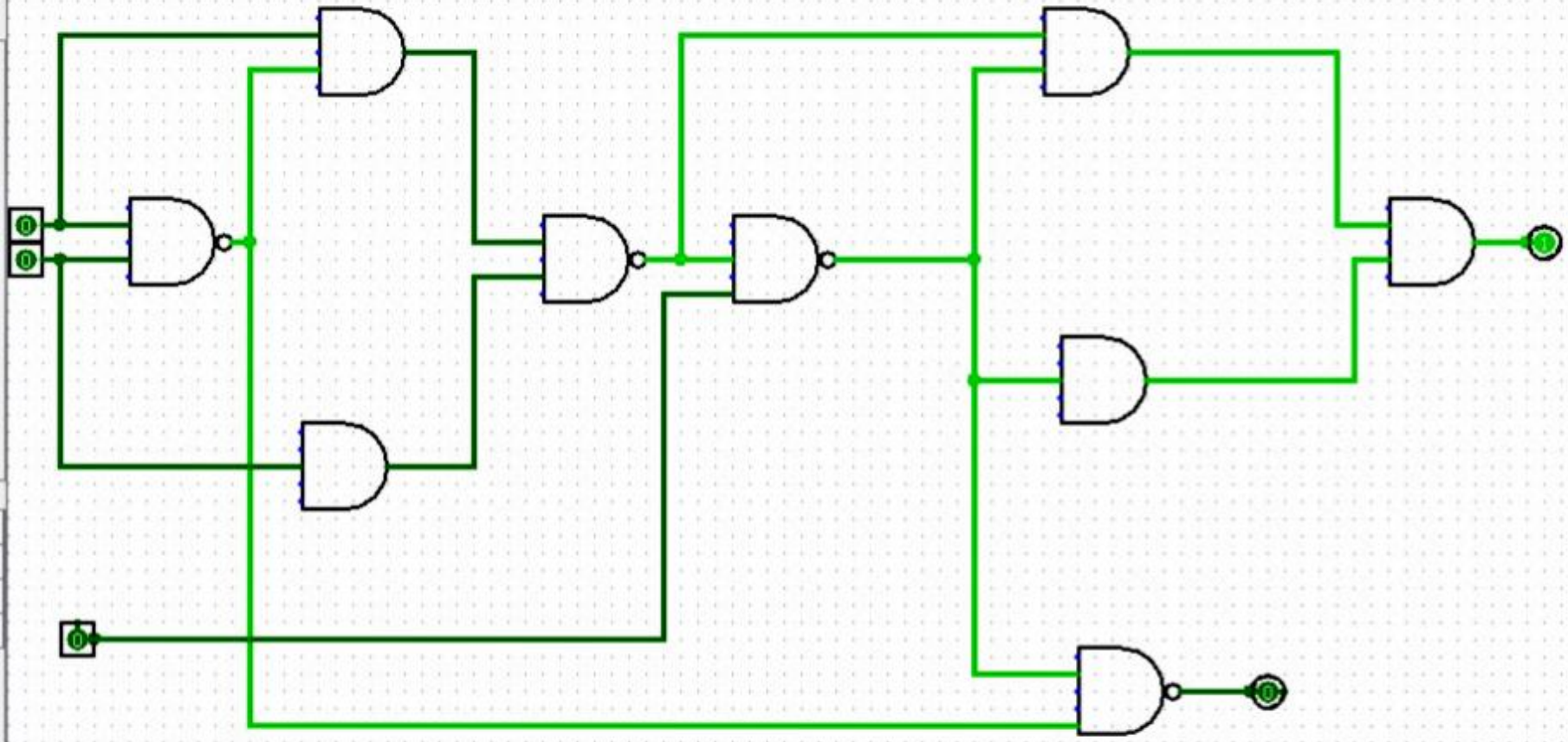
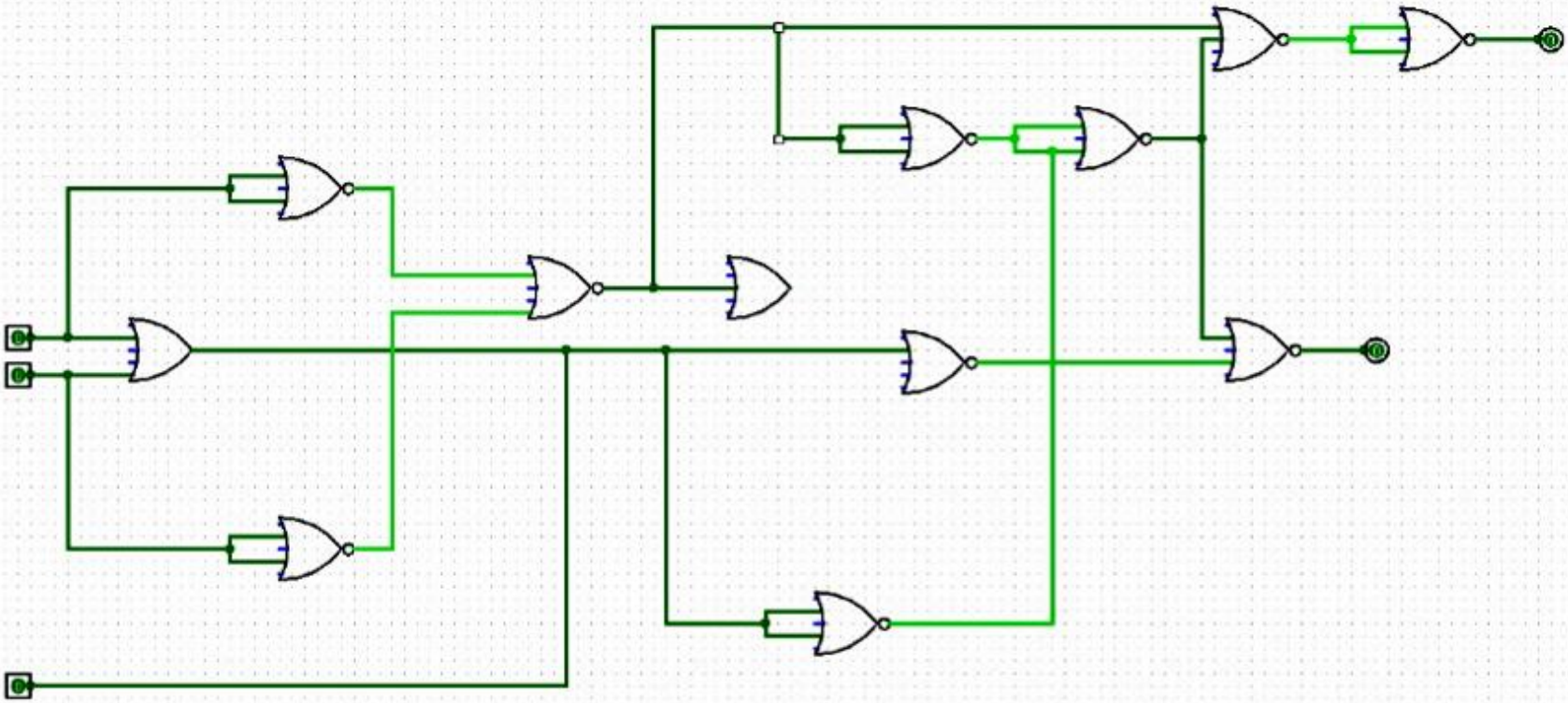




Load me at

```
1  LXI H,8050
2  MOV A,M
3  MVI C,00
4  MOV D,A
5  MOV E,A
6  L2: MOV B,D
7  L1: CMP B
8  JC LABEL
9  SUB B
10 JNZ L1
11 LABEL: CPI 00
12 JNZ SKIP
13 INR C
14 SKIP: MOV A,E
15 DCR D
16 JNZ L2
17 MOV A,C
18 STA 8051
19 HLT
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	5
1F73	8051	2
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

Line No	Assembler Message
0	Program assembled successfully

```
1  LDA 8050
2  MOV B,A
3  LDA 8051
4  CMP B
5  JC STORE
6  MOV A,B
7  STORE: STA 8052
8  HLT
```

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	50
1F73	8051	30
1F74	8052	30
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

Line No	Assembler Message
0	Program assembled successfully

elp

Load me at

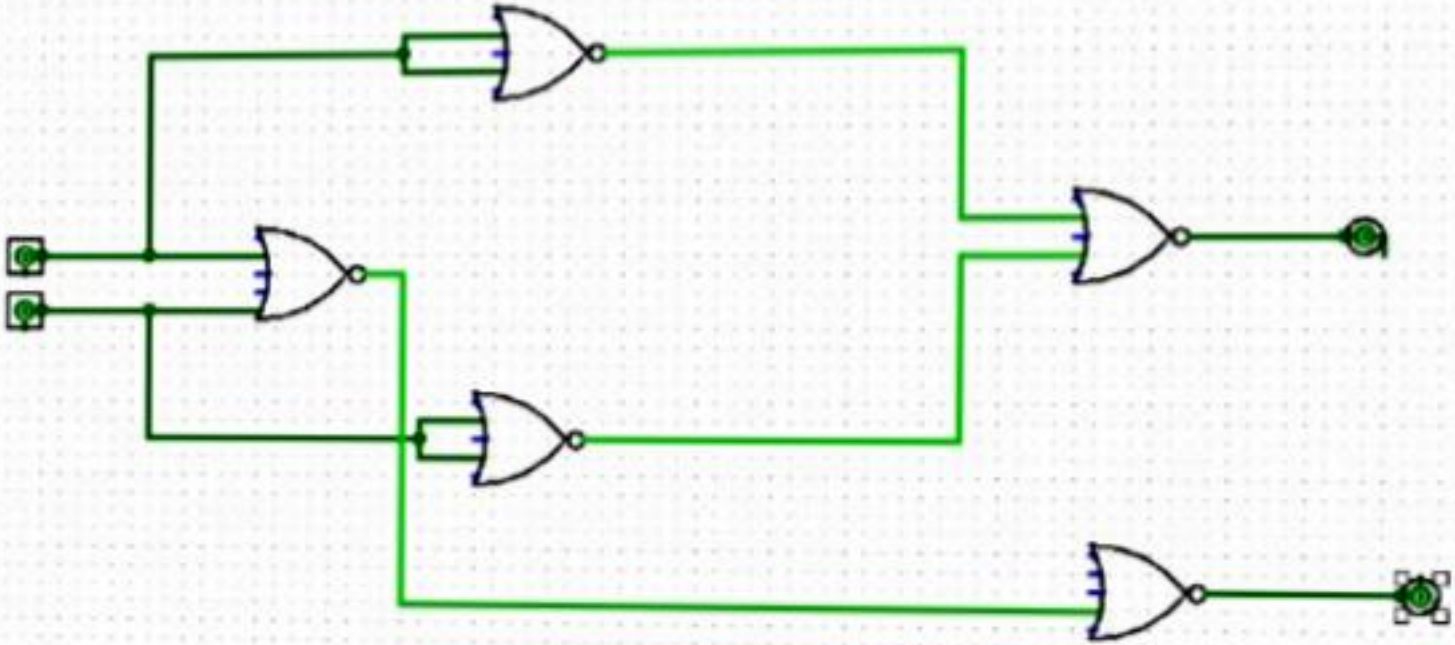
```
1  LDA 8050
2  MOV B,A
3  LDA 8051
4  CMP B
5  JC STORE
6  MOV A,B
7  STORE: STA 8052
8  HLT
```

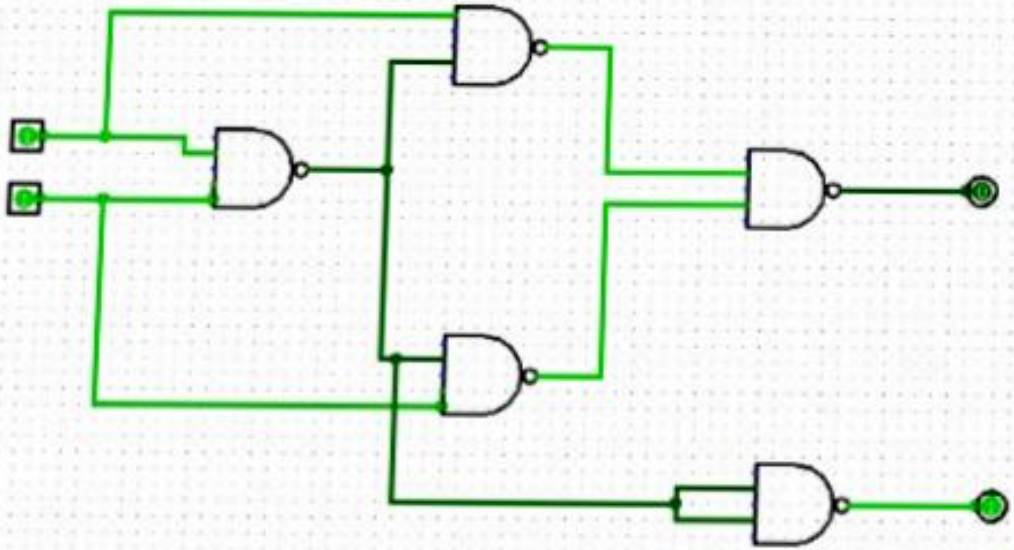
Data | Stack | KeyPad | **Memory** | I/O Ports

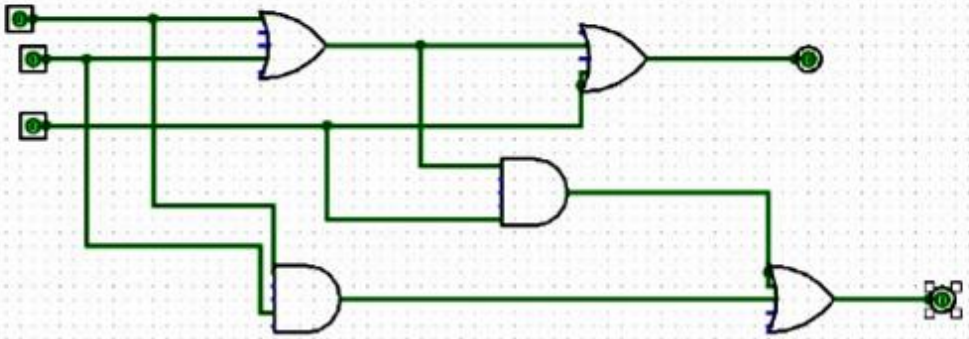
Start 8052 OK

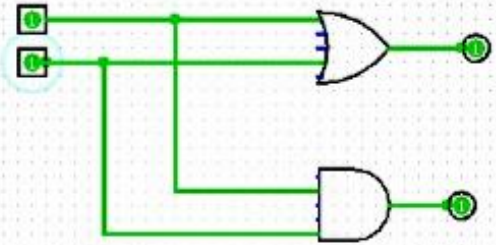
Address (Hex)	Address	Data
1F74	8052	21
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0
1F7E	8062	0
1F7F	8063	0
1F80	8064	0
1F81	8065	0

Line No	Assembler Message
0	Program assembled successfully









GNUSim8085 - 8085 Microprocessor Simulator

elp

Load me at

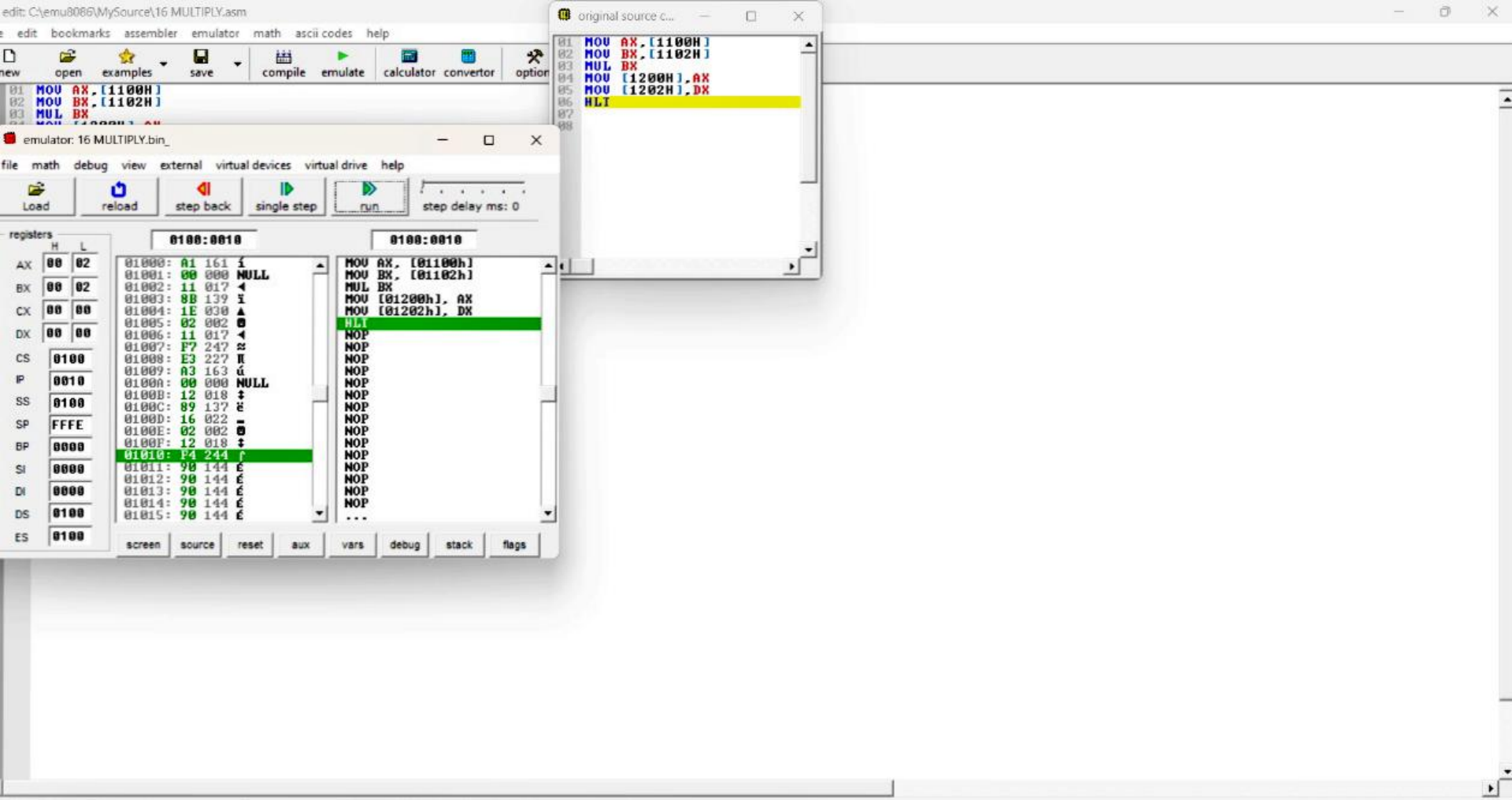
```
1   LXI H,8050
2   MOV C,M
3   INX H
4   MOV B,M
5   DCR C
6   LOOP: INX H
7   MOV A,M
8   CMP B
9   JC SKIP
10  MOV B,A
11  SKIP: DCR C
12  JNZ LOOP
13  LXI H,8510
14  MOV M,B
15  HLT
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start OK

Address (Hex)	Address	Data
2134	8500	0
2135	8501	0
2136	8502	0
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	10
213F	8511	0
2140	8512	0
2141	8513	0

Line No	Assembler Message
0	Program assembled successfully


edit: C:\emu8086\MySource\16 MULTIPLY.asm
edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor option
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 MUL BX
original source c...
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 MUL BX
04 MOV [1200H],AX
05 MOV [1202H],DX
06 HLT
07
08
emulator: 16 MULTIPLY.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 02
BX 00 02
CX 00 00
DX 00 00
CS 0100
IP 0010
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:0010
0100:0010
01000: A1 161 í
01001: 00 000 NULL
01002: 11 017 ◄
01003: 8B 139 ï
01004: 1E 030 ▲
01005: 02 002 ☻
01006: 11 017 ◄
01007: F7 247 ≈
01008: E3 227 π
01009: A3 163 ú
0100A: 00 000 NULL
0100B: 12 018 ↕
0100C: 89 137 ë
0100D: 16 022 ▬
0100E: 02 002 ☻
0100F: 12 018 ↕
01010: F4 244 ⌠
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
MOV AX, [01100h]
MOV BX, [01102h]
MUL BX
MOV [01200h], AX
MOV [01202h], DX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags

File Reset Assembler Debug Help

Registers

A	0B	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	13
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 0
AC 1
P 0
C 0

Decimal - Hex Convertion

Decimal	Hex
0	0

To Hex | To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

8050 − + 05

Update Memory

Load me at

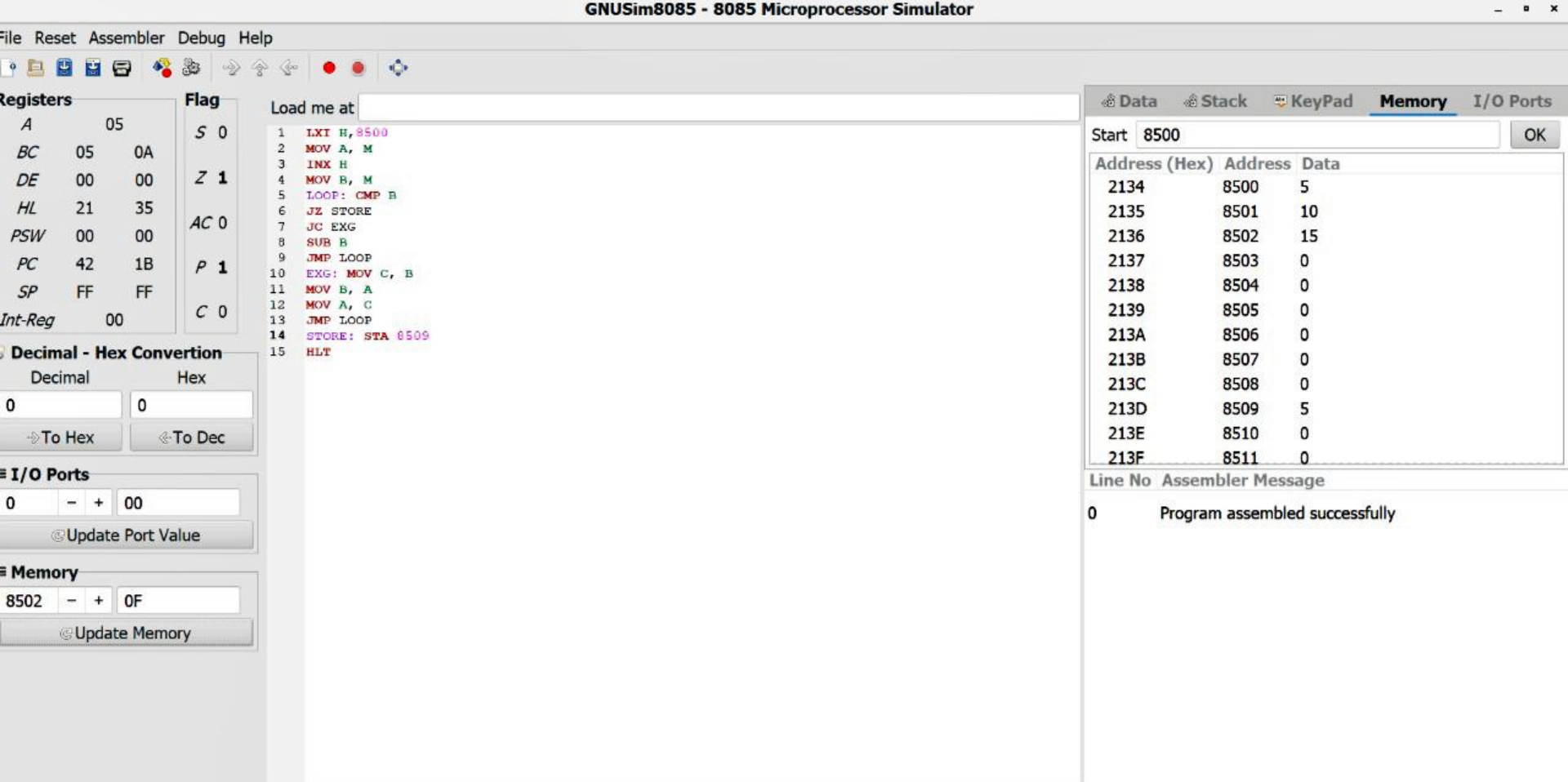
```
1  LDA 8050
2  ANI 01
3  JZ LOOP1
4  MVI A,11
5  JMP LOOP2
6  LOOP1: MVI A,22
7  LOOP2: STA 8051
8  HLT
```

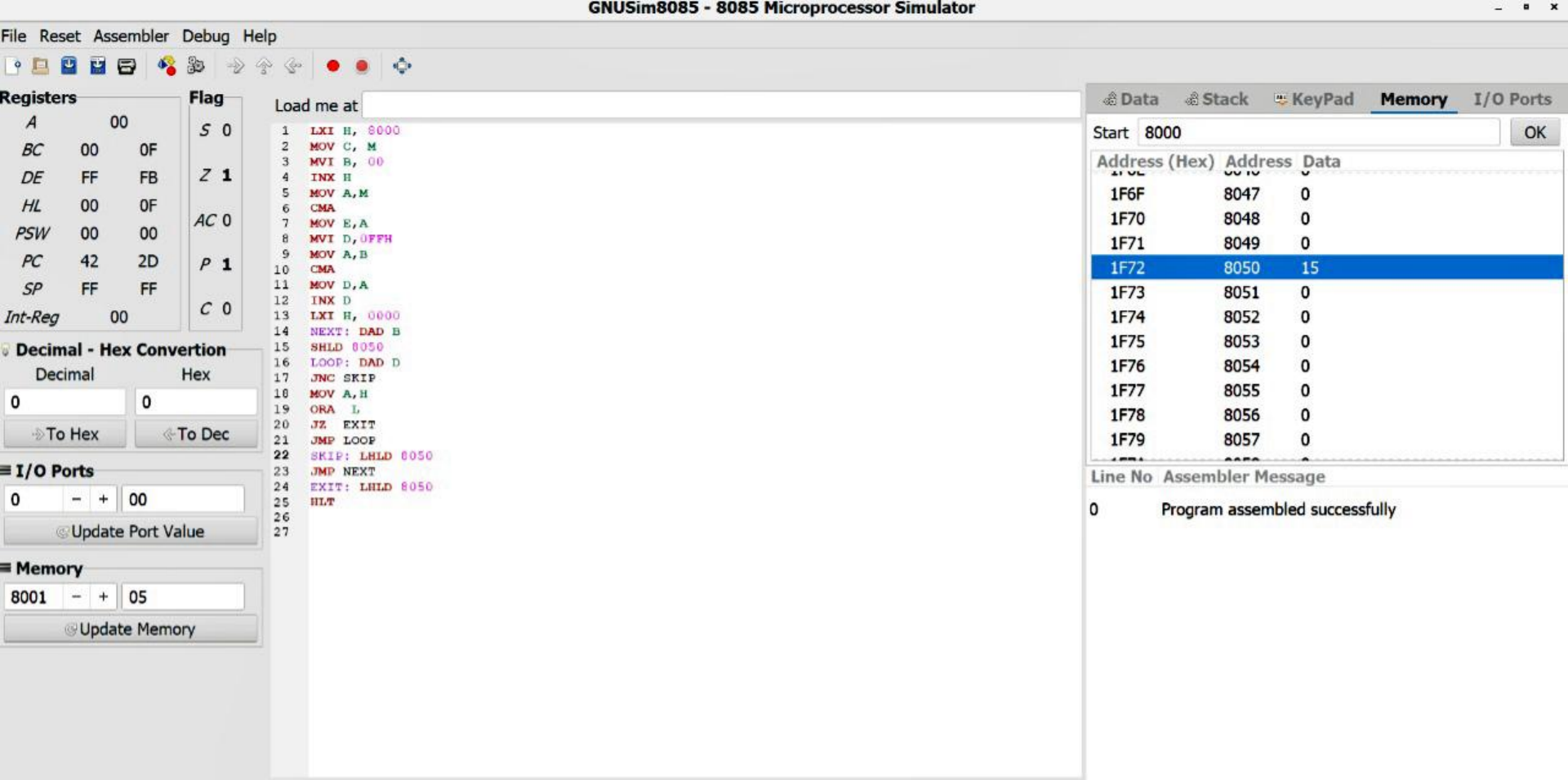
Data | Stack | KeyPad | **Memory** | I/O Ports

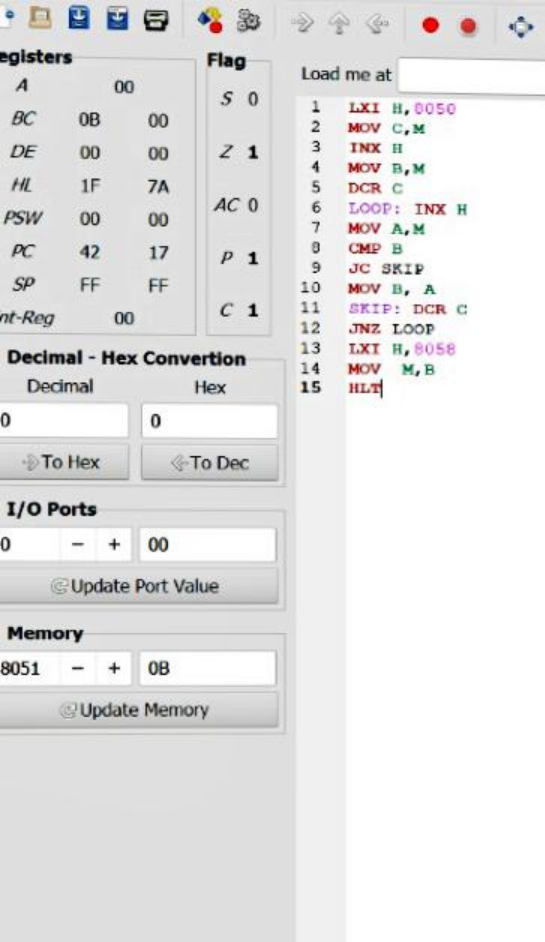
Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	5
1F73	8051	11
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

Line No	Assembler Message
0	Program assembled successfully


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Registers
A 05
BC 05 0A
DE 00 00
HL 21 35
PSW 00 00
PC 42 1B
SP FF FF
Int-Reg 00
Flag
S 0
Z 1
AC 0
P 1
C 0
Decimal - Hex Convertion
Decimal
Hex
0
0
To Hex
To Dec
I/O Ports
0 − + 00
Update Port Value
Memory
8502 − + 0F
Update Memory
Load me at
1 LXI H,8500
2 MOV A, M
3 INX H
4 MOV B, M
5 LOOP: CMP B
6 JZ STORE
7 JC EXG
8 SUB B
9 JMP LOOP
10 EXG: MOV C, B
11 MOV B, A
12 MOV A, C
13 JMP LOOP
14 STORE: STA 8509
15 HLT
Data
Stack
KeyPad
Memory
I/O Ports
Start 8500
OK
Address (Hex) Address Data
2134 8500 5
2135 8501 10
2136 8502 15
2137 8503 0
2138 8504 0
2139 8505 0
213A 8506 0
213B 8507 0
213C 8508 0
213D 8509 5
213E 8510 0
213F 8511 0
Line No Assembler Message
0 Program assembled successfully


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Registers
A 00
BC 00 0F
DE FF FB
HL 00 0F
PSW 00 00
PC 42 2D
SP FF FF
Int-Reg 00
Flag
S 0
Z 1
AC 0
P 1
C 0
Decimal - Hex Convertion
Decimal
Hex
0
0
To Hex
To Dec
I/O Ports
0 - + 00
Update Port Value
Memory
8001 - + 05
Update Memory
Load me at
1 LXI H, 8000
2 MOV C, M
3 MVI B, 00
4 INX H
5 MOV A,M
6 CMA
7 MOV E,A
8 MVI D,0FFH
9 MOV A,B
10 CMA
11 MOV D,A
12 INX D
13 LXI H, 0000
14 NEXT: DAD B
15 SHLD 8050
16 LOOP: DAD D
17 JNC SKIP
18 MOV A,H
19 ORA L
20 JZ EXIT
21 JMP LOOP
22 SKIP: LHLD 8050
23 JMP NEXT
24 EXIT: LHLD 8050
25 HLT
26
27
Data
Stack
KeyPad
Memory
I/O Ports
Start 8000
OK
Address (Hex) Address Data
1F6F 8047 0
1F70 8048 0
1F71 8049 0
1F72 8050 15
1F73 8051 0
1F74 8052 0
1F75 8053 0
1F76 8054 0
1F77 8055 0
1F78 8056 0
1F79 8057 0
Line No Assembler Message
0 Program assembled successfully



egisters

A	00	
BC	0B	00
DE	00	00
HL	1F	7A
PSW	00	00
PC	42	17
SP	FF	FF
nt-Reg	00	

Flag

S 0
Z 1
AC 0
P 1
C 1

Decimal - Hex Convertion

Decimal: 0 — To Hex
Hex: 0 — To Dec

I/O Ports

0 − + 00
Update Port Value

Memory

8051 − + 0B
Update Memory

Load me at

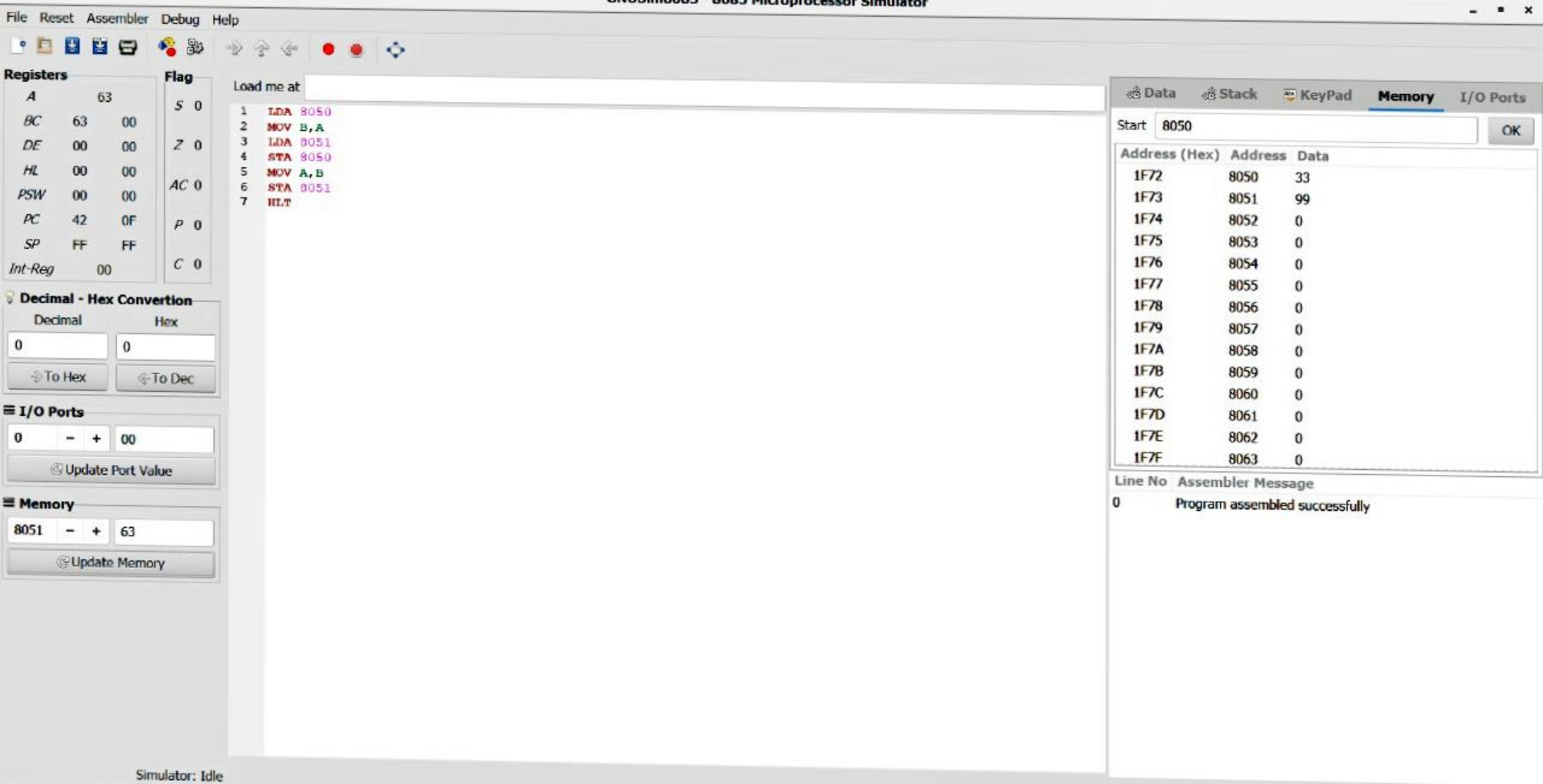
```
1  LXI H,8050
2  MOV C,M
3  INX H
4  MOV B,M
5  DCR C
6  LOOP: INX H
7  MOV A,M
8  CMP B
9  JC SKIP
10 MOV B, A
11 SKIP: DCR C
12 JNZ LOOP
13 LXI H,8058
14 MOV  M,B
15 HLT
```

Data | Stack | KeyPad | Memory | I/O Po

Start 8050

Address (Hex)	Address	Data
1F72	8050	15
1F73	8051	11
1F74	8052	0
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	11
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0
1F7E	8062	0
1F7F	8063	0

Line No	Assembler Message
0	Program assembled successfully


File Reset Assembler Debug Help
Registers
A 63
BC 63 00
DE 00 00
HL 00 00
PSW 00 00
PC 42 0F
SP FF FF
Int-Reg 00
Flag
S 0
Z 0
AC 0
P 0
C 0
Decimal - Hex Convertion
Decimal
Hex
0
0
To Hex
To Dec
I/O Ports
0 - + 00
Update Port Value
Memory
8051 - + 63
Update Memory
Load me at
1 LDA 8050
2 MOV B,A
3 LDA 8051
4 STA 8050
5 MOV A,B
6 STA 8051
7 HLT
Data
Stack
KeyPad
Memory
I/O Ports
Start 8050
OK
Address (Hex) Address Data
1F72 8050 33
1F73 8051 99
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
Line No Assembler Message
0 Program assembled successfully
Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	48	
BC	18	00
DE	00	00
HL	08	04
PSW	00	00
PC	42	0D
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 0
AC 0
P 1
C 0

Decimal - Hex Convertion

Decimal: 0 — To Hex

Hex: 0 — To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

2050 − + 0C

Update Memory

Load me at

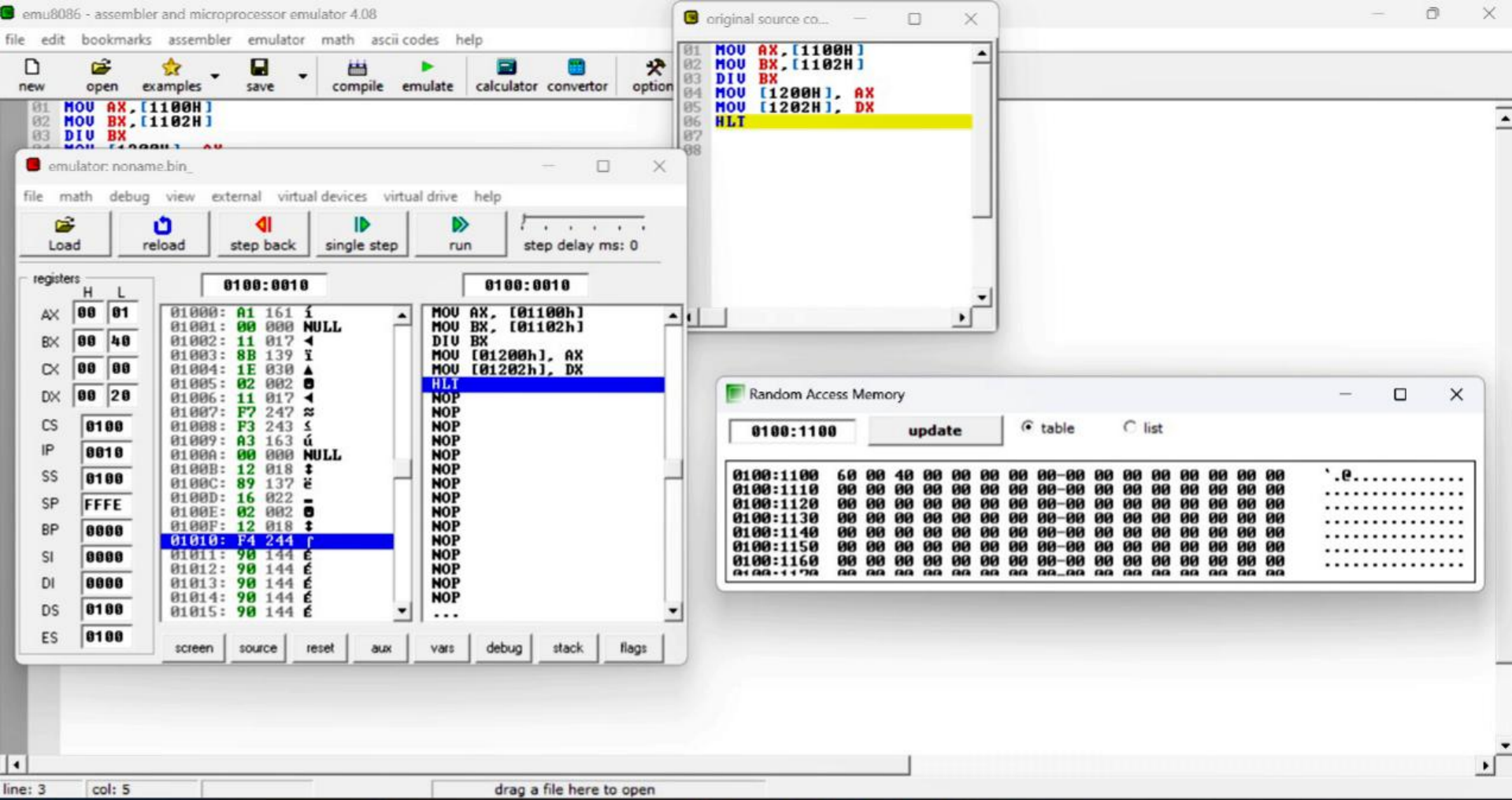
```
1  LXI H,2050
2  MOV A,M
3  ADD A
4  MOV B,A
5  ADD A
6  ADD B
7  INX H
8  ADD M
9  INX H
10 MOV M,A
11 HLT
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 2050 OK

Address (Hex)	Address	Data
0802	2050	12
0803	2051	0
0804	2052	72
0805	2053	0
0806	2054	0
0807	2055	0
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0
080E	2062	0
080F	2063	0

Line No	Assembler Message
0	Program assembled successfully


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor option
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 DIV BX
original source co...
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 DIV BX
04 MOV [1200H], AX
05 MOV [1202H], DX
06 HLT
07
08
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 01
BX 00 40
CX 00 00
DX 00 20
CS 0100
IP 0010
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:0010
01000: A1 161 í
01001: 00 000 NULL
01002: 11 017 ◄
01003: 8B 139 ï
01004: 1E 030 ▲
01005: 02 002 ☻
01006: 11 017 ◄
01007: F7 247 ≈
01008: F3 243 ≤
01009: A3 163 ú
0100A: 00 000 NULL
0100B: 12 018 ↕
0100C: 89 137 ë
0100D: 16 022 ▬
0100E: 02 002 ☻
0100F: 12 018 ↕
01010: F4 244 ⌠
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
0100:0010
MOV AX, [01100h]
MOV BX, [01102h]
DIV BX
MOV [01200h], AX
MOV [01202h], DX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags
Random Access Memory
0100:1100
update
table
list
0100:1100 60 00 40 00 00 00 00 00-00 00 00 00 00 00 00 00 `.@.....
0100:1110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
line: 3
col: 5
drag a file here to open

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	19	
BC	19	00
DE	00	00
HL	1F	74
PSW	00	00
PC	42	1D
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 1
AC 0
P 1
C 0

Decimal - Hex Convertion

Decimal: 0 — To Hex
Hex: 0 — To Dec

I/O Ports

0 − + 00
Update Port Value

Memory

8055 − + 03
Update Memory

Load me at

```
1  LXI H,8085
2  MOV C,M
3  DCR C
4  LOOP3: MOV D,C
5  LXI H,8051
6  LOOP2: MOV A,M
7  INX H
8  CMP M
9  JNC LOOP1
10 MOV B,M
11 MOV M,A
12 DCX H
13 MOV M,B
14 INX H
15 LOOP1: DCR D
16 JNZ LOOP2
17 DCR C
18 JNZ LOOP3
19 HLT
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	0
1F73	8051	25
1F74	8052	17
1F75	8053	12
1F76	8054	10
1F77	8055	3
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	05	
BC	03	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	0C
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 0
AC 0
P 1
C 0

Decimal - Hex Convertion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

8501 − + 02

Update Memory

Load me at

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

Data Stack KeyPad **Memory** I/O Ports

Start 8500 OK

Address (Hex)	Address	Data
2134	8500	3
2135	8501	2
2136	8502	5
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

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	02	
BC	09	00
DE	00	02
HL	00	04
PSW	00	00
PC	42	1A
SP	FF	FF
Int-Reg	00	

Flag

S 1
Z 0
AC 0
P 1
C 1

Decimal - Hex Convertion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

8502 − + 3

Update Memory

Load me at

```
1  LDA 8501
2  MOV B,A
3  LDA 8500
4  MVI C,00
5  LOOP: CMP B
6  JC LOOP1
7  SUB B
8  INR C
9  JMP LOOP
10 STA 8503
11 DCR C
12 MOV A,C
13 LOOP1: STA 8502
14
```

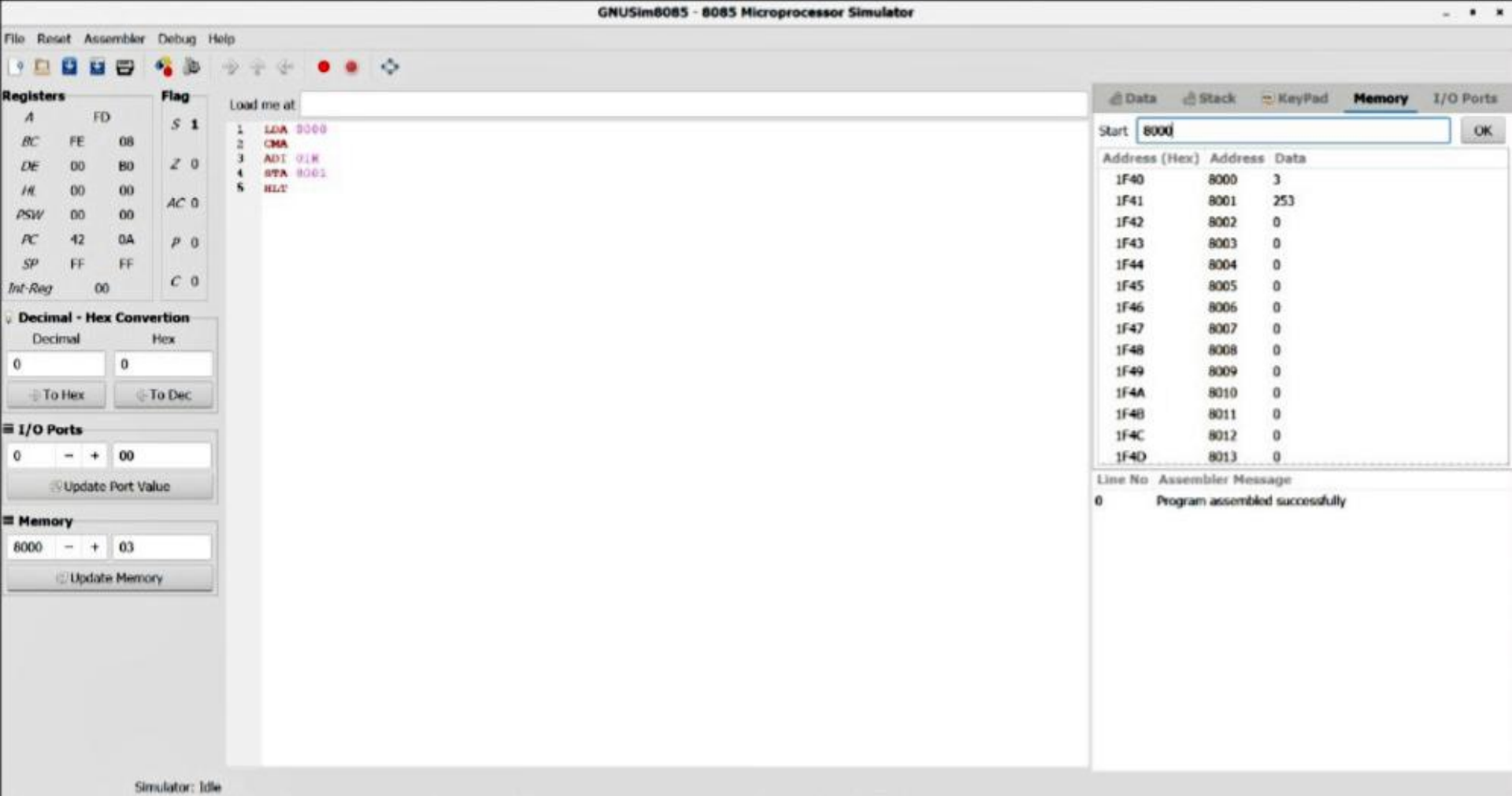
Data Stack KeyPad **Memory** I/O Ports

Start 8501 OK

Address (Hex)	Address	Data
2135	8501	9
2136	8502	3
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

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Registers
A FD
BC FE 08
DE 00 B0
HL 00 00
PSW 00 00
PC 42 0A
SP FF FF
Int-Reg 00
Flag
S 1
Z 0
AC 0
P 0
C 0
Decimal - Hex Convertion
Decimal
Hex
0
0
To Hex
To Dec
I/O Ports
0 - + 00
Update Port Value
Memory
8000 - + 03
Update Memory
Load me at
1 LDA 8000
2 CMA
3 ADI 01H
4 STA 8001
5 HLT
Data
Stack
KeyPad
Memory
I/O Ports
Start 8000
OK
Address (Hex) Address Data
1F40 8000 3
1F41 8001 253
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
Line No Assembler Message
0 Program assembled successfully
Simulator: Idle

File Reset Assembler Debug Help

Registers

A	F9	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	08
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 0
AC 0
P 0
C 0

Decimal - Hex Convertion

Decimal	Hex
0	0

To Hex | To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

8000 − + 06

Update Memory

Load me at

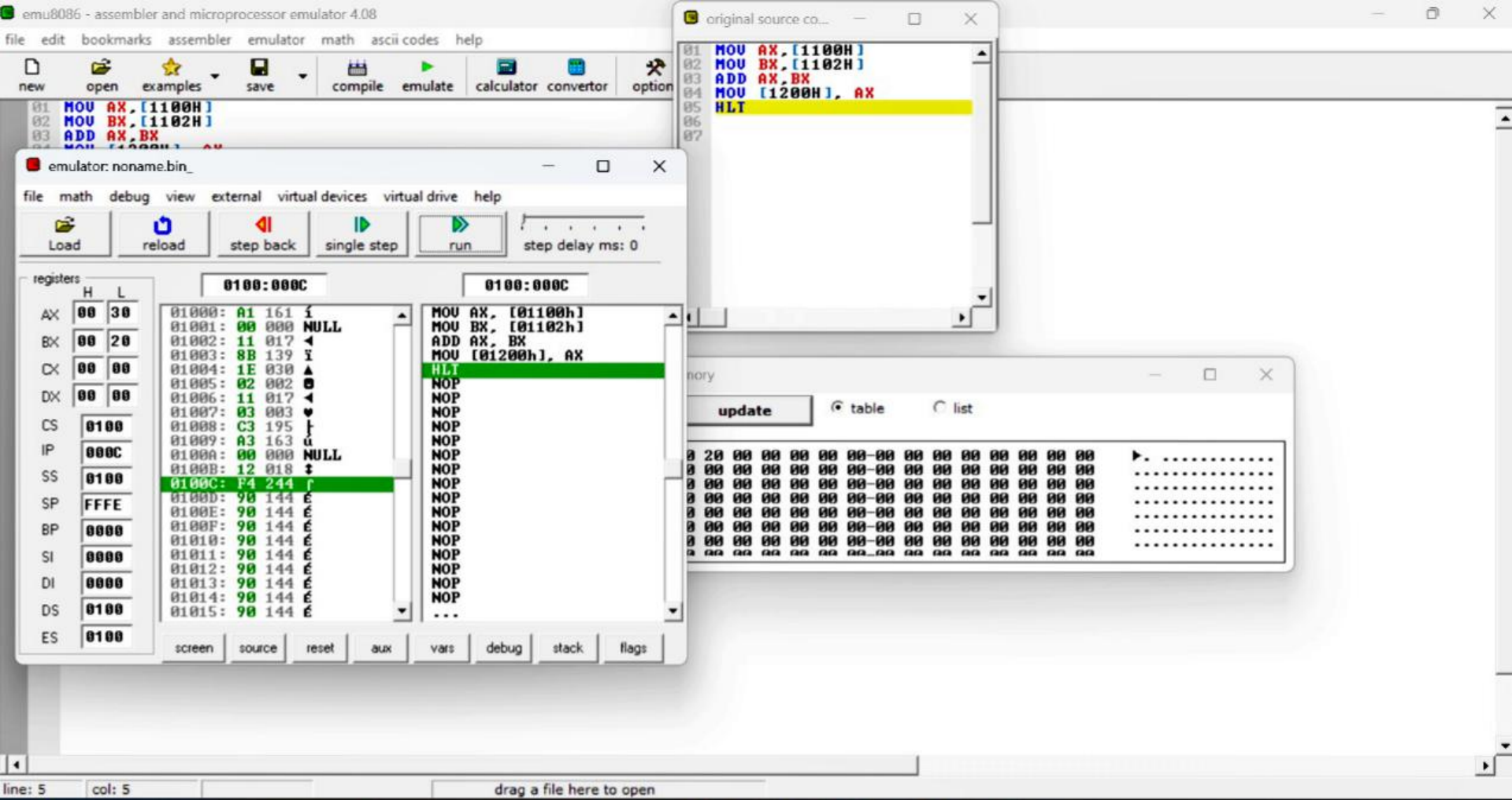
```
1  LDA 8000
2  CMA
3  STA 8001
4  HLT
5
```

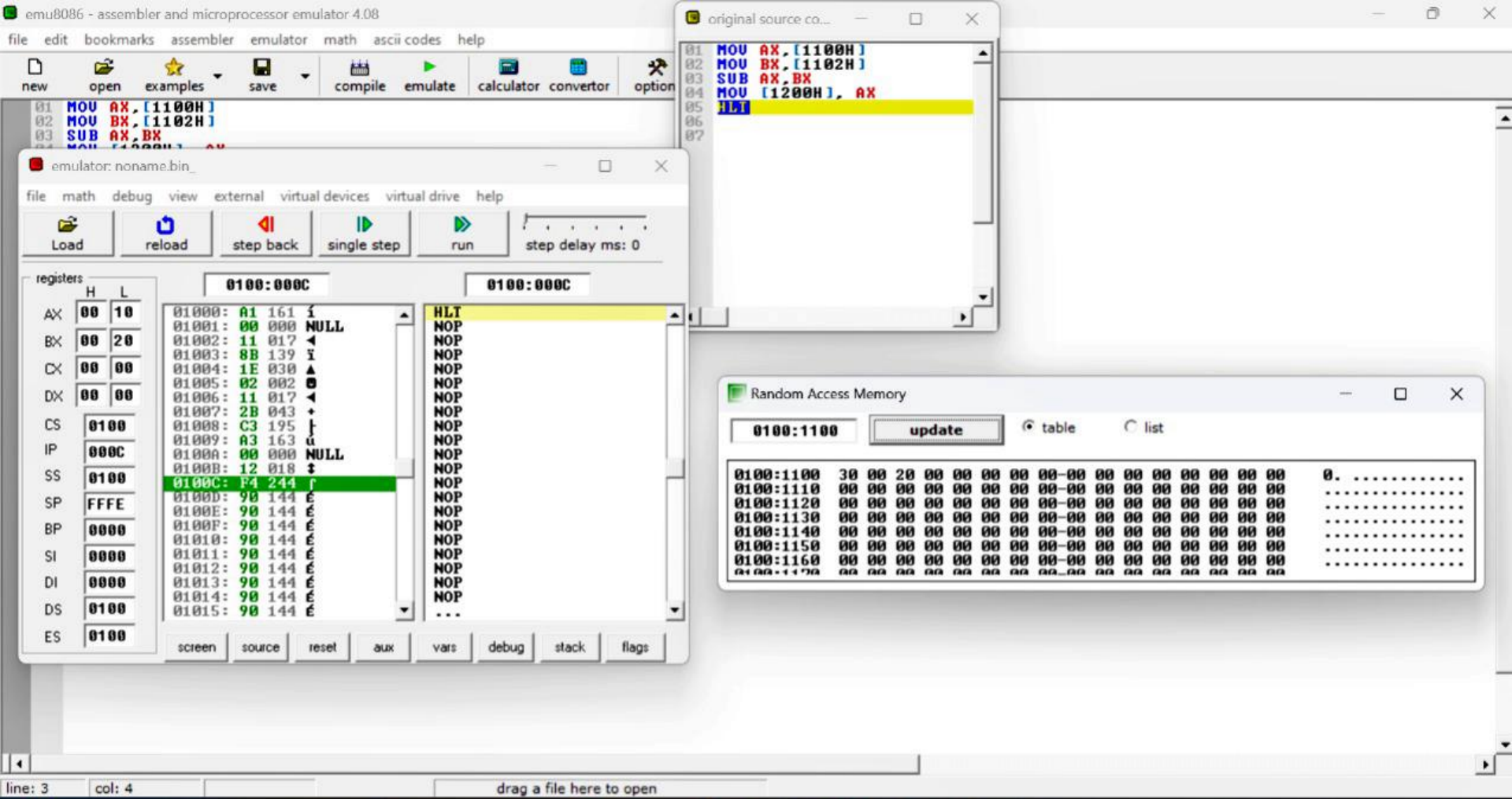
Data | Stack | KeyPad | **Memory** | I/O Ports

Start 8000 OK

Address (Hex)	Address	Data
1F40	8000	6
1F41	8001	249
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

Line No	Assembler Message
0	Program assembled successfully


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor option
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 ADD AX,BX
original source co...
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 ADD AX,BX
04 MOV [1200H], AX
05 HLT
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 30
BX 00 20
CX 00 00
DX 00 00
CS 0100
IP 000C
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:000C
01000: A1 161 í
01001: 00 000 NULL
01002: 11 017 ◄
01003: 8B 139 ï
01004: 1E 030 ▲
01005: 02 002 ☻
01006: 11 017 ◄
01007: 03 003 ♥
01008: C3 195 ├
01009: A3 163 ú
0100A: 00 000 NULL
0100B: 12 018 ↕
0100C: F4 244 ⌠
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
0100:000C
MOV AX, [01100h]
MOV BX, [01102h]
ADD AX, BX
MOV [01200h], AX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags
update
table
list
0 20 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 00
line: 5
col: 5
drag a file here to open


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor option
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 SUB AX,BX
original source co...
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 SUB AX,BX
04 MOV [1200H], AX
05 HLT
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 10
BX 00 20
CX 00 00
DX 00 00
CS 0100
IP 000C
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:000C
01000: A1 161 í
01001: 00 000 NULL
01002: 11 017 ◄
01003: 8B 139 ï
01004: 1E 030 ▲
01005: 02 002 ☻
01006: 11 017 ◄
01007: 2B 043 +
01008: C3 195 ├
01009: A3 163 ú
0100A: 00 000 NULL
0100B: 12 018 ↕
0100C: F4 244 ⌠
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
0100:000C
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags
Random Access Memory
0100:1100 update table list
0100:1100 30 00 20 00 00 00 00 00-00 00 00 00 00 00 00 00 0. ....
0100:1110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
0100:1160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....
line: 3
col: 4
drag a file here to open

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	02	
BC	02	00
DE	00	02
HL	00	04
PSW	00	00
PC	42	16
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 1
AC 0
P 1
C 0

Decimal - Hex Convertion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

2201 − + 02

Update Memory

Load me at

```
1  LDA 2200
2  MOV E,A
3  MVI D,00
4  LDA 2201
5  MOV C,A
6  LXI H,0000
7  BACK: DAD D
8  DCR C
9  JNZ BACK
10 SHLD 2202
11 HLT
```

Data Stack KeyPad **Memory** I/O Ports

Start 2200 OK

Address (Hex)	Address	Data
0898	2200	2
0899	2201	2
089A	2202	4
089B	2203	0
089C	2204	0
089D	2205	0
089E	2206	0
089F	2207	0
08A0	2208	0
08A1	2209	0
08A2	2210	0
08A3	2211	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	00	
BC	03	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	0C
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 1
AC 0
P 1
C 0

Decimal - Hex Convertion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

8000 − + 03

Update Memory

Load me at

```
1  LDA 8000
2  MOV B.,A
3  LDA 8001
4  SUB B
5  STA 8002
6  RST 1
7
```

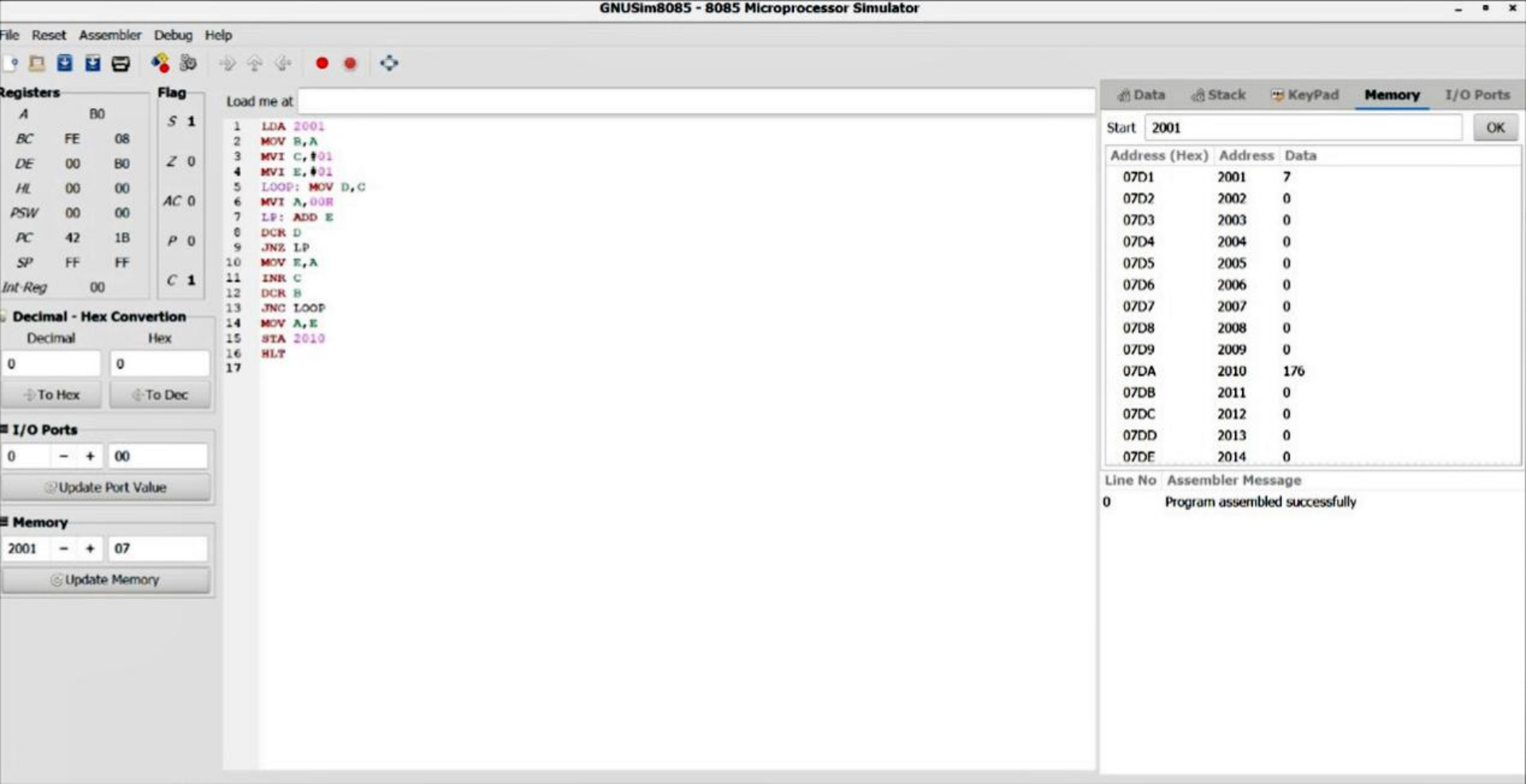
Data Stack KeyPad **Memory** I/O Ports

Start 8000 OK

Address (Hex)	Address	Data
1F40	8000	3
1F41	8001	3
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

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle


GNUSim8085 - 8085 Microprocessor Simulator
File Reset Assembler Debug Help
Registers
A B0
BC FE 08
DE 00 B0
HL 00 00
PSW 00 00
PC 42 1B
SP FF FF
Int-Reg 00
Flag
S 1
Z 0
AC 0
P 0
C 1
Decimal - Hex Convertion
Decimal Hex
0 0
To Hex To Dec
I/O Ports
0 - + 00
Update Port Value
Memory
2001 - + 07
Update Memory
Load me at
1 LDA 2001
2 MOV B,A
3 MVI C,01
4 MVI E,01
5 LOOP: MOV D,C
6 MVI A,00H
7 LP: ADD E
8 DCR D
9 JNZ LP
10 MOV E,A
11 INR C
12 DCR B
13 JNC LOOP
14 MOV A,E
15 STA 2010
16 HLT
17
Data Stack KeyPad Memory I/O Ports
Start 2001 OK
Address (Hex) Address Data
07D1 2001 7
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 176
07DB 2011 0
07DC 2012 0
07DD 2013 0
07DE 2014 0
Line No Assembler Message
0 Program assembled successfully

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers

A	19	
BC	19	00
DE	00	00
HL	1F	74
PSW	00	00
PC	42	1D
SP	FF	FF
Int-Reg	00	

Flag

S 0
Z 1
AC 0
P 1
C 0

Decimal - Hex Convertion

Decimal: 0 — To Hex
Hex: 0 — To Dec

I/O Ports

0 − + 00
Update Port Value

Memory

8055 − + 03
Update Memory

Load me at

```
1  LXI H,8085
2  MOV C,M
3  DCR C
4  LOOP3: MOV D,C
5  LXI H,8051
6  LOOP2: MOV A,M
7  INX H
8  CMP M
9  JNC LOOP1
10 MOV B,M
11 MOV M,A
12 DCX H
13 MOV M,B
14 INX H
15 LOOP1: DCR D
16 JNZ LOOP2
17 DCR C
18 JNZ LOOP3
19 HLT
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	0
1F73	8051	25
1F74	8052	17
1F75	8053	12
1F76	8054	10
1F77	8055	3
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

File Reset Assembler Debug Help

Registers

A	16	
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	12
SP	FF	FF
Int-Reg	00	

Flag

S 1
Z 0
AC 1
P 0
C 0

Decimal - Hex Convertion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0 − + 00

Update Port Value

Memory

8500 − + 42

Update Memory

Load me at

```
1  MVI A,9AH
2  ANI 00H
3  JZ NEG
4  MVI A,22
5  JMP STO
6  NEG: MVI A,11
7  STO: STA 8501
8  HLT
```

Data | Stack | KeyPad | **Memory** | I/O Ports

Start 8500 OK

Address (Hex)	Address	Data
2134	8500	66
2135	8501	22
2136	8502	0
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

Line No	Assembler Message
0	Program assembled successfully