



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

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	-	<input type="text" value="00"/>
<input type="button" value="Update Memory"/>		

Load me at

```
1  LHLD 2050
2  SPHL
3  LHLD 2052
4  XCHG
5  LXI H,0000H
6  LXI B,0000H
7  AGAIN: DAD SP
8  JNC START
9  INX B
10 START: DCK D
11 MOV A,E
12 ORA D
13 JNZ AGAIN
14 SHLD 2054
15 MOV L,C
16 MOV H,B
17 SHLD 2055
18 HLT
19
20
21
22
23
24
25
26
27
28
29
30
```

Start

Address (Hex)	Address	Data
0802	2050	10
0803	2051	0
0804	2052	5
0805	2053	0
0806	2054	50
0807	2055	5
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No	Assembler Message
0	Program assembled successfully

FileResetAssemblerDebugHelp

Registers

A

F9

S 0

BC

00

00

Z 0

DE

00

00

AC 0

HL

00

00

P 0

PSW

00

00

C 0

PC

42

08

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

LD A

8050

2

CMA

3

STA

8053

4

HLT

Data

Stack

Keypad

Memory

I/O Ports

Start

8050

OK

Address (Hex)

Address

Data

1F72

8050

6

1F73

8051

249

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

2 cm of rain

Monday

Search

ENG

IN

11:14

13-11-2025

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers			Flag	
A	54		S	0
BC	44	00	Z	0
DE	78	00	AC	0
HL	08	04	P	0
PSW	00	00	C	0
PC	42	0E		
SP	FF	FF		
Int-Reg	00			

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	<input type="button" value="-"/>	<input type="button" value="+"/>	<input type="text" value="00"/>
<input type="button" value="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 A
7  ADD B
8  INX H
9  ADD M
10 INX H
11 MOV M, A
12 HLT

```

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

Start

Address (Hex)	Address	Data
0802	2050	34
0803	2051	0
0804	2052	84
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

Line No	Assembler Message
0	Program assembled successfully

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers

A	00	00	00	S	0
BC	00	00	00	Z	0
DE	00	00	00	AC	0
HL	00	00	00	P	0
PSW	00	00	00	C	0
PC	00	00	00		
SP	00	00	00		
Int-Reg	00	00	00		

Decimal - Hex Conversion

Decimal: Hex:

I/O Ports

-

Memory

-

Load me at

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11 START: NOP
12 LHLI 2050
13 XCBO
14 LHLI 2052
15 MVI C,00
16 MOV A, C
17 SUB L
18 STA 2054
19 MOV A, D
20 SUB H
21 STA 2055
22 HLT
23 hit
    
```

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

Start 2050

Address (Hex)	Address	Data
0802	2050	2
0803	2051	0
0804	2052	3
0805	2053	0
0806	2054	1
0807	2055	1
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

Simulator: Idle

Heavy rain
Today



Search



ENG
IN

11:43
06-11-2025



Registers			Flag	
A	00		S	0
BC	05	00	Z	1
DE	00	00	AC	0
HL	00	00	P	1
PSW	00	00	C	0
PC	42	15		
SP	FF	FF		
Int-Reg	00			

Decimal - Hex Conversion

Decimal: Hex:

I/O Ports

Memory

Load me at:

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11 LDA 8050
12 MOV B,A
13 LDA 8051
14 ADD B
15 STA 8052
16 hlt

```

Start:

Address (Hex)	Address	Data
1F72	8050	3
1F73	8051	5
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



Registers

A	00	Flag	S	0
BC	00 00		Z	0
DE	00 00		AC	0
HL	00 00		P	0
PSW	00 00		C	0
PC	00 00			
SP	00 00			
Int-Reg	00			

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="2202"/>	-	+	<input type="text" value="8"/>
<input type="button" value="Update Memory"/>			

Load me at

```
1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11 LDA 2200
12 MOV R,A
13 MVI D,00
14 LDA 2201
15 MOV C,A
16 LXI B,0000
17 BACK: DAD D
18 DCR C
19 JNZ BACK
20 SHLD 2202
21 HLT
22 hit
```

Data Stack Keypad Memory I/O Ports

Start 2200

OK

Address (Hex) Address Data

0898	2200	4
0899	2201	2
089A	2202	8
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
08A4	2212	0
08A5	2213	0

Line No. Assembler Message

Simulator: Idle



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

Load me at

```

1 MVI A, 9AH
2 ANI 80H
3 JZ NEG
4 MVI A, 22
5 JMP ST0
6 NEG: MVI A, 11
7 ST0: STA 8501
8 HLT
    
```

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	-	+	<input type="text" value="00"/>
<input type="button" value="Update Memory"/>			

Start

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	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

Line No	Assembler Message
0	Program assembled successfully

GNUSim8085 - 8085 Microprocessor Simulator



File Reset Assembler Debug Help



Registers

A	16	
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	1
AC	1
P	1
C	0

Load me at

```

1  LDA 8050H
2  ANI 01
3  JZ LOOP1
4  MVI A,11
5  JMP LOOP2
6  LOOP1: MVI A,22
7  LOOP2: STA 8051
8  HLT
9

```

Decimal - Hex Conversion

Decimal

Hex

0

0

→ To Hex

← To Dec

I/O Ports

0

-

+

00

Update Port Value

Memory

0

-

+

00

Update Memory

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

Start 8050

OK

Address (Hex)	Address	Data
1F72	8050	20
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

Flag

```

1
2 ;<Program title>
3
4 jmp start
5
6 ;data
7
8
9 ;code
10 start: nop
11 START: NOP
12 LDA 3050
13 MOV B,A
14 LDA 3051
15 ADD B
16 STA 3052
17 LDA 3053
18 MOV B,A
19 LDA 3054
20 ADC B
21 STA 3055
22 HLT
23 hlt

```

Decimal	Hex
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
10	0A
11	0B
12	0C
13	0D
14	0E
15	0F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
31	1F
32	20
33	21
34	22
35	23
36	24
37	25
38	26
39	27
40	28
41	29
42	2A
43	2B
44	2C
45	2D
46	2E
47	2F
48	30
49	31
50	32
51	33
52	34
53	35
54	36
55	37
56	38
57	39
58	3A
59	3B
60	3C
61	3D
62	3E
63	3F
64	40
65	41
66	42
67	43
68	44
69	45
70	46
71	47
72	48
73	49
74	4A
75	4B
76	4C
77	4D
78	4E
79	4F
80	50
81	51
82	52
83	53
84	54
85	55
86	56
87	57
88	58
89	59
90	5A
91	5B
92	5C
93	5D
94	5E
95	5F
96	60
97	61
98	62
99	63
100	64
101	65
102	66
103	67
104	68
105	69
106	6A
107	6B
108	6C
109	6D
110	6E
111	6F
112	70
113	71
114	72
115	73
116	74
117	75
118	76
119	77
120	78
121	79
122	7A
123	7B
124	7C
125	7D
126	7E
127	7F
128	80
129	81
130	82
131	83
132	84
133	85
134	86
135	87
136	88
137	89
138	8A
139	8B
140	8C
141	8D
142	8E
143	8F
144	90
145	91
146	92
147	93
148	94
149	95
150	96
151	97
152	98
153	99
154	9A
155	9B
156	9C
157	9D
158	9E
159	9F
160	A0
161	A1
162	A2
163	A3
164	A4
165	A5
166	A6
167	A7
168	A8
169	A9
170	AA
171	AB
172	AC
173	AD
17	

0

④ To Dec

0	-	+	0
---	---	---	---

⑤ Update Port Value

3055	-	+	10
------	---	---	----

Update Memory

 **Rain warning**
in effect

```

01 MOV AX, [1100H]
02 MOV BX, [1102H]
03 DIV BX
04 MOV [1200H], AX
05 MOV [1202H], DX
06 HLT

```

The screenshot shows the x86-64 emulator window titled "emulator: noname.bin_". The menu bar includes file, math, debug, view, external, virtual devices, virtual drive, and help. Below the menu are several control buttons: Load, reload, step back, single step, run, and a slider for step delay ms: 0.

Registers:

	H	L
AX	00	01
ECX	00	10
EDX	00	00
EAX	00	00
CS	0100	
IP	0010	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

Memory Dump (Address 0100:0010):

Address	Hex	ASCII
010000	A1 16 1	i
010001	00 000	NUL
010002	11 017	<
010003	0B 139	I
010004	1E 030	A
010005	02 002	B
010006	11 017	<
010007	P7 247	s
010008	F3 243	f
010009	A3 163	a
01000A	00 000	NUL
01000B	12 018	t
01000C	89 137	e
01000D	16 022	n
01000E	02 002	B
01000F	12 018	t
010010	P4 244	r
010011	90 144	E
010012	90 144	E
010013	90 144	E
010014	90 144	E
010015	90 144	E

Instruction Stream:

```

MOV AX, [01000h]
MOV BX, [01002h]
DIV BX
MOV [01000h], AX
MOV [01002h], BX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

```

At the bottom, there are tabs for screen, source, reset, aux, vars, debug, stack, and flags.

[illegible]

GNUSim8085 - 8085 Microprocessor Simulator



File Reset Assembler Debug Help



Registers

A	00	
BC	05	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	15
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	1
AC	0
P	1
C	0

Decimal - Hex Conversion

Decimal

Hex

0 0

→ To Hex

← To Dec

I/O Ports

0 - + 00

Update Port Value

Memory

0 - + 00

Update Memory

Load me at

```

1  MVI D, 00
2      MVI A, 00
3      LXI H, 4150      ;loading first
4      MOV B,M
5      INX H
6      MOV C,M
7  LOOP:  ADD B
8          JNC NEXT
9          INR D
10 NEXT:  DCR C
11          JNZ LOOP
12          STA 4152
13  HLT
    
```

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

Start 4150

OK

Address (Hex)	Address	Data
1036	4150	8
1037	4151	6
1038	4152	48
1039	4153	0
103A	4154	0
103B	4155	0
103C	4156	0
103D	4157	0
103E	4158	0
103F	4159	0
1040	4160	0
1041	4161	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	0
DE	2C	04	AC	0
HL	08	08	P	0
PSW	00	00	C	0
PC	07	DD		
SP	08	08		
Int-Reg	00			

Decimal - Hex Conversion

Decimal: Hex:

I/O Ports

Memory

Load me at

```

1  LHLD 2050
2  SPHL
3  LHLD 2052
4  XCHG
5  LXI H,0000H
6  LXI B,0000H
7  DAD SP
8  JNC 2013
9  INX B
10 DCX D
11 MOV A,E
12 ORA D
13 JNZ 2008
14 SHLD 2054
15 MOV L,C
16 MOV H,B
17 SHLD 2056
18 hlt
  
```

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

Start

Address (Hex)	Address	Data
0802	2050	11
0803	2051	11
0804	2052	4
0805	2053	44
0806	2054	44
0807	2055	0
0808	2056	0
0809	2057	0
080A	2058	0
080B	2059	0
080C	2060	0
080D	2061	0

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle

Registers

A	01
BC	02 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	0
C	0

Decimal - Hex Conversion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0	-	+	00
Update Port Value			

Memory

0	-	+	00
Update Memory			

Load me at:

1 LDA 8050

2 MOV B,A

3 LDA 8051

4 SUB B

5 STA 8052

6 HLT

Data Stack Keypad Memory I/O Ports

Start 8050 OK

Address (Hex)	Address	Data
1F72	8050	2
1F73	8051	3
1F74	8052	1
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



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

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	<input type="button" value="-"/>	<input type="button" value="+"/>	<input type="text" value="04"/>
<input type="button" value="Update Memory"/>			

Load me at

```

1  Start: 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
11  REC:  SUB B
12      CMP B
13      JZ OP
14      JNC REC
15      MOV C, A
16      MOV A, B
17      MOV B, C
18      JMP REC
19
20  OP:   OUT 02H
21      HLT
  
```

Start

Address (Hex)	Address	Data
0000	0	4
0001	1	24
0002	2	4
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

Line No	Assembler Message
0	Program assembled successfully

A	00	S	0
BC	00 00	Z	0
DE	04 03	AC	0
HL	08 08	P	0
PSW	00 00	C	0
PC	42 0C		
SP	FF FF		
Int-Reg	00		

```

1  LHALD 2050
2  XCHG
3  LHALD 2052
4  DAD D
5  SHLD 2054
6  HLT

```

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	-	+	<input type="text" value="00"/>
<input type="button" value="Update Memory"/>			

Start

Address (Hex)	Address	Data
0802	2050	3
0803	2051	4
0804	2052	5
0805	2053	4
0806	2054	8
0807	2055	8
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

```
01 MOV AX,[1100H]
02 MOV BX,[1102H]
03 DIV BX
04 MOV [1200H],AX
05 MOV [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: 0

registers

	H	L
AX	00	02
BX	00	03
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

01000: A1 161 i	MOV AX, [01100h]
01001: 00 000 NULL	MOV BX, [01102h]
01002: 11 017	DIV BX
01003: 0B 139 i	MOV [01200h], AX
01004: 1E 030	MOV [01202h], DX
01005: 02 002	HLT
01006: 11 017	NOP
01007: F7 247	NOP
01008: F3 243 i	NOP
01009: A3 163	NOP
0100A: 00 000 NULL	NOP
0100B: 12 018	NOP
0100C: 89 137 E	NOP
0100D: 16 022	NOP
0100E: 02 002	NOP
0100F: 12 018	NOP
01010: F4 244 i	NOP
01011: 90 144 E	NOP
01012: 90 144 E	NOP
01013: 90 144 E	NOP
01014: 90 144 E	NOP
01015: 90 144 E	NOP

0100:0010

screen source reset aux vars debug stack flags

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
```

Random Access Memory

0100:1100 update table list

0100:1100	06 00 03 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

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers			Flag	
A	63		S	1
BC	63	00	Z	0
DE	00	00	AC	0
HL	00	00	P	0
PSW	00	00	C	1
PC	42	10		
SP	FF	FF		
Int-Reg	00			

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	<input type="button" value="-"/>	<input type="button" value="+"/>	<input type="text" value="10"/>
<input type="button" value="Update Memory"/>			

Load me at

```

1  LDA 2050
2  MOV B,A
3  LDA 2051
4  CMP B
5  JNC STORE
6  MOV A,B
7  STORE: STA 2052
8  HLT

```

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

Start

Address (Hex)	Address	Data
0802	2050	99
0803	2051	22
0804	2052	99
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

Line No	Assembler Message
0	Program assembled successfully

Simulator: Idle



Registers			Flag		
A	B4		S	0	
BC	14	09	Z	1	
DE	B4	00	AC	0	
HL	00	00	P	1	
PSW	00	00	C	0	
PC	42	17			
SP	FF	FF			
Int-Reg	00				

Decimal - Hex Conversion

Decimal: Hex:

I/O Ports

2

Memory

0

```

1 ; Program to compute the LCM of three 8-bit numbers
2
3
4 START: IN 00H          ; Input Integer 1 from I/O port
5         MOV C, A       ; Move A → C
6         MOV D, A       ; Move A → D
7         IN 01H         ; Input Integer 2 from I/O port
8         MOV B, A       ; Move A → B
9         MOV A, C       ; Move C → A
10
11        Call LCM        ; Call the LCM subroutine
12
13        IN 02H         ; Input Integer 3 from I/O port
14        MOV C, A       ; Move A → C
15        MOV D, A       ; Move A → D
16        IN 03H         ; Input old LCM from I/O port
17        MOV B, A       ; Move A → B
18        MOV A, C       ; Move C → A
19
20        Call LCM        ; Call the LCM subroutine
21
22        HLT            ; Terminate
23
24 LCM:    SUB B          ; Store A - B → A
25        JNC LCM        ; Jump to LCM if not carry (A > 0)
26        ADD B          ; Store A + B → A
27        CPI 00         ; Compare A with 00H
28        JZ OP          ; Jump to OP if zero (A = 00H)
29        MOV A, D       ; Move D → A
30        ADD C          ; Store A+C → A
31        MOV D, A       ; Move A → D
32        JMP LCM        ; Unconditional Jump to LCM

```

I/O Ports		
Start <input type="text"/>		
Address (Hex)	Address	Data
00	0	4
01	1	5
02	2	9
03	3	180
04	4	0
05	5	0
06	6	0
07	7	0
08	8	0
09	9	0
0A	10	0
0B	11	0
0C	12	0
0D	13	0
0E	14	0
0F	15	0
10	16	0
11	17	0

Line No Assembler Message

0 Program assembled successfully



Registers			Flag
A	54		S 0
BC	44	00	
DE	04	03	Z 0
HL	08	04	
PSW	00	00	AC 0
PC	42	0E	P 0
SP	FF	FF	
Int-Reg	00		C 0

Decimal - Hex Conversion

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

I/O Ports

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

Memory

<input type="text" value="0"/>	-	<input type="text" value="00"/>
<input type="button" value="Update Memory"/>		

Load me at

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

Start

Address (Hex)	Address	Data
0802	2050	34
0803	2051	0
0804	2052	84
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

GNUSim8085 - 8085 Microprocessor Simulator

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 Conversion

Decimal: Hex:

I/O Ports

-

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

Address (Hex)	Address	Data
1F72	8050	15
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

Engage, main of Untitled

File Edit Project Simulate Window Help

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

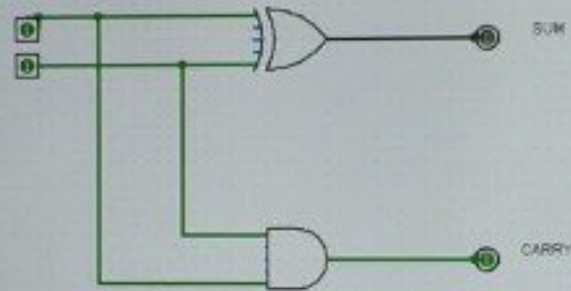
Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

Alt selection and add wires (Ctrl-Z)

HALF ADDER



Label

Text	
Font	SanSerif Plain 12
Horizontal Alignment	Center
Vertical Alignment	Base



Search



