

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help



Registers			Flag
A	1E		S 0
BC	0A 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		

Load me at

```
1 LDA 8050
2 MOV B,A
3 LDA 8051
4 ADD B
5 STA 8052
6 RST
```

Decimal - Hex Conversion

Decimal Hex
0 0
To Hex To Dec

I/O Ports

0 - + 00
Update Port Value

Memory

8051 - + 14
Update Memory

Data Stack KeyPad Memory I/O Ports

Start 8050

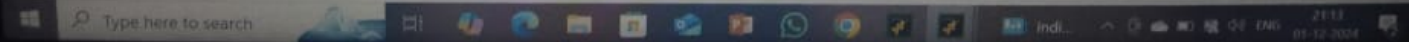
Address (Hex)	Address	Data
1F72	8050	10
1F73	8051	20
1F74	8052	30
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

Line No Assembler Message

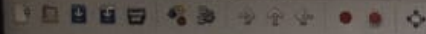
0 Program assembled successfully

8 bit add I

Simulator: Idle



File Reset Assembler Debug Help



Registers

A	01	S	0
BC	26 01	Z	0
DE	00 00	AC	0
HL	00 00	P	0
PSW	00 00	C	1
PC	42 18		
SP	FF FF		
Int-Reg	00		

Decimal - Hex Conversion

Decimal	Hex
35	23
To Hex	To Dec

I/O Ports

0	-	+	00
Update Port Value			

Memory

8051	-	+	23
Update Memory			

Load me at

```

1 MVI C, 30
2 LDA 8050
3 MOV B, A
4 LDA 8051
5 SUB B
6 JNC LOOP
7 CMA
8 INR A
9 INR C
10 LOOP: STA 8052
11 MOV A, C
12 STA 8053
13 HLT
    
```

Data Stack KeyPad Memory I/O Port

Start 8050

Address (Hex)	Address	Data
1F72	8050	38
1F73	8051	35
1F74	8052	3
1F75	8053	1
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

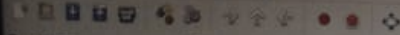
0 Program assembled successfully

Simulator: Idle

8 bit sub^I

Type here to search





Registers	Flag
A B4	S 0
BC 2D 00	Z 1
DE 00 00	AC 0
HL 00 00	P 1
PSW 00 00	C 0
PC 42 12	
SP FF FF	
Int-Reg 00	

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
    
```

Decimal - Hex Conversion

Decimal	Hex
26	1A

To Hex To Dec

I/O Ports

0 - + 00

Update Port Value

Memory

8051 - + 04

Update Memory

Data Stack KeyPad Memory I/O Ports

Start 8050

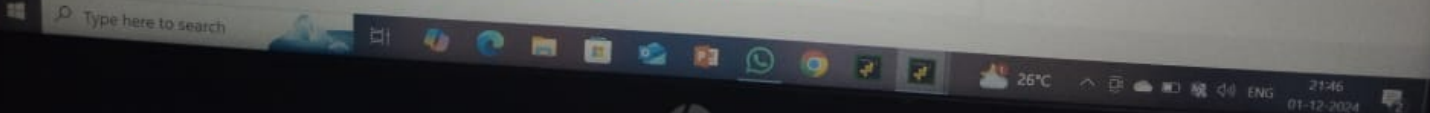
Address (Hex)	Address	Data
1F72	8050	45
1F73	8051	4
1F74	8052	180
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

Line No Assembler Message

0 Program assembled successfully

8 bit mul

Simulator: Idle



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

Decimal - Hex Conversion

Decimal Hex

0 0

To Hex

To Dec

I/O Ports

0 - + 00

Update Port Value

Memory

8500 - + 09

Update Memory

Load me at

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

Start 8500

Address (Hex)	Address	Data
2134	8500	9
2135	8501	3
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

Line No Assembler Message

0 Program assembled successfully

8 bit div

Simulator: Idle




```
01 MOV AX, [1100H]
02 MOV BX, [1102H]
03 ADD AX, BX
04 MOV [1200H], AX
05 HLT
06
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	7A
BX	00	67
CX	00	00
DX	00	00
CS	0100	
IP	000C	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0003

```
0100: 01 161 f
0100: 00 000 NULL
0100: 11 017
0100: 0B 139 f
0100: 1E 030
0100: 02 002
0100: 11 017
0100: 03 003
0100: C3 195
0100: A3 163
0100: 00 000 NULL
0100: 12 018
0100: F4 244 f
0100: 90 144 E
0100: 90 144 E
0100: 90 144 E
0100: 90 144 E
0100: 90 144 E
0100: 90 144 E
0100: 90 144 E
0100: 90 144 E
```

0100:0003

```
MOV BX, [01102h]
ADD AX, BX
MOV [01200h], AX
HLT
NOP
```

Random Access Memory

0100:1100

update

table

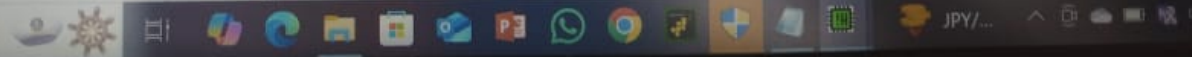
list

```
0100:1100 13 00 67 00 00 00 00 00-00 00 00 00 00 00 00 00 !!.g.....
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 .....
```

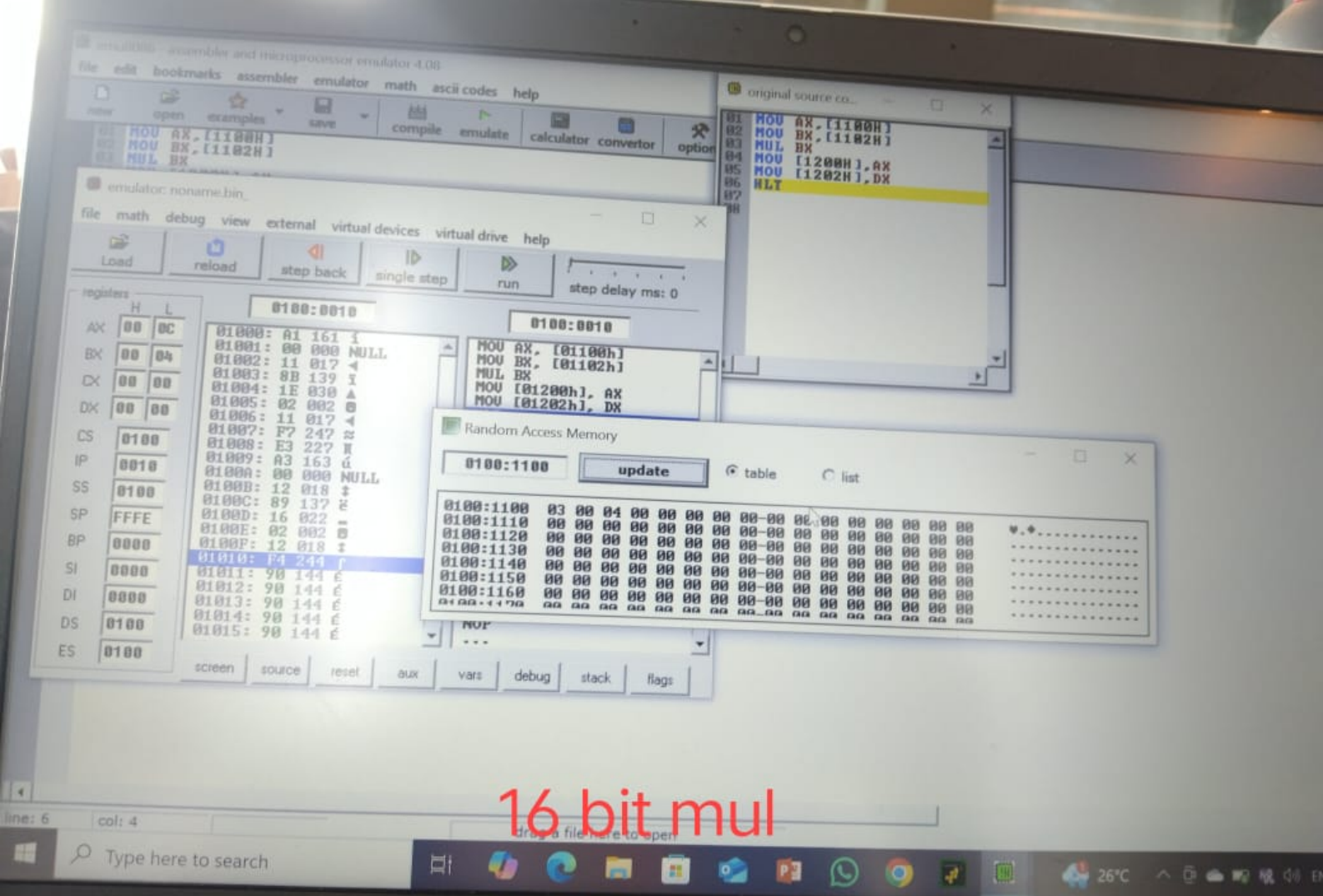
16 bit add

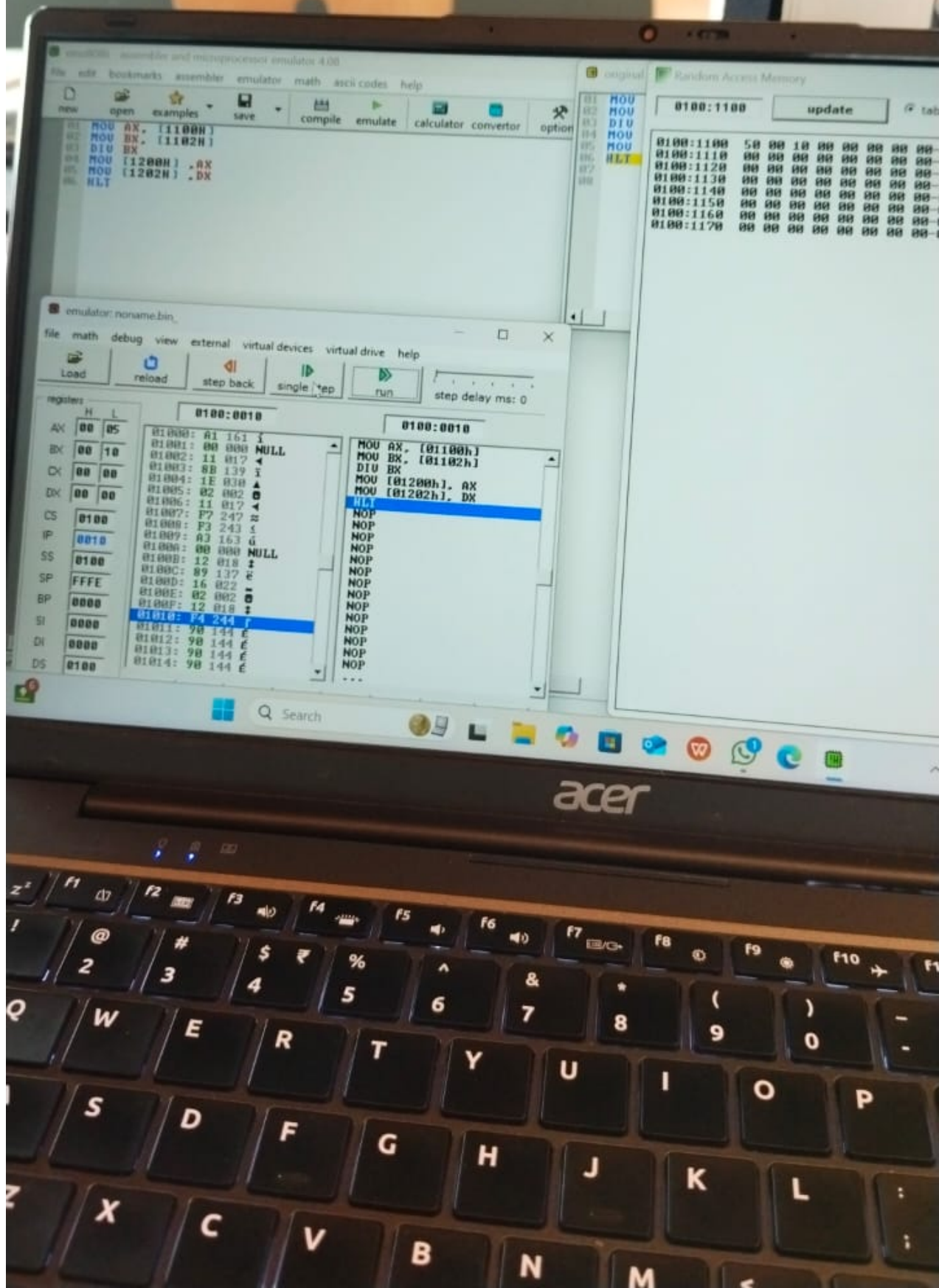
line: 5 col: 4 drag a file here to open

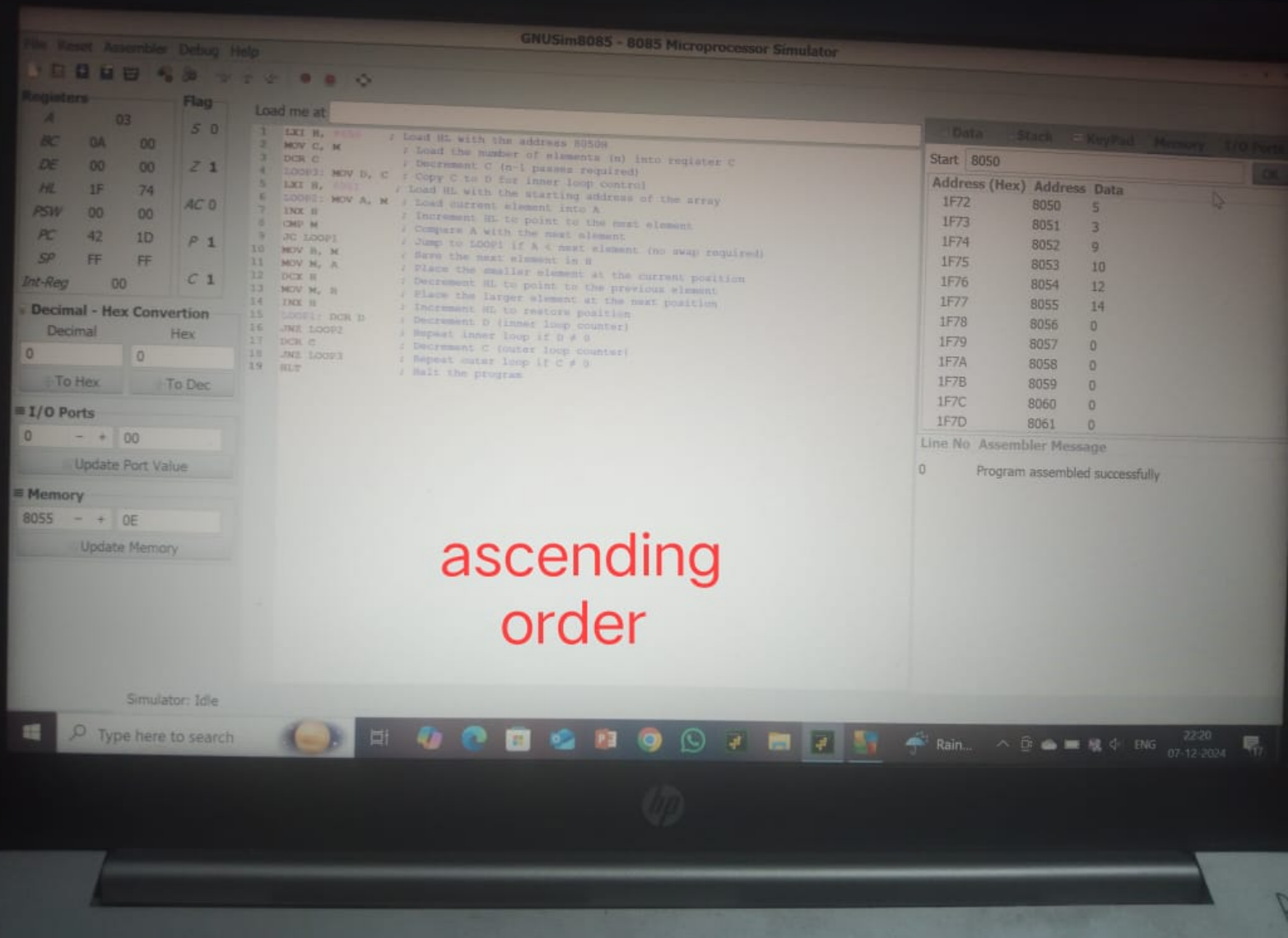
Type here to search



16 bit sub







File View Assembler Debug Help

Register

Register	Value	Flag
A	00	S 0
BC	0A 00	Z 1
DE	00 00	
HL	1F 74	AC 0
PSW	00 00	
PC	42 1D	P 1
SP	FF FF	C 0

Decimal - Hex Conversion

Decimal	Hex
0	0

To Hex To Dec

I/O Ports

0 - + 00

Update Port Value

Memory

8055 - + 06

Update Memory

Load me at

```
1. LDI B, 0005 ; Load BL with the address 8050H
2. MOV C, B ; Load the number of elements (n) into register C
3. DCR C ; Decrement C (n-1 passes required)
4. LOOP1: MOV D, C ; Copy C to D for inner loop control
5. LDI B, 0001 ; Load BL with the starting address of the array
6. LOOP2: MOV A, M ; Load current element into A
7. INR B ; Increment BL 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 required)
10. MOV B, M ; Save the next element in B
11. MOV M, A ; Place the larger element at the current position
12. DCR B ; Decrement BL to point to the previous element
13. MOV M, B ; Place the smaller element at the next position
14. INR B ; Increment BL to restore position
15. LOOP1: DCR D ; Decrement D (inner loop counter)
16. JNE LOOP2 ; Repeat inner loop if D != 0
17. DCR C ; Decrement C (outer loop counter)
18. JNE LOOP1 ; Repeat outer loop if C != 0
19. HLT ; Halt the program
```

Start 8050

Address (Hex)	Address	Data
1F72	8050	5
1F73	8051	13
1F74	8052	12
1F75	8053	10
1F76	8054	9
1F77	8055	6
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

descending
order

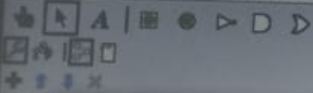
Simulator: Idle

Type here to search



22:23

07-12-2024



Combinational Analysis

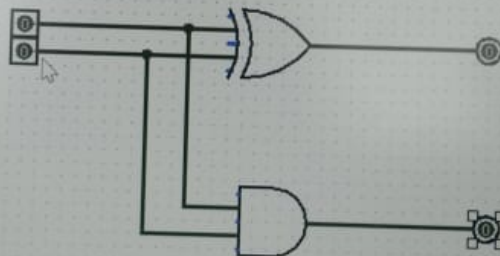
File Edit Project Simulate Window Help

Inputs Outputs Table Expression Minimized

a	b	x	y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Build Circuit

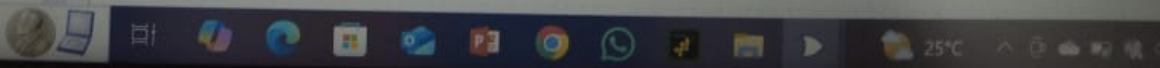
Facing	West
Output?	Yes
Data Bits	1
Three-state?	Yes
Pull Behavior	Unchanged
Label	
Label Location	East
Label Font	SansSerif Plain 12



half adder

100%

Type here to search



File Edit Project Simulate Window Help

Combinational Analysis

Inputs Outputs Table Expression Minimized

a	b	x	y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

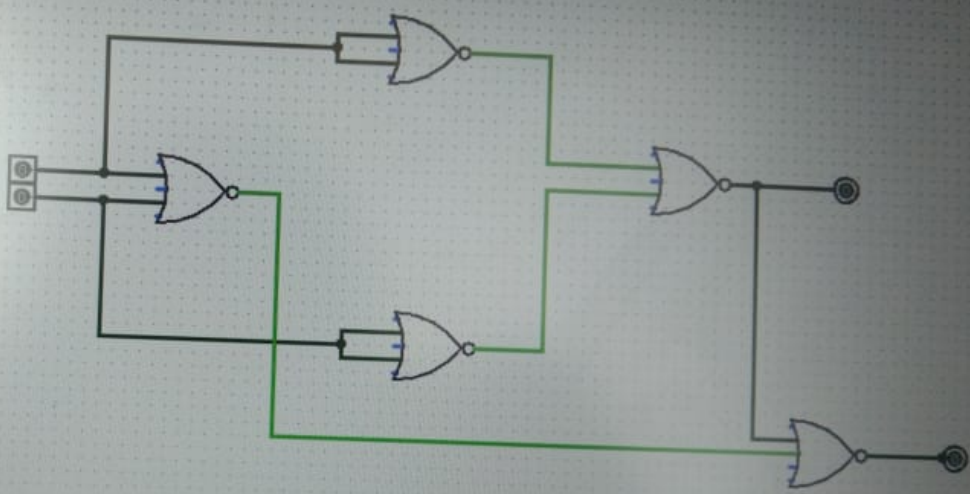
Build Circuit

Circuit Name: main

Shared Label

Shared Label Facing: East

Shared Label Font: SansSerif Plain 12



2 bit half
adder

File Edit Project Simulate Window Help

LogicSim

main

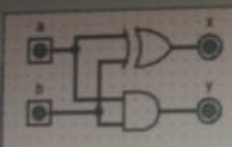
Wiring

Gates

- NOT Gate
- Buffer
- AND Gate
- OR Gate
- NAND Gate
- NOR Gate
- XOR Gate
- XNOR Gate
- Odd Parity
- Even Parity
- Controlled Buffer
- Controlled Inverter

Circuit: main

Circuit Name	main
Shared Label	
Shared Label Facing	East
Shared Label Font	San serif Plain 12



Combinational Analysis

File Edit Project Simulate Window Help

Inputs Outputs Table Expression Minimized

a	b	x	y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Build Circuit

Half adder

Register	Value	Flag
PC	03	S 0
DE	00	0 1
BC	1F	73
PSW	00	00
AC	42	18
SP	FF	FF
IR-Reg	00	C 0

Decimal - Hex Conversion

Decimal: Hex:

To Hex To Dec

I/O Ports

0 - + - 00

Update Port Value

Memory

8051 - + - 03

Update Memory

```

1  LDI B, 1234 ; Load the address 1234 into the B register pair
2  MOV A, M ; Move the contents of the memory location 8050H into the accumulator (A)
3  INX B ; Increment B to point to the next memory location (8051H)
4  MOV B, M ; Move the contents of the memory location 8051H into the B register
5
6  LOOP: CMP B ; Compare A with B
7      JZ STORE ; If A == B, jump to STORE
8      JC EXG ; If A < B, jump to EXG
9      SUB B ; A = A - B
10     JMP LOOP ; Repeat the loop
11
12 EXG: MOV C, B ; Store B in C (temporary storage)
13     MOV B, A ; Move A to B
14     MOV A, C ; Move C (original B) to A
15     JMP LOOP ; Repeat the loop
16
17 STORE: STA 8050H ; Store the GCD (value of A) at memory location 8050H
18     HLT ; Halt the program
    
```

Data Stack KeyPad Memory I/O Ports

Start 8050

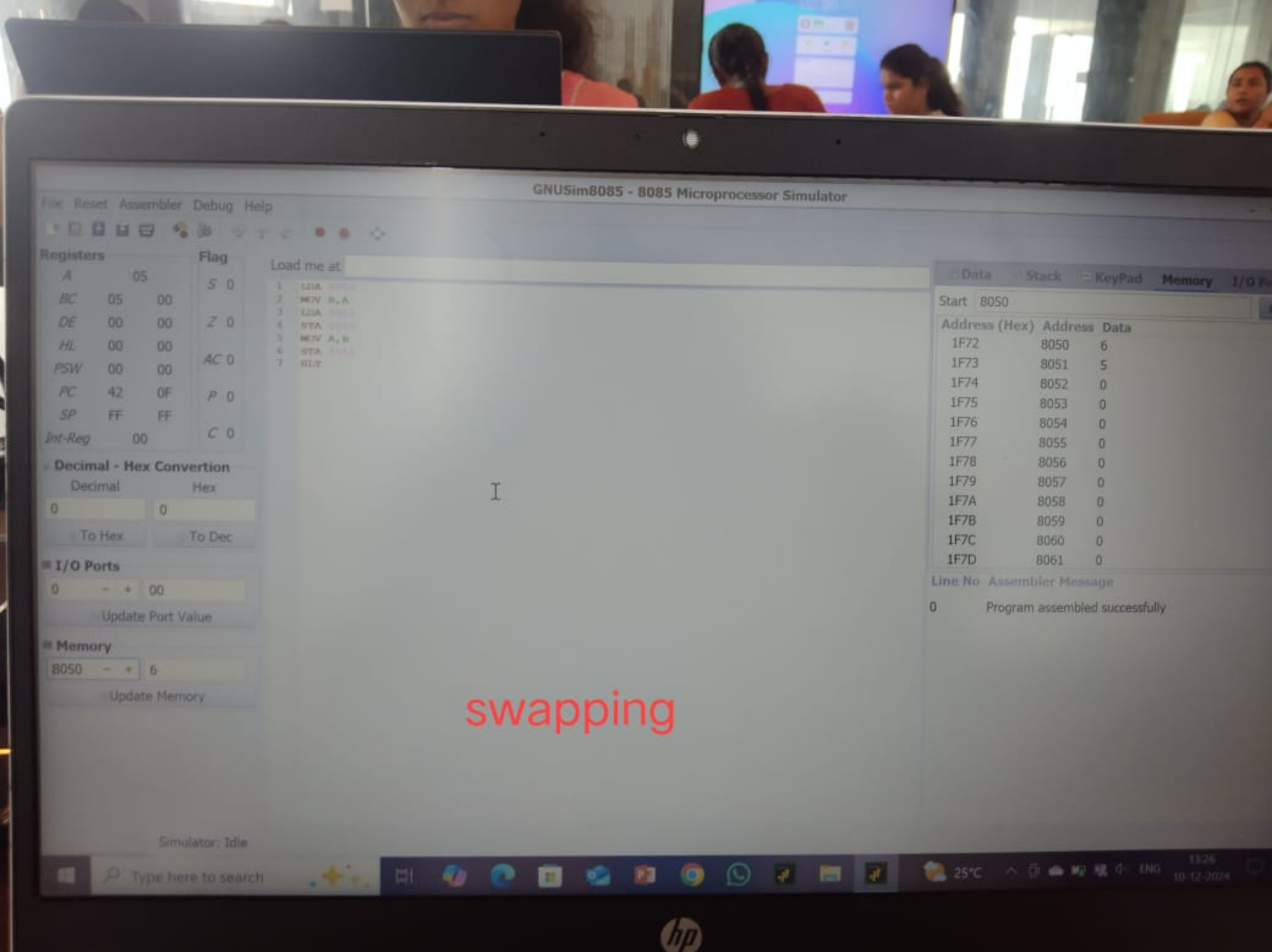
Address (Hex)	Address	Data
1F72	8050	12
1F73	8051	3
1F74	8052	0
1F75	8053	0
1F76	8054	0
1F77	8055	0
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	3
1F7C	8060	0
1F7D	8061	0

Line No Assembler Message

0 Program assembled successfully

GCD

Simulator: Idle



Register	Flag
A	VC
BC	DE
CD	00
HE	00
PSW	00
PC	42
SP	FF
Car Flag	00

Load me at

```

1 LDA 0000
2 MOV A,A
3 LDA 0000
4 INP B
5 JNC STORE
6 STORE: MOV A,B
7 STA 0000
8 HLT
    
```

Decimal - Hex Conversion

Decimal: 0 Hex: 0

To Hex To Dec

I/O Ports

0 - + 00

Update Port Value

Memory

8051 - + 28

Update Memory

Data	Stack	Register	Memory
------	-------	----------	--------

Start: 8050

Address (Hex)	Address	Data
1F72	8050	30
1F73	8051	40
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

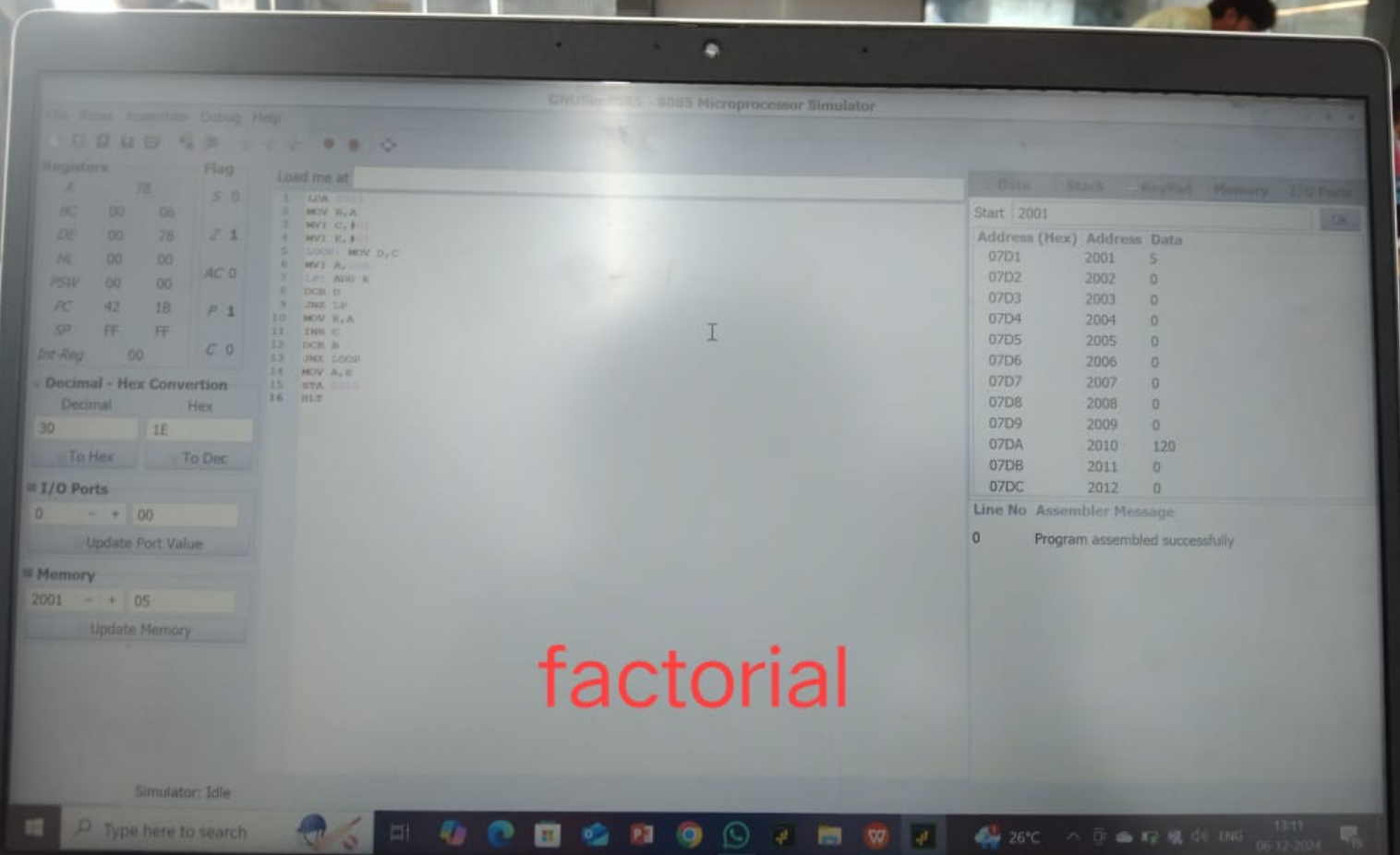
Line No: Assembly Message

0 Program assembled successfully

smallest
number.

Simulator: Idle





Registers

A	04	
BC	03	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	10
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	0
AC	0
P	0
C	0

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 8052
8 HLT
```

I

Decimal - Hex Conversion

Decimal	Hex
0	0

To Hex

To Dec

I/O Ports

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

Update Port Value

Memory

8051	-	+	04
------	---	---	----

Update Memory

Start 8050

Address (Hex)	Address	Data
1F72	8050	3
1F73	8051	4
1F74	8052	4
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

greatest
number

Simulator: Program running

Type here to search



26°C

1402

05-12-2024

ENG

Registers			Flag
A	AC		S 1
BC	56	00	
DE	00	00	Z 0
HL	08	04	AC 0
PSW	00	00	
PC	42	0C	P 1
SP	FF	FF	
Int-Reg	00		C 0

Load me at

1. **LDI R, C**
2. MOV A, R
3. ADD A, A
4. MOV B, A
5. ADD B
6. INR B
7. ADD M
8. INR M
9. MOV M, A
10. HLT

Decimal - Hex Conversion

Decimal Hex

0 0

To Enter a hexadecimal number

I/O Ports

0 - + 00

Update Port Value

Memory

2050 - + 2B

Update Memory

Data Stack KeyPad Men

Start 2050

Address (Hex)	Address	Data
0802	2050	43
0803	2051	0
0804	2052	172
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

Decimal to hexa

Simulator: Program running

Type here to search



GNUsim8085 - 8085 Microprocessor Simulator

File View Assembly Debug Help

Registers: A 02, B 00, C 00, D 00, E 34, H 00, L 00, PC 42, SP FF, In-Reg 00, Flag Z 1, AC 0, P 1, C 1

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

I/O Ports: 0, Update Port Value

Memory: 8055, Update Memory

Load file as:

- 1. GET BL
- 2. MOV CL, B
- 3. INC B
- 4. MOV BL, B
- 5. DEC C
- 6. JNZ, INC B
- 7. MOV BL, B
- 8. INC B
- 9. JNC, INC C
- 10. MOV BL, B
- 11. JZ, DEC C
- 12. INC, INC C
- 13. GET BL, 8055
- 14. MOV BL, B
- 15. HLT

Data Stack KeyPad Memory I/O Ports

Start: 8050

Address (Hex)	Address	Data
1F72	8050	5
1F73	8051	56
1F74	8052	76
1F75	8053	45
1F76	8054	31
1F77	8055	2
1F78	8056	0
1F79	8057	0
1F7A	8058	0
1F7B	8059	0
1F7C	8060	0
1F7D	8061	0

Line No: 0, Assembler Message: Program assembled successfully

smallest no.
array



Registers		Flag
A	FB	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	

Decimal - Hex Conversion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0 - + 00

Enter new port value and click Update

Memory

8050 - + 04

Update Memory

Load me at

```
1 LDA #05H
2 CMA
3 STA #051
4
```

Data Stack KeyPad Memory

Start 8050

Address (Hex)	Address	Data
1F72	8050	4
1F73	8051	251
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

one's
complement

Simulator: Idle

Type here to search



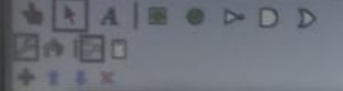
25°C



ENG

13:49

10-12-20



Combinational Analysis

File Edit Project Simulate Window Help

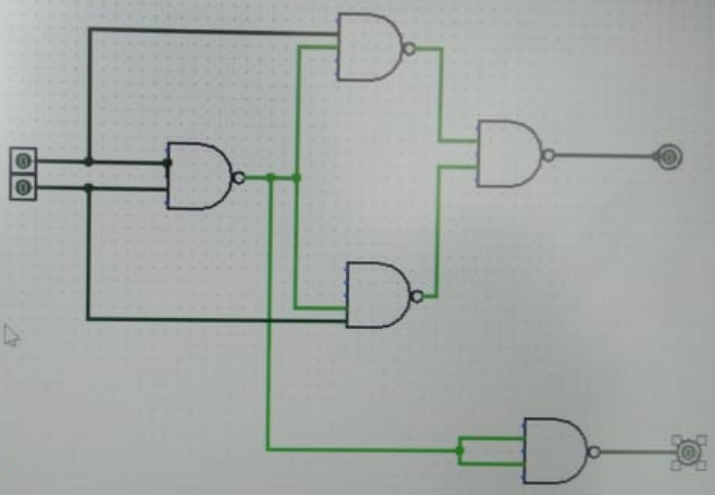
Inputs Outputs Table Expression Minimized

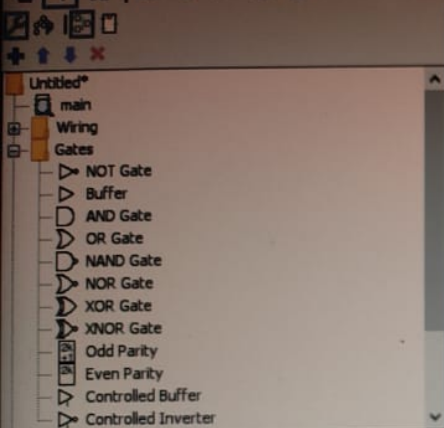
a	b	x	y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Build Circuit

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

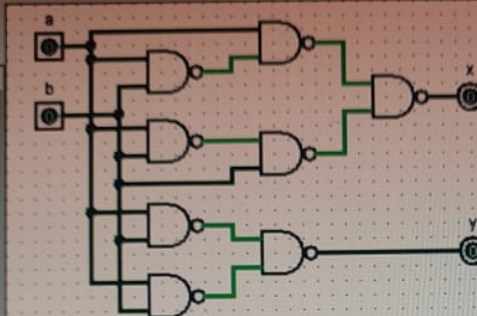
2 bit half adder





Circuit: main

Circuit Name	main
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12



Combinational Analysis

File Edit Project Simulate Window Help

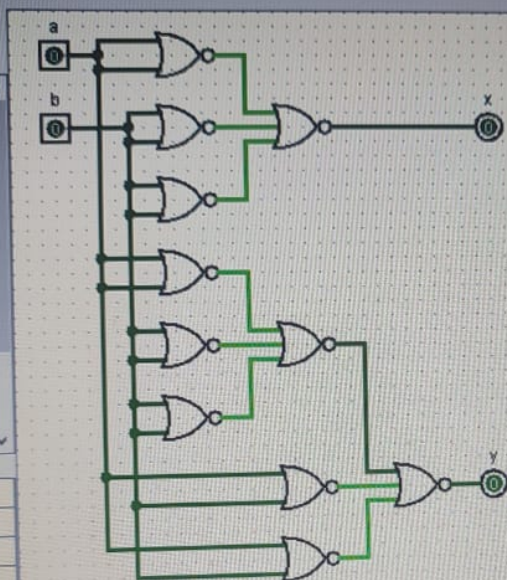
Inputs Outputs Table Expression Minimized

a	b	x	y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Build Circuit

Circuit: main

Name	main
Label	
Label Facing	East
Label Font	SansSerif Plain 12



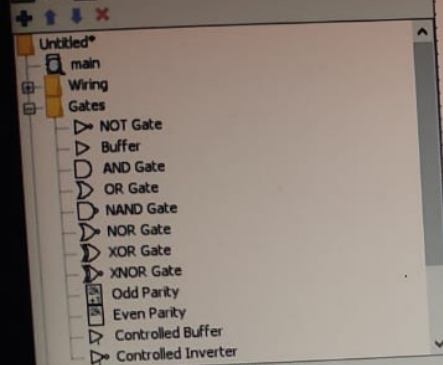
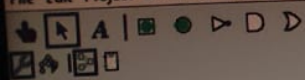
Combinational Analysis

File Edit Project Simulate Window Help

Inputs Outputs Table Expression Minimized

a	b	x	y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Build Circuit



Circuit: main

Circuit Name	main
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12



Combinational Analysis

File Edit Project Simulate Window Help

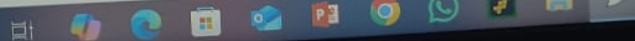
Inputs Outputs Table Expression Minimized

a	b	x	y
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Build Circuit

100%

Type here to search



26°C

ENG

23:12
07-12-2024

SHOT ON MI 10I

main of Untitled

Project Simulate Window Help

NOT Gate
Buffer
AND Gate
OR Gate
NAND Gate
NOR Gate
XOR Gate
XNOR Gate
Odd Parity
Even Parity
Controlled Buffer
Controlled Inverter

Circuit: main

ne	main
bel	
bel Facing	East
bel Font	SansSerif Plain 12

Combinational Analysis

File Edit Project Simulate Window Help

Inputs Outputs Table Expression Minimized

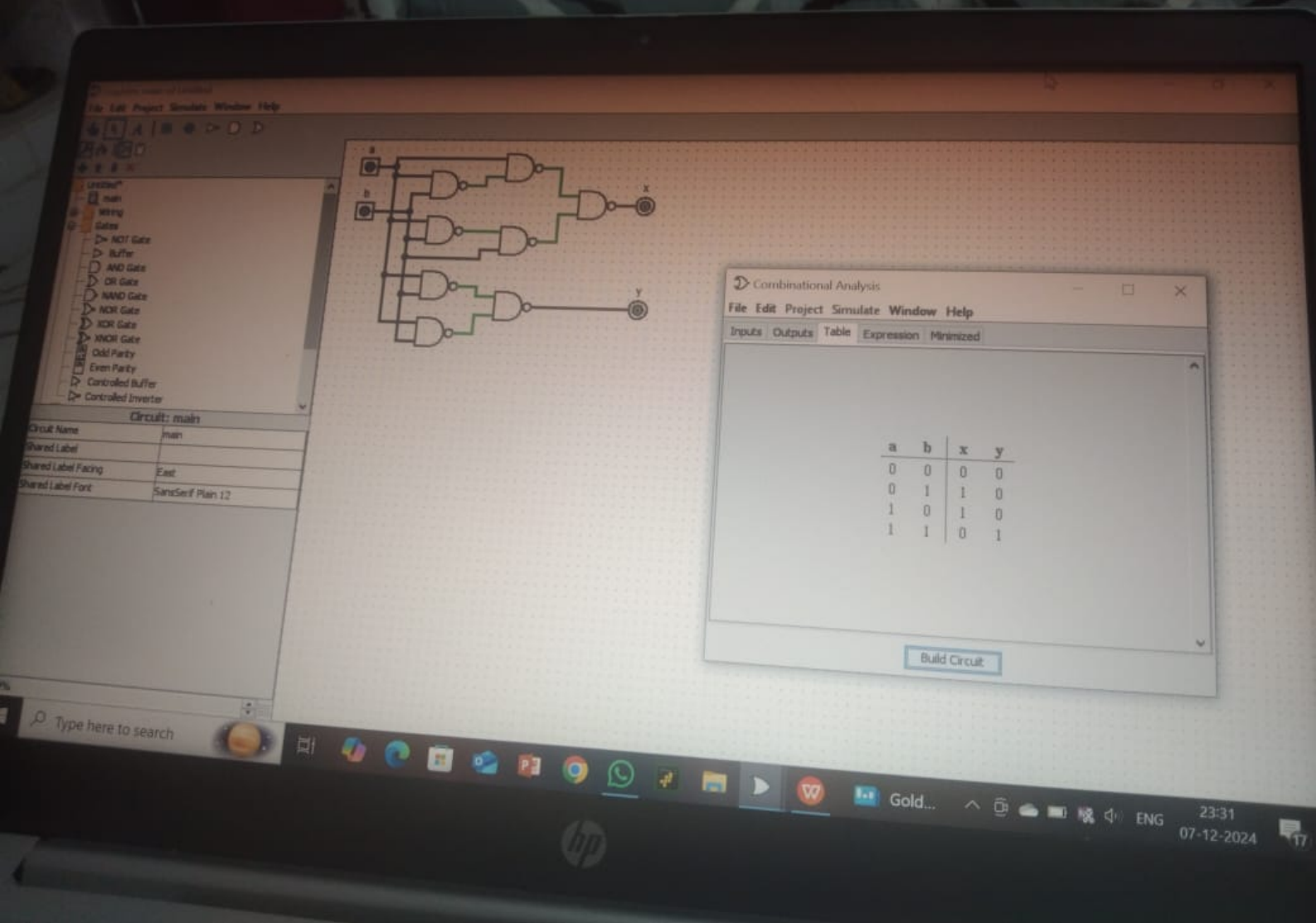
a	b	c	x	y
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Build Circuit

Type here to search

11:17
08-12-2024

SHOT ON MI 101





Registers			Flag	
A	5C		S	0
BC	5E	00	Z	1
DE	00	00	AC	0
HL	21	34	P	1
PSW	00	00	C	1
PC	42	17		
SP	FF	FF		
Int-Reg	00			

Decimal - Hex Conversion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0	-	+	00
Update Port Value			

Memory

8056	-	+	00
Update Memory			

Load me at

```
1 LXL B,8056
2 MOV C,M
3 INX B
4 MOV B,M
5 DCR C
6 LOOP: INX B
7 MOV A,M
8 CMP B
9 JC BRIP
10 MOV B,A
11 DCR: DCR C
12 INX LOOP
13 LXL B,8500
14 MOV M,B
15 HLT
```

I

largest no. in
array

Data Stack KeyPad Memory I/O Ports

Start 8050

Address (Hex)	Address	Data
2130	8496	0
2131	8497	0
2132	8498	0
2133	8499	0
2134	8500	94
2135	8501	0
2136	8502	0
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0

Line No Assembler Message

0 Program assembled successfully

Simulator: Idle



Type here to search



25°C



ENG

13:21

10-12-2024



Registers			Flag
A	00	00	S 0
BC	00	00	Z 0
DE	00	00	AC 1
HL	00	00	P 0
PSW	00	00	C 0
PC	42	13	
SP	FF	FF	
Int-Reg	00		

Decimal - Hex Conversion

Decimal	Hex
0	0
To Hex	To Dec

I/O Ports

0	-	+	00
Update Port Value			

Memory

8050	-	+	0B
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 [Odd number]
5 JMP LOOP2 ;Jump to LOOP2
6 LOOP1: MVI A,22 ;Move immediate value 22 into the accumulator [Even number]
7 LOOP2: STA 8051 ;Store the accumulator content at memory location 8051H
8 HLT ;Halt the processor
```

Data Stack Keypad Memory I/O

Start 8050

Address (Hex) Address Data

1F72	8050	11
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

Simulator: Idle

Type here to search

HP

25°C

ENG

13:44

10-12-2024