

Hamza Tariq  
i21-0707  
Y

COAL Fall, 2022

NUCES, Islamabad

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING  
SCIENCES ISLAMABAD

COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGE, Fall 2022

**ASSIGNMENT 03**

Due Date: 11:59 AM 7<sup>th</sup> November 2022

Hard Copy submission : 8<sup>th</sup> November 2022

1. Consider last four digits of your ID and ADD 5555H to all digits and create your new COAL ID in HEX as shown in example below

1	2	8	9	Example ID
+	5	5	5	Added Value
6	7	D	E	Example COAL ID
0110	0111	1101	1110	

**CREATE NEW COAL ID**

0	7	0	7	Your ID
5	5	5	5	Added Value
5	C	,	5	COAL NEW ID
0101	1100	0101	1100	Binary
<b>MS-BCD</b>				<b>LS-BCD</b>

2. Consider the following code and fill given registers and memory accordingly?

**NOTE:** **Multiplicand** is last digit of your NEW COAL ID to it whereas **Multiplier** is second last digital of your New COAL ID?

```

7   multiplicand db      ;last digit of your NEW COAL ID LS BCD
8   multiplier  db      ;Second last digit of your NEW COAL ID LS BCD
9   result db 0
10 .code
11   mov ax,@data
12   mov ds,ax
13   mov ax,0
14   mov cx,4
15   mov al,multiplicand
16   mov bl,multiplier
17 checkbit:
18   shr bl,1
19   jnc skip
20   add result,al
21
22 skip:
23   shl al,1
24   loop checkbit

```

00 00 1100      000 0,0 010,1

      al                  bl

CF	multiplicand	Multiplier	CF	result
----	--------------	------------	----	--------

0	00 01 10 00	00 00 00 10	1	00 00 11 00
0	00 11 00 00	00 00 00 01	0	00 00 11 00
0	01 10 00 00	00 00 00 00	1	00 11 11 00
0	11 00 00 00	00 00 00 00	0	00 11 11 00

3      C

$$5 \times 12 = (60)_{10} = (3C)_{16}$$

4. Change above given code for following data declaration?

```
4 .data  
5 | multiplicand DQ 012345678H  
6 | multiplier DB 05AH  
7 | result DQ 0
```

.Code

mov ax, @ data

mov ds,ax

mov ax,b

mov dl, Multiplier

mov cx, size of Multiplicand

L1:

Sir Al, I

Inc noadd

mov ax , word ptr [multiplicand]

add word ptr [result], ax

mov bx, word ptr [multiplicand + 2]

adc word ptr[result + 2], bx

mov bx, word ptr [multiplicand + 4]

adc word pbr[result + 4], bx

MOV bx, word PTR [multipliand + 6]

adc word ptr [result+6], bx

noadd:

SQL word pbs [multiplicand]

rcd word ptr [multiplicand + 2], 1  
rcd word ptr [multiplicand + 4], 1  
rcd word ptr [multiplicand + 6], 1

Loop L1.

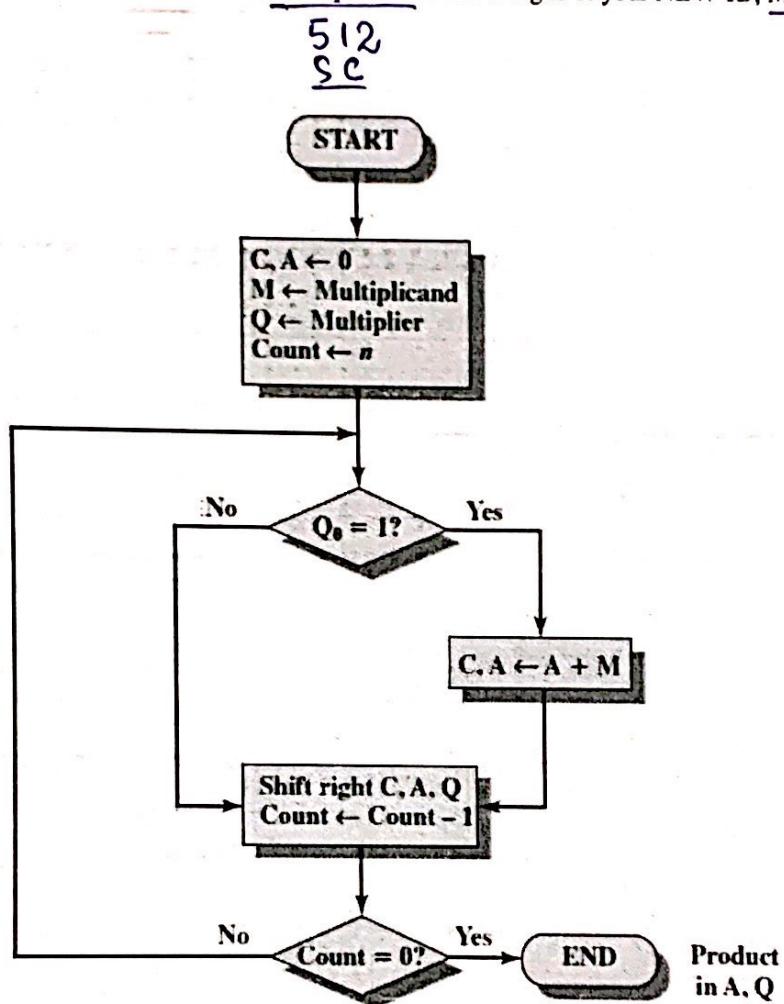
mov ah, 4ch

int arr

end.

5. Perform unsigned binary multiplication using following given flow chart?

NOTE: Your computer width is 8-bit, Multiplicand is last 2 digits of your NEW ID, Multiplier is **0A5H**



COAL Fall, 2022

C	A(ACCUMULATOR)	Q(MULTIPLIER)	M(MULTICAND)	NUCES, Islamabad
0	0101 1100	1010 0101	0101 1100	Add 8
0	0010 1110	0101 0010		Shift
C	A(ACCUMULATOR)	Q(MULTIPLIER)	M(MULTICAND)	
0	0001 0111	0010 1001		Add 7
0	0111 0011	0010 1001		Shift
0	0011 1001	1001 0100		Add 6
C	A(ACCUMULATOR)	Q(MULTIPLIER)	M(MULTICAND)	
0	0001 1100	1100 1010		Add 5
0	0001 1110	0110 0101		Shift
C	A(ACCUMULATOR)	Q(MULTIPLIER)	M(MULTICAND)	
0	0000 1110	0110 0101		Add 4
0	0110 1010	0110 0101		Shift
0	0011 0101	0011 0110		Add 3
C	A(ACCUMULATOR)	Q(MULTIPLIER)	M(MULTICAND)	
0	0001 1010	1001 1001		Add 2
0	0001 1010	0100 1100		Shift
C	A(ACCUMULATOR)	Q(MULTIPLIER)	M(MULTICAND)	
0	0111 0110	1001 1001		Add 1
0	0011 1011	0100 1100		Shift

3      B      4      C

$$(3B4C)_{16}$$

$$\begin{aligned} A5 &= 165 \\ SC &= 92 \end{aligned}$$

$$\begin{array}{r} 165 \\ 92 \\ \hline (15180)_{10} \end{array}$$

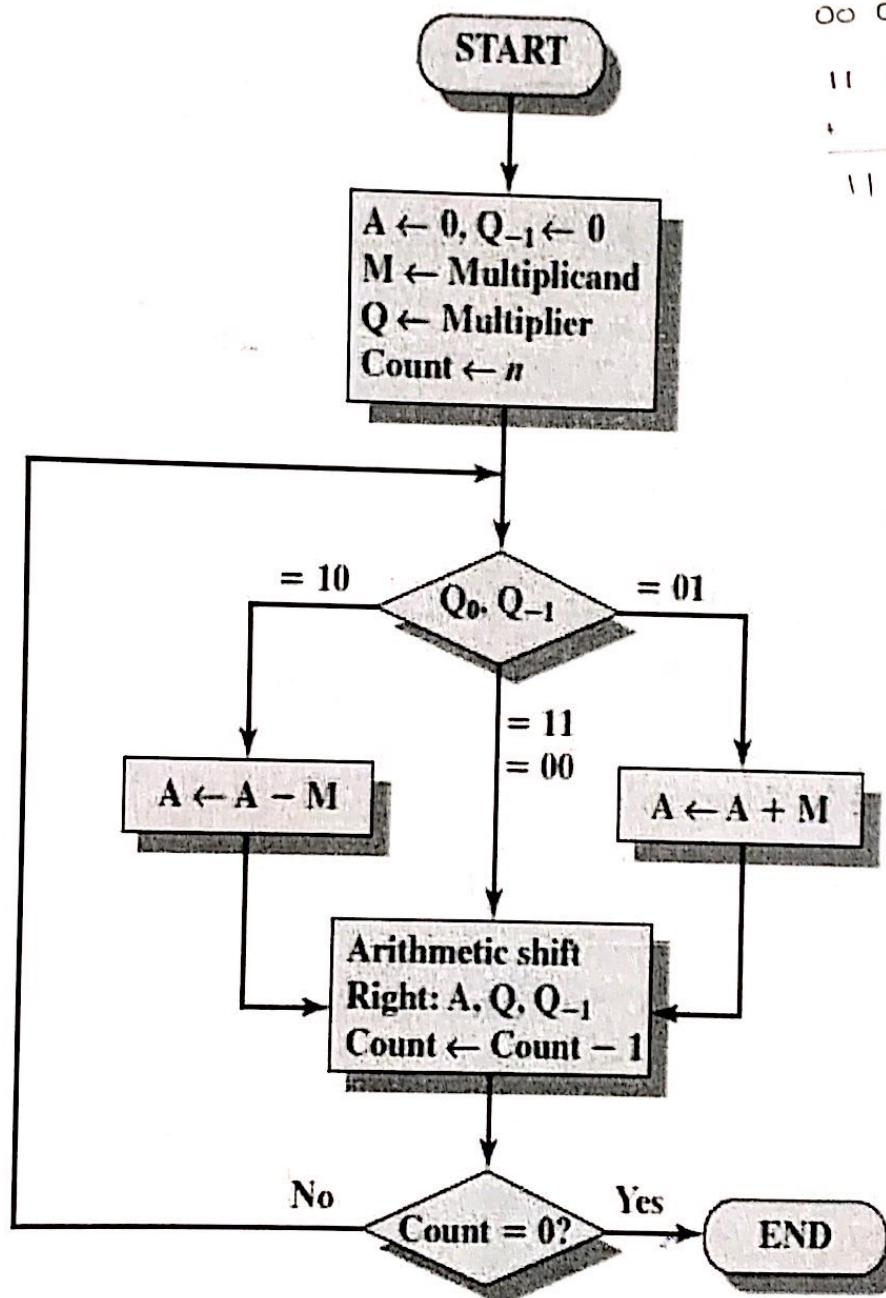
$$(15180)_{10} = (3B4C)_{16}.$$

6. Perform Multiplication using Booth's algorithm?

NOTE: Your computer width is 10-bit, **Multiplicand** is 2's complement of last 2 digits of your  
**NEW ID** (Means it's a signed and negative number), **Multiplier** is **0A5H**

51C

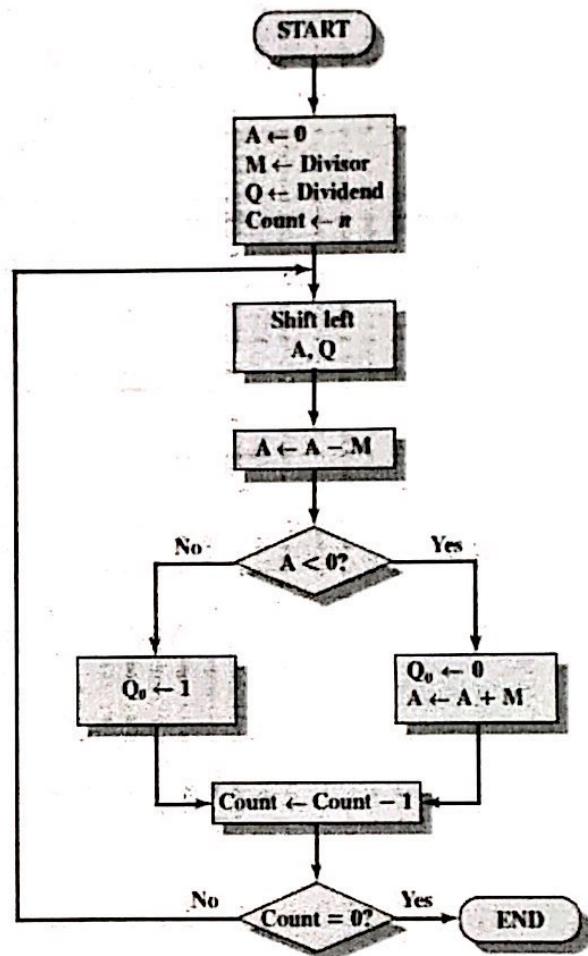
$$\begin{array}{r}
 00 \ 0101 \ 1100 \rightarrow M' \\
 11 \ 1010 \ 0011 \\
 \hline
 1110100100
 \end{array}$$



A (ACCUMULATOR)	Q (MULTIPLIER)	Q-1	M (MULTIPLICAND)	COUT
00 0101 1100	00 1010 0101	0	11 1010 0100	Add 10
00 0010 1110	00 0101 0010	1	/	Shift
11 1101 0010	00 0101 0010	1	/	Add 9
11 1110 1001	00 0010 1001	0	/	Shift
00 0100 0101	00 0010 1001	0	/	Add 8
00 0010 0010	10 0001 0100	1	/	Shift
11 1100 0110	10 0001 0100	1	/	Add 7
11 1110 0011	01 0000 1010	0	/	Shift
/	/	0	/	Add 6
11 1111 0001	10 1000 0101	0	/	Shift
00 0100 1101	10 1000 0101	0	/	Add 5
00 0010 0110	01 0100 0010	1	/	Shift
11 1100 1010	01 0100 0010	1	/	Add 4
11 1110 0101	01 1010 0001	0	/	Shift
00 0100 0001	01 1010 0001	0	/	Add 3
00 0010 0000	10 1101 0000	1	/	Shift
11 1100 0100	10 1101 0000	1	/	Add 2
11 1110 0010	01 0110 1000	0	/	Shift
/	/	0	/	Add 1
11 1111 0001	00 1011 0100	0	11 1010 0100	Shift
↓		$\begin{array}{r} 165 \\ +92 \\ \hline -15,180 \end{array}$		
Answer:		$\begin{array}{r} 165 \\ +92 \\ \hline -15,180 \end{array}$		
<u>-15,180</u>				

## 7. Perform Unsigned binary division?

NOTE: Your computer width is 8-bit, where dividend is last 2 digits of your NEW ID, Divisor is 003H



Date:

 $-M = 1111 \ 1101$ 

0000 0000	0101 1100	0	0000 0011	
S A	Q	Q <sub>0</sub>	M	n
0000 0000	1011 1000	0	0000 0011	8
-M 1111 1101		1	1111 1101 0000 0000	
1111 1101	1011 1000	0	1111 1101 0000 0000	
M 0000 0011			1111 1101 0000 0000	
0000 0000			1111 1101 0000 0000	
0000 0001	0111 0000	1	1111 1101 0000 0000	7
-M 1111 1101			1111 1101 0000 0000	
1111 1110	0111 0000	0	1111 1101 0000 0000	
M 0000 0011			1111 1101 0000 0000	
0000 0001			1111 1101 0000 0000	
0000 0010	1110 0000		1111 1101 0000 0000	6
-M 1111 1101			1111 1101 0000 0000	
1111 1111	1110 0000	0	1111 1101 0000 0000	
M 0000 0011			1111 1101 0000 0000	
0000 0010			1111 1101 0000 0000	
0000 0101	1100 0000		1111 1101 0000 0000	5
-M 1111 1101			1111 1101 0000 0000	
0000 0010	1100 0001	1	1111 1101 0000 0000	
0000 0101	1000 0010		1111 1101 0000 0000	4
-M 1111 1101			1111 1101 0000 0000	
0000 0010	1000 0011	1	1111 1101 0000 0000	
0000 0101	0000 0110		1111 1101 0000 0000	3
-M 1111 1101			1111 1101 0000 0000	
0000 0010	0000 0111	1	1111 1101 0000 0000	

Date:-

0000 0100	0000 1110	0	0000 1110	0000 0000
-M 1111 1101		0	1111 1101	1111 1111 M-
0000 0001	0000 1111	1		1011 1111
0000 0010	0001 1110	0	1011 1101	1011 1111 M-
-M 1111 1101		0	1100 1101	1100 1111 M-
1111 1111				0000 0000
M 0000 0011	0001 1110	0	0000 1110	1000 0000
0000 0010		0	0000 1110	1011 1111 M-
0000 0001	0001 1110	0	0000 1110	1111 1111 M-
0000 0000		0	0000 1110	1111 1111 M-
0000 0010	0001 1110	0	Answer	1100 0000 M
		↓ ≈30		

8. Apply following piece of code and update register accordingly?

3 **mov al, 0x5f**  
4 **shr al, 1**

0	0	1	0	1	1	1	1
---	---	---	---	---	---	---	---

CF  
1

6 **mov al, 0xf5**  
7 **sar al, 1**

1	1	1	1	1	0	1	0
---	---	---	---	---	---	---	---

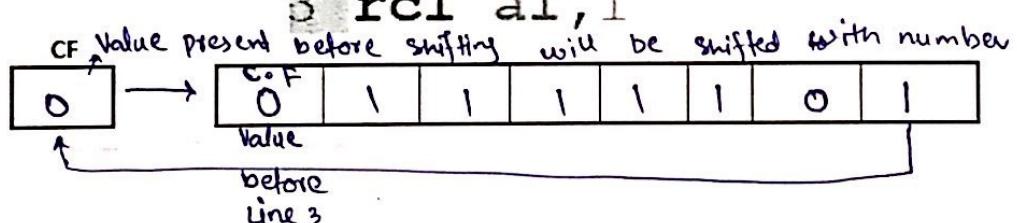
CF  
1

9 **mov al, 0x5f** (0101 1111)  
10 **rol al, 1**

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

CF  
0

2 **mov al, 0xf5** (1111 1010)  
3 **rcl al, 1**



Date: \_\_\_\_\_

9. Save Your New ID in BX register and calculate Number of 1's

- mode small
- stack 100h
- data
- code

mov bx, 0SCSCH

mov cx, 16

mov ax, 0

L1:

shr bx, 16

Jc increment

Jmp exit

increment:

add ax, 1

exit:

Loop L1

mov ax, 04ch

int 21h

end

Date: \_\_\_\_\_

10

.data

Array db 1, 3, 7, 8, 9, 6, 5, 11, 13, 15  
odd db 0.

.code

mov ax, @data

mov ds, ax

mov ax, 0.

mov si, offset Array

mov cx, \$C - Array

mov al, 0

L1:

mov bl, [si]

shr bl, 1

inc Notodd

add al, 1

Notodd :

inc si

Loop L1:

al shows total number of odd numbers  
in Array.

Date: \_\_\_\_\_

11.data

```

Id : dw SCSCH
Id1 db (0)
Id2 db (0)
Id3 db (0)
Id4 db (0)

```

.Code

```

mov al, @data
mov DS, AX
mov AX, 0
mov al, byte ptr [si]
mov bl, 0
mov cx, 4
clc

```

L1:

```

shl al, 1
rcl bl, 1

```

Loop L1

```
mov Id3, bl
```

```
mov bl, 0
```

```
clc
```

```
mov cx, 4
```

Date:

mov al, byte ptr [si]

L4:

L2:

shr al, 1

shr al, 1

rcr bl, 1

rcr bl, 1

Loop L2

(a)

mov Id1, bl

shl bl, 4

mov Id4, bl

mov ah, 4ch

mov al, byte ptr [si+1]

end

mov bl, 0

mov cx, 4

clc

Value's as a byte  
in Id1, Id2, Id3 & Id4  
according to order.

L3:

shl al, 1

rcr bl, 1

Loop L3.

mov Id2, bl

mov bl, 0

mov cx, 4

mov al, byte ptr [si+1]

clc