

Registers

Register	Value	Flag
A	09	S 0
BC	00 00	Z 1
DE	00 00	AC 0
HL	09 03	P 1
PSW	00 00	C 0
PC	42 15	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

2057 - + 08

Assembly Code:

```

1 LXI H, 2050
2 MOV C, H
3 DCR C
4 INX H
5 MOV A, H
6 LOOP1: INX H
7 CMP M
8 JNC LOOP
9 MOV A, H
10 LOOP: DCR C
11 JNZ LOOP1
12 STA 2050
13 HLT
14

```

Memory View (Start: 2058):

Address (Hex)	Address	Data
080A	2058	9
080B	2059	0
080C	2060	0
080D	2061	0
080E	2062	0
080F	2063	0
0810	2064	0
0811	2065	0
0812	2066	0
0813	2067	0
0814	2068	0
0815	2069	0
0816	2070	0
0817	2071	0

Assembler Message:

```

Line No Assembler Message
0 Program assembled successfully

```

Registers

Register	Value	Flag
A	07	S 0
BC	07 08	Z 0
DE	00 00	AC 0
HL	00 00	P 0
PSW	00 00	C 0
PC	42 10	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

2002 - + 08

Assembly Code:

```

1 LDA 2001
2 MOV B, A
3 LDA 2002
4 MOV C, A
5 STA 2003
6 MOV A, B
7 STA 2004
8 HLT
9

```

Memory View (Start: 2003):

Address (Hex)	Address	Data
07D3	2003	8
07D4	2004	7
07D5	2005	0
07D6	2006	0
07D7	2007	0
07D8	2008	0
07D9	2009	0
07DA	2010	0
07DB	2011	0
07DC	2012	0
07DD	2013	0
07DE	2014	0
07DF	2015	0
07E0	2016	0

Assembler Message:

```

Line No Assembler Message
0 Program assembled successfully

```

Registers

Register	Value	Flag
A	00	S 0
BC	00 00	Z 0
DE	00 00	AC 0
HL	1F 41	P 0
PSW	00 00	C 0
PC	42 0F	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

8003 - + 05

Assembly Code:

```

1 LXI H, 8000
2 MOV B, H
3 INX H
4 MOV A, H
5 CMP B
6 JC STORE
7 MOV A, B
8 STORE: STA 8050
9 HLT
10
11
12
13
14
15
16
17

```

Memory View (Start: 8050):

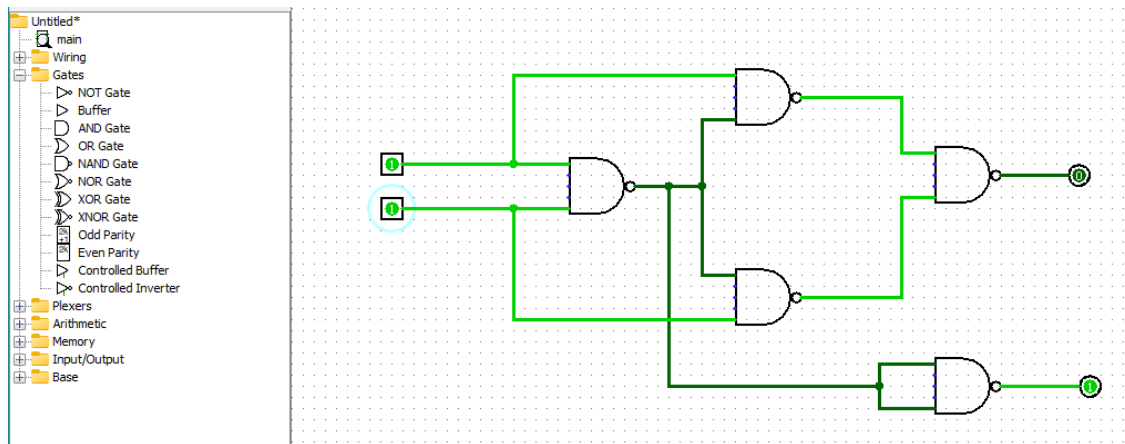
Address (Hex)	Address	Data
1F72	8050	0
1F73	8051	0
1F74	8052	0
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0
1F7E	8062	0
1F7F	8063	0

Assembler Message:

```

Line No Assembler Message
0 Program assembled successfully

```



Registers

Register	Value	Flag
A	19	S 0
BC	00 00	Z 1
DE	00 00	AC 0
HL	1F 40	P 1
PSW	00 00	C 0
PC	42 0E	
SP	FF FF	
Int-Reg	00	

Load me at

```

1 LXI H, 8000
2 XRA A
3 MOV B, M
4 LOOP: ADD M
5 DCR B
6 JNZ LOOP
7 STA 8001
8 HLT
9

```

Memory

Address (Hex)	Address	Data
1F41	8001	25
1F42	8002	0
1F43	8003	0
1F44	8004	0
1F45	8005	0
1F46	8006	0
1F47	8007	0
1F48	8008	0
1F49	8009	0
1F4A	8010	0
1F4B	8011	0
1F4C	8012	0
1F4D	8013	0
1F4E	8014	0

Line No. 0: Program assembled successfully

Registers

Register	Value	Flag
A	FB	S 1
BC	00 00	Z 0
DE	00 00	AC 0
HL	00 00	P 0
PSW	00 00	C 0
PC	42 0D	
SP	FF FF	
Int-Reg	00	

Load me at

```

1 LDA 3000
2 CMA
3 STA 3001
4 ADI 01
5 STA 3002
6 HLT
7

```

Memory

Address (Hex)	Address	Data
0B89	3001	250
0B8A	3002	251
0B8B	3003	0
0B8C	3004	0
0B8D	3005	0
0B8E	3006	0
0B8F	3007	0
0B90	3008	0
0B91	3009	0
0B92	3010	0
0B93	3011	0
0B94	3012	0
0B95	3013	0
0B96	3014	0

Line No. 0: Program assembled successfully

Registers

Register	Value	Flag
A	30	S 0
BC	04 00	Z 0
DE	00 00	AC 0
HL	00 00	P 0
PSW	00 00	C 0
PC	42 0A	
SP	FF FF	
Int-Reg	00	

Load me at

```

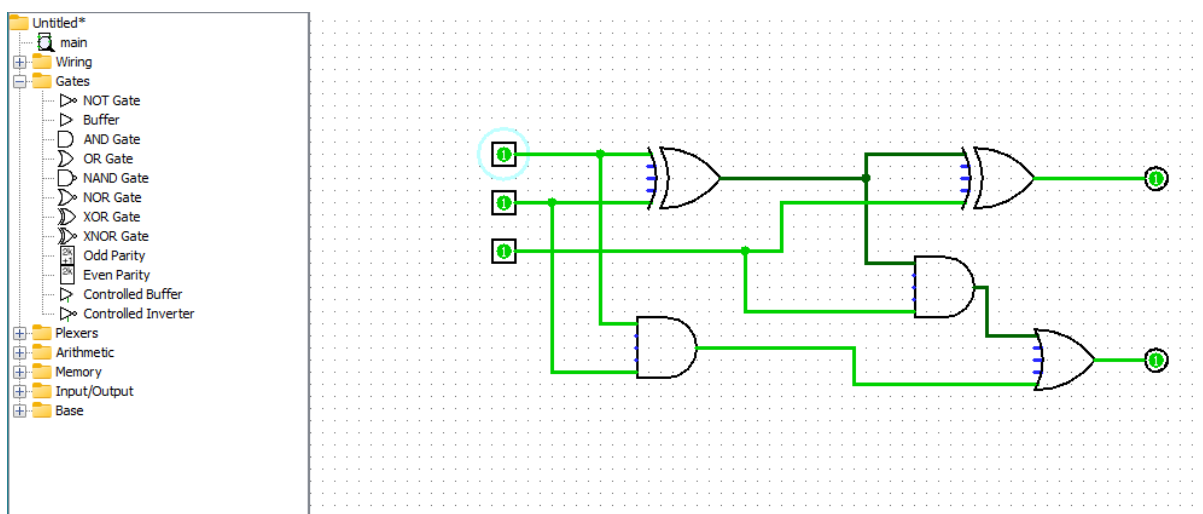
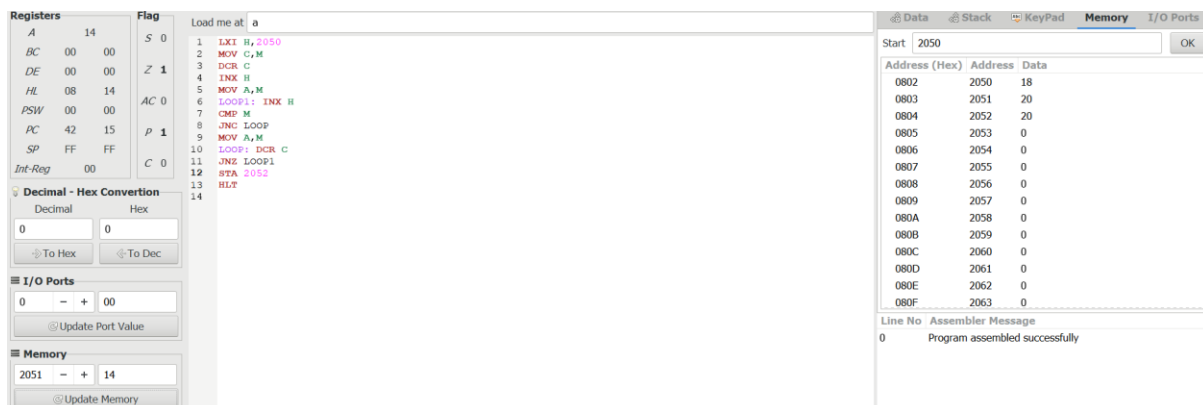
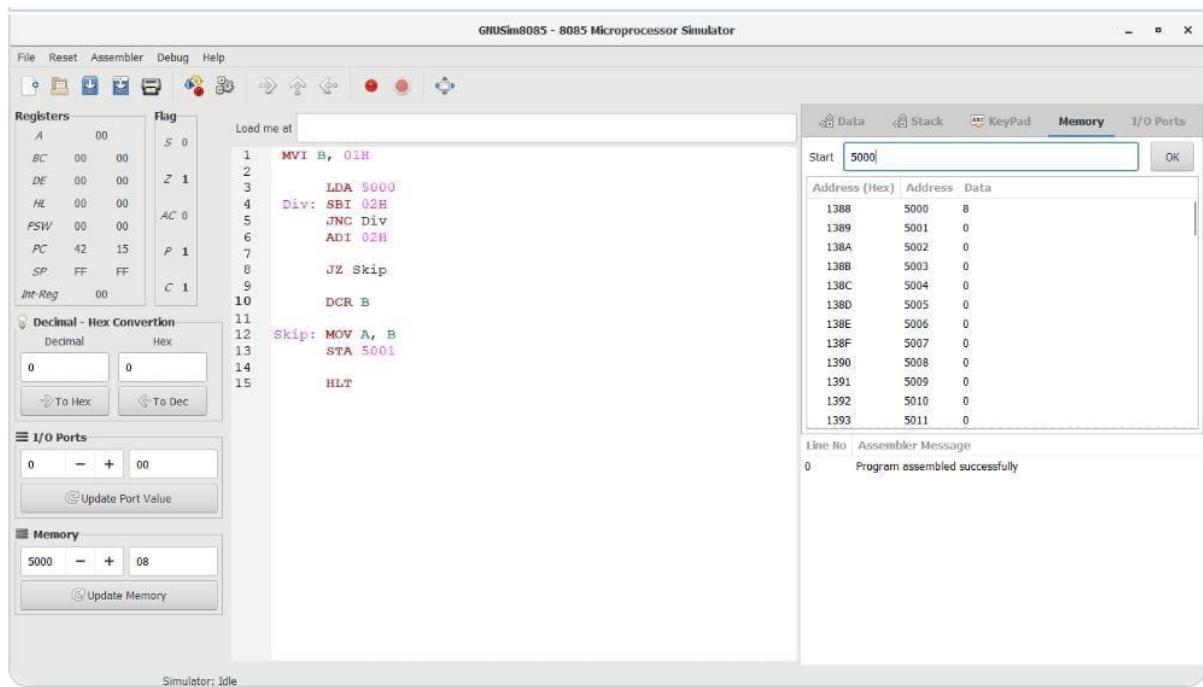
1 MVI A, 03
2 RRC
3 RRC
4 RRC
5 RRC
6 STA 2000
7 HLT
8
9

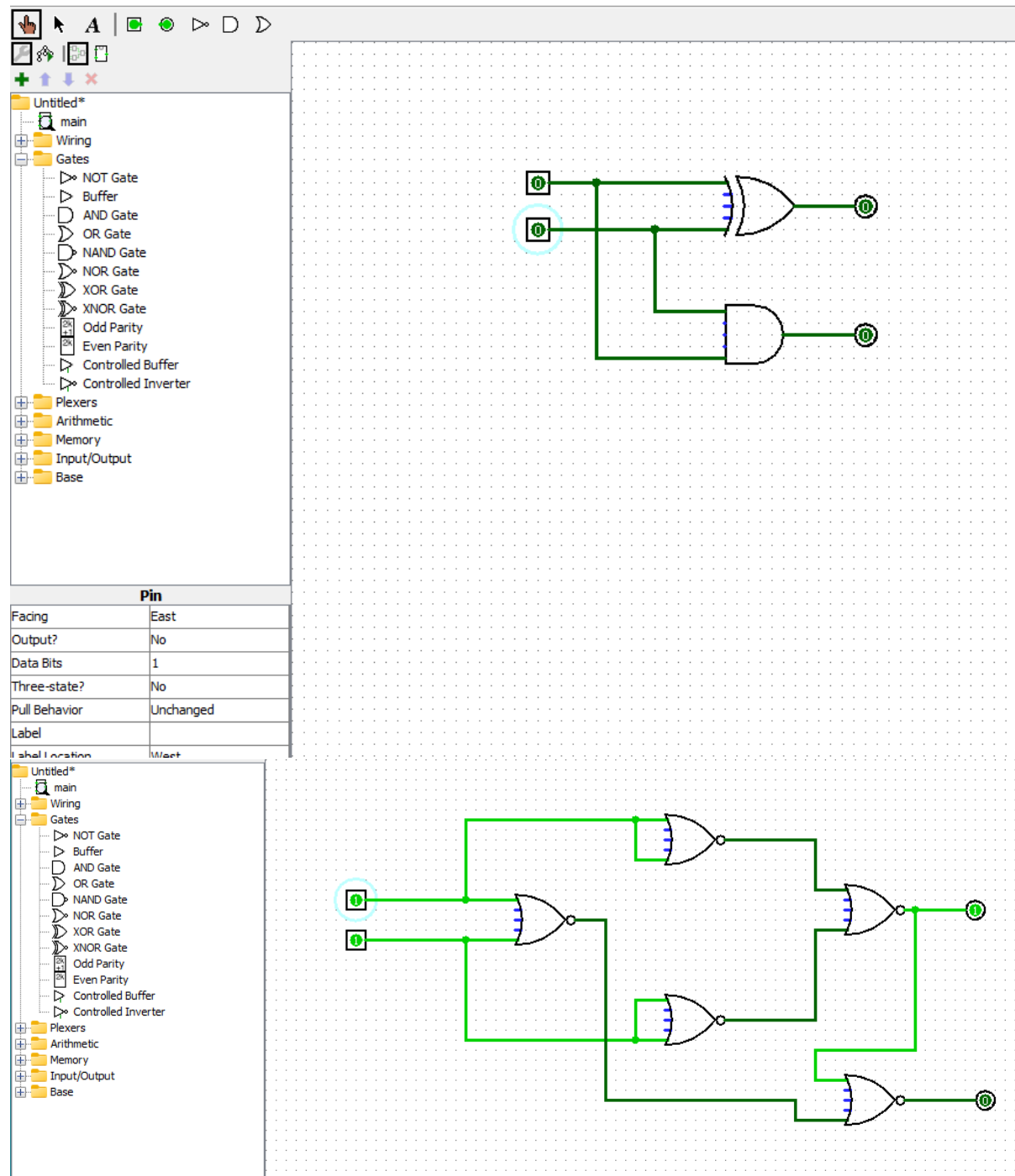
```

Memory

Address (Hex)	Address	Data
07D0	2000	48
07D1	2001	0
07D2	2002	0
07D3	2003	0
07D4	2004	0
07D5	2005	0
07D6	2006	0
07D7	2007	0
07D8	2008	0
07D9	2009	0
07DA	2010	0
07DB	2011	0

Line No. 0: Program assembled successfully





GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers: A 20, BC 04 00, DE 00 00, HL 00 00, PSW 00 00, PC 42 0A, SP FF FF, Int-Reg 00. Flag: S 0, Z 0, AC 0, P 0, C 0.

Decimal - Hex Conversion: Decimal 0, Hex 0. To Hex, To Dec.

I/O Ports: 0, 00. Update Port Value.

Memory: 0, 00. Update Memory.

Load me at: 1 MVI A, 02, 2 RLC, 3 RLC, 4 RLC, 5 RLC, 6 STA 2000, 7 HLT, 8, 9, 10.

Start: 2000. OK.

Address (Hex)	Address	Data
07D0	2000	32
07D1	2001	0
07D2	2002	0
07D3	2003	0
07D4	2004	0
07D5	2005	0
07D6	2006	0
07D7	2007	0
07D8	2008	0
07D9	2009	0
07DA	2010	0
07DB	2011	0

Line No Assembler Message
0 Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers: A 04, BC 04 00, DE 00 00, HL 00 00, PSW 00 00, PC 42 20, SP FF FF, Int-Reg 00. Flag: S 0, Z 1, AC 0, P 1, C 0.

Decimal - Hex Conversion: Decimal 0, Hex 0. To Hex, To Dec.

I/O Ports: 2, 4. Update Port Value.

Memory: 65535, 00.

Load me at: 1 IN 00H, 2 MOV B, A, 3 IN 01H, 4 CMP B, 5 JZ OP, 6 JNC REC, 7 MOV C, A, 8 MOV A, B, 9 MOV B, C, 10 REC: SUB B, 11 CMP B, 12 JZ OP, 13 JNC REC, 14 MOV C, A, 15 MOV A, B, 16 MOV B, C, 17 JMP REC, 18 OP: OUT 02H, 19 HLT.

Start: 02. OK.

Address (Hex)	Address	Data
0002	2	0
0003	3	0
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0
000E	14	0
000F	15	0

Line No Assembler Message
0 Program assembled successfully

Load me at: 1 MVI A, 07, 2 MVI B, 06, 3 ORA B, 4 STA 2000, 5 HLT.

Start: 2000. OK.

Address (Hex)	Address	Data
07D0	2000	7
07D1	2001	0
07D2	2002	0
07D3	2003	0
07D4	2004	0
07D5	2005	0
07D6	2006	0
07D7	2007	0
07D8	2008	0
07D9	2009	0
07DA	2010	0
07DB	2011	0
07DC	2012	0
07DD	2013	0

Registers

A	04
BC	04 00
DE	00 00
HL	00 00
PSW	00 00
PC	42 09
SP	FF FF
Int-Reg	00

Flag

S	0
Z	0
AC	1
P	0
C	0

I/O Ports

Port 0	04
--------	----

Assembly Code

Load me at

```

1  MOV A, 06
2  MOV B, 04
3  ANA B
4  STA 2500
5  HLT

```

Decimal - Hex Conversion

Decimal

0

Hex

0

→ To Hex

← To Dec

I/O Ports

1

-

+

04

Update Port Value

Memory

0

-

+

00

Update Memory

Memory Dump

Start

Z500

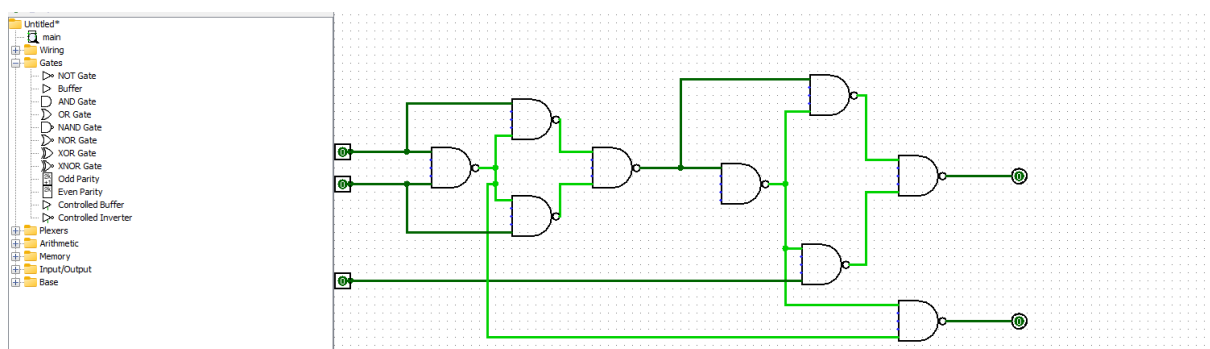
OK

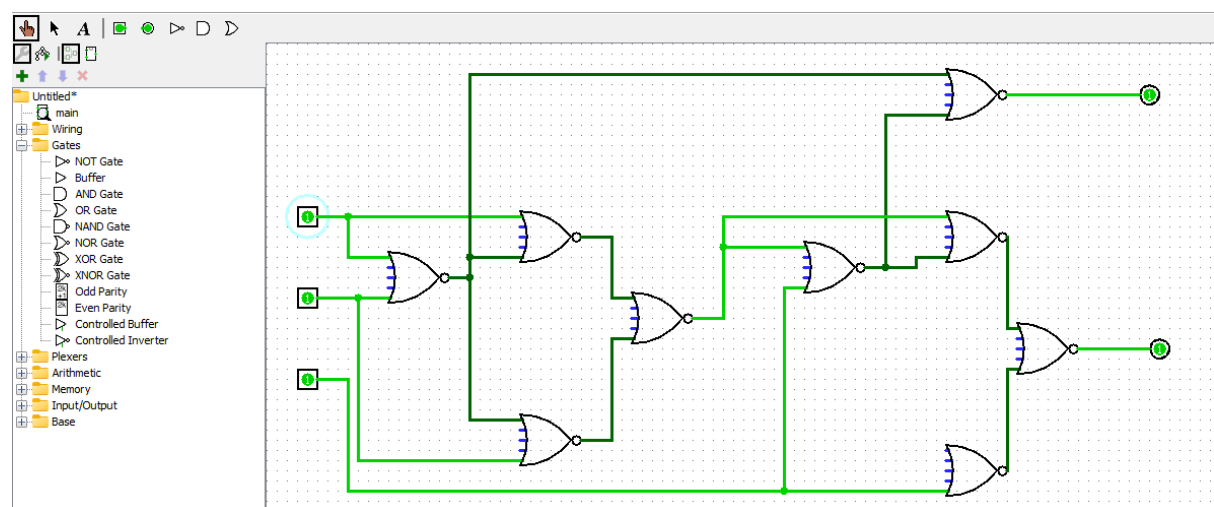
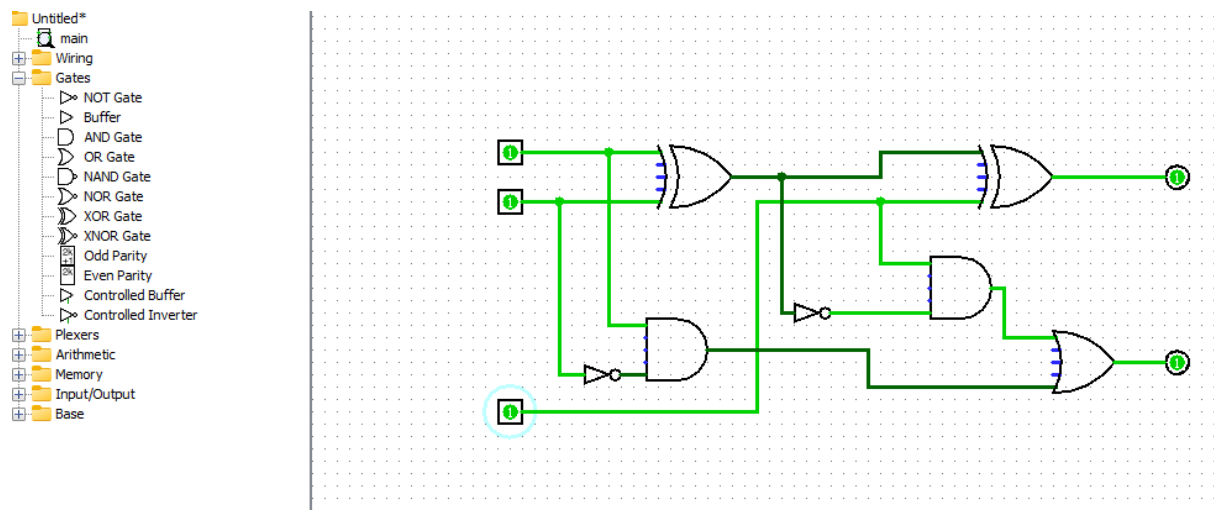
Address (Hex)	Address	Data
09C4	2500	4
09C5	2501	0
09C6	2502	0
09C7	2503	0
09C8	2504	0
09C9	2505	0
09CA	2506	0
09CB	2507	0
09CC	2508	0
09CD	2509	0
09CE	2510	0
09CF	2511	0
09D0	2512	0
09D1	2513	0

Line No

Assembler Message

0 Program assembled successfully





Registers

Register	Value	Flag
A	18	S 0
BC	00 05	Z 1
DE	00 18	AC 0
HL	00 00	P 1
PSW	00 00	C 0
PC	42 18	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

2001 - + 04

Load me at

```

1  LDA 2001
2  MOV B,A
3  MVI C,#01
4  MVI R,#01
5  LOOP: MOV D,C
6  MVI A,DOR
7  LP: ADD E
8  DCR D
9  JNZ LP
10 MOV B,A
11 INR C
12 DCR B
13 JNZ LOOP
14 MOV A,B
15 STA 2010
16 HLT
17

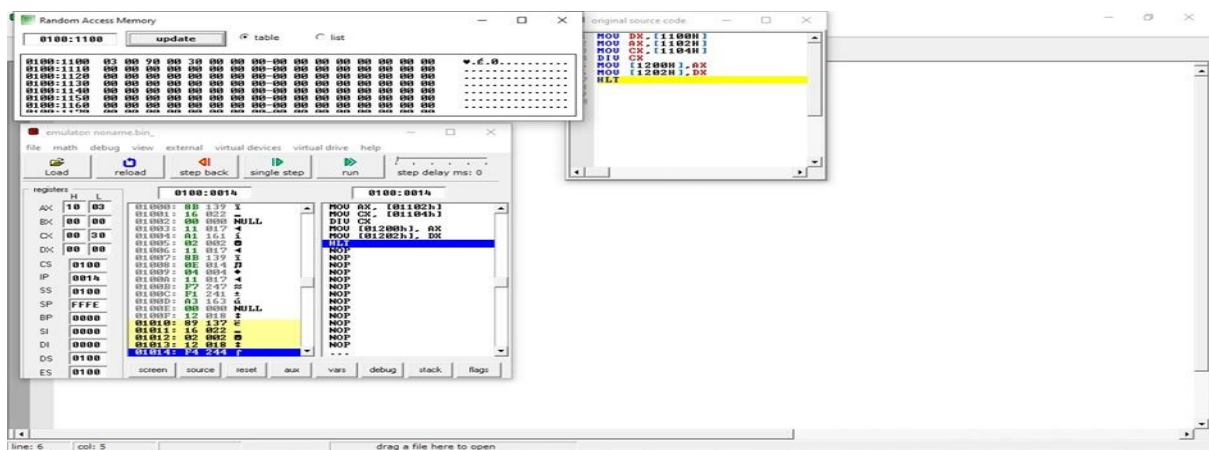
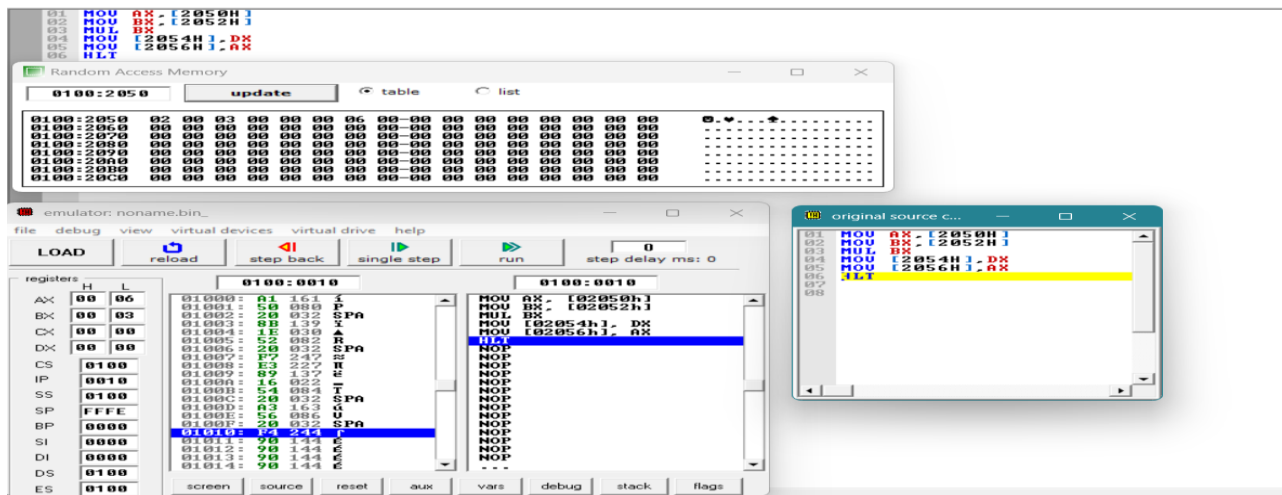
```

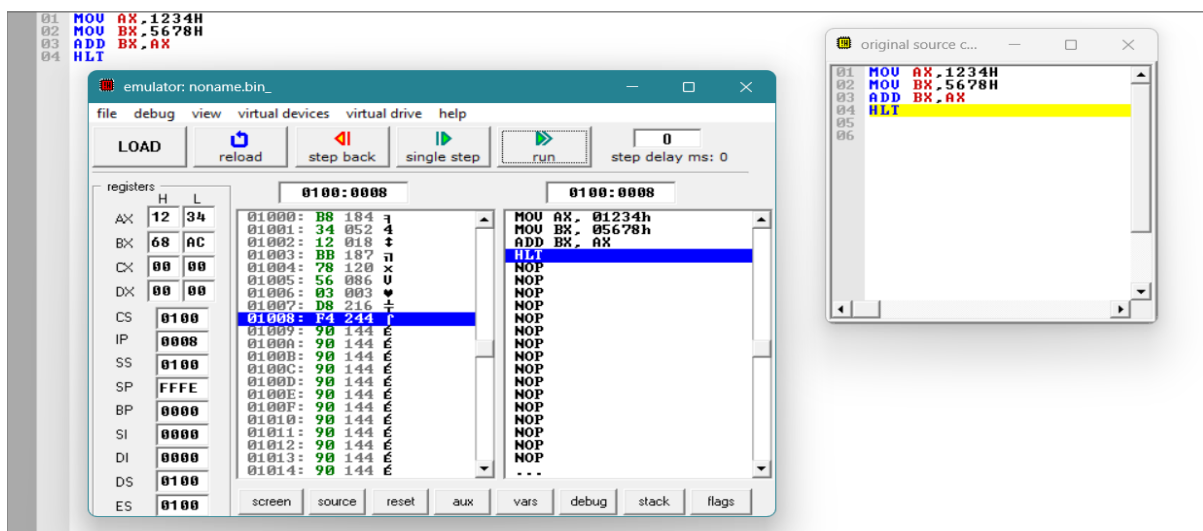
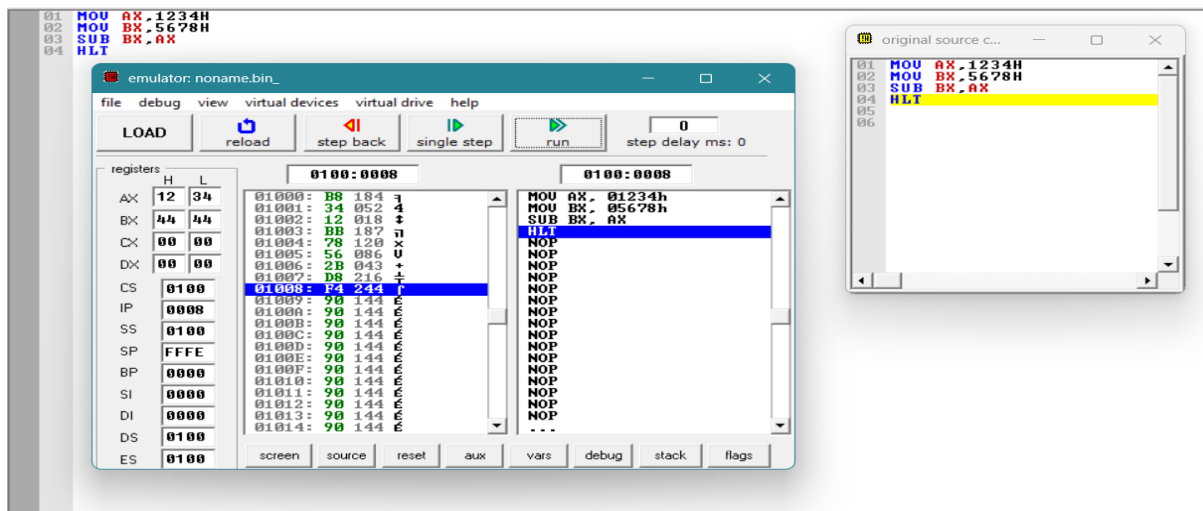
Memory

Address (Hex)	Address	Data
07DA	2010	24
07DB	2011	0
07DC	2012	0
07DD	2013	0
07DE	2014	0
07DF	2015	0
07E0	2016	0
07E1	2017	0
07E2	2018	0
07E3	2019	0
07E4	2020	0
07E5	2021	0
07E6	2022	0
07E7	2023	0

Line No Assembler Message

0 Program assembled successfully





Registers

Register	Value	Flag
A	06	S 0
BC	02 00	Z 0
DE	00 00	AC 0
HL	00 00	P 1
PSW	00 00	C 0
PC	42 0C	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

501 - + 08

Load me at

```

1 LDA 8500
2 MOV B, A
3 LDA 8501
4 SUB B
5 STA 8502
6 RST 1
7

```

Data

Address (Hex)	Address	Data
2136	8502	6
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0
2140	8512	0
2141	8513	0
2142	8514	0
2143	8515	0

Line No Assembler Message

0 Program assembled successfully

Registers

Register	Value	Flag
A	09	S 0
BC	08 00	Z 0
DE	00 00	AC 0
HL	00 00	P 1
PSW	00 00	C 0
PC	42 0C	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

8501 - + 01

Load me at

```

1 LDA 8500
2 MOV B, A
3 LDA 8501
4 ADD B
5 STA 8502
6 RST 1
7

```

Data

Address (Hex)	Address	Data
2136	8502	9
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0
2140	8512	0
2141	8513	0
2142	8514	0
2143	8515	0

Line No Assembler Message

0 Program assembled successfully

Registers

Register	Value	Flag
A	05	S 1
BC	04 05	Z 0
DE	00 00	AC 0
HL	13 89	P 1
PSW	00 00	C 1
PC	42 19	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

5001 - + 14

Load me at

```

1 LXI H, 5000
2 MOV B, M ;Get the dividend in B - reg.
3 MVI C, 00 ;Clear C - reg for quotient
4 INX H ;Increment HL pair of registers
5 MOV A, M ;Get the divisor in A - reg
6 NEXT: CMP B ;Compare A - reg with register B.
7 JC LOOP ;Jump on carry to LOOP
8 SUB B ;Subtract A - reg from B - reg.
9 INR C ;Increment content of register C.
10 JMP NEXT ;Jump to NEXT
11 LOOP: STA 5002 ;Store the remainder in Memory
12 MOV A, C ;Move Content of C - Reg to A - Reg
13 STA 5003 ;Store the quotient in memory
14 HLT ;Terminate the program.

```

Data

Address (Hex)	Address	Data
138A	5002	0
138B	5003	5
138C	5004	0
138D	5005	0
138E	5006	0
138F	5007	0
1390	5008	0
1391	5009	0
1392	5010	0
1393	5011	0
1394	5012	0
1395	5013	0
1396	5014	0
1397	5015	0

Line No Assembler Message

0 Program assembled successfully

Registers

Register	Value	Flag
A	06	S 0
BC	02 00	Z 1
DE	00 00	AC 0
HL	00 00	P 1
PSW	00 00	C 0
PC	42 1A	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal: 0 Hex: 0

I/O Ports

0 - + 00

Memory

8501 - + 03

Load me at

```

1 LDA 8500
2 MOV B, A
3 LDA 8501
4 MOV C, A
5 CPI 00
6 JZ LOOP
7 XRA A
8 LOOP1: ADD B
9 DCR C
10 JZ LOOP
11 JMP LOOP1
12 LOOP: STA 8502
13 RST 1
14

```

Data

Address (Hex)	Address	Data
2136	8502	6
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0
2140	8512	0
2141	8513	0
2142	8514	0
2143	8515	0

Line No Assembler Message

0 Program assembled successfully