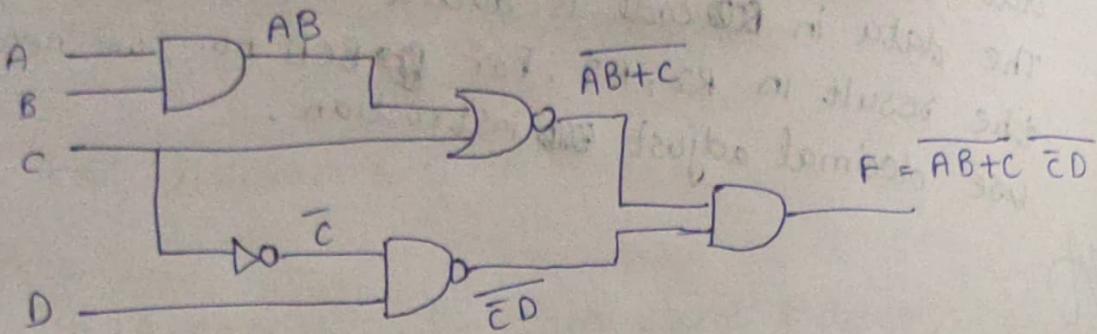


## Exp 2.1a

Aim: To develop, execute a program for 8051 microcontroller for following digital circuit design



Software used: Keil uVision 5.0

Program : ORG 0000H ; set the origin of Program at address at 0000H

```

SETB ACC.0      ; set bit 0 of accumulator
SETB ACC.1      ; set bit 1 of accumulator
CLR ACC.2       ; clear bit 2 of accumulator
SETB ACC.3       ; set bit 3 of accumulator
                  (Input:- 1101)
MOV C,ACC.0      ; move bit 0 of Acc into carry flag
ANL C,ACC.1      ; Perform bitwise AND between
                  carry and bit 1 of Acc.
ORL C,ACC.2      ; Perform OR between C,bit 2 of Acc
CPL C            ; Complement the carry flag
MOV ACC.7,C      ; move carry flag value into bit 7 of
                  Accumulator.
MOV C,ACC.2      ; move bit 2 of A into carry flag
CPL C            ; Complement the carry flag
MOV ACC.3        ; store value of acc in bit 3 of acc
ANL C,ACC.3      ; Perform bitwise AND between
                  carry & bit 3 of Acc
CPL C            ; Complement the carry flag
ANL ACC.7,C      ; Perform and between Acc & carry flag

```

HALT: stops HALT ; infinite loop to halt condition  
END : End of Program

manual (Theoretical)

calculation :

Input A=1; B=1; C=0; D=0

$$AB = 1$$

$$AB + C = 1$$

$$\overline{AB+C} = 0$$

$$C = 0$$

$$\bar{C} = 1$$

$$\overline{CD} = 1$$

$$\overline{\overline{CD}} = 0$$

$$F = \overline{AB+C} \cdot \overline{\overline{CD}}$$

$$F = 0 \iff \underline{\underline{C\bar{y} = 0}}$$

Practical  
output

$$C\bar{y}(B607) = 0$$

carry bit

Activities Kell uVision5

Feb 5 23:39

C:\users\matlab\Desktop\22bct0059.uvproj - uVision [Non-Commercial Use License]

SR1 HALT ;S0MP HALT

C:00011B	00	NOP
C:00011C	00	NOP
C:00012D	00	NOP
C:00011E	00	NOP
Instruction	00	00

22BCT0059.asm

```
1 ORG (0000)
2 CLR ACC:0
3 CLR ACC:1
4 CLR ACC:2
5 CLR ACC:3
6 MOV C,ACC:0
7 ANL C,ACC:1
8 ORL C,ACC:2
9 XRL C,ACC:3
10 MOV ACC:7,C
11 MOV ACC:7,C
12 CPL C
13 ANI C,ACC:3
14 CPL C
15 ANI C,ACC:7
16 HALT ;S0MP HALT
17 END
```

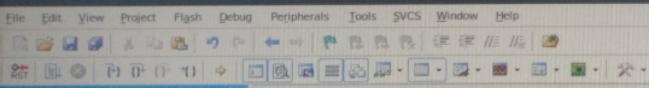
Call Stack + Locals

Name	Locality/Value	Type
22BCT0059	C:000003	

ARM ASSEMBLY Breakable Breakable Breakable Breakable Breakable Breakable Breakable Breakable COVERAGE COVFILE DEFINE DIN Display



It is dhruv



Register	Value
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
dptr	0x0000
PC \$	C:0x0019
states	16592793
sec	4.97783840
psw	0x01
p	1
f1	0
ov	0
rs	0
r0	0
ac	0
cy	0

16: HALT: SJMP HALT  
C:0x0019 80FE SJMP HALT(C:0019)

22BCT0059.asm

```

1 ORG 0000H
2 SETB ACC.0
3 SETB ACC.1
4 CLR ACC.2
5 SETB ACC.3
6 MOV C,ACC.0
7 ANL C,ACC.1
8 ORL C,ACC.2
9 CPL C
10 MOV ACC.7,C
11 MOV C,ACC.2
12 CPL C
13 ANL C,ACC.3
14 CPL C
15 ANL C,ACC.7
16 HALT: SJMP HALT
17 END

```



Running with Code Size Limit: 2K  
Load "C:\\users\\matlab\\Desktop\\Objects\\22bct0059"

Name	Location/Value	Type
22BCT0059	C:0x0019	

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display

### Exp 2.1 b

Aim: To develop and execute assembly language program for 8051 micro-controller to implement half adder and half subtractor. Store sum/diff in PSW.1, carry in PSW.5 and borrow in ACC.0.

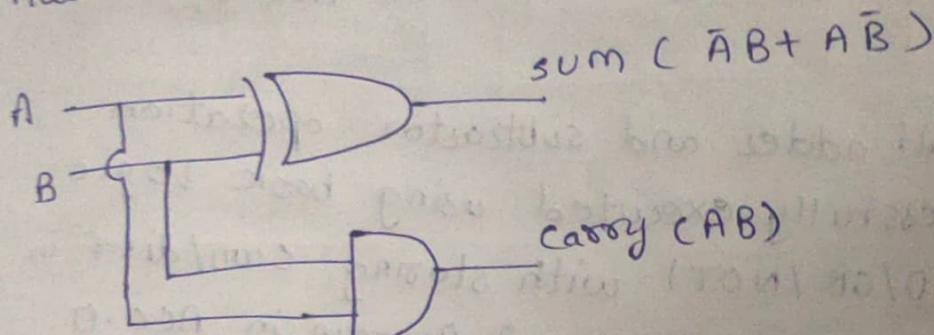
Software: Keil UVision 5.0 used.

Program:

```
ORG 0000H ; set origin to address 0000H
CLR ACC.0 ; clear bit 0 of Accumulator
CLR ACC.1 ; clear bit 1 of Accumulator
MOV C, ACC.0 ; move bit 0 of A into carry flag
ANL C, ACC.1 ; Perform AND b/w carry and bit 1 of A
MOV PSW.5, C ; store value of C in PSW.5
CPL ACC.1 ; complement bit 1 of Acc
MOV C, ACC.0 ; move bit 0 of A into carry flag
ANL ACC.0 C, ACC.1 ; Perform bitwise AND b/w Acc and carry flag
MOV ACC.7, C ; move carry flag value into Acc
CPL ACC.0 ; complement bit 0 of ACC
MOV ACC.0 ; store carry flag value into Acc
MOV ACC.0
CPL ACC.1 ; complement bit 1 of A
MOV C, ACC.0 ; mov bit 0 of A into carry
ANL ACC.0 C, ACC.1 ; Perform bitwise AND b/w A8
MOV ACC.0, C ; move value of C into 0 bit of A
ORL ACC.0 C, ACC.7 ; Perform bitwise OR b/w bit 1 of A and carry flag
MOV PSW.1, C ; store final carry flag value into bit 1 of PSW
HALT: SJMP HALT ; Infinite loop to halt execution
END ; End of Program
```

manual / Theoretical calculation

half adder



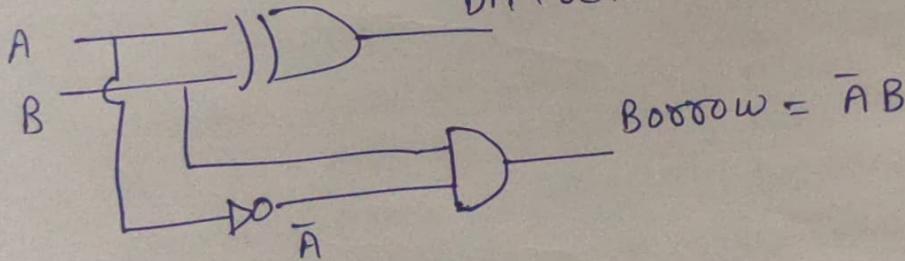
Truth table (Half adder)

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table (Half subtractor)

A	B	diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

half subtractor



we have take CLR ACC.0, CLR ACC.1 ∴ 00

∴ According to truth table.

for 00 → sum = 0 ∴ PSW.1 = 0  
or  
diff = 0

Practical output  
PSW.1 = 0

carry = 0 ∴ PSW.5 = 0 PSW.5 = 0  
borrow = 0 ∴ ACC.0 = 0 ACC.0 = 0

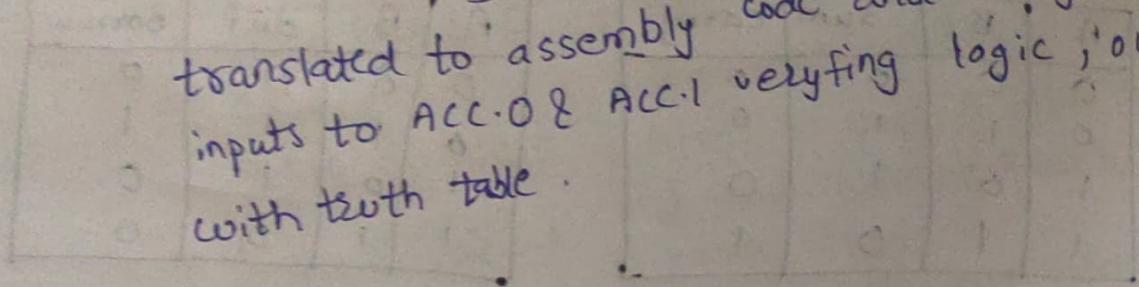
for CLR Acc.0, SETB ACC.1

## Practical output

~~Q1~~  $\Rightarrow$  sum/difference = 1  
carry = 0  
borrow = 1

PSW. T = 1  
PSW. S = 0  
ACL. O = 1

~~A~~ Conclusion: The half adder and subtractor operation were successfully executed using basic logic gates (AND/OR/NOT) with storing sum/diff in PSW.1, carry in PSW.5 & Borrow in ACC.0. Additionally given digital circuit was accurately translated to assembly code and assigning inputs to ACC.0 & ACC.1 verifying logic, output with truth table.





Dhruv Agrawal

22BCT0059

HOSTELLER

Registers      Value

R0	0x0000
R1	0x0000
R2	0x0000
R3	0x0000
R4	0x0000
R5	0x0000
R6	0x0000
R7	0x0000
S0	0x0000
S1	0x0000
S2	0x0000
PC	0x0000
status	19707274
SPC	511218279
SW	0x0000
R	0
R1	0
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0

18: HALT: SJMP HALT

C:\0x0022	00F	00F	HALT(C:0020)
C:\0x0022	00	NOP	
C:\0x0023	00	NOP	
C:\0x0024	00	NOP	
C:\0x0025	00	NOP	

22BCT0059.asm

```
1 ORG 0000H
2 CLR ACC.0
3 CLR ACC.1
4 MOV CACC.0
5 ANL CACC.1
6 MOV PSW.5.C
7 CPL ACC.1
8 MOV CACC.0
9 ANL CACC.1
10 MOV ACC.7.C
11 CPL ACC.0
12 CPL ACC.1
13 MOV CACC.0
14 ANL CACC.1
15 MOV ACC.0.C
16 ORL CACC.7
17 MOV PSW.1.C
18 HALT: SJMP HALT
19 END
```

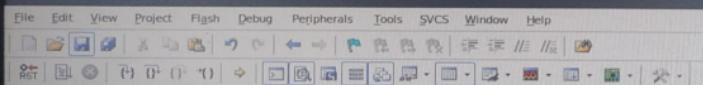
Command

Running with Code Size Limit: 2K

Last used "C:\Users\matlab\Desktop\Objects\22bct0059"

Call Stack + Locals

Name	Location/Value	Type
22BCT0059	C:000000	



**Registers**

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00

**Sys**

a	0x03
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC	\$ C:0x0020
states	16195288
sec	4.85858689
psw	0x82
p	0
f1	1
ov	0
rs	0
f0	0
ac	0
cy	1

**Disassembly**

```

18: HALT: SJMP HALT
C:0x0020 80FE SJMP HALT(C:0020)
C:0x0022 00 NOP
C:0x0023 00 NOP
C:0x0024 00 NOP
C:0x0025 00 NOP
C:0x0026 00 NOP

```

**22BCT0059.asm**

```

1 ORG 0000H
2 CLR ACC.0
3 SETB ACC.1
4 MOV C,ACC.0
5 ANL C,ACC.1
6 MOV PSW,5,C
7 CPL ACC.1
8 MOV C,ACC.0
9 ANL C,ACC.1
10 MOV ACC.7,C
11 CPL ACC.0
12 CPL ACC.1
13 MOV C,ACC.0
14 ANL C,ACC.1
15 MOV ACC.0,C
16 ORL C,ACC.7
17 MOV PSW,1,C
18 HALT: SJMP HALT
19 END

```

**Project**   **Registers**

**Command**

Running with Code Size Limit: 2K

Load "C:\users\matlab\Desktop\Objects\22bct0059"

Name	Location/Value	Type
22BCT0059	C:0x0020	

Call Stack + Locals

Call Stack + Locals

Memory 1





VIT

Vellore Institute of Technology  
One of the leading universities in the Asia-Pacific region

Dhruv Agrawal

22BCT0059

HOSTELLER

Registers

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x03
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0020
states	16195288
sec	4.85858683
psw	0x82
p	0
f1	1
ov	0
rs	0
r0	0
ac	0
cy	1

Disassembly

```

18: HALT: SJMP HALT
C:0x0020 80FE SJMP HALT(C:0:0020)
C:0x0022 00 NOP
C:0x0023 00 NOP
C:0x0024 00 NOP
C:0x0025 00 NOP
C:0x0026 00 NOP

```

22BCT0059.asm

```

1 ORG 0000H
2 CLR ACC.0
3 SETB ACC.1
4 MOV C,ACC.0
5 ANL C,ACC.1
6 MOV PSW.5,C
7 CPL ACC1
8 MOV C,A ACC.1 (D:0xE0.1) = 1
9 ANL C,ACC.1
10 MOV ACC.7,C
11 CPL ACC.0
12 CPL ACC.1
13 MOV C,ACC.0
14 ANL C,ACC.1
15 MOV ACC.0,C
16 ORL C,ACC.7
17 MOV PSW.1,C
18 HALT: SJMP HALT
19 END

```

Project Registers

Command

Running with Code Size Limit: 2K  
Load "C:\users\matlab\Desktop\Objects\22bct0059"

Call Stack + Locals

Name	Location/Value	Type
22BCT0059	C:0x0020	

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVTOFILE DEFINE DIR Display

## Exp 2.2 a

Aim: To develop and execute assembly lang program to toggle all bits of P0, P1, P2 continuously sending 55H and AAH to these ports. Also put delay b/w ON and OFF states.

Software : Keil uVision 5.0  
used

Program :

```
ORG 0000H ; Set origin to address 0000H
HERE: MOV P0, #55H ; Turn off all LEDs. (all bits 0101)
      MOV P1, #55H ; Turn off all LEDs (all bits 0101)
      MOV P2, #55H ; Turn off all LEDs (all bits 0101)
      ACALL DELAY ; call delay subroutine.

      MOV P0, #0AAH ; Turn on all EED (Toggle)
      MOV P1, #0AAH ; Turn on all LED (Toggle).
      MOV P2, #0AAH ; Turn on all LED (Toggle)
      ACALL DELAY ; call delay subroutine.

      SJMP HERE ; Infinite loop to repeat
                  ; the process.

DELAY: MOV R1, #04H ; move 04H in R1
BACK:  MOV R2, #20H ; move 20H in R2.
AGAIN: DJNZ R2, AGAIN ; Decrement R2 until R2=0
      -          DJNZ R1, BACK. ; Decrement R1 until R1=0
      RET         ; Return delay
```

MANUAL / Theoretical:  
calculation

Port 1							
1	0	1	1	1	0	0	0
7	6	5	4	3	2	1	0
LSB	MSB						

Port 2							
1	0	1	1	1	0	0	0
7	6	5	4	3	2	1	0
LSB	MSB						

Port 3							
1	0	1	1	1	0	0	0
7	6	5	4	3	2	1	0
LSB	MSB						

Port 1							
1	0	1	1	1	0	0	0
7	6	5	4	3	2	1	0

Port 2							
1	0	1	1	1	0	0	0
7	6	5	4	3	2	1	0

Port 3							
1	0	1	1	1	0	0	0
7	6	5	4	3	2	1	0



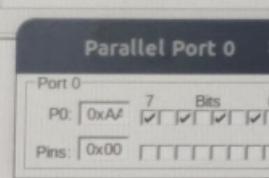
Registers	
Register	Value
r0	0x00
r1	0x01
r2	0x06
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x09
sp_max	0x09
dptr	0x0000
PC \$	C:0x001C
states	472298382
sec	141.68952877
psw	
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Disassembly

```

13: AGAIN:DJNZ R2, AGAIN
    :0x001C DAFE DJNZ R2, AGAIN(C:001C)
14: DJNZ R1, BACK
    :0x001E D9FA DJNZ R1, BACK(C:001A)
15: RET
    :0x0020 22 RET
    :0x0021 00 NOP

```

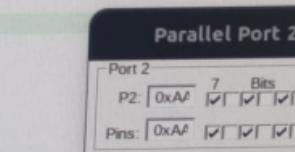
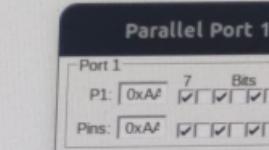


2\_2A.asm

```

1 ORG 0000H
2 HERE: MOV P0,#55H
3 MOV P1,#55H
4 MOV P2,#55H
5 ACALL DELAY
6 MOV P0,#0AAH
7 MOV P1,#0AAH
8 MOV P2,#0AAH
9 ACALL DELAY
10 SJMP HERE
11 DELAY: MOV R1,#04H
12 BACK:MOV R2,#20H
13 AGAIN:DJNZ R2, AGAIN
14 DJNZ R1, BACK
15 RET

```



File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

	Value
r0	0x00
r1	0x04
r2	0x14
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x09
sp_max	0x09
dptr	0x0000
PC \$	C:0x001C
states	419709394
sec	125.91283079
psw	0x00
p	0
f1	0
ov	0
rs	0
r0	0
ac	0
cy	0

Disassembly

```

13: AGAIN:DJNZ R2, AGAIN
  :0x001C DAFE DJNZ R2, AGAIN(C:001C)
14: DJNZ R1, BACK
  :0x001E D9FA DJNZ R1, BACK(C:001A)
15: RET
  :0x0020 22 RET
  :0x0021 00 NOP

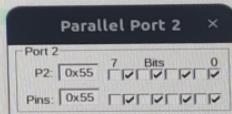
```

2\_2A.asm

```

1 ORG 0000H
2 HERE: MOV P0,#55H
3 MOV P1,#55H
4 MOV P2,#55H
5 ACALL DELAY
6 MOV P0,#0AAH
7 MOV P1,#0AAH
8 MOV P2,#0AAH
9 ACALL DELAY
10 SJMP HERE
11 DELAY: MOV R1,#04H
12 BACK:MOV R2,#20H
13 AGAIN:DJNZ R2, AGAIN
14 DJNZ R1, BACK
15 RET

```



Project Registers

Command

Running with Code Size Limit: 2K  
Load "C:\\users\\matlab\\Desktop\\22bct0059\\Objects\\22BCT0059"

Call Stack + Locals

Name	Location/Value	Type
+ 2_2A	C:0x001C	

## Exp 2.2 b

Aim: To design and implement assembly language program to get data from Port 1 and send it to port 2

Software used : Keil uVision 5.0

Program:

```

MOV A, #0FFH
MOV P1, A

```

HERE: MOV A, P1

MOV P2, A

SJMP here

END

- Move value FFH into accumulator
- move P1 in Accumulator (value)
- get data from P1
- send it to P2
- infinite loop

Manual /  
Theoretical calculation:

Port 1							
1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0

Port 2							
1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0

Port 1							
1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0

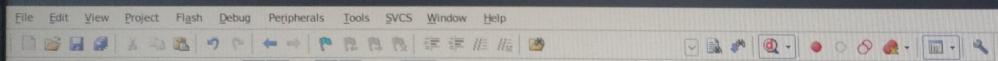
Port 2							
1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0

Activities

Keil uVision5

Feb 12 23:32

C:\users\matlab\Desktop\22bct0059\22BCT0059.uvproj - µVision [Non-Commercial Use License]



Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0xffff
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0006
states	96322123
sec	28.89663979
psw	0x00
p	0
f1	0
ov	0
rs	0
r0	0
ac	0
cy	0

```

Disassembly
      5:    MOV P2,A
      6:    F5A0  MOV P2(0xA0),A
      7:    SJMP HERE
      8:    80FA  SJMP HERE(C:0004)
      9:    00    NOP
     10:   00    NOP
     11:   00    NOP
  
```

```

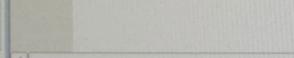
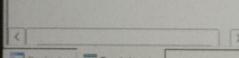
2_2A.asm
1 ORG 0000H
2 MOV A,#0FFH
3 MOV P1,A
4 HERE:MOV A,P1
5 MOV P2,A
6 SJMP HERE
7 END
  
```

## Parallel Port 1

Port 1	P1: 0xDF	7 Bits	0				
Pins:	0xDF	<input checked="" type="checkbox"/>					

## Parallel Port 2

Port 2	P2: 0xDF	7 Bits	0				
Pins:	0xDF	<input checked="" type="checkbox"/>					



Command

Running with Code Size Limit: 2K

Load "C:\\users\\matlab\\Desktop\\22bct0059\\Objects\\22BCT0059"

Call Stack + Locals

Name	Location/Value	Type
------	----------------	------

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display

Call Stack + Locals

Memory 1

Simulation



File Edit View Project Flash Debug Peripherals Tools SVCS Window Help



## Registers

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0xdd
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0006
states	441403775
sec	132.42114574
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

## Disassembly

```

5:      MOV P2,A
       F5A0    MOV P2(0xA0),A
6:      SJMP HERE
       ::0x0008  80FA    SJMP HERE(C:0004)
       ::0x000A  00      NOP
       ::0x000B  00      NOP
       ::0x000C  00      NOP

```

```

1 ORG 0000H
2 MOV A,#0FFH
3 MOV P1,A
4 HERE:MOV A,P1
5 MOV P2,A
6 SJMP HERE
7 END

```

## Parallel Port 1

Port 1—  
 P1: 0xDE 7 Bits 0  
 Pins: 0xDE ✓✓✓✓✓✓✓✓

## Parallel Port 2

Port 2—  
 P2: 0xDE 7 Bits 0  
 Pins: 0xDE ✓✓✓✓✓✓✓✓



Project Registers

Command  
Running with Code Size Limit: 2K  
Load "C:\users\matlab\Desktop\22bct0059\Objects\22BCT0059"

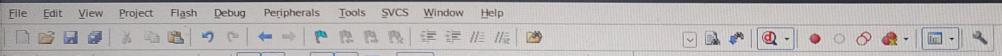
Call Stack + Locals

Name	Location/Value	Type
------	----------------	------

Activities Keil uVision5

Feb 12 23:34

C:\users\matlab\Desktop\22bct0059\22BCT0059.uvproj - uVision [Non-Commercial Use License]



Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
<hr/>	
Sys	
a	0xfd
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0006
states	769107175
sec	230.73217557
<hr/>	
psw	0x01
p	1
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

```

Disassembly
5:    MOV P2,A
       F5A0  MOV P2(0xA0),A
6:    SJMP HERE
       :0x0008 80FA  SJMP HERE(C:0004)
       :0x000A 00    NOP
       :0x000B 00    NOP
       :0x000C 00    NOP

2_2A.asm
1 ORG 0000H
2 MOV A,#0FFH
3 MOV P1A
4 HERE-MOV A,P1
5 MOV P2,A
6 SJMP HERE
7 END

```



Project Registers

Command  
Running with Code Size Limit: 2K  
Load "C:\\users\\matlab\\Desktop\\22bct0059\\Objects\\22BCT0059"

Call Stack + Locals

Name Location/Value Type

Call Stack + Locals Memory 1

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display



## Exp 2.2 C

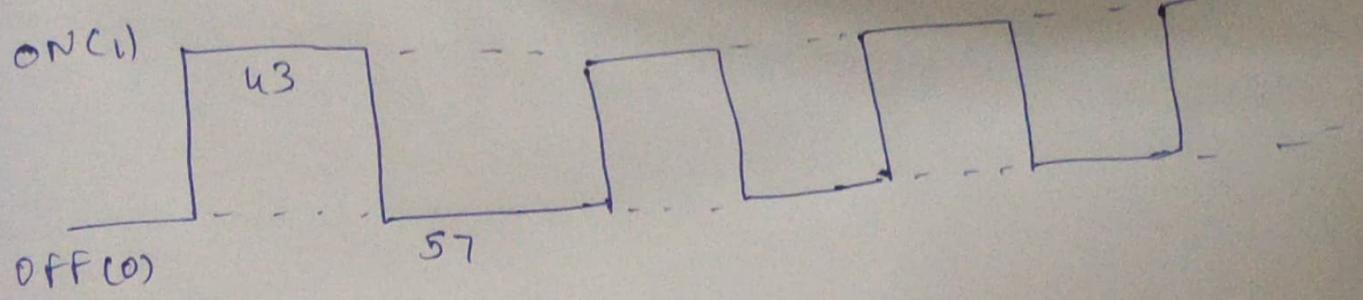
Aim: To design and implement assembly code for 8051 microcontroller to generate waveforms of duty cycle ~~43%~~ 43%. without using timer.

Software : Keil uVision 5.0  
used

Program

ORG 0000H	; set origin to 0000H
HERE: MOV P1, #0FFH	; set P1 as input port
MOV R2, #2BH	; move 2BH value in R2
BACK1: ACALL DELAY	; call delay subroutine
DJNZ R2, Back1	; Decrement R2 until R2=0 goto BACK1
MOV P1, #00H	; move value 00H in P1
MOV R3, #39H	; move value 39H in R3
BACK2: ACALL DELAY	; call delay subroutine
DJNZ R3, BACK2	; Decrement R3 until R3=0 goto BACK2
SJMP HERE	; Infinite loop.
DELAY: MOV R1, #04H	; Move value 04H in R1
BACK: DJNZ R1, BACK	; Decrement R1 goto BACK until R1=0
RET	; Return to delay (ISR)

Manual/Theoretical  
calculation :



Q3  $\frac{43}{100} \Rightarrow \text{Duty cycle} = \frac{T_{ON}}{T_{ON} + T_{OFF}}$

$$\frac{43}{100} = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$

$$\therefore T_{ON} = 43; T_{OFF} = 57.$$

$$T_{ON} = 28\text{Hz}; T_{OFF} = 39\text{Hz}$$

Practical output  
 $T_{ON} = 42.86$   
 $T_{OFF} = 57.14$

~~Conclusion:~~ we implement assembly lang program to toggle bits of Port 0, Port 1, Port 2 using STH & AAH value. In second Program we defined Port 1 as input Port and passed same value in output Port 2 using Accumulator. In third program we generated wave cycle for duty = 43%. and clock frequency 110592Hz without using timer. We used delay subroutine & DJNZ only.

Logic Analyzer

Setup Load Save... Min Time Max Time Grid Zoom In Out All Min/Max Update Screen Transition Jump to Signal Info Amplitude Timestamps Enable

Zoom

In

Out

All

Min/Max

Update

Screen

Transition

Jump to

Code

Trace

Prev

Next

Start

Clear

Undo

Signal Info

Amplitude

Timestamps

Enable

Show Cycles

Cursor

1

P1

0

Y: 1, d:y: 0

Y: 0

968.8542 s 3229513780

968.8546 s 3229514919, d: 215

968.8558 s 3229519113

3229514919, d: 215

Disassembly Logic Analyzer

2\_2.asm

```

1 ORG 0000H
2 HERE: MOV P1,#0FFH
3 MOV R2,#2BH
4 BACK1:ACALL DELAY
5 DJNZ R2,BACK1
6 MOV P1,#00H
7 MOV R3,#39H
8 BACK2:ACALL DELAY
9 DJNZ R3,BACK2
10 SJMP HERE
11 DELAY: MOV R1,#04H
12 BACK: DJNZ R1,BACK
13 RET
14

```

**Parallel Port 1**

Port 1

P1:	0x00	7 Bits	0
Pins:	0x00		

Dhruv Agrawal

22BCT0059

HOSTELLER

15

Limit: 2K  
\\Desktop\\22bct0059\\Objects\\22BCT0059"

Call Stack + Locals

Name	Location/Value	Type
+ 2_2A	C:0x0005	

File BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display

Call Stack + Locals

Memory 1

Activities Keil uVisions

Feb 12 23:51

C:\users\matlab\Desktop\22bct0059\22BCT0059.uvproj - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Logic Analyzer

Register	Value
r0	0x00
r1	0x04
r2	0x29
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00

Sys

a	0x00
b	0x00
sp	0x09
sp_max	0x09
dptr	0x0000
PC \$	C:0x0016
states	558426008
sec	167.52781915
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Parallel Port 1

Port 1	P1 [0x00]	7 Bits	0
Pins	[0x00]		

2\_2A.asm

```

1 ORG 0000H
2 HERE: MOV P1,#0FFH
3 MOV R2,#2BH
4 BACK1:ACALL DELAY
5 DJNZ R2,BACK1
6 MOV P1,#00H
7 MOV R3,#39H
8 BACK2:ACALL DELAY
9 DJNZ R3,BACK2
10 SJMP HERE
11 DELAY: MOV R1,#04H
12 BACK: DJNZ R1,BACK
13 RET
14

```

Project Registers

Command

Running with Code Size Limit: 2K  
Load "C:\\users\\matlab\\Desktop\\22bct0059\\Objects\\22BCT0059"  
LA (P1 & 0x00000001)

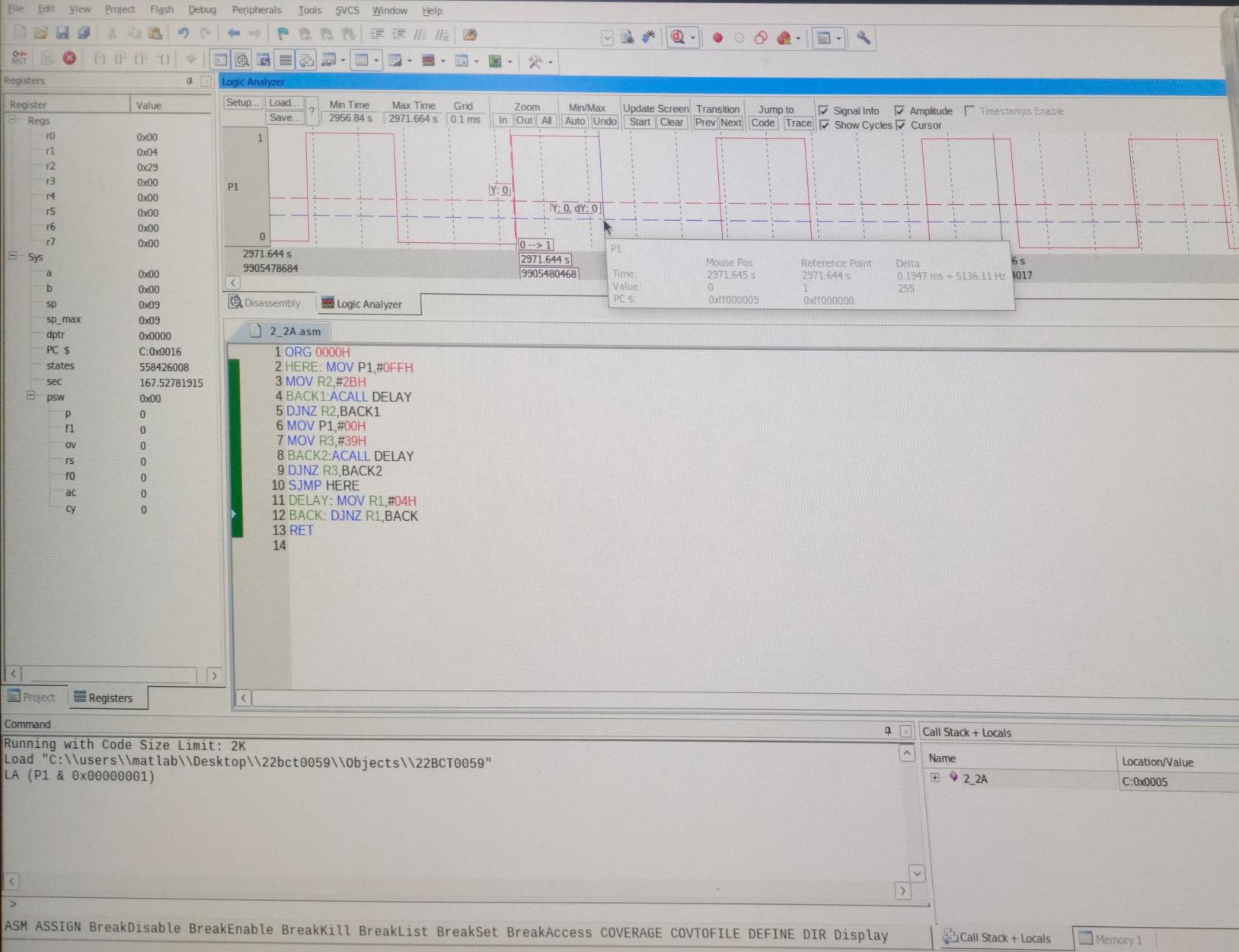
Call Stack + Locals

Name	Location/Value	Type
2_2A	C:0x0005	

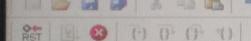
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display

VIT®  
Vellore Institute of Technology  
Grant to be University under section 3 of the UGC Act, 1956  
VELLORE CAMPUS

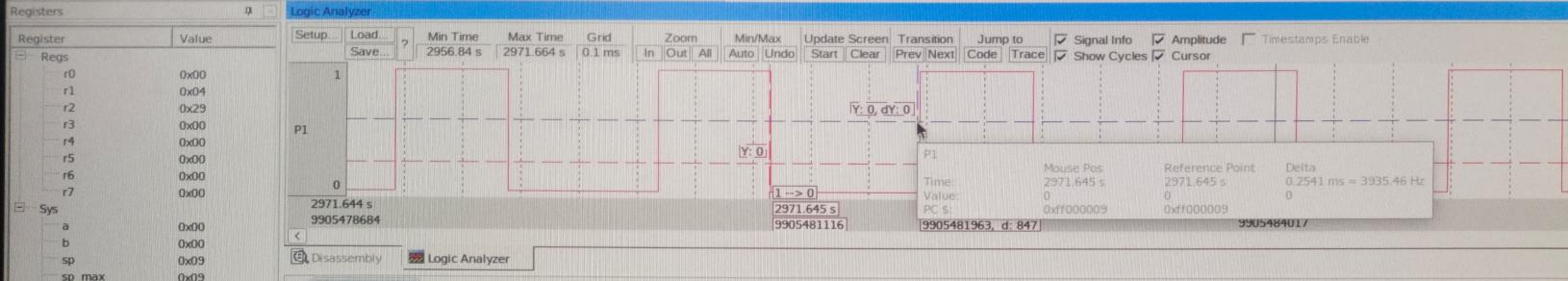
Dhruv Agrawal  
22BCT0059  
HOSTELLER



File Edit View Project Flash Debug Peripherals Tools SVCS Window Help



## Logic Analyzer



## Disassembly

## 2\_2A.asm

```

1 ORG 0000H
2 HERE: MOV P1,#0FFH
3 MOV R2,#2BH
4 BACK1:ACALL DELAY
5 DJNZ R2,BACK1
6 MOV P1,#00H
7 MOV R3,#39H
8 BACK2:ACALL DELAY
9 DJNZ R3,BACK2
10 SJMP HERE
11 DELAY: MOV R1,#04H
12 BACK: DJNZ R1,BACK
13 RET
14

```

Project Registers

Command

Running with Code Size Limit: 2K  
Load "C:\\\\users\\\\matlab\\\\Desktop\\\\22bct0059\\\\Objects\\\\22BCT0059"  
LA (P1 & 0x00000001)

## Call Stack + Locals

Name	Location/Value	Type
+ 2_2A	C:0x0005	