



Name : Salma Rani

Sap ID : 54194

Instructor:Mam Ayesha

Lab submission:9

Class Work:

The screenshot displays a 68000 assembly simulator interface. It includes a menu bar (File, Edit, Modify, Execute, Help), a 'Data' dropdown set to 'Hex', and a 'Registers' table. The 'Registers' table lists AC, AR, DR, E, I, IR, PC, and S with their respective widths and data values. The central pane shows assembly code for 'AND.a X' with instructions like START, INP, STA NUM, INP, AND NUM, OUT, HLT, and a data definition for NUM. The right pane shows a memory dump for 'MAIN' with addresses from 000 to 010 and their corresponding hex data values. At the bottom, a status window shows the execution progress, including prompts for input and the final halt message.

Name	Width	Data
AC	16	0000
AR	12	001
DR	16	0000
E	1	0
I	1	0
IR	16	E001
PC	12	006
S	1	1

```
1 START:
2 INP
3 STA NUM
4 INP
5 AND NUM
6 OUT
7 HLT
8
9 NUM: .data 1 0
10
11
12
```

Addr	Data
000	F800
001	6006
002	F800
003	0006
004	F400
005	E001
006	0000
007	0000
008	0000
009	0000
00A	0000
00B	0000
00C	0000
00D	0000
00E	0000
00F	0000
010	0000

EXECUTING...
Enter Inputs, the first of which must be an Integer: 0
Enter Inputs, the first of which must be an Integer: 0
Output: 0
EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]

Task 1:OR Operation

DataHex

Registers

Name	Width	Data
AC	16	0001
AR	12	E00
DR	16	0000
E	1	0
I	1	0
IR	16	E001
PC	12	006
S	1	1

ORa X

```
1 START:
2 INP
3 STA NUM
4 INP
5 OR NUM
6 OUT
7 HLT
8
9 NUM: .data 1 0
```

AddrHexDataHex

MAIN

Addr	Data
000	F800
001	6006
002	F800
003	3006
004	F400
005	E001
006	0000
007	0000
008	0000
009	0000
00A	0000
00B	0000
00C	0000
00D	0000
00E	0000
00F	0000
010	0000

EXECUTING...

Enter Inputs, the first of which must be an Integer: 0

Enter Inputs, the first of which must be an Integer: 1

Output: 1

EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]

Task 2:NAND Operation

Registers

Name	Width	Data
AC	16	FFFE
AR	12	E00
DR	16	0000
E	1	0
I	1	0
IR	16	E001
PC	12	006
S	1	1

START:

```
1 START:
2 INP
3 STA NUM
4 INP
5 NAND NUM
6 OUT
7 HLT
8
9 NUM: .data 1 0
10
```

MAIN

Addr	Data
000	F800
001	6006
002	F800
003	4006
004	F400
005	E001
006	0000
007	0000
008	0000
009	0000
00A	0000
00B	0000
00C	0000
00D	0000
00E	0000
00F	0000
010	0000

EXECUTING...

Enter Inputs, the first of which must be an Integer: 1

Enter Inputs, the first of which must be an Integer: 1

Output: -2

EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]

Task 3: NOR Operation

DataHex

Registers

Name	Width	Data
AC	16	FFFF
AR	12	E00
DR	16	0000
E	1	0
I	1	0
IR	16	E001
PC	12	006
S	1	1

NOR.a X

```
1 START:
2   INP
3   STA NUM
4   INP
5   NOR NUM
6   OUT
7   HLT
8
9 NUM: .data 1 0
10
```

AddrHexDataHex

MAIN

Addr	Data
000	F800
001	6006
002	F800
003	5006
004	F400
005	E001
006	0000
007	0000
008	0000
009	0000
00A	0000
00B	0000
00C	0000
00D	0000
00E	0000
00F	0000
010	0000

EXECUTING...

Enter Inputs, the first of which must be an Integer: 0

Enter Inputs, the first of which must be an Integer: 0

Output: -1

EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]

Task 4: XOR Operation

DataHex

Registers

Name	Width	Data
AC	16	0001
AR	12	E00
DR	16	0000
E	1	0
I	1	0
IR	16	E001
PC	12	006
S	1	1

XOR.a X

```
1 START:
2   INP
3   STA NUM
4   INP
5   XOR NUM
6   OUT
7   HLT
8
9 NUM: .data 1 0
10
```

AddrHexDataHex

MAIN

Addr	Data
000	F800
001	6006
002	F800
003	7006
004	F400
005	E001
006	0000
007	0000
008	0000
009	0000
00A	0000
00B	0000
00C	0000
00D	0000
00E	0000
00F	0000
010	0000

EXECUTING...

Enter Inputs, the first of which must be an Integer: 0

Enter Inputs, the first of which must be an Integer: 1

Output: 1

EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]

Task 5:NOT Operation

DataHex

Registers

Name	Width	Data
AC	16	FFFF
AR	12	E00
DR	16	0000
E	1	0
I	1	0
IR	16	E001
PC	12	005
S	1	1

NOT.a X

```
1 START:
2   INF
3   STA NUM
4   NOT NUM
5   OUT
6   HLT
7
8 NUM: .data 1 0
9
```

AddrHexDataHex

MAIN

Addr	Data
000	F800
001	6005
002	8005
003	F400
004	E001
005	0000
006	0000
007	0000
008	0000
009	0000
00A	0000
00B	0000
00C	0000
00D	0000
00E	0000
00F	0000
010	0000

EXECUTING...
Enter Inputs, the first of which must be an Integer: 0
Output: -1
EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]