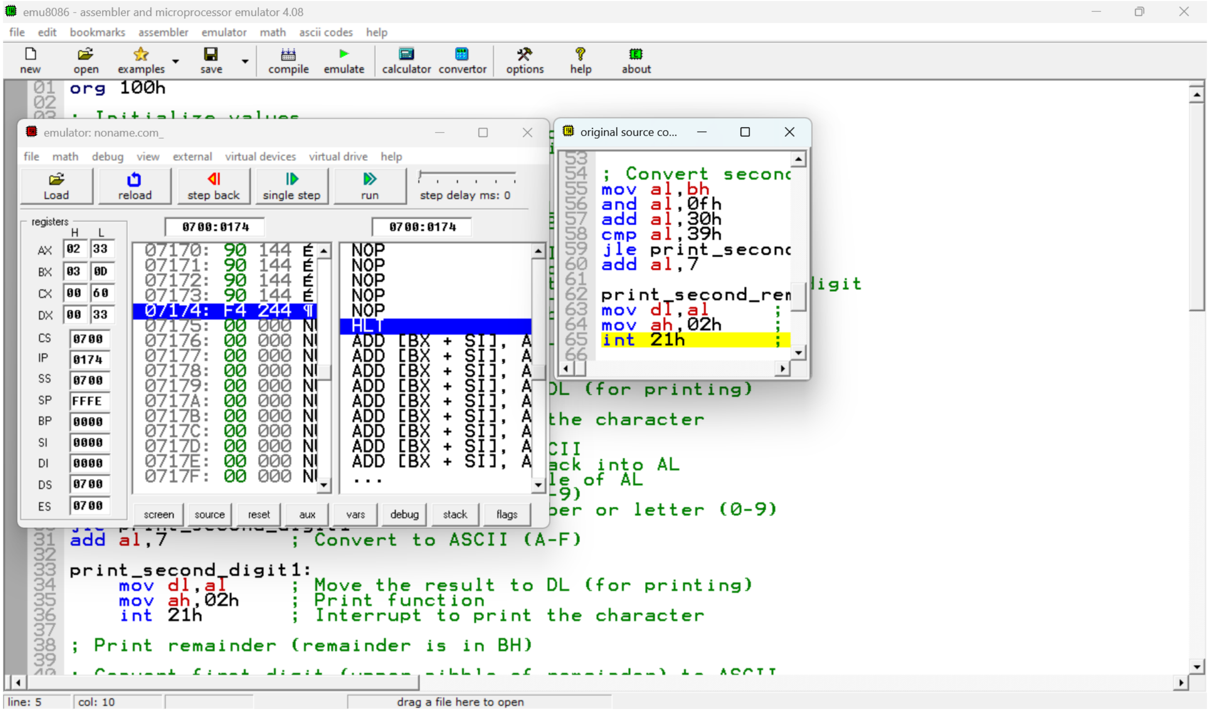
```
0D03
```

registers

	H	L
AX	02	33
BX	03	0D
CX	00	60
DX	00	33

CS	0700
IP	0174
SS	0700
SP	FFFE
BP	0000
SI	0000
DI	0000
DS	0700
ES	0700

emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
0700:0174
original source co...
clear screen change font
line: 5 col: 10 drag a file here to open



```
org 100h
```

emulator: noname.com_

```
AX 02 33
BX 03 0D
CX 00 60
DX 00 33
CS 0700
IP 0174
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
```

```
0700:0174
07170: 90 144 É
07171: 90 144 É
07172: 90 144 É
07173: 90 144 É
07174: F4 244 ¶
07175: 00 000 NU
07176: 00 000 NU
07177: 00 000 NU
07178: 00 000 NU
07179: 00 000 NU
0717A: 00 000 NU
0717B: 00 000 NU
0717C: 00 000 NU
0717D: 00 000 NU
0717E: 00 000 NU
0717F: 00 000 NU
```

```
0700:0174
NOP
NOP
NOP
NOP
NOP
HLT
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
...
```

original source co...

```
; Convert seconc
mov al,bh
and al,0fh
add al,30h
cmp al,39h
jle print_seconc
add al,7

print_second_rem
mov dl,al        ;
mov ah,02h       ;
int 21h          ;
```

```
add al,7                ; Convert to ASCII (A-F)

print_second_digit1:
    mov dl,al           ; Move the result to DL (for printing)
    mov ah,02h          ; Print function
    int 21h             ; Interrupt to print the character

; Print remainder (remainder is in BH)
```

emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers H L
screen source reset aux vars debug stack flags
line: 5 col: 10 drag a file here to open

2.Write a program in assembly language to perform division of 16-bit data.

Code:

org 100h

mov ax,2567h

mov bx,1400h

div bx

 mov bx,ax

 mov cx,dx

 mov ah,ch

 and ah,0f0h

 shr ah,4

 add ah,30h

 cmp ah,39h

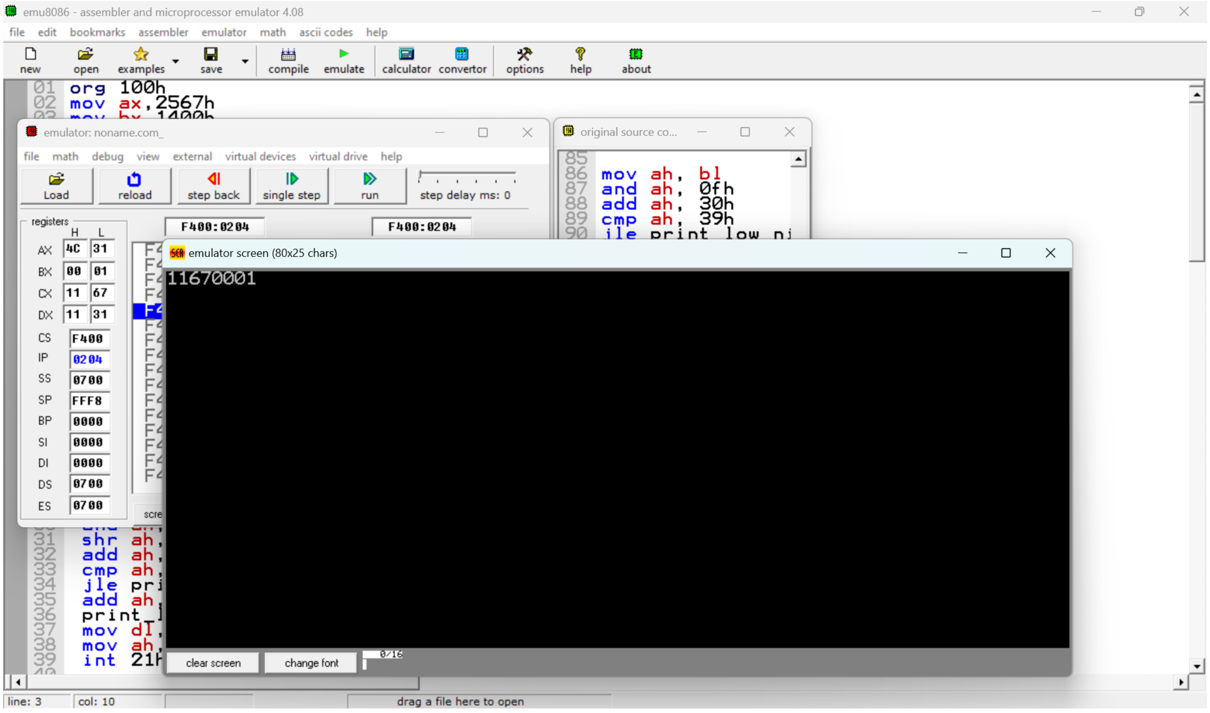
 jle print_high_nibble32

 add ah,7

```
print_high_nibble32:
  mov dl,ah
  mov ah,02h
  int 21h
mov ah,ch
and ah,0fh
add ah,30h
cmp ah,39h
jle print_low_nibble32
add ah,7
print_low_nibble32:
mov dl,ah
mov ah,02h
int 21h

mov ah,cl
and ah,0f0h
shr ah,4
add ah,30h
cmp ah,39h
jle print_low_nibble24
add ah,7
print_low_nibble24:
mov dl,ah
mov ah,02h
int 21h

mov ah,cl
and ah,0fh
```

```
 add ah,30h
 cmp ah,39h
 jle print_high_nibble24:
 add ah,7
 print_high_nibble24:
 mov dl,ah
 mov ah,02h
 int 21h


 mov ah, bh
 shr ah, 4
 add ah, 30h
 cmp ah, 39h
 jle print_high_nibble
 add ah, 7
print_high_nibble:
 mov dl, ah
 mov ah, 02h
 int 21h

 mov ah, bh
 and ah, 0fh
 add ah, 30h
 cmp ah, 39h
 jle print_low_nibble
 add ah, 7
print_low_nibble:
 mov dl, ah
 mov ah, 02h
 int 21h

 mov ah, bl
 shr ah, 4
 add ah, 30h
 cmp ah, 39h
 jle print_high_nibble2
 add ah, 7
print_high_nibble2:
 mov dl, ah
 mov ah, 02h
 int 21h

 mov ah, bl
 and ah, 0fh
 add ah, 30h
 cmp ah, 39h
 jle print_low_nibble2
 add ah, 7
print_low_nibble2:
 mov dl, ah
 mov ah, 02h
 int 21h
 mov ah,4ch
 int 21h
```

Output:


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
emulator screen (80x25 chars)
clear screen change font
line: 3 col: 10 drag a file here to open

```
org 100h
mov ax,2567h
mov bx,1400h
```

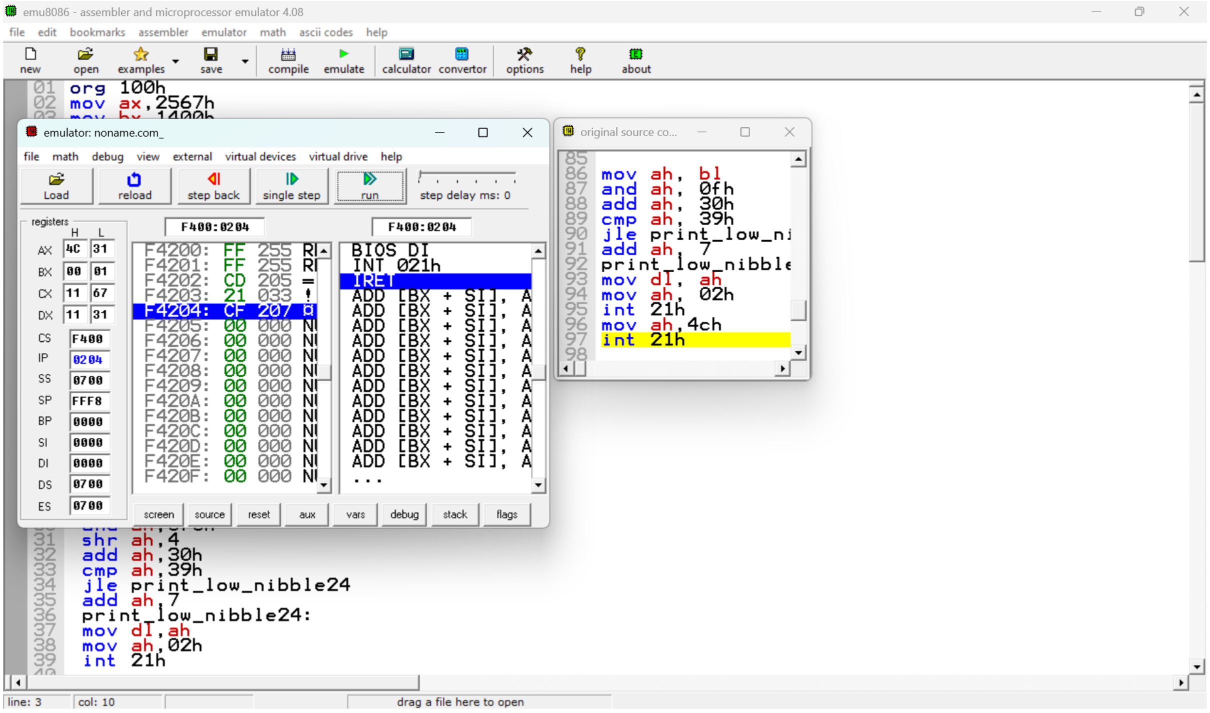
```
shr ah,
add ah,
cmp ah,
jle pri
add ah
print_l
mov dl,
mov ah,
int 21h
```

```
mov ah, bl
and ah, 0fh
add ah, 30h
cmp ah, 39h
jle print_low_ni
```

```
registers
     H   L
AX   4C  31
BX   00  01
CX   11  67
DX   11  31
CS   F400
IP   0204
SS   0700
SP   FFF8
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

F400:0204 F400:0204

```
11670001
```


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source co...
screen source reset aux vars debug stack flags
line: 3 col: 10 drag a file here to open

```
org 100h
mov ax,2567h
mov bx,1400h
```

```
registers
     H   L
AX   4C  31
BX   00  01
CX   11  67
DX   11  31
CS   F400
IP   0204
SS   0700
SP   FFF8
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

F400:0204 F400:0204

```
F4200: FF 255 RI
F4201: FF 255 RI
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ¤
F4205: 00 000 NI
F4206: 00 000 NI
F4207: 00 000 NI
F4208: 00 000 NI
F4209: 00 000 NI
F420A: 00 000 NI
F420B: 00 000 NI
F420C: 00 000 NI
F420D: 00 000 NI
F420E: 00 000 NI
F420F: 00 000 NI
```

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
...
```

```
mov ah, bl
and ah, 0fh
add ah, 30h
cmp ah, 39h
jle print_low_ni
add ah, 7
print_low_nibble
mov dl, ah
mov ah, 02h
int 21h
mov ah,4ch
int 21h
```

```
shr ah,4
add ah,30h
cmp ah,39h
jle print_low_nibble24
add ah,7
print_low_nibble24:
mov dl,ah
mov ah,02h
int 21h
```