Assignment

COEN 311

Computer Organization and Software

Assignment #3

Sunil Kublalsingh

ID: 40212432

Professor: Anjali Agarwal

“I certify that this submission is my original work and meets the Faculty’s Expectations of Originality”

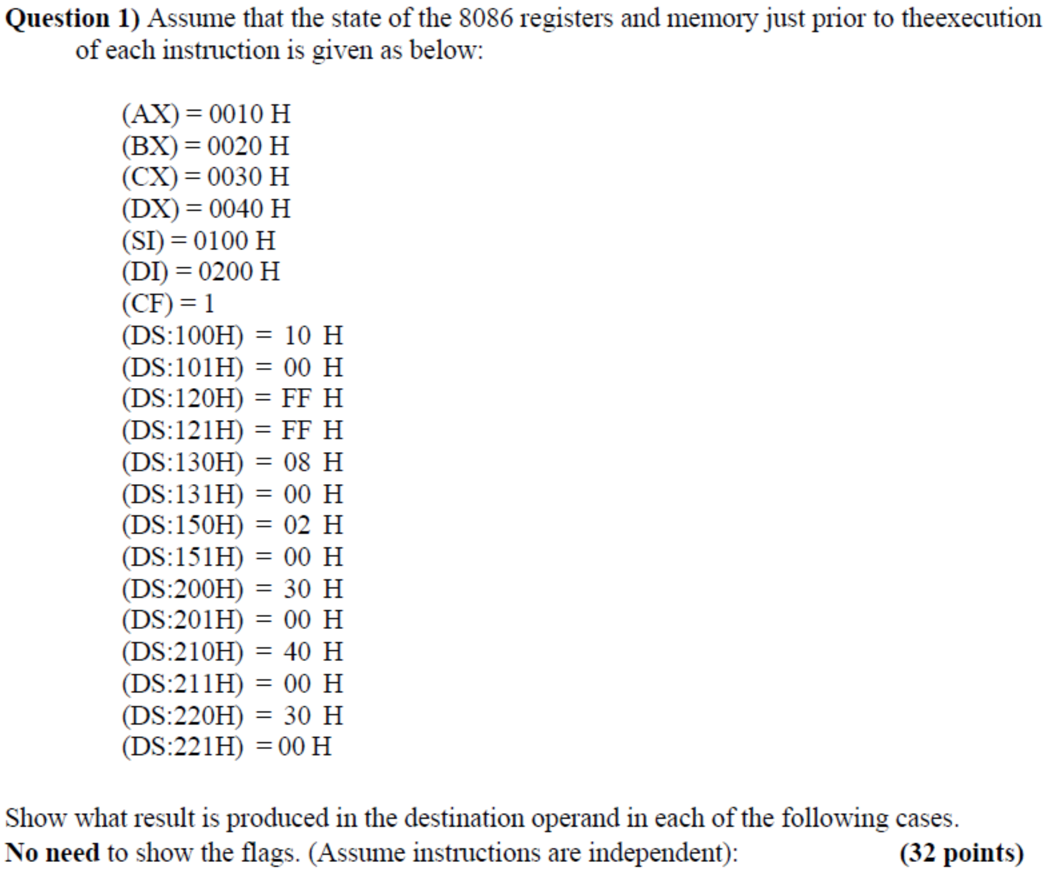
Sunil Kublalsingh
Student ID: 40212432

February 28, 2022

Sunil Kublalsingh
Student ID: 40212432
Course: COEN 311 W – Computer Organization and Software
Professor: Anjali Agarwal

Short answer questions

Question 1



Question 1) Assume that the state of the 8086 registers and memory just prior to theexecution of each instruction is given as below:

(AX) = 0010 H
(BX) = 0020 H
(CX) = 0030 H
(DX) = 0040 H
(SI) = 0100 H
(DI) = 0200 H
(CF) = 1
(DS:100H) = 10 H
(DS:101H) = 00 H
(DS:120H) = FF H
(DS:121H) = FF H
(DS:130H) = 08 H
(DS:131H) = 00 H
(DS:150H) = 02 H
(DS:151H) = 00 H
(DS:200H) = 30 H
(DS:201H) = 00 H
(DS:210H) = 40 H
(DS:211H) = 00 H
(DS:220H) = 30 H
(DS:221H) = 00 H

Show what result is produced in the destination operand in each of the following cases. **No need** to show the flags. (Assume instructions are independent): **(32 points)**

a) ADD AX, 00FF H
b) ADC SI, AX
c) INC BYTE PTR [0100H]
d) SUB DL, BL

e) SBB DL, [0200H]
f) MUL DX
g) IMUL BYTE PTR [SI]
h) IDIV BX

a)

$$\begin{array}{r} {\scriptstyle 1} \\ 00FF \\ +\ 0010 \\ \hline 010F \end{array}$$

$$\boxed{(AX) = 010F\ H}$$

b)

$$\begin{array}{r} 0010 \\ 0100 \\ +\ 0001 \\ \hline 0111 \end{array}$$

$$\boxed{(SI) = 0111\ H}$$

c)

$$[0100] = 10\ H$$

$$\begin{array}{r} 10 \\ +\ 1 \\ \hline 11\ H \end{array}$$

$$\boxed{ANS = 11H}$$

d)

$$\begin{array}{r} 40 \\ -20 \\ \hline 20 \end{array}$$

$$\begin{array}{r} FF \\ -\ 20 \\ \hline DF \\ +\ \ 1 \\ \hline E0 \end{array}$$

$$\boxed{(DL) = 20\ H}$$

$$\begin{array}{r} {}^{1}\ \ \ \ \\ 40 \\ +E0 \\ \hline 20 \end{array}$$

$4+E = 18$

$= 1 \times 16 + 2$

e)

$$\begin{array}{r} {}^{3\ 16} \\ \cancel{4}0 \\ 30 \\ -01 \\ \hline 0F \end{array}$$

$$\boxed{(DL) = 0F\ H}$$

f)

$$\begin{array}{r}
0040 \\
\times\ 0010 \\
\hline
0000 \\
0040\text{x} \\
0000\text{xx} \\
+\ 0000\text{xxx} \\
\hline
00000400
\end{array}$$

$$\boxed{\begin{array}{l}
(EDX) = 0000\text{ H} \\
(EAX) = 0400\text{ H}
\end{array}}$$

g)

AL = 10H

$$\begin{array}{r}
10 \\
\times\ 10 \\
\hline
00 \\
+\ 10\text{x} \\
\hline
100
\end{array}$$

$$\boxed{\text{ANS} = 0100\text{ H}}$$

h)

$$BX = 0020\,H = 0000\ 0000\ 0010\ 0000 = 2^5 = 32$$

$$AX = 0010\,H = 0000\ 0000\ 0001\ 0000 = 2^4 = 16$$

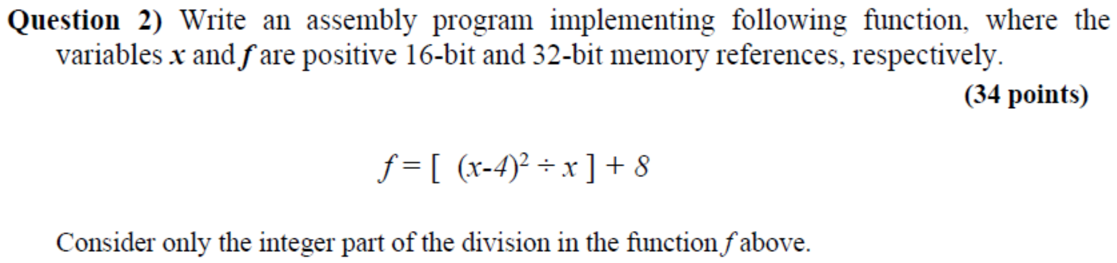
$$\frac{AX}{BX} = \frac{16}{32} = \frac{1}{2} = \begin{cases} Q = (0)_{10} \\ R = (1)_{10} \end{cases} \Rightarrow \begin{cases} Q = 00\,H \\ R = 01\,H \end{cases}$$

16 bits divided = 8 bit quotient + 8 bit remainder

$$\boxed{\begin{aligned} AH &= 00\,H \\ AL &= 01\,H \end{aligned}}$$

Sunil Kublalsingh
Student ID: 40212432
Course: COEN 311 W – Computer Organization and Software
Professor: Anjali Agarwal

Question 2



Question 2) Write an assembly program implementing following function, where the variables ***x*** and ***f*** are positive 16-bit and 32-bit memory references, respectively.

(34 points)

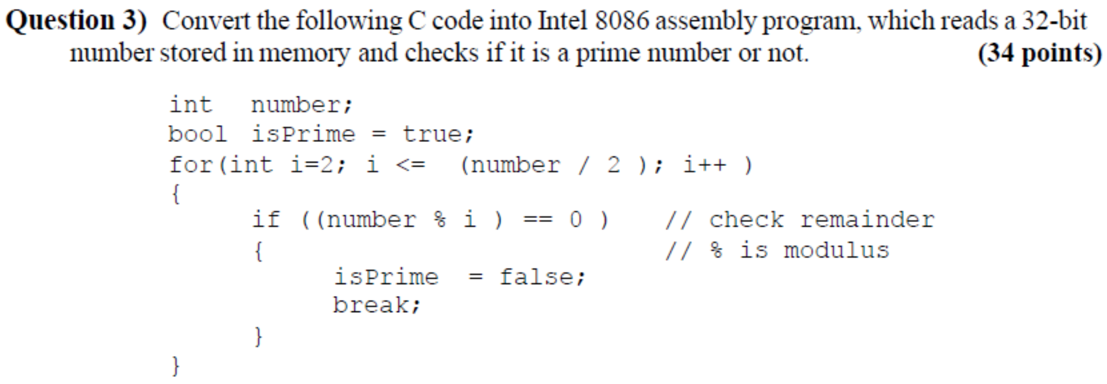
$$f = [\ (x\text{-}4)^2 \div x\ ] + 8$$

Consider only the integer part of the division in the function *f* above.

```
MOV AX, [X]      ;Place the value of x in register AX
SUB AX, 4        ; Subtract 4 from AX
IMUL AX (should go into DX: AX)          ;Multiply AX by AX

IDIV [X]         ; Divide AX by X
ADD AX, 8        ;Add 8 to the answer
MOV f, AX        ; Store result in f
```

Question 3



Question 3) Convert the following C code into Intel 8086 assembly program, which reads a 32-bit number stored in memory and checks if it is a prime number or not. **(34 points)**

```
int   number;
bool  isPrime = true;
for(int i=2; i <=  (number / 2 ); i++ )
{
      if ((number % i ) == 0 )    // check remainder
      {                           // % is modulus
            isPrime  = false;
            break;
      }
}
```

```
section .data
        isPrime db $FF
        number dd 8
section .bss

section .text
        global _start

_start:
        MOV EAX, [number]     ; Place the number in register EAX
        IDIV 2 ; Divide EAX by 2, places back into EAX
        MOV EBX, EAX ; Place the endpoint in register EBX
        MOV EAX, [number]      ;Place the original number back into EAX
        MOV ECX, 2      ; Use register CX as counter i with starting point i=2
MyLoop:
        CMP EBX, ECX     ; Check if i has reached the end point
        je endLoop    ; exit the loop if i==number/2
        IDIV ECX   ; Divide EAX by ECX, which is number / i, remainder should go into EDX
        CMP EDX, 0   ; Check if the remainder is = 0
        MOV EAX, [number]       ;Place the number back in EAX
        je then    ; Jump to the stuff in the if condition
        INC ECX    ; Increment the counter i
        jmp MyLoop      ; jump back to the start of the loop
then:
        MOV [isPrime], $00    ; Set isPrime to false
endLoop:

_exit:
     mov eax,1 ; The system call for exit (sys_exit)
     mov ebx,0 ; Exit with return code of 0 (no error)
     int 80h
```