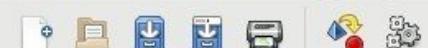


GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	00	S 1
BC	27 00	Z 0
DE	00 00	
HL	00 00	AC 1
PSW	00 00	
PC	42 16	P 1
SP	FF FF	
Int-Reg	00	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0

I/O Ports

0	-	+	00
---	---	---	----

Memory

8050	-	+	27
------	---	---	----

Load me at

```

1 MVI C,00
2 LDA 8050
3 MOV B,A
4 LDA 8051
5 ADD B
6 JNC LOOP
7 INR C
8 LOOP: STA 8052
9 MOV A,C
10 STA 8053
11 HLT

```

Data Stack KeyPad Memory I/O Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	39
1F73	8051	90
1F74	8052	129
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

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	00	S 1
BC	08 00	Z 0
DE	00 00	AC 0
HL	00 00	P 1
PSW	00 00	SP FF FF
PC	42 18	Int-Reg 00
SP	FF FF	C 1

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8050	-	+	8
<input type="button" value="Update Memory"/>			

Load me at

```

1 MVI C,00
2 LDA 8050
3 MOV B,A
4 LDA 8051
5 SUB B
6 JC LOOP
7 CMA
8 INR A
9 INR C
10 LOOP: STA 8052
11 MOV A,C
12 STA 8053
13 HLT

```

Memory

Start 8050

Address (Hex)	Address	Data
1F72	8050	8
1F73	8051	2
1F74	8052	6
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

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	08	S 0
BC	04 00	Z 1
DE	00 00	
HL	00 00	AC 0
PSW	00 00	
PC	42 12	P 1
SP	FF FF	
Int-Reg	00	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8051	-	+	02
<input type="button" value="Update Memory"/>			

Load me at

```

1 LDA 8050
2 MOV B,A
3 LDA 8051
4 MOV C,A
5 XRA A
6 LOOP: ADD B
7 DCR C
8 JNZ LOOP
9 STA 8052
10 HLT

```

Memory

Start 8050

OK

Address (Hex)	Address	Data
1F72	8050	4
1F73	8051	2
1F74	8052	8
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

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



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

Decimal - Hex Conversion	
Decimal	Hex
0	0

Load me at

```

1 LDA 8050
2 MOV B,A
3 LDA 8051
4 MVI C,00
5 LOOP: CMP B
6 JC LOOP1
7 SUB B
8 INR C
9 JMP LOOP
10 LOOP1: STA 8052
11 MOV A,C
12 STA 8053
13 HLT

```

Data	Stack	KeyPad	Memory	I/O Ports
Start 8050				OK
Address (Hex)		Address	Data	
1F72		8050	8	
1F73		8051	2	
1F74		8052	4	
1F75		8053	0	
1F76		8054	0	
1F77		8055	0	
1F78		8056	0	
1F79		8057	0	
1F7A		8058	0	
1F7B		8059	0	

emu8086 - assembler and microprocessor emulator 4.08

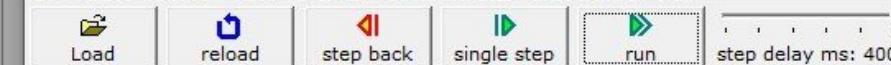
file edit bookmarks assembler emulator math ascii codes help



01 MOU AX, [1100H]
02 MOU BX, [1102H]
03 ADD AX, BX
04 MOU [1200H], AX
05 HLT

emulator: noname.bin_

file math debug view external virtual devices virtual drive help



registers

	H	L
AX	00	07
BX	00	03
CX	00	00
DX	00	00
CS	0100	
IP	0009	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0009

0100:0009

```

01000: A1 161 i
01001: 00 000 NULL
01002: 11 017 ↶
01003: 8B 139 i
01004: 1E 030 ▲
01005: 02 002 Ⓜ
01006: 11 017 ↶
01007: 03 003 ▼
01008: C3 195 ↴
01009: A3 163 u
0100A: 00 000 NULL
0100B: 12 018 ↳
0100C: F4 244 ↸
0100D: 90 144 E
0100E: 90 144 E
0100F: 90 144 E
01010: 90 144 E
01011: 90 144 E
01012: 90 144 E
01013: 90 144 E
01014: 90 144 E
01015: 90 144 E
...
```

screen source reset aux vars debug stack flags

original source code

```

01 MOU AX, [1100H]
02 MOU BX, [1102H]
03 ADD AX, BX
04 MOU [1200H], AX
05 HLT
06
07

```

Random Access Memory

0100:1100

update

table list

0100:1100	04	00	03	00	00	00	00-00	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	00	00	00	00
0100:1120	00	00	00	00	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	00	00	00	00
0100:1140	00	00	00	00	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	00	00	00	00
0100:1160	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00	00	00	00
0100:1170	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00	00	00	00

line: 4 col: 11 drag a file here to open

Type here to search



16:40
11-12-2024

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 MOU AX, [1100H]
 02 MOU BX, [1102H]
 03 SUB AX, BX
 04 MOU [1200H], AX

emulator: noname.bin_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 400

registers

	H	L	
AX	00	06	
BX	00	03	
CX	00	00	
DX	00	00	
CS	0100		
IP	0009		
SS	0100		
SP	FFFE		
BP	0000		
SI	0000		
DI	0000		
DS	0100		
ES	0100		

0100:0009

0100:0009

 01000: A1 161 i
 01001: 00 000 NULL
 01002: 11 017 ↲
 01003: 8B 139 i
 01004: 1E 030 ▲
 01005: 02 002 Ⓜ
 01006: 11 017 ↲
 01007: 2B 043 +
 01008: C3 195 ↴
 01009: A3 163 u
 0100A: 00 000 NULL
 0100B: 12 018 ↳

 0100C: F4 244 ↵
 0100D: 90 144 E
 0100E: 90 144 E
 0100F: 90 144 E
 01010: 90 144 E
 01011: 90 144 E
 01012: 90 144 E
 01013: 90 144 E
 01014: 90 144 E
 01015: 90 144 E
 ...

screen source reset aux vars debug stack flags

line: 4

col: 16

drag a file here to open



Type here to search



ENG

17:12
11-12-2024

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 MOU AX, [1100H]
 02 MOU BX, [1102H]
 03 MUL BX
 04 MOU [1200H], AX
 05 MOU [1202H], DX
 06 HLT

emulator: noname.bin_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 400

 registers H L
 AX 00 08
 BX 00 02
 CX 00 00
 DX 00 00
 CS 0100
 IP 0009
 SS 0100
 SP FFFE
 BP 0000
 SI 0000
 DI 0000
 DS 0100
 ES 0100

 01000: A1 161 i
 01001: 00 000 NULL
 01002: 11 017 ▲
 01003: 8B 139 i
 01004: 1E 030 ▲
 01005: 02 002 ⊖
 01006: 11 017 ▲
 01007: F7 247 ≈
 01008: E3 227 II
 01009: A3 163 ü
 0100A: 00 000 NULL
 0100B: 12 018 ♦
 0100C: 89 137 e
 0100D: 16 022 -
 0100E: 02 002 ⊖
 0100F: 12 018 ♦
 01010: F4 244 ↑
 01011: 90 144 E
 01012: 90 144 E
 01013: 90 144 E
 01014: 90 144 E
 01015: 90 144 E
 ...

screen source reset aux vars debug stack flags

 original source code
 01 MOU AX, [1100H]
 02 MOU BX, [1102H]
 03 MUL BX
 04 MOU [1200H], AX
 05 MOU [1202H], DX
 06 HLT

Random Access Memory

0100:1100 update table list

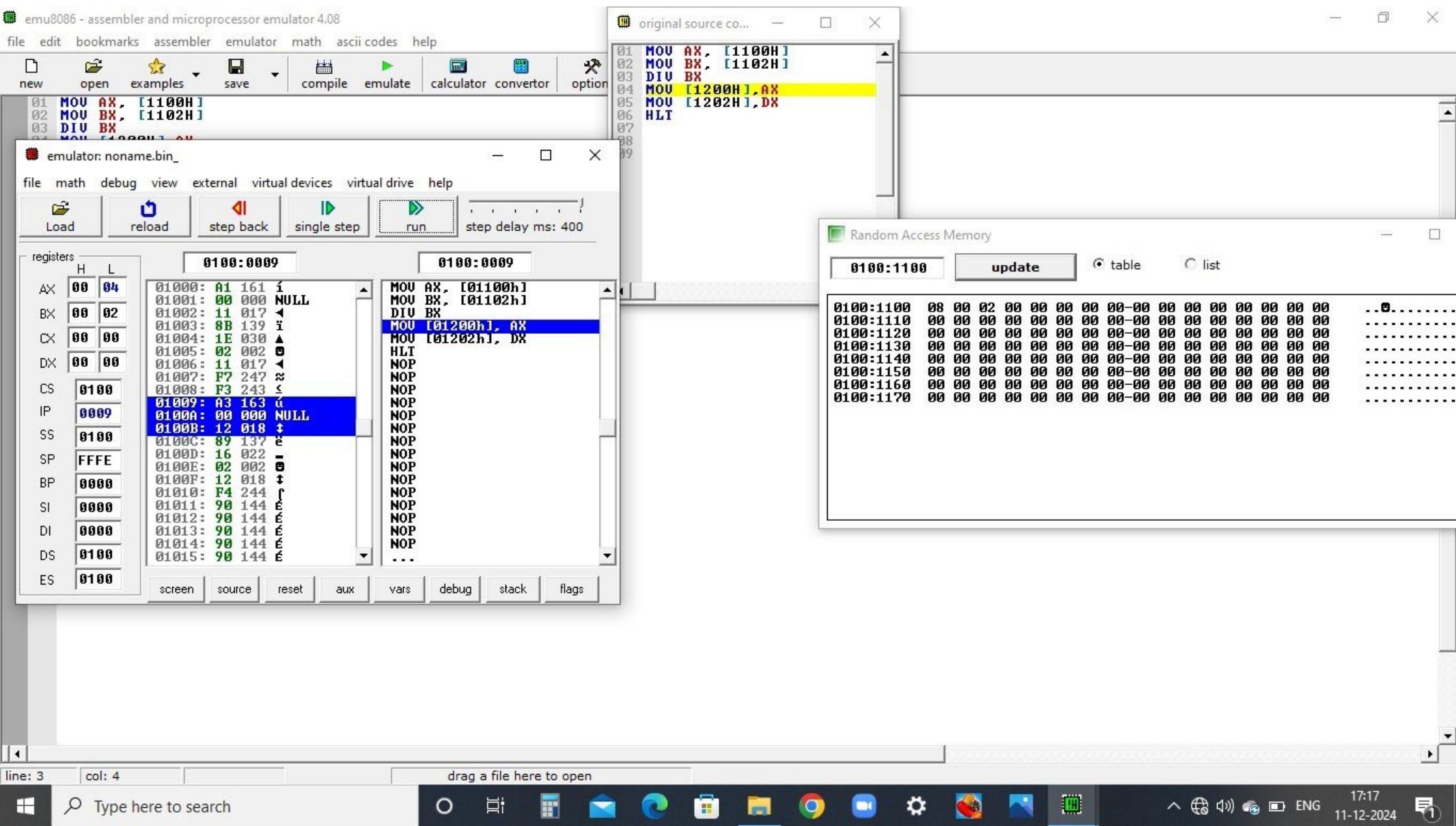
 0100:1100 04 00 02 00 00 00 00-00 00 00 00 00 00 00 00 00
 0100:1110 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
 0100:1120 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
 0100:1130 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
 0100:1140 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00
 0100:1150 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
 0100:1170 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00

line: 4

col: 12

drag a file here to open


 17:15
 11-12-2024 ENG



GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	02	S 0
BC	02 2D	Z 0
DE	FF FB	
HL	1F 73	AC 0
PSW	00 00	
PC	42 10	P 1
SP	FF FF	C 0
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8052	-	+	00
<input type="button" value="Update Memory"/>			

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 8055
8 HLT

```

Memory

Start 8050

OK

Address (Hex)	Address	Data
1F72	8050	2
1F73	8051	8
1F74	8052	0
1F75	8053	0
1F76	8054	0
1F77	8055	2
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		Flag
A	00	S 0
BC	00 04	Z 1
DE	FF FB	
HL	00 14	AC 0
PSW	00 00	
PC	42 2D	P 1
SP	FF FF	
Int-Reg	00	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8001	-	+	05
<input type="button" value="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, OFFH
9 MOV A, B
10 CMA
11 MOV D, A
12 INX D
13 LXI H, 0000
14 NEXT: DAD B
15 SHLD 8010
16 LOOP: DAD D
17 JNC SKIP
18 MOV A, H
19 ORA L
20 JZ EXIT
21 JMP LOOP
22 SKIP: LHLD 8010
23 JMP NEXT
24 EXIT: LHLD 8010
25 HLT

```

Memory

Start OK

Address (Hex)	Address	Data
1F40	8000	4
1F41	8001	5
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	20
1F4B	8011	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	0F	S 0
BC	0F 2D	Z 1
DE	FF FB	
HL	1F 73	AC 0
PSW	00 00	
PC	42 1B	P 1
SP	FF FF	
Int-Reg	00	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8051	-	+	3C
<input type="button" value="Update Memory"/>			

Load me at

```

1 LXI H, 8050
2 MOV A, M
3 INX H
4 MOV B, M
5
6 LOOP: CMP B
7 JZ STORE
8 JC EXG
9 SUB B
10 JMP LOOP
11
12 EXG: MOV C, B
13 MOV B, A
14 MOV A, C
15 JMP LOOP
16
17 STORE: STA 8059
18 HLT

```

Memory

Start

Address (Hex)	Address	Data
1F72	8050	45
1F73	8051	60
1F74	8052	0
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	15
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		Flag
A	03	S 0
BC	03 00	Z 1
DE	00 FB	AC 0
HL	1F 74	
PSW	00 00	P 1
PC	42 1D	
SP	FF FF	C 1
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8055	-	+	00
<input type="button" value="Update Memory"/>			

Load me at

```

1 LXI H, 8050
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 JC 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

```

Memory

Start 8050

Address (Hex)	Address	Data
1F72	8050	4
1F73	8051	3
1F74	8052	5
1F75	8053	8
1F76	8054	9
1F77	8055	0
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	4E	S	0
BC	1A 00	Z	0
DE	00 00		
HL	08 04		
PSW	00 00	AC	0
PC	42 0D	P	1
SP	FF FF		
Int-Reg	00	C	0

Flag

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
12

```

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="→ To Hex"/> <input type="button" value="← To Dec"/>	

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

2050	-	+	0D
<input type="button" value="Update Memory"/>			

Simulator: Idle

Data	Stack	KeyPad	Memory	I/O Ports																																
Start <input type="text" value="2050"/>				OK																																
<table border="1"> <thead> <tr> <th>Address (Hex)</th> <th>Address</th> <th>Data</th> </tr> </thead> <tbody> <tr><td>0802</td><td>2050</td><td>13</td></tr> <tr><td>0803</td><td>2051</td><td>0</td></tr> <tr><td>0804</td><td>2052</td><td>78</td></tr> <tr><td>0805</td><td>2053</td><td>0</td></tr> <tr><td>0806</td><td>2054</td><td>0</td></tr> <tr><td>0807</td><td>2055</td><td>0</td></tr> <tr><td>0808</td><td>2056</td><td>0</td></tr> <tr><td>0809</td><td>2057</td><td>0</td></tr> <tr><td>080A</td><td>2058</td><td>0</td></tr> <tr><td>080B</td><td>2059</td><td>0</td></tr> </tbody> </table>				Address (Hex)	Address	Data	0802	2050	13	0803	2051	0	0804	2052	78	0805	2053	0	0806	2054	0	0807	2055	0	0808	2056	0	0809	2057	0	080A	2058	0	080B	2059	0
Address (Hex)	Address	Data																																		
0802	2050	13																																		
0803	2051	0																																		
0804	2052	78																																		
0805	2053	0																																		
0806	2054	0																																		
0807	2055	0																																		
0808	2056	0																																		
0809	2057	0																																		
080A	2058	0																																		
080B	2059	0																																		
Address (Hex)	Address	Data																																		
0802	2050	13																																		
0803	2051	0																																		
0804	2052	78																																		
0805	2053	0																																		
0806	2054	0																																		
0807	2055	0																																		
0808	2056	0																																		
0809	2057	0																																		
080A	2058	0																																		
080B	2059	0																																		

Line No Assembler Message

0 Program assembled successfully

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	08	S 0
BC	02 2D	Z 0
DE	FF FB	
HL	1F 73	AC 0
PSW	00 00	
PC	42 10	P 1
SP	FF FF	C 0
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8052	-	+	0
<input type="button" value="Update Memory"/>			

Load me at

```

1 LDA 8050
2 MOV B,A
3 LDA 8051
4 CMP B
5 JNC STORE
6 MOV A,B
7 STORE: STA 8055
8 HLT

```

Memory

Start 8050

OK

Address (Hex)	Address	Data
1F72	8050	2
1F73	8051	8
1F74	8052	0
1F75	8053	0
1F76	8054	0
1F77	8055	8
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



Untitled*

- main
- Wiring
- Gates
 - NOT Gate
 - Buffer
 - AND Gate
 - OR Gate
 - NAND Gate
 - NOR Gate
 - XOR Gate
 - XNOR Gate
 - Odd Parity
 - Even Parity
 - Controlled Buffer
 - Controlled Inverter
- Plexers
- Arithmetic

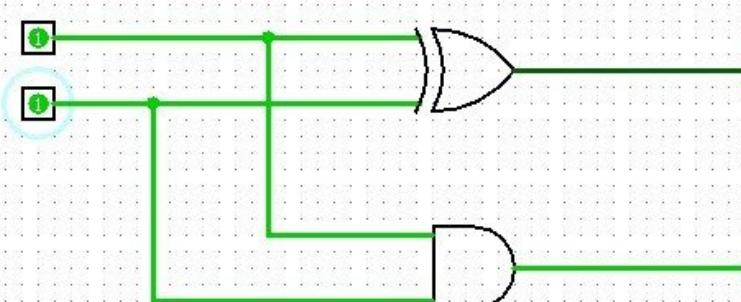
Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

100%



Type here to search



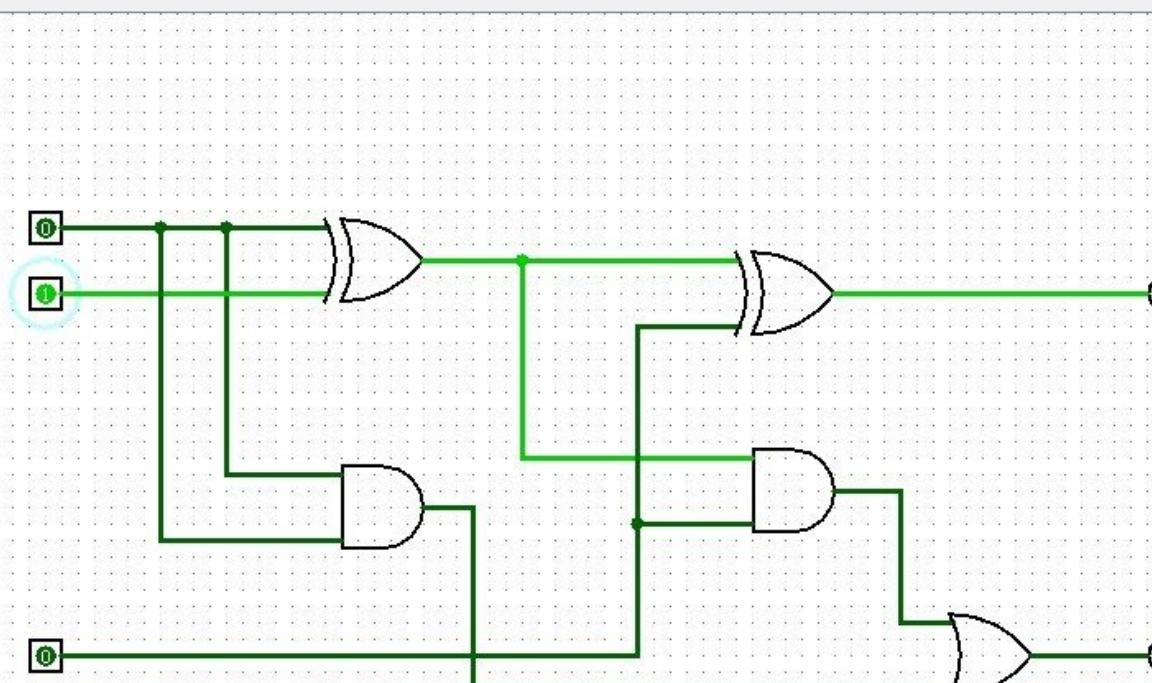


Untitled*

- main
- Wiring
- Gates
 - NOT Gate
 - Buffer
 - AND Gate
 - OR Gate
 - NAND Gate
 - NOR Gate
 - XOR Gate
 - XNOR Gate
 - Odd Parity
 - Even Parity
 - Controlled Buffer
 - Controlled Inverter
- Plexers
- Arithmetic

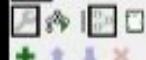
Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12



Logisim: main of Untitled

File Edit Project Simulate Window Help



Untitled*

main

Wiring

Gates

NOT Gate

Buffer

AND Gate

OR Gate

NAND Gate

NOR Gate

XOR Gate

XNOR Gate

Odd Parity

Even Parity

Controlled Buffer

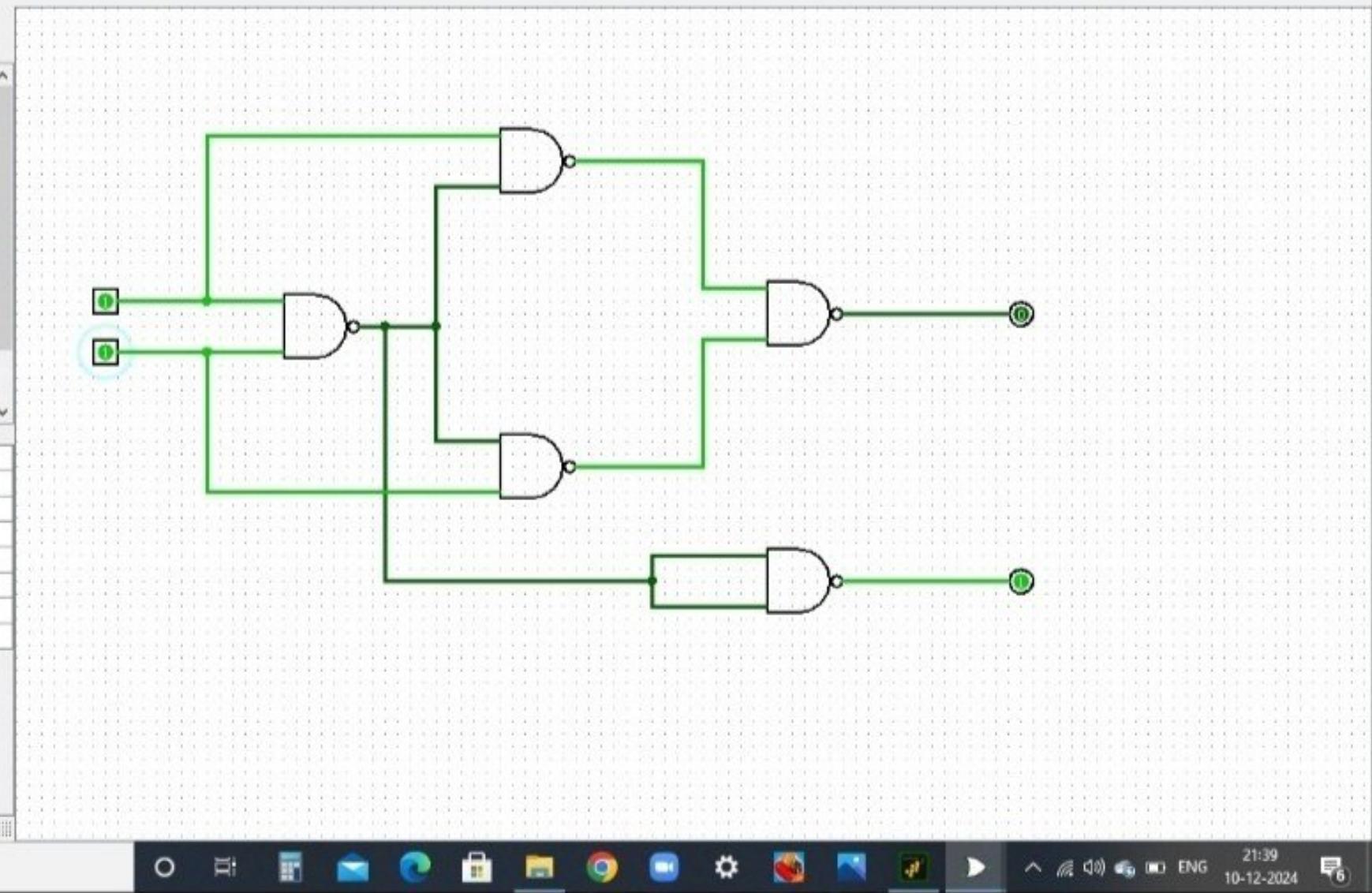
Controlled Inverter

Plexers

Arithmetic

Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12



100%

Type here to search

21:39
10-12-2024

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



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

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

2001	-	+	04
<input type="button" value="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 JNZ LOOP
14 MOV A,E
15 STA 2010
16 HLT

```

Memory

Start

Address (Hex)	Address	Data
07D1	2001	4
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	24
07DB	2011	0
07DC	2012	0

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	22	S 0
BC	22 00	Z 1
DE	00 00	AC 0
HL	1F 7A	
PSW	00 00	P 1
PC	42 17	
SP	FF FF	C 0
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8050	-	+	8
<input type="button" value="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 Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	8
1F73	8051	4
1F74	8052	2
1F75	8053	12
1F76	8054	34
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	34
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		Flag
A	16	S 0
BC	10 00	Z 1
DE	00 00	
HL	80 51	AC 1
PSW	00 00	
PC	42 13	P 1
SP	FF FF	
Int-Reg	00	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8050	-	+	04
<input type="button" value="Update Memory"/>			

Load me at

```

1 LDA 8050      ;Load the accumulator with the content of memory location 8050
2 ANI 01        ;Logical AND operation with accumulator and immediate value 01
3 JZ LOOP1      ;Jump to LOOP1 if the result of the AND operation is zero
4 MVI A,11      ;Move immediate value 11 into the accumulator
5 JMP LOOP2      ;Jump to LOOP2
6 LOOP1: MVI A,22 ;Move immediate value 22 into the accumulator
7 LOOP2: STA 8051 ;Store the accumulator content at memory location 8051
8 HLT           ;Halt the processor

```

Memory

Data	Stack	KeyPad	Memory	I/O Ports																																							
Start 8050																																											
<table border="1"> <thead> <tr> <th>Address (Hex)</th> <th>Address</th> <th>Data</th> </tr> </thead> <tbody> <tr><td>1F72</td><td>8050</td><td>4</td></tr> <tr><td>1F73</td><td>8051</td><td>22</td></tr> <tr><td>1F74</td><td>8052</td><td>0</td></tr> <tr><td>1F75</td><td>8053</td><td>0</td></tr> <tr><td>1F76</td><td>8054</td><td>0</td></tr> <tr><td>1F77</td><td>8055</td><td>0</td></tr> <tr><td>1F78</td><td>8056</td><td>0</td></tr> <tr><td>1F79</td><td>8057</td><td>0</td></tr> <tr><td>1F7A</td><td>8058</td><td>0</td></tr> <tr><td>1F7B</td><td>8059</td><td>0</td></tr> <tr><td>1F7C</td><td>8060</td><td>0</td></tr> <tr><td>1F7D</td><td>8061</td><td>0</td></tr> </tbody> </table>					Address (Hex)	Address	Data	1F72	8050	4	1F73	8051	22	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
Address (Hex)	Address	Data																																									
1F72	8050	4																																									
1F73	8051	22																																									
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																																									

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	16	S 1
BC	10 00	Z 0
DE	00 00	
HL	80 51	AC 1
PSW	00 00	
PC	42 12	P 0
SP	FF FF	C 0
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8050	-	+	09
<input type="button" value="Update Memory"/>			

Load me at

```

1 MVI A, 9AH      ;Move immediate value 9AH into the accumulator
2 ANI 80H          ;Logical AND operation with accumulator and
3 JZ NEG           ;Jump to NEG if the result of the AND operat
4 MVI A, 22         ;Move immediate value 22 into the accumulato
5 JMP STO          ;Jump to STO
6 NEG: MVI A, 11    ;Move immediate value 11 into the accumulato
7 STO: STA 8501     ;Store the accumulator content at memory loc
8 HLT              ;Halt the processor

```

Memory

Start 8050

Address (Hex)	Address	Data
1F72	8050	9
1F73	8051	22
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

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	00	S 0
BC	00 00	Z 1
DE	00 00	AC 0
HL	1F 7A	
PSW	00 00	P 1
PC	42 17	
SP	FF FF	C 0
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8053	-	+	12
<input type="button" value="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 JNC 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 Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	8
1F73	8051	4
1F74	8052	2
1F75	8053	12
1F76	8054	34
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	2
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

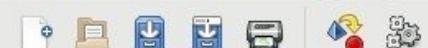
Line No Assembler Message

0 Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	F8	S 1
BC	10 00	Z 0
DE	00 00	AC 1
HL	80 51	P 0
PSW	00 00	SP FF FF
PC	42 08	Int-Reg 00
SP	FF FF	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8000	-	+	07
<input type="button" value="Update Memory"/>			

Load me at

```

1 LDA 8000 ; Load the number from memory location 8000H i
2 CMA ; Complement the number in the accumulator (o
3 STA 8001 ; Store the result (one's complement) in memor
4
5 HLT ; Halt the program
    
```

Data	Stack	KeyPad	Memory	I/O Ports
Start 8000				<input type="button" value="OK"/>
Address (Hex)	Address	Data		
1F40	8000	7		
1F41	8001	248		
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

192411091
Poojasree.B

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	F9	S 1
BC	10 00	Z 0
DE	00 00	AC 0
HL	80 51	P 1
PSW	00 00	C 0
PC	42 0A	
SP	FF FF	
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8000	-	+	07
<input type="button" value="Update Memory"/>			

Load me at

```

1 LDA 8000 ; Load the number from memory location 8000H i
2 CMA ; Complement the accumulator (one's complemen
3 ADI 01H ; Add 1 to the accumulator (to get two's comp
4 STA 8001 ; Store the result (two's complement) in memor
5
6 HLT ; Halt the program

```

Memory

Data	Stack	KeyPad	Memory	I/O Ports																																							
Start 8000				OK																																							
<table border="1"> <thead> <tr> <th>Address (Hex)</th> <th>Address</th> <th>Data</th> </tr> </thead> <tbody> <tr><td>1F40</td><td>8000</td><td>7</td></tr> <tr><td>1F41</td><td>8001</td><td>249</td></tr> <tr><td>1F42</td><td>8002</td><td>0</td></tr> <tr><td>1F43</td><td>8003</td><td>0</td></tr> <tr><td>1F44</td><td>8004</td><td>0</td></tr> <tr><td>1F45</td><td>8005</td><td>0</td></tr> <tr><td>1F46</td><td>8006</td><td>0</td></tr> <tr><td>1F47</td><td>8007</td><td>0</td></tr> <tr><td>1F48</td><td>8008</td><td>0</td></tr> <tr><td>1F49</td><td>8009</td><td>0</td></tr> <tr><td>1F4A</td><td>8010</td><td>0</td></tr> <tr><td>1F4B</td><td>8011</td><td>0</td></tr> </tbody> </table>					Address (Hex)	Address	Data	1F40	8000	7	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
Address (Hex)	Address	Data																																									
1F40	8000	7																																									
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																																									

192411091Poojasree.B

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	09	S 0
BC	09 00	Z 1
DE	00 FB	AC 0
HL	1F 74	P 1
PSW	00 00	SP FF FF
PC	42 1D	Int-Reg 00
SP	FF FF	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8055	-	+	00
<input type="button" value="Update Memory"/>			

Load me at

```

1 LXI H, 8050 ; Load HL with the address 8050H
2 MOV C, M ; Load the number of elements (n) into register C
3 DCR C ; Decrement C (n-1 passes required)
4 LOOP3: MOV D, C ; Copy C to D for inner loop control
5 LXI H, 8051 ; Load HL with the starting address of the array
6 LOOP2: MOV A, M ; Load current element into A
7 INX H ; Increment HL to point to the next element
8 CMP M ; Compare A with the next element
9 JNC LOOP1 ; Jump to LOOP1 if A ≥ next element (no swap)
10 MOV B, M ; Save the next element in B
11 MOV M, A ; Place the larger element at the current position
12 DCX H ; Decrement HL to point to the previous element
13 MOV M, B ; Place the smaller element at the next position
14 INX H ; Increment HL to restore position
15 LOOP1: DCR D ; Decrement D (inner loop counter)
16 JNZ LOOP2 ; Repeat inner loop if D ≠ 0
17 DCR C ; Decrement C (outer loop counter)
18 JNZ LOOP3 ; Repeat outer loop if C ≠ 0
19 HLT

```

Data Stack KeyPad Memory I/O Ports

Start 8050

OK

Address (Hex)	Address	Data
1F72	8050	4
1F73	8051	9
1F74	8052	8
1F75	8053	5
1F76	8054	3
1F77	8055	0
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

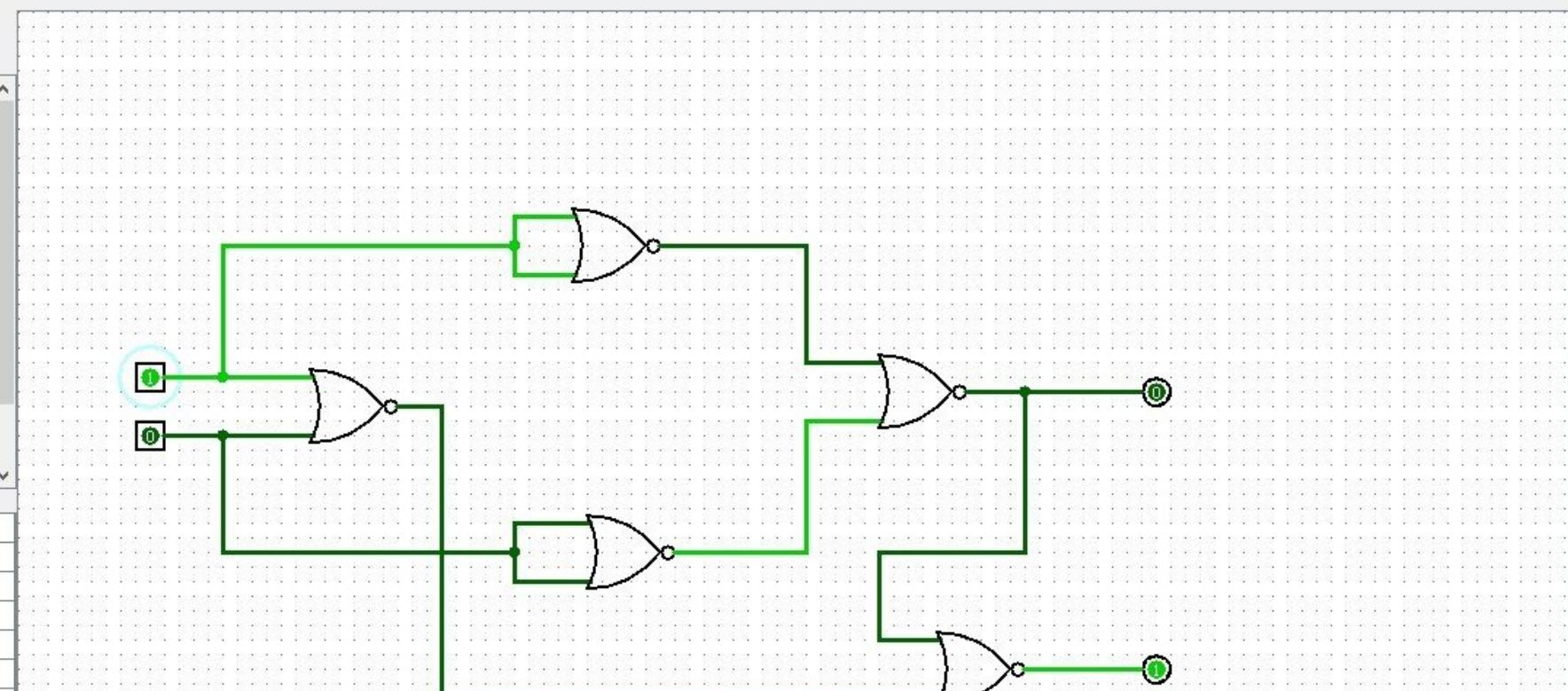


- Untitled*
- main
- Wiring
- Gates
 - NOT Gate
 - Buffer
 - AND Gate
 - OR Gate
 - NAND Gate
 - NOR Gate
 - XOR Gate
 - XNOR Gate
 - Odd Parity
 - Even Parity
 - Controlled Buffer
 - Controlled Inverter
- Plexers
- Arithmetic

Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

100%

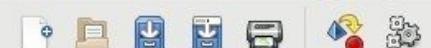


Type here to search



GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers		Flag
A	06	S 0
BC	06 00	Z 0
DE	00 00	AC 0
HL	00 00	P 0
PSW	00 00	
PC	42 0F	
SP	FF FF	C 0
Int-Reg	00	

Decimal - Hex Conversion

Decimal	Hex
0	0
<input type="button" value="To Hex"/>	<input type="button" value="To Dec"/>

I/O Ports

0	-	+	00
<input type="button" value="Update Port Value"/>			

Memory

8052	-	+	0
<input type="button" value="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

```

Memory

Address (Hex)	Address	Data
1F72	8050	3
1F73	8051	6
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

Line No Assembler Message

0 Program assembled successfully

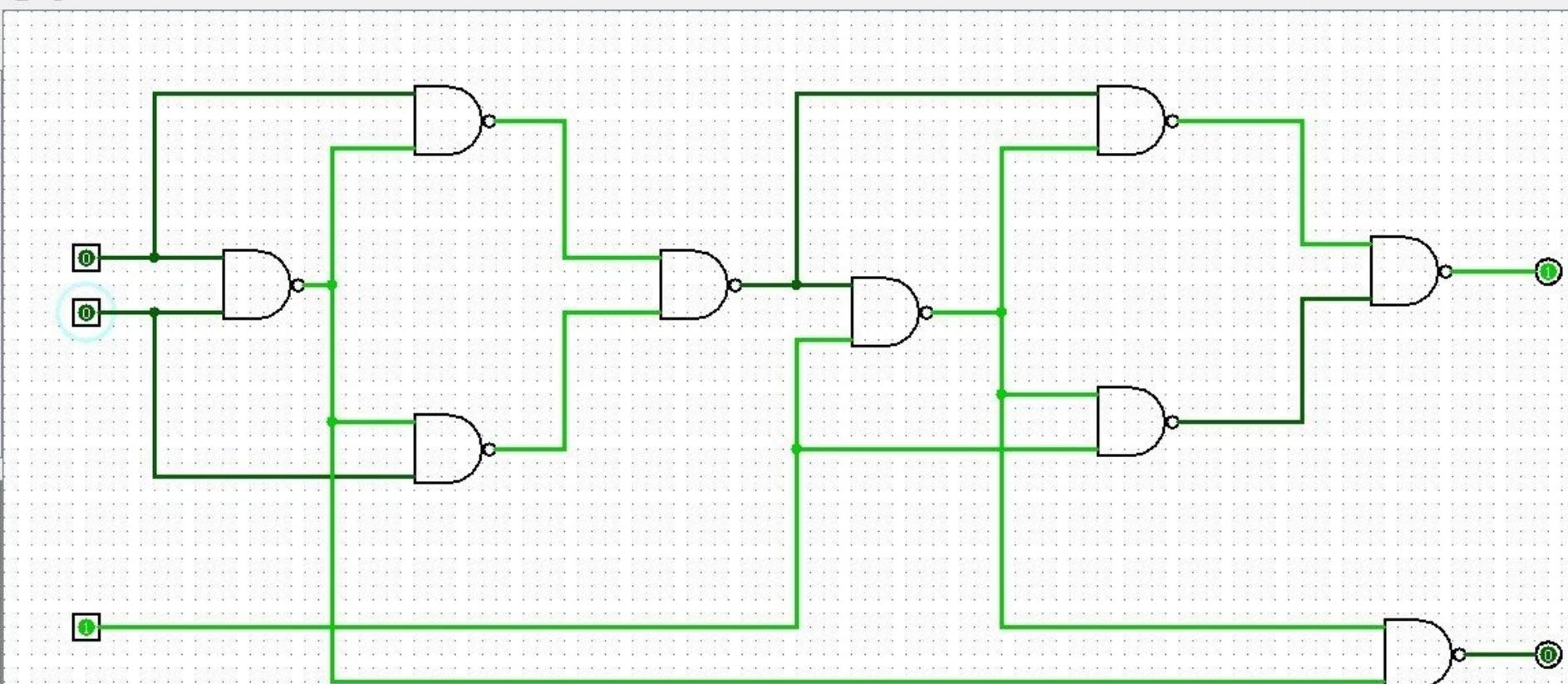
Simulator: Idle



Untitled 2*
main
Wiring
Gates
NOT Gate
Buffer
AND Gate
OR Gate
NAND Gate
NOR Gate
XOR Gate
XNOR Gate
Odd Parity
Even Parity
Controlled Buffer
Controlled Inverter
Divers

Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain...





Untitled*

main

Wiring

Gates

NOT Gate

Buffer

AND Gate

OR Gate

NAND Gate

NOR Gate

XOR Gate

XNOR Gate

Odd Parity

Even Parity

Controlled Buffer

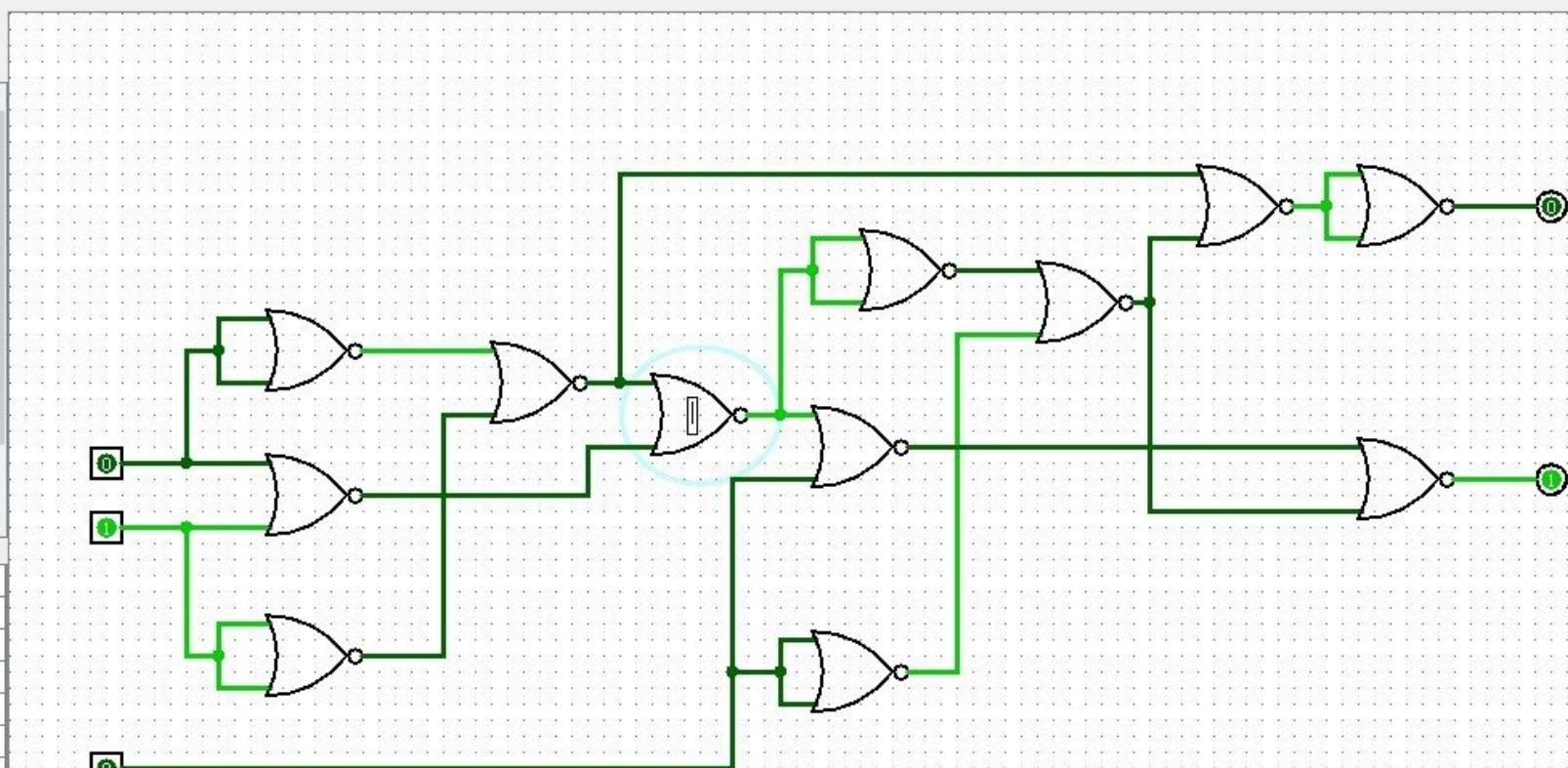
Controlled Inverter

Plexers

Arithmetic

NOR Gate

Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	SansSerif Plain 12
Negate 1 (Top)	No
Negate 2 (Bottom)	No





Untitled*

- main
- Wiring
- Gates
 - NOT Gate
 - Buffer
 - AND Gate
 - OR Gate
 - NAND Gate
 - NOR Gate
 - XOR Gate
 - XNOR Gate
 - Odd Parity
 - Even Parity
 - Controlled Buffer
 - Controlled Inverter
- Plexers
- Arithmetic

Selection: Pin

Facing	West
Output?	Yes
Data Bits	1
Three-state?	Yes
Pull Behavior	Unchanged
Label	
Label Location	East
Label Font	SansSerif Plain 12

100%

