

LAB 5

Simulation of RTC using 8085 MP

Q1. 12 hour clock

Code:

START: LXI H, 00H

HOURLMIN: SHLD 5001H

XRA A

SECMINHOURL: STA 5000H

CALL SECDELAY

LDA 5000H

ADI 01H

DAA

MOV E, A

CPI 60

JNZ SECMINHOURL

XRA A

STA 5000H

MOV E, A

LHLD 5001H

MOV A, L

ADI 01H

DAA

MOV L, A

CPI 60

JNZ HOURMIN

XRA A

STA 5001H

MOV L,A

MOV A, H

ADI 01H

DAA

MOV H, A

CPI 12

JNZ HOURMIN

JMP START

SECDELAY: MVI D, 02H

OUTERLOOP: LXI B, FFFFH

INNERLOOP: DCX B

MOV A, B

ORA C

JNZ INNERLOOP

DCR D

JNZ OUTERLOOP

RET

Output:

Compile

The screenshot shows the Sim8085 emulator interface. On the left, the 'Registers' panel displays the following values: A/PSW (0x0002), BC (0x0000), DE (0x0000), HL (0x0000), SP (0x0000), PC (0x0800), Z (unchecked), S (unchecked), P (unchecked), C (unchecked), and AC (unchecked). The 'Flags' panel shows Z, S, P, C, and AC flags, all of which are unchecked. The central 'main.asm' window contains the following assembly code:

```
1 START: LXI H, 00H
2 HOURMIN: SHLD 5001H
3           XRA A
4 SECMINHOURL: STA 5000H
5           CALL SECDELAY
6           LDA 5000H
7           ADI 01H
8           DAA
9           MOV E, A
10          CPI 60
11          JNZ SECMINHOURL
12          XRA A
13          STA 5000H
14          MOV E, A
15          LHLD 5001H
16          MOV A, L
17          ADI 01H
18          DAA
19          MOV L, A
20          CPI 60
```

The 'Memory V' panel on the right shows a memory dump starting at address 000, with all bytes set to 00. A green notification box at the top right states: "Your code has been compiled and loaded to memory location 0x0800. Now you need to execute it to see the results."

The screenshot shows the Sim8085 emulator interface after execution. The 'Registers' panel displays the following values: A/PSW (0x0002), BC (0x0000), DE (0x0000), HL (0x0000), SP (0x0000), PC (0x0800), Z (unchecked), S (unchecked), P (unchecked), C (unchecked), and AC (unchecked). The 'Flags' panel shows Z, S, P, C, and AC flags, all of which are unchecked. The central 'main.asm' window contains the following assembly code:

```
21          JNZ HOURMIN
22          XRA A
23          STA 5001H
24          MOV L, A
25          MOV A, H
26          ADI 01H
27          DAA
28          MOV H, A
29          CPI 12
30          JNZ HOURMIN
31          JMP START
32 SECDELAY: MVI D, 02H
33 OUTERLOOP: LXI B, FFFFH
34 INNERLOOP: DCX B
35            MOV A, B
36            ORA C
37            JNZ INNERLOOP
38            DCR D
39            JNZ OUTERLOOP
40            RET
```

The 'Memory V' panel on the right shows a memory dump starting at address 000, with all bytes set to 00. A red error message box at the top right states: "Program terminated with errors! Looks like you have an infinite loop in your code. Did you forget the HLT instruction? This software is still in alpha phase. I want to work on this more but working on this is not my highest priorities right now. If you like to help, please file a bug or contribute to fix this bug."

Q2. 24 hour clock

Code:

START: LXI H, 00H

HOURLMIN: SHLD 5001H

XRA A

SECMINHOURL: STA 5000H

CALL SECDELAY

LDA 5000H

ADI 01H

DAA

MOV E, A

CPI 60

JNZ SECMINHOURL

XRA A

STA 5000H

MOV E, A

LHLD 5001H

MOV A, L

ADI 01H

DAA

```
MOV L, A
CPI 60
JNZ HOURMIN
XRA A
STA 5001H
MOV L,A
MOV A, H
ADI 01H
DAA
MOV H, A
CPI 24
JNZ HOURMIN
JMP START
```

```
SECDELAY: MVI D, 02H
```

```
OUTERLOOP: LXI B, FFFFH
```

```
INNERLOOP: DCX B
```

```
MOV A, B
```

```
ORA C
```

```
JNZ INNERLOOP
```

```
DCR D
```

```
JNZ OUTERLOOP
```

RET

Output:

The screenshot shows the Sim8085 web simulator interface. The assembly code in `main.asm` is as follows:

```
1 START: LXI H, 00H
2 HOURMIN: SHLD 5001H
3 XRA A
4 SECMIN HOUR: STA 5000H
5 CALL SECDELAY
6 LDA 5000H
7 ADI 01H
8 DAA
9 MOV E, A
10 CPI 60
11 JNZ SECMIN HOUR
12 XRA A
13 STA 5000H
14 MOV E, A
15 LHLD 5001H
16 MOV A, L
17 ADI 01H
18 DAA
19 MOV L, A
20 CPI 60
```

The Registers panel shows the following values:

Register	Value
A/PSW	0x0002
BC	0x0000
DE	0x0000
HL	0x0000
SP	0x0000
PC	0x0800

The Flags panel shows the following values:

Flag	Value
Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input type="checkbox"/>
C	<input type="checkbox"/>
AC	<input type="checkbox"/>

The Memory panel shows the following values:

Address	Value
000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
001	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
006	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
008	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
009	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

A green notification box states: "Your code has been compiled and loaded to memory location 0x0800. Now you need to execute it to see the results."

The screenshot shows the Sim8085 web simulator interface. The assembly code in `main.asm` is as follows:

```
1 START: LXI H, 00H
2 HOURMIN: SHLD 5001H
3 XRA A
4 SECMIN HOUR: STA 5000H
5 CALL SECDELAY
6 LDA 5000H
7 ADI 01H
8 DAA
9 MOV E, A
10 CPI 60
11 JNZ SECMIN HOUR
12 XRA A
13 STA 5000H
14 MOV E, A
15 LHLD 5001H
16 MOV A, L
17 ADI 01H
18 DAA
19 MOV L, A
20 CPI 60
```

The Registers panel shows the following values:

Register	Value
A/PSW	0x0002
BC	0x0000
DE	0x0000
HL	0x0000
SP	0x0000
PC	0x0800

The Flags panel shows the following values:

Flag	Value
Z	<input type="checkbox"/>
S	<input type="checkbox"/>
P	<input type="checkbox"/>
C	<input type="checkbox"/>
AC	<input type="checkbox"/>

The Memory panel shows the following values:

Address	Value
000	00 00 00
001	00 00 00
002	00 00 00
003	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
005	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
006	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
007	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
008	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
009	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

A red notification box states: "Program terminated with errors! Looks like you have an infinite loop in your code. Did you forget the HLT instruction? This software is still in alpha phase. I want to work on this more but working on this is not my highest priorities right now. If you like to help, please file a bug or contribute to fix this bug."

LAB-5

Q) START: LXI H, 000H

Hour_min: SHLD 500H
XRA A

Sec_min_Hour: STA 5000H
CALL Sec_delay

LDA 5000H

ADI 01H

DAA

MOV E, A

CPI 60

JNZ Sec_min_Hour

XRA A

STA 5000H

MOV E, A

LHLD 500H

MOV A, L

ADI 01H

DAA

MOV L, A

CPZ 60

JNZ Hour_min

XRA A

STA 5000H

MOV A, L

MOV A, H

ADI 01H

DAA

MOV H, A

CPI 24

JNZ Hour_min

JMP START

Sec_delay: MVI 0, 01H

outer_loop: LXI B, FF0H

Inner-loop: DCR B
 MOV A, B
 ORA C
 JNZ Inner-loop
 DCR D
 JNZ outer-loop
 RET

12 Hour clock:

AM-PM:

START

MVI D, 00H
 LXI H, 0000H
 MOV A, D
 CMA
 ANI 0FH
 STA 5005H
 MOV D, A

Hour-min:

SHLD 5001H
 XRA A

Sec-min-hour:

STA 5000H
 call sec delay
 LDA 5000H
 ADI 01H
 DAA
 MOV E, A
 CPI 60
 JNZ Sec-min-hour
 XRA A
 STA 5000H
 MOV E, A
 LHLD 5001H
 MOV A, L
 ADI 01H
 DAA


```

MOV     L, A
CPI     80
JNZ     Hour-min
XRA     A
STA     5000H
MOV     L, A
MOV     A, H
ADD     01H
DAA
MOV     H, A
CPI     12
JNZ     Hour-min
JMP     START

```

```

Sec delay:  MVI     D, 02H
Outer loop:  LXI     A, 0FFH
Inner loop:  DCR     B
             MOV     A, B
             ORA     C
             JNZ     inner-loop
             DCR     D
             JNZ     Outer-loop
             RET.

```

MOV A, B
 RAR
 RLC
 RRC