

U19CS076 MIT ASSIGNMENT-4

KRITHIKHA BALAMURUGAN

Q1. Write a program to check the 4th bit of 8-numbers stored from location 2000H.

The screenshot shows an 8085 assembler interface. On the left, the Registers window shows A=01, BC=00, DE=00, HL=00, PSW=00, PC=42, SP=FF, and Int-Reg=00. The Flag window shows S=0, Z=0, AC=1, P=0, and C=0. The main window contains the following assembly code:

```
1 ;<Program title>
2
3
4 ;q1
5 ; Write a program to check the 4th bit of 8-numbers stored from location 2000H
6 lda 3000h
7 ani 08h ;to get 4th bit
8 rrc
9 rrc
10 rrc
11 sta 3001h; stores 4th bit which is in place of 1st bit
12
13 hlt
```

On the right, the Memory window shows the following data:

Address (Hex)	Address	Data
3000	12288	13
3001	12289	1
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

The I/O Ports window shows 0 and 00. The Line No Assembler Message window shows "0 Program assembled successfully".

Q2. Write a program to swap lower 4 bit nibble with upper 4 bit nibble of 8 bit data at memory

location 2100H and place a result to location 2101H.

The screenshot shows an 8085 assembler interface. On the left, the Registers window shows A=39, BC=93, DE=00, HL=00, PSW=00, PC=42, SP=FF, and Int-Reg=00. The Flag window shows S=0, Z=0, AC=0, P=1, and C=0. The main window contains the following assembly code:

```
1 ;<Program title>
2
3 ;q2
4 ;Write a program to swap lower 4 bit nibble with upper 4 bit nibble of 8 bit data
5 ;at memory location 2100H and place a result to location 2101H.
6 lda 2100h ;data is 147 = 93H
7 mov b,a
8 ani 0f0h; mask lower 4 bits, only upper 4 bits present =90H
9 rlc
10 rlc
11 rlc
12 rlc; finished shifting upper 4 bits in place of lower 4 bits =09H
13 mov c,a
14 mov a,b
15 ani 0fh; mask upper 4 bits, only lower 4 bits present=03H
16 rlc
17 rlc
18 rlc
19 rlc; finised shifting lower 4 bits in place of upper 4 bits=30H
20 add c; combining upper and lower nibbles=39H
21 sta 2101h
22
23 hlt
24
```

On the right, the Memory window shows the following data:

Address (Hex)	Address	Data
2100	8448	147
2101	8449	57
2102	8450	0
2103	8451	0
2104	8452	0
2105	8453	0
2106	8454	0
2107	8455	0
2108	8456	0
2109	8457	0
210A	8458	0
210B	8459	0

The I/O Ports window shows 0 and 00. The Line No Assembler Message window shows "0 Program assembled successfully".

The screenshot shows a "Decimal - Hex Conversion" tool. The Decimal input field contains "147" and the Hex output field contains "93". The "To Hex" button is highlighted.

Q3. Write a Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair

The screenshot shows an 8085 assembly simulator. The assembly code is as follows:

```

1 ;q4
2 lxi h,1230h; hl now holds 16-bit data 2034h
3 ora a; set carry to 0
4 dad h; adds hl to hl and stores to hl =hl+hl=2*hl
5 ; effectively doubles hl i.e, shift 16-bit data by 1 bit left
6 shld 2000h; result stored at memory location 2000h
7 hlt

```

The registers window shows the following values:

Register	Value
A	00
BC	00 00
DE	00 00
HL	24 60
PSW	00 00
PC	42 09
SP	FF FF
Int-Reg	00

The flag window shows: S 0, Z 1, AC 0, P 1, C 0.

The memory window shows the following data:

Address (Hex)	Address	Data
2000	8192	96
2001	8193	36
2002	8194	0
2003	8195	0
2004	8196	0
2005	8197	0
2006	8198	0
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0

The I/O Ports window shows: 0 - + 00.

The assembler message window shows: 0 Program assembled successfully.

Q4. Write a Program to calculate the factorial of a number between 0 to 8.

The screenshot shows an 8085 assembly simulator. The assembly code is as follows:

```

1 ;<Program title>
2 ;q4
3 ;Write a Program to calculate the factorial of a number between 0 to 8.
4
5 lda 3000h
6 cpi 02h; compare to check if no is 0/1
7 jc down; factorial of 0 and 1 is 1
8 ;and we jump to directly output that result
9
10 mov e,a
11 mvi d,00h; to create 16-bit data in DE register pair
12 dcr a; decrease by 1
13 call facto; subroutine for calculating factorial
14 shld 3001h; after return of value from subroutine,
15 ; output to memory location 3251h
16 jmp end
17
18 down: lxi h,0001h
19 shld 3001h;data of e(lower bits) goes to 3251h and d(higher bits) goes to 3252h
20
21 end: hlt
22
23 facto: lxi h,0000h; initial hl pair with 0 for further addition
24 mov c,a; move decremented value to c
25
26 up: dad d ;data of hl pair+data of de pair=data of hl pair
27 dcr c
28 jnz up ;if c is not 0
29 xchg ;exchange data of HL and DE pair
30 dcr a
31 cnz facto ;till a not 0
32 ret
33
34

```

The registers window shows the following values:

Register	Value
A	00
BC	00 00
DE	00 18
HL	00 18
PSW	00 00
PC	42 1C
SP	FF FF
Int-Reg	00

The flag window shows: S 0, Z 1, AC 0, P 1, C 0.

The memory window shows the following data:

Address (Hex)	Address	Data
3000	12288	4
3001	12289	24
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

The I/O Ports window shows: 0 - + 00.

The Memory window shows: 0 - + 00.

The assembler message window shows: 0 Program assembled successfully.

Data
Stack
KeyPad
Memory
I/O Ports

Start
3000h
OK

Address (Hex)	Address	Data
3000	12288	8
3001	12289	128
3002	12290	157
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

Line No
Assembler Message

0
Program assembled successfully

Q5. Write a program to Split 8 bit HEX data into two nibbles and store it in memory.

Registers
Flag

A 03 S 0
BC 23 00 Z 0
DE 00 00
HL 00 00 AC 1
PSW 00 00
PC 42 14 P 1
SP FF FF
Int-Reg 00 C 0

Decimal - Hex Conversion
Decimal Hex
0 0
To Hex To Dec

I/O Ports
0 - + 00
Update Port Value

Load me at

```

1 ;<Program title>
2 ;q5
3 ;Write a program to Split 8 bit HEX data into two nibbles and store it in memory.
4
5 lda 2000h ;35=0010 0011
6 mov b,a
7 ani 0f0h; mask lower nibble,only higher nibble present
8 rrc
9 rrc
10 rrc
11 rrc ;bring it in place of lower nibble-0010
12 sta 2002h
13 mov a,b
14 ani 0fh; mask upper nibble,only lower nibble present-0011
15 sta 2001h
16 hlt
17
18
19
20
21
22
23
24
25
26
27
28

```

Data
Stack
KeyPad
Memory
I/O Ports

Start
2000h
OK

Address (Hex)	Address	Data
2000	8192	35
2001	8193	3
2002	8194	2
2003	8195	0
2004	8196	0
2005	8197	0
2006	8198	0
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0

Line No
Assembler Message

0
Program assembled successfully