VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE ENGINEERING

WINTER SEMESTER 2022-2023

LAB ASSIGNMENT - 2

Slot: L11 – L12

Class: VL2022230504038

Programme Name & Branch: B. Tech CSE

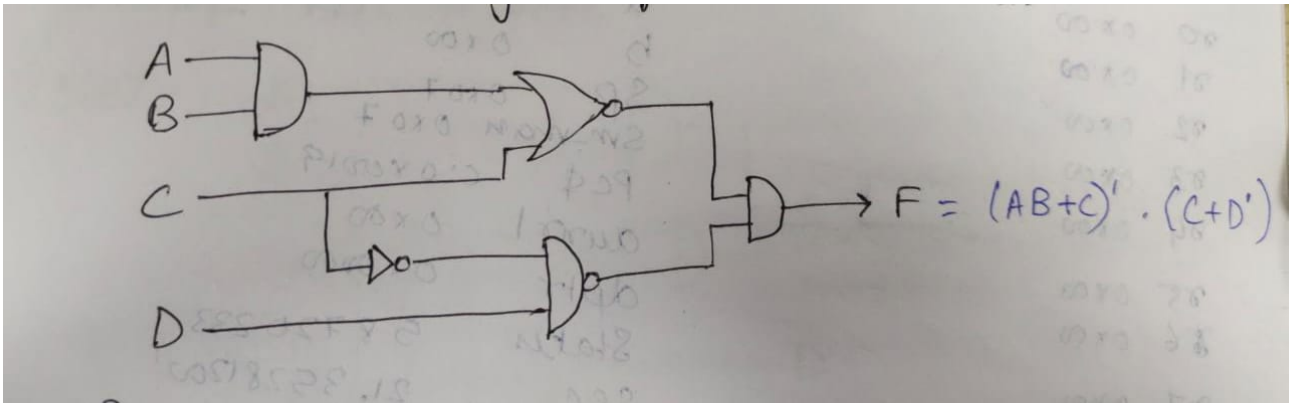
Course code & Title: BECE204P – Microprocessors and Microcontrollers Lab

Faculty Name: Venu Allapakam

Task 2: Implementation of digital circuit and port programming

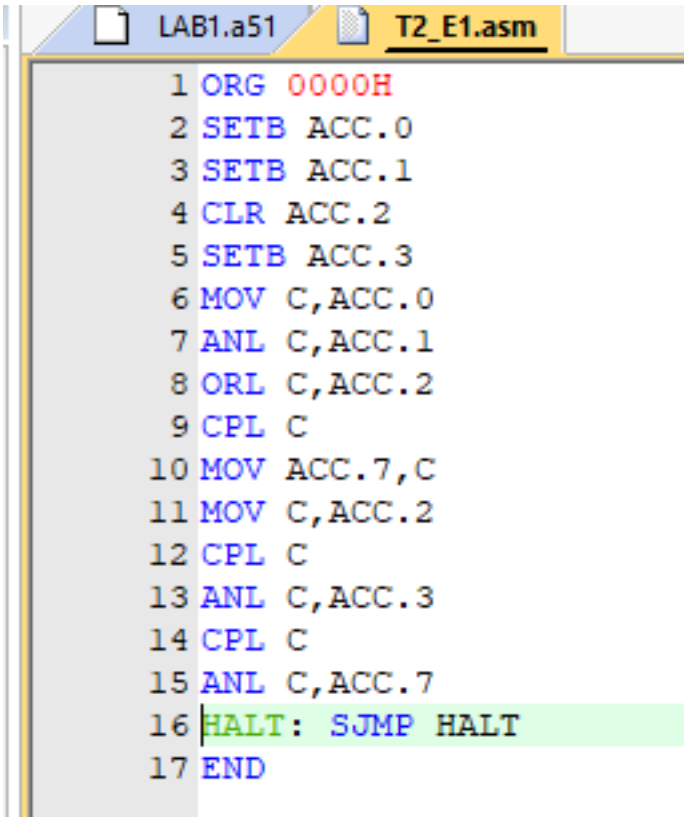
Program 1: Digital Circuit 1

Aim: To implement the given digital logic circuit using keil software for 8051 micro controllers



Software Requirement: Keil Software

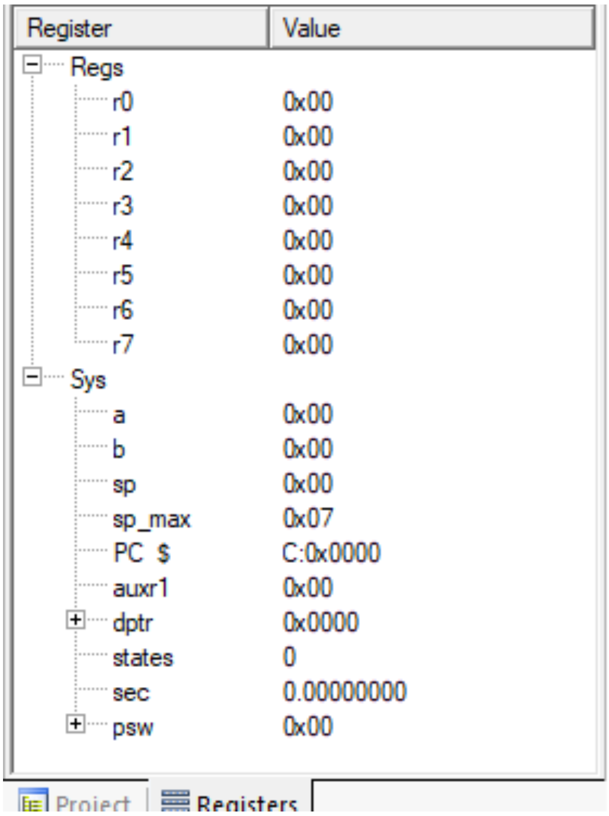
Program:



```
ORG 0000H
SETB ACC.0
SETB ACC.1
CLR ACC.2
SETB ACC.3
MOV C,ACC.0
ANL C,ACC.1
ORL C,ACC.2
CPL C
MOV ACC.7,C
MOV C,ACC.2
CPL C
ANL C,ACC.3
CPL C
ANL C,ACC.7
HALT: SJMP HALT
END
```

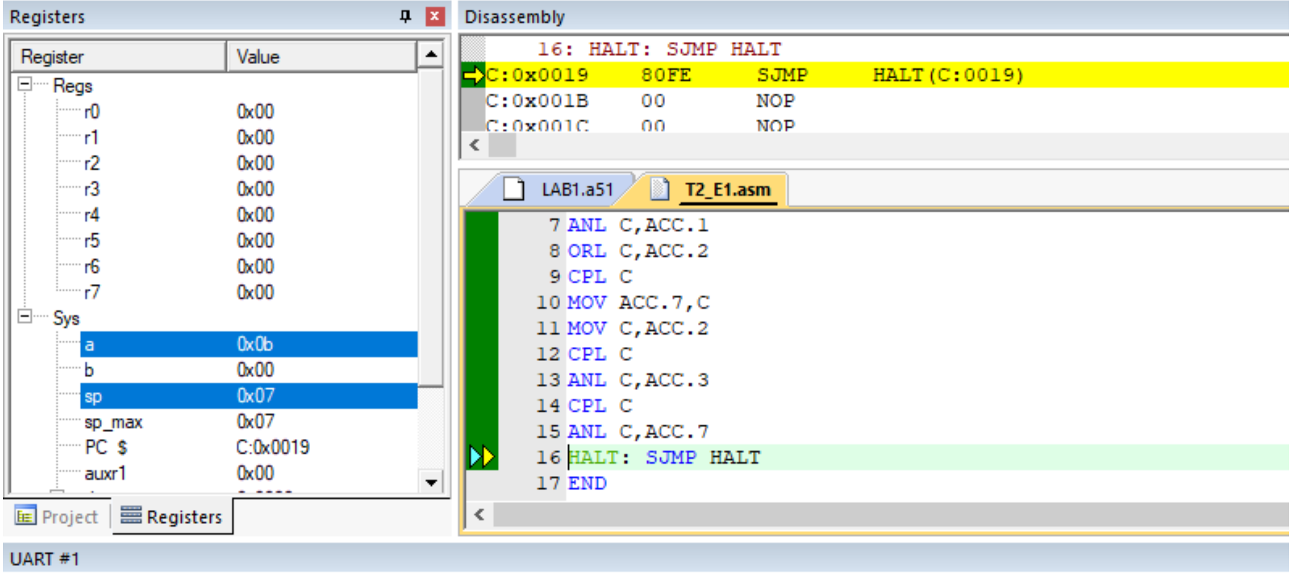
Output:

Before Execution Register status:



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
PC $	C:0x0000
auxr1	0x00
dptr	0x0000
states	0
sec	0.00000000
psw	0x00

After Execution:

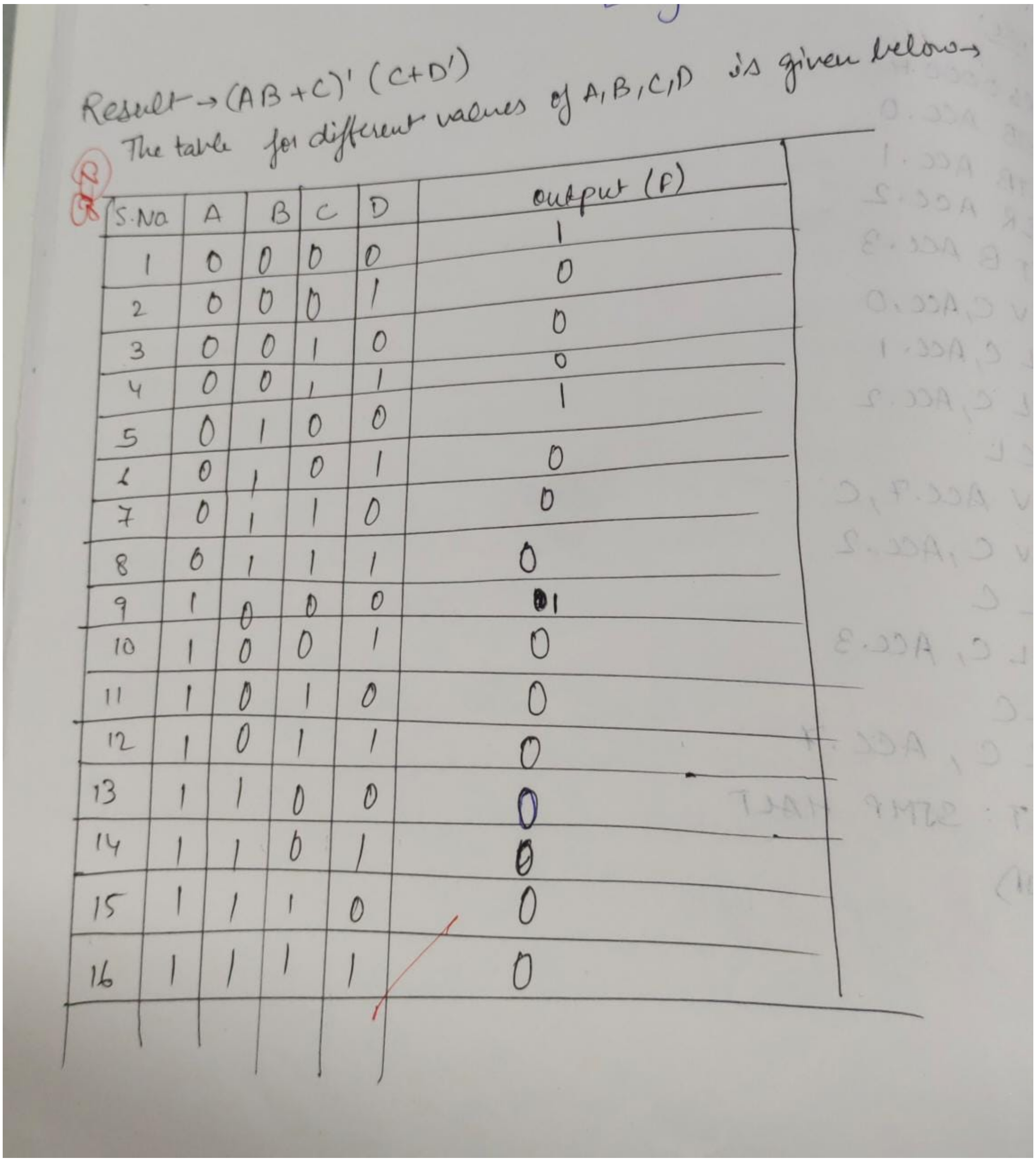


Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x0b
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0019
auxr1	0x00

```
    16: HALT: SJMP HALT
C:0x0019    80FE     SJMP     HALT(C:0x0019)
C:0x001B    00       NOP
C:0x001C    00       NOP
```

```
ANL C,ACC.1
ORL C,ACC.2
CPL C
MOV ACC.7,C
MOV C,ACC.2
CPL C
ANL C,ACC.3
CPL C
ANL C,ACC.7
HALT: SJMP HALT
END
```

Result-



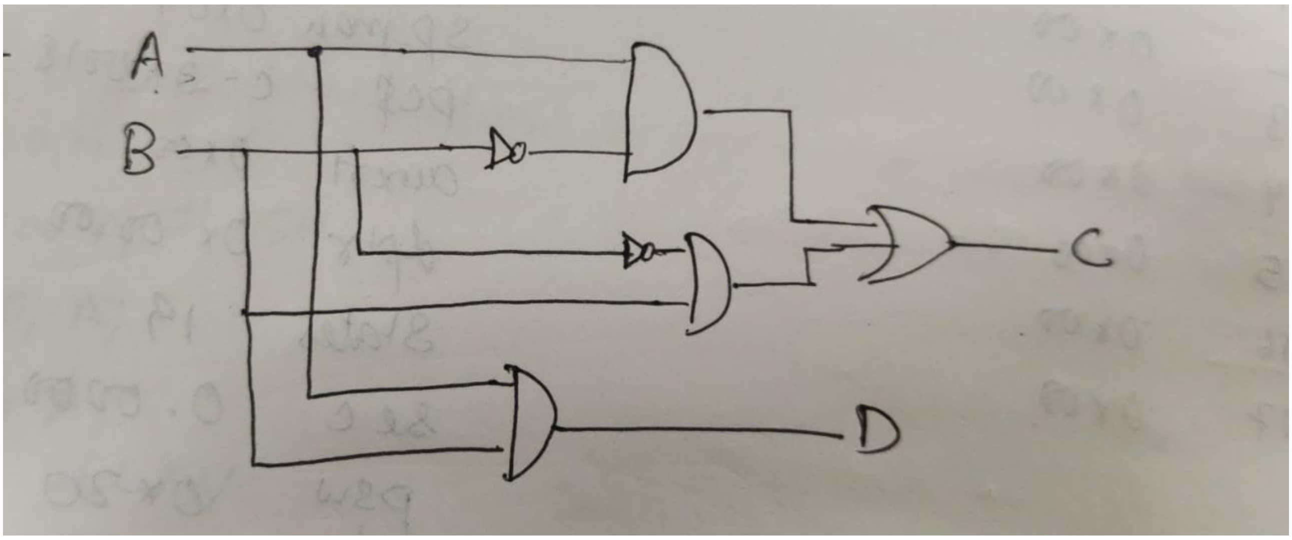
Result → (AB + C)' (C+D')

The table for different values of A, B, C, D is given below →

S.No	A	B	C	D	output (F)
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	0
5	0	1	0	0	1
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	1
10	1	0	0	1	0
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0

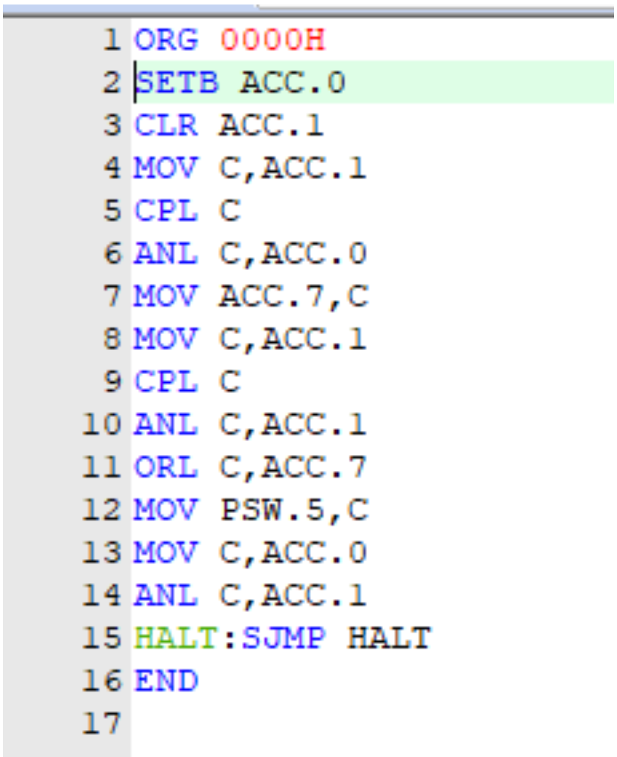
Program 2: Digital Circuit implementation

Aim: To Implement the given digital logic circuit using keil software for 8051 micro controllers



Software Requirement: Keil Software

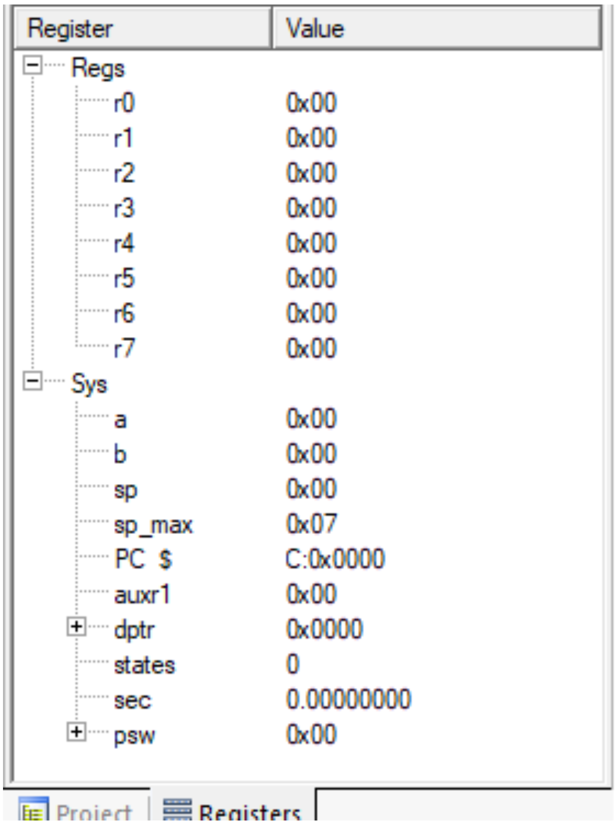
Program:



```
ORG 0000H
SETB ACC.0
CLR ACC.1
MOV C,ACC.1
CPL C
ANL C,ACC.0
MOV ACC.7,C
MOV C,ACC.1
CPL C
ANL C,ACC.1
ORL C,ACC.7
MOV PSW.5,C
MOV C,ACC.0
ANL C,ACC.1
HALT:SJMP HALT
END
```

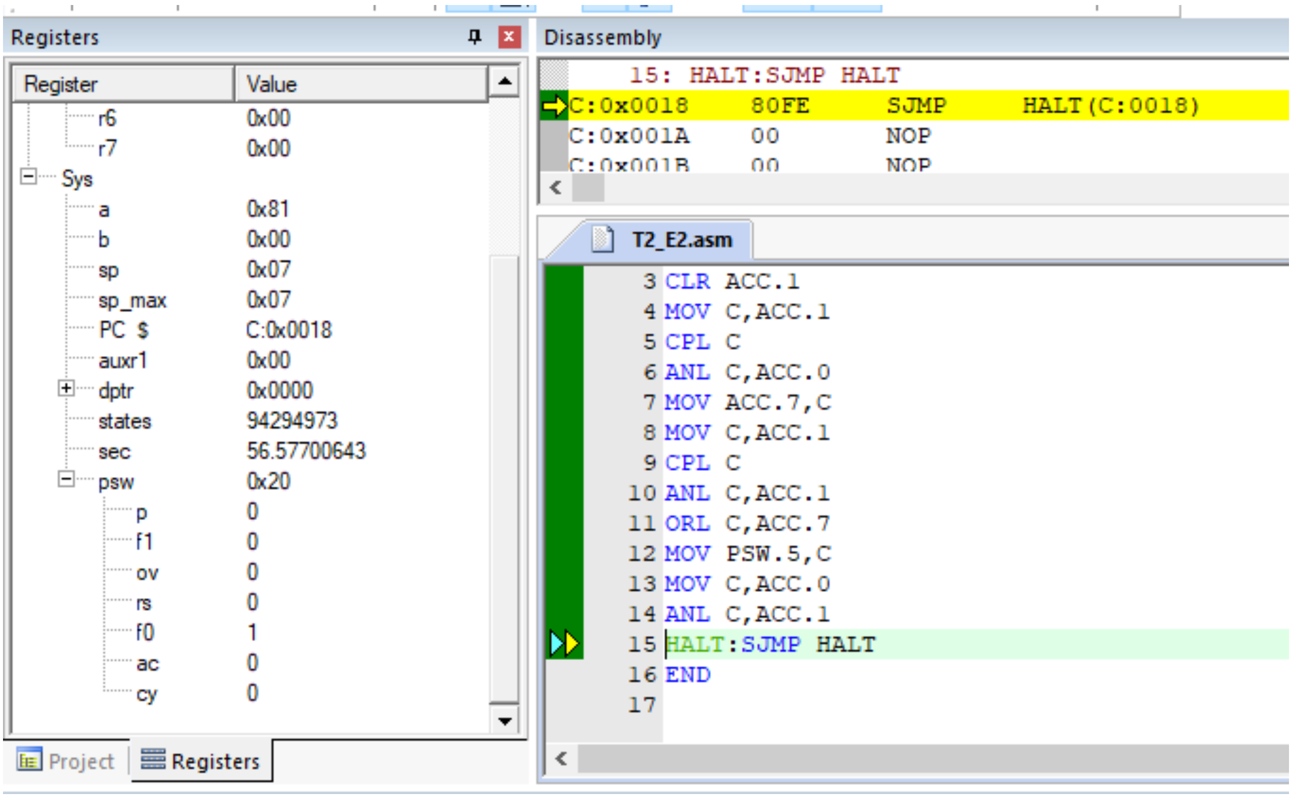
Output:

Before Execution Register status:



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
PC $	C:0x0000
auxr1	0x00
dptr	0x0000
states	0
sec	0.00000000
psw	0x00

After Execution:

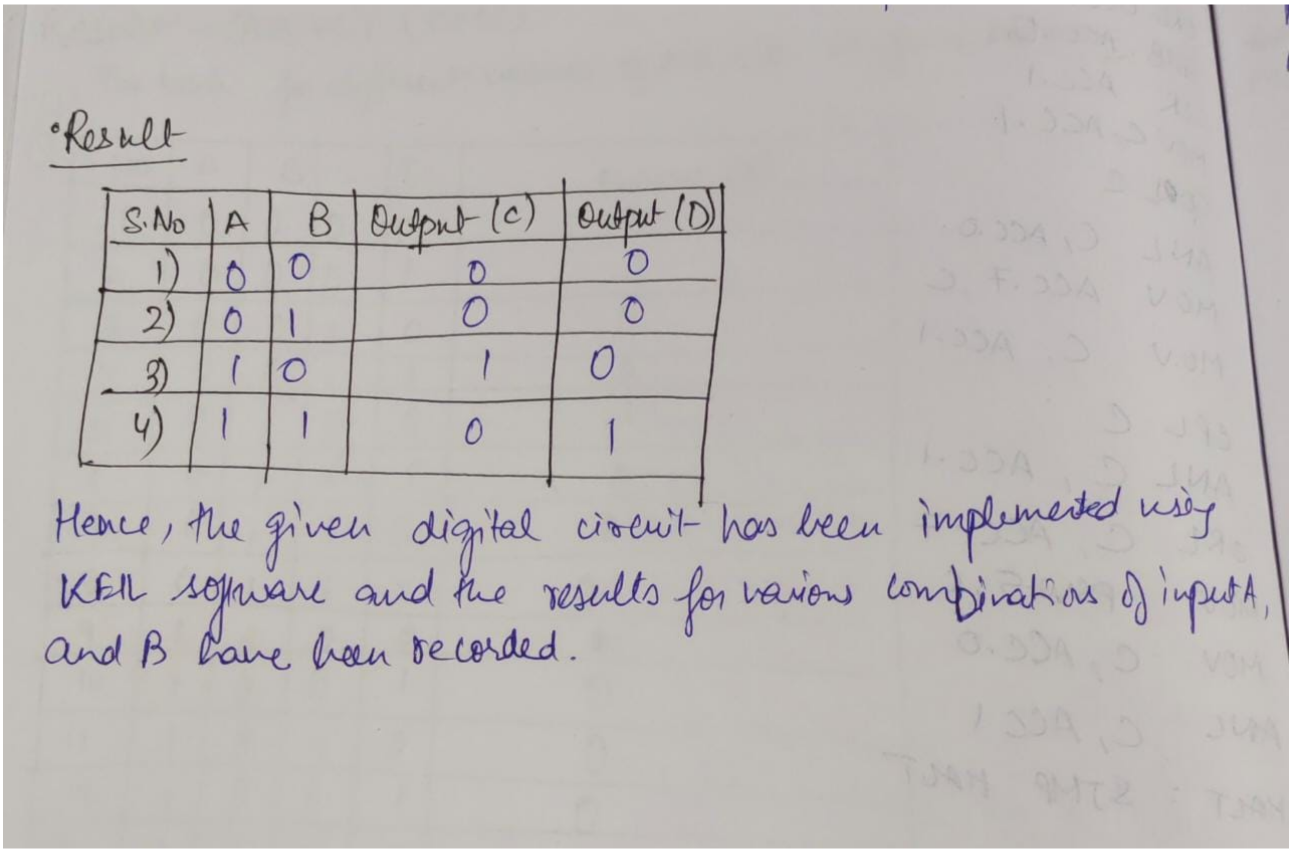


Register	Value
r6	0x00
r7	0x00
Sys	
a	0x81
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0018
auxr1	0x00
dptr	0x0000
states	94294973
sec	56.57700643
psw	0x20
p	0
f1	0
ov	0
rs	0
f0	1
ac	0
cy	0

```
    15: HALT:SJMP HALT
C:0x0018    80FE     SJMP     HALT(C:0018)
C:0x001A    00       NOP
C:0x001B    00       NOP
```

```
CLR ACC.1
MOV C,ACC.1
CPL C
ANL C,ACC.0
MOV ACC.7,C
MOV C,ACC.1
CPL C
ANL C,ACC.1
ORL C,ACC.7
MOV PSW.5,C
MOV C,ACC.0
ANL C,ACC.1
HALT:SJMP HALT
END
```

Result—



•Result

S.No	A	B	Output (C)	Output (D)
1)	0	0	0	0
2)	0	1	0	0
3)	1	0	1	0
4)	1	1	0	1

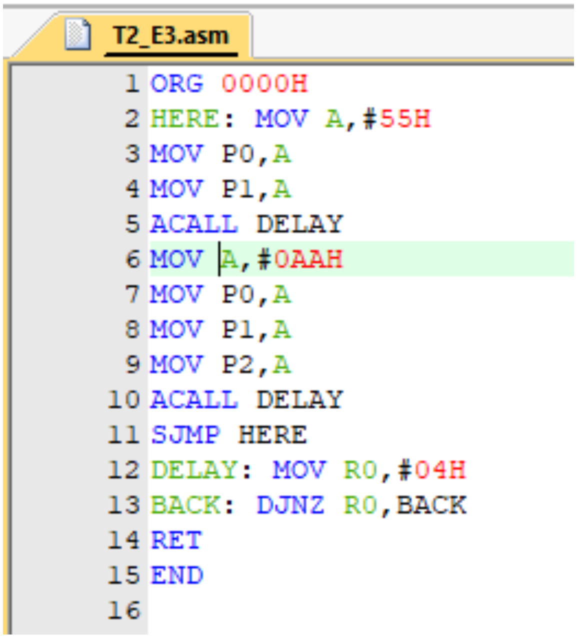
Hence, the given digital circuit has been implemented using KEIL software and the results for various combinations of input A, and B have been recorded.

Program 3:

Aim: To write an ALP to toggle all bits of P0, P1, and P2 continuously by sending 55H and AAH to these ports.

Software Requirement: Keil Software

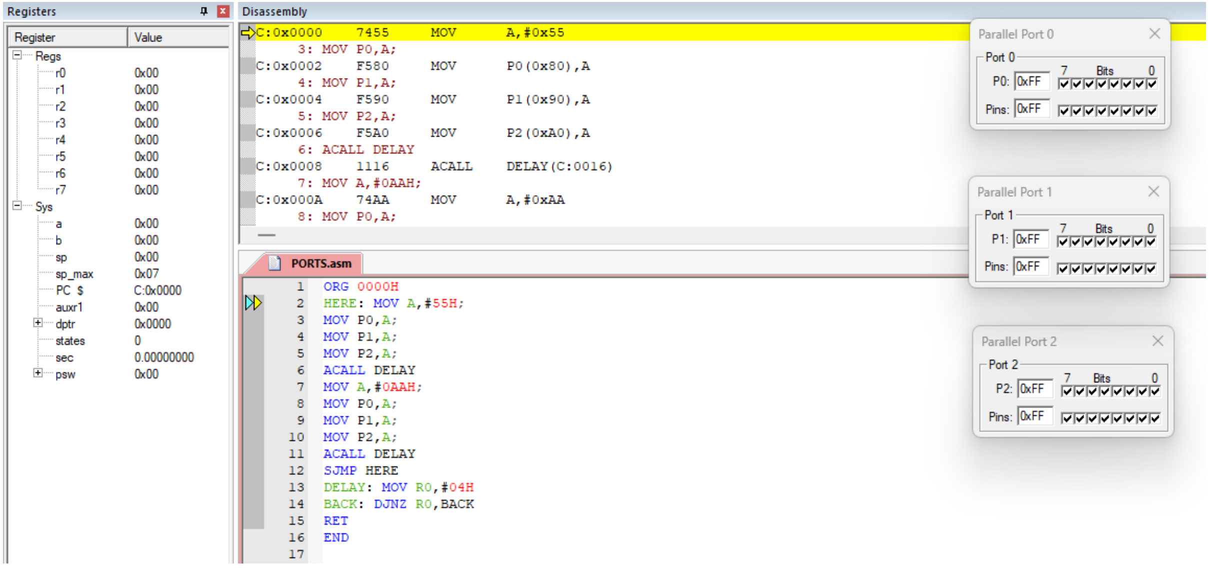
Program:


T2_E3.asm

```
ORG 0000H
HERE: MOV A,#55H
MOV P0,A
MOV P1,A
ACALL DELAY
MOV A,#0AAH
MOV P0,A
MOV P1,A
MOV P2,A
ACALL DELAY
SJMP HERE
DELAY: MOV R0,#04H
BACK: DJNZ R0,BACK
RET
END
```

Output:

BEFORE & AFTER Execution Register status:



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
PC $	C:0x0000
auxr1	0x00
dptr	0x0000
states	0
sec	0.00000000
psw	0x00

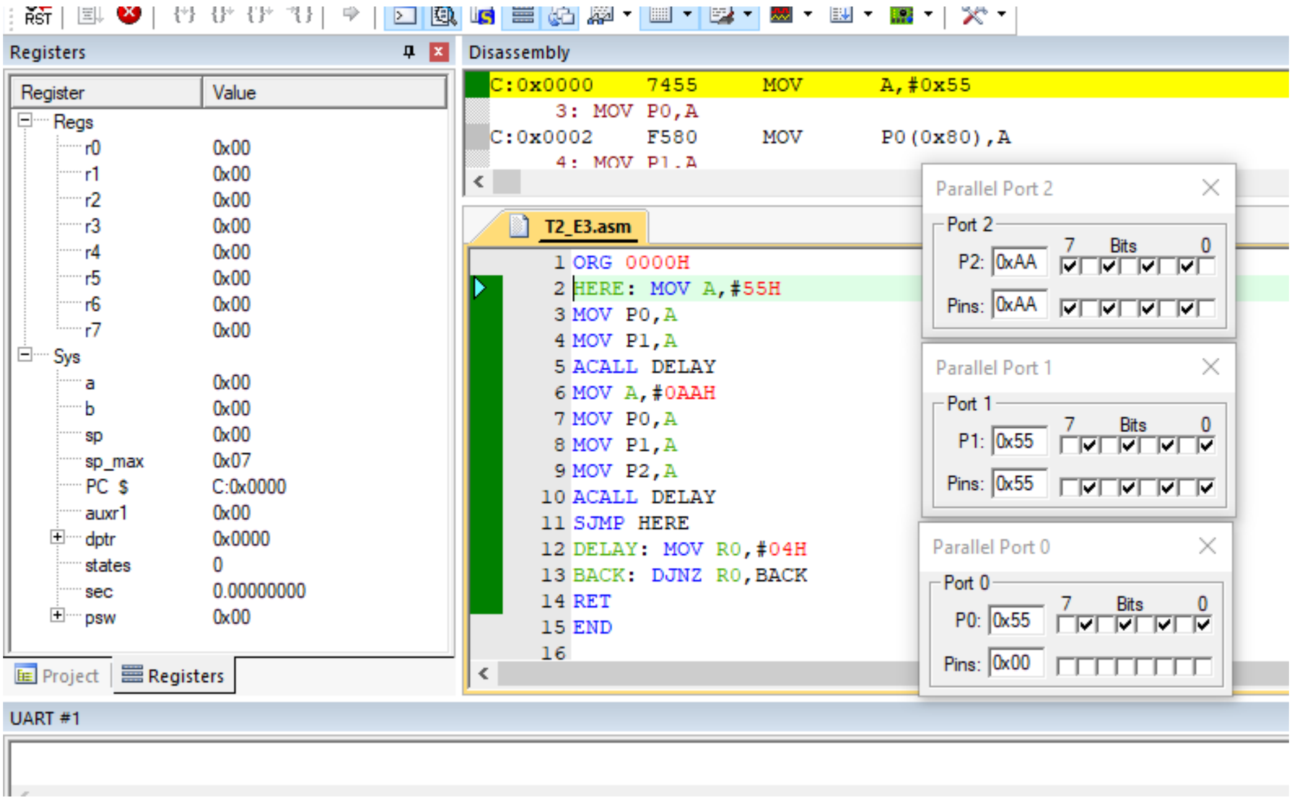
```
C:0x0000    7455     MOV      A,#0x55
     3: MOV P0,A;
C:0x0002    F580     MOV      P0(0x80),A
     4: MOV P1,A;
C:0x0004    F590     MOV      P1(0x90),A
     5: MOV P2,A;
C:0x0006    F5A0     MOV      P2(0xA0),A
     6: ACALL DELAY
C:0x0008    1116     ACALL    DELAY(C:0016)
     7: MOV A,#0AAH;
C:0x000A    74AA     MOV      A,#0xAA
     8: MOV P0,A;
```

```
ORG 0000H
HERE: MOV A,#55H;
MOV P0,A;
MOV P1,A;
MOV P2,A;
ACALL DELAY
MOV A,#0AAH;
MOV P0,A;
MOV P1,A;
MOV P2,A;
ACALL DELAY
SJMP HERE
DELAY: MOV R0,#04H
BACK: DJNZ R0,BACK
RET
END
```

Parallel Port 0
Port 0
P0: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑
Pins: 0xFF ☑☑☑☑☑☑☑☑

Parallel Port 1
Port 1
P1: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑
Pins: 0xFF ☑☑☑☑☑☑☑☑

Parallel Port 2
Port 2
P2: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑
Pins: 0xFF ☑☑☑☑☑☑☑☑



Registers

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
PC $	C:0x0000
auxr1	0x00
dptr	0x0000
states	0
sec	0.00000000
psw	0x00

Disassembly

```
C:0x0000    7455     MOV      A,#0x55
     3: MOV P0,A
C:0x0002    F580     MOV      P0(0x80),A
     4: MOV P1,A
```

T2_E3.asm

```
ORG 0000H
HERE: MOV A,#55H
MOV P0,A
MOV P1,A
ACALL DELAY
MOV A,#0AAH
MOV P0,A
MOV P1,A
MOV P2,A
ACALL DELAY
SJMP HERE
DELAY: MOV R0,#04H
BACK: DJNZ R0,BACK
RET
END
```

Parallel Port 2

Port 2 — 7 Bits 0

P2: 0xAA ☑☐☑☐☑☐☑☐

Pins: 0xAA ☑☐☑☐☑☐☑☐

Parallel Port 1

Port 1 — 7 Bits 0

P1: 0x55 ☐☑☐☑☐☑☐☑

Pins: 0x55 ☐☑☐☑☐☑☐☑

Parallel Port 0

Port 0 — 7 Bits 0

P0: 0x55 ☐☑☐☑☐☑☐☑

Pins: 0x00 ☐☐☐☐☐☐☐☐

UART #1

Result—

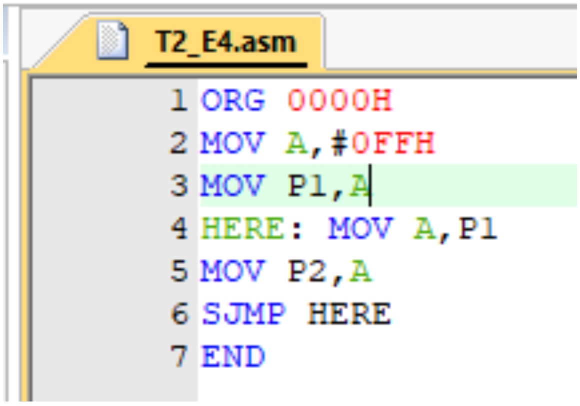
Hence all bits of P0,P1 and P2 are continuously being toggled by sending 55H and AAH to these port.

Program 4: Ports

Aim: To get data from P1 port and send it to port P2.

Software Requirement: Keil Software

Program:

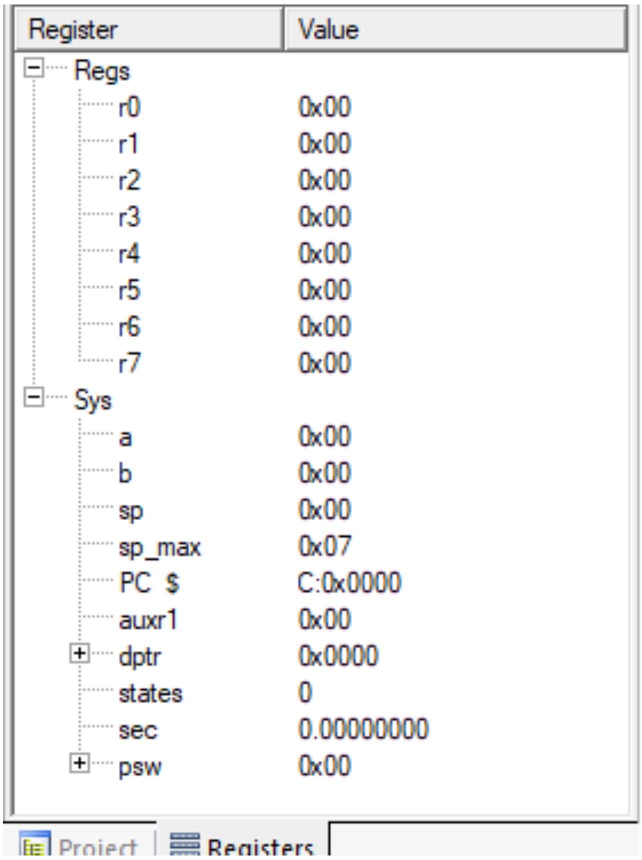


T2_E4.asm

```
ORG 0000H
MOV A,#0FFH
MOV P1,A
HERE: MOV A,P1
MOV P2,A
SJMP HERE
END
```

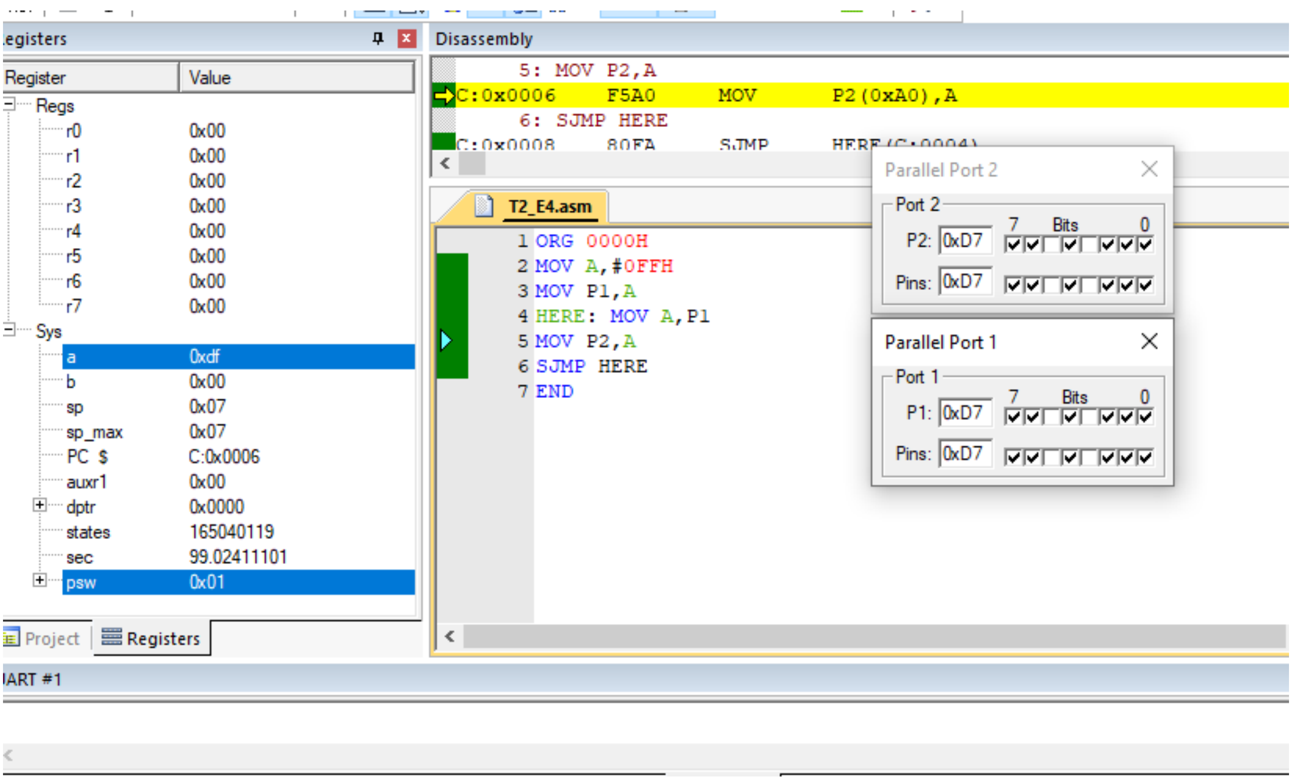
Output:

Before Execution Register status:



```
Register        Value
Regs
   r0           0x00
   r1           0x00
   r2           0x00
   r3           0x00
   r4           0x00
   r5           0x00
   r6           0x00
   r7           0x00
Sys
   a            0x00
   b            0x00
   sp           0x00
   sp_max       0x07
   PC $         C:0x0000
   auxr1        0x00
   dptr         0x0000
   states       0
   sec          0.00000000
   psw          0x00
```

After Execution:



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0xdf
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0006
auxr1	0x00
dptr	0x0000
states	165040119
sec	99.02411101
psw	0x01

```
     5: MOV P2,A
C:0x0006    F5A0     MOV      P2(0xA0),A
     6: SJMP HERE
C:0x0008    80FA     SJMP     HERE(C:0004)
```

```
ORG 0000H
MOV A,#0FFH
MOV P1,A
HERE: MOV A,P1
MOV P2,A
SJMP HERE
END
```

Parallel Port 2
Port 2
P2: 0xD7 7 Bits 0 ☑☑□☑□☑☑☑
Pins: 0xD7 ☑☑□☑□☑☑☑

Parallel Port 1
Port 1
P1: 0xD7 7 Bits 0 ☑☑□☑□☑☑☑
Pins: 0xD7 ☑☑□☑□☑☑☑

Result—

Hence, the P1 has been transformed as an input port and the data collected from it is continuously being sent to port P2. This is being done continuously.