19BEC0358

VIT®

Vellore Institute of Technology

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

School of Electronics Engineering ,VIT, Vellore

Reg.No	19BEC0358		
Student Name	ARPIT PATAWAT		
Course Code	ECE3003	**Slot & Semester**	L35+L36 FALL -- 2021-22
Course Name	Microcontroller and its applications		
Program Title	PORTS AND TIMERS		
Date of Exp.	20-09-2021	**Date of Submission**	23-10-2021
Faculty	A.Karthikeyan		

Question

1. Write and assemble a program to toggle all the bits of P0, P1, and P2 continuously by sending 55H and AAH to these ports. Put a time delay between the "on" and "off" states.

   Then using the simulator, single-step through the program and examine the ports. Do not single-step through the time delay call.

2. Write and assemble a program to Get the Data From Port P1 and Send it to Port P2, Note:P1 as input Port and P2 as Output Port

3. Write a program using timer 1 mode 1 to generate a 1 kHz square wave frequency on one of the pins of P1. Then examine the frequency using the oscilloscope.

4. Write a program using timer 1 mode 1 to generate a 500Hz square wave frequency on one of the pins of P1. Then examine the frequency using the oscilloscope.

5. Assuming that clock pulses are fed into pin T1, write a program for counter 1 in mode 2 to count the pulses and display the state of the TL1 count on P2, which connects to 8 LEDs.

TASK 1) -

Aim: To write an 8051 ALP to perform to toggle all the bits of p0, p1 and p2 using keil software and to verify the result manually.

Tools Required: Keil Micro vision Software

Algorithm:

1. 55 is 0101 0101 and AA is 1010 1010
2. load 55 to the ports P0, P1 and P2
3. make a delay by using ACALL to move to subroutine
4. in delay programme, we are first decrementing R2 20 times and repeating this process 10 times (total 200 times)
5. after delay function execution, we are again moving back to our main function
6. load AA to ports P0, P1 and P2 (basically complementing)
7. again delaying and repeating this process from step 2

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	0000H					Defining origin of the program	NONE
HERE	MOV	P0, #55H	Immediate	1	2	Data Transfer	Load 55 to port 0	NONE
	MOV	P1, #55H	Immediate	1	2	Data Transfer	Load 55 to port 1	NONE
	MOV	P2, #55H	Immediate	1	2	Data Transfer	Load 55 to port 2	NONE
	ACALL	DELAY		2	2	Program branching	Absolute delay subroutine is called	NONE

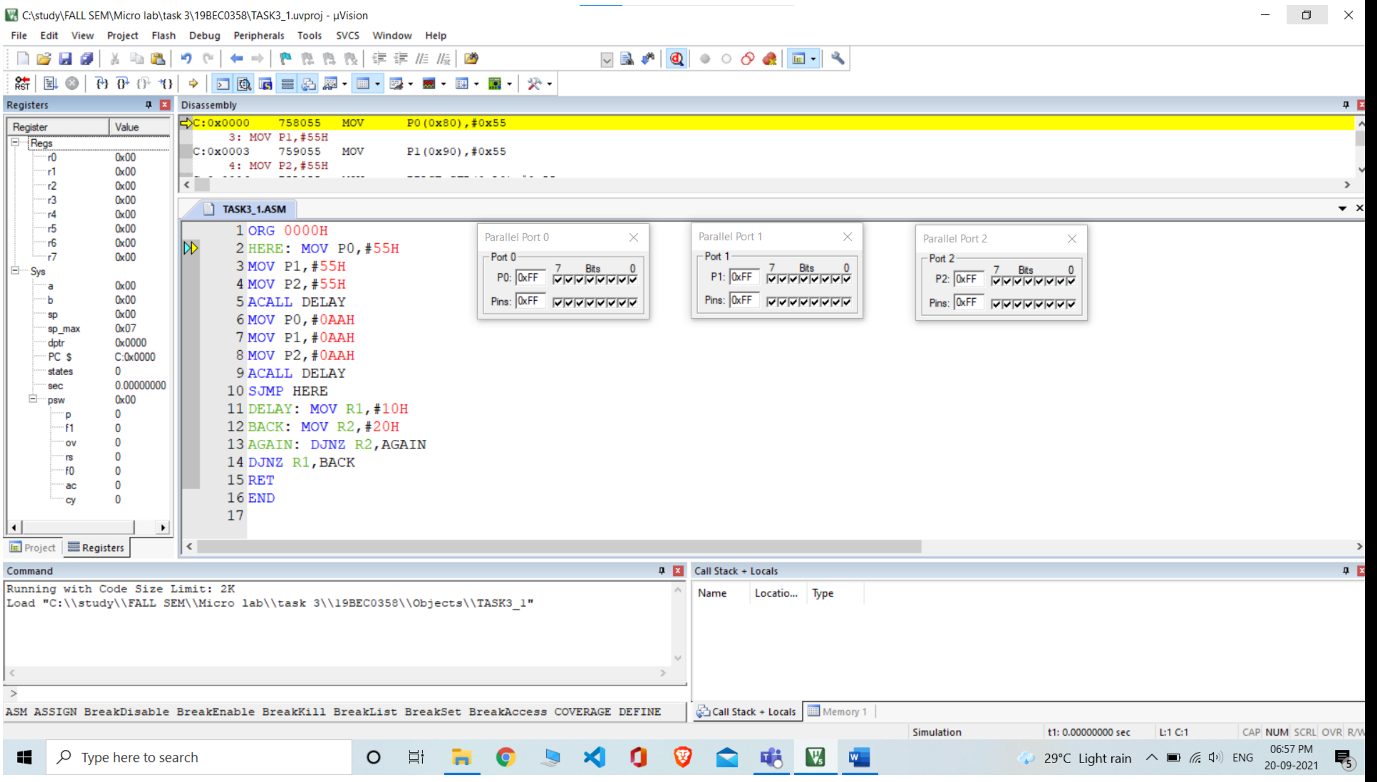
	MOV	P0, #0AAH	Immediate	1	2	Data Transfer	Load AA to port 0	NONE
	MOV	P1, #0AAH	Immediate	1	2	Data Transfer	Load AA to port 1	NONE
	MOV	P2, #0AAH	Immediate	1	2	Data Transfer	Load AA to port 2	NONE
	ACALL	DELAY		2	2	Program branching	Absolute delay subroutine is called	NONE
	SJMP	HERE		2	2	Program branching	Make an unconditional jump to HERE label	NONE
DELAY	MOV	R1, #10H	Immediate	1	2	Data Transfer	Load 10 to port R1	NONE
BACK	MOV	R2, #20H	Immediate	1	2	Data Transfer	Load 20 to port R2	NONE
AGAIN	DJNZ	R2, AGAIN	Direct	2	2	Program branching	Decrement R2 and jump to label if non zero	NONE
	DJNZ	R1, BACK	direct	2	2	Program branching	Decrement R2 and jump to label if non zero	NONE
	RET					Program branching	Return to instruction after ACALL	NONE
	END						End of the program	NONE

Output : Ports containing the Result: P0= P1= P2 = 55 or AA

Manual Calculation : NONE

Results and Observations

Program and registers before execution:



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

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
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Disassembly

```
C:0x0000    758055   MOV      P0(0x80),#0x55
     3: MOV P1,#55H
C:0x0003    759055   MOV      P1(0x90),#0x55
     4: MOV P2,#55H
```

TASK3_1.ASM

```
ORG 0000H
HERE: MOV P0,#55H
MOV P1,#55H
MOV P2,#55H
ACALL DELAY
MOV P0,#0AAH
MOV P1,#0AAH
MOV P2,#0AAH
ACALL DELAY
SJMP HERE
DELAY: MOV R1,#10H
BACK: MOV R2,#20H
AGAIN: DJNZ R2,AGAIN
DJNZ R1,BACK
RET
END
```

Parallel Port 0 — Port 0: P0: 0xFF, Pins: 0xFF, Bits 7–0: ☑☑☑☑☑☑☑☑ / ☑☑☑☑☑☑☑☑

Parallel Port 1 — Port 1: P1: 0xFF, Pins: 0xFF, Bits 7–0: ☑☑☑☑☑☑☑☑ / ☑☑☑☑☑☑☑☑

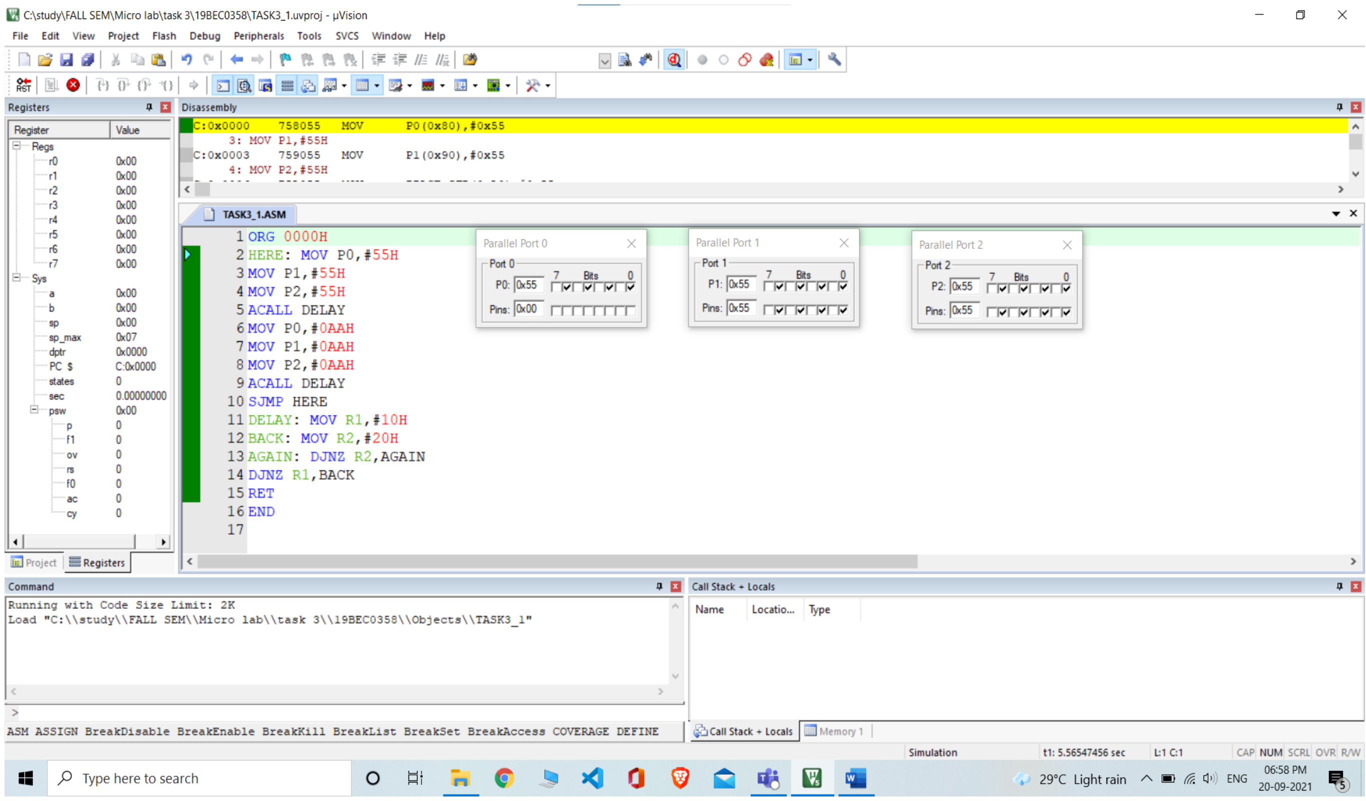
Parallel Port 2 — Port 2: P2: 0xFF, Pins: 0xFF, Bits 7–0: ☑☑☑☑☑☑☑☑ / ☑☑☑☑☑☑☑☑

Command

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
```

```
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
```

Program and registers after execution: FINAL STEP



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

Disassembly

```
C:0x0000    758055   MOV      P0(0x80),#0x55
     3: MOV P1,#55H
C:0x0003    759055   MOV      P1(0x90),#0x55
     4: MOV P2,#55H
```

TASK3_1.ASM

```
ORG 0000H
HERE: MOV P0,#55H
MOV P1,#55H
MOV P2,#55H
ACALL DELAY
MOV P0,#0AAH
MOV P1,#0AAH
MOV P2,#0AAH
ACALL DELAY
SJMP HERE
DELAY: MOV R1,#10H
BACK: MOV R2,#20H
AGAIN: DJNZ R2,AGAIN
DJNZ R1,BACK
RET
END
```

Parallel Port 0 — P0: 0x55, Pins: 0x00
Parallel Port 1 — P1: 0x55, Pins: 0x55
Parallel Port 2 — P2: 0x55, Pins: 0x55

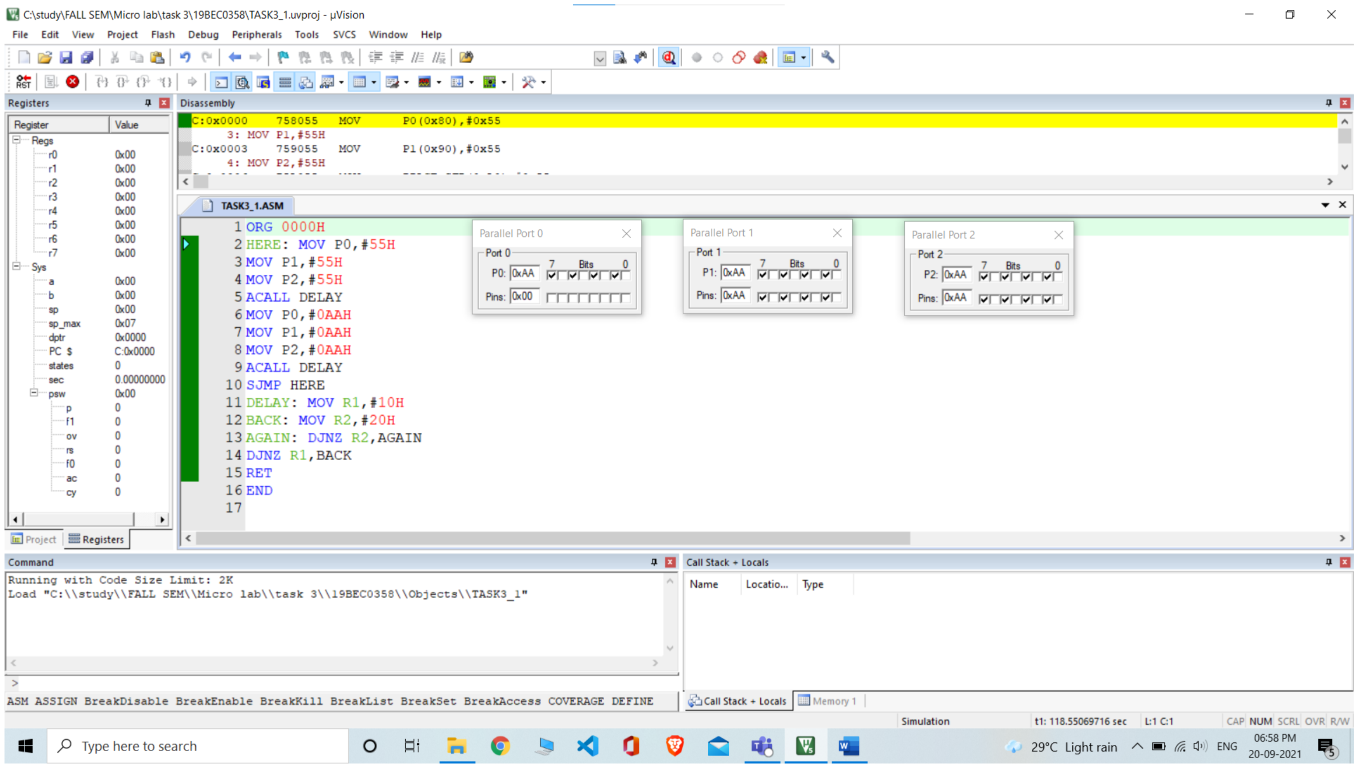
Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Command

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
```

Simulation t1: 5.56547456 sec L:1 C:1

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Project
Registers
Call Stack + Locals
Name Locatio... Type
Memory 1
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
Type here to search
29°C Light rain
ENG
06:58 PM
20-09-2021



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

Disassembly

```
C:0x0000    758055   MOV      P0(0x80),#0x55
     3: MOV P1,#55H
C:0x0003    759055   MOV      P1(0x90),#0x55
     4: MOV P2,#55H
```

TASK3_1.ASM

```
ORG 0000H
HERE: MOV P0,#55H
MOV P1,#55H
MOV P2,#55H
ACALL DELAY
MOV P0,#0AAH
MOV P1,#0AAH
MOV P2,#0AAH
ACALL DELAY
SJMP HERE
DELAY: MOV R1,#10H
BACK: MOV R2,#20H
AGAIN: DJNZ R2,AGAIN
DJNZ R1,BACK
RET
END
```

Parallel Port 0 — P0: 0xAA, Pins: 0x00
Parallel Port 1 — P1: 0xAA, Pins: 0xAA
Parallel Port 2 — P2: 0xAA, Pins: 0xAA

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Command

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
```

Simulation t1: 118.55069716 sec L:1 C:1

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Project
Registers
Call Stack + Locals
Name Locatio... Type
Memory 1
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
Type here to search
29°C Light rain
ENG
06:58 PM
20-09-2021

Inferences:

1. About PSW VALUES – they remain unchanged

2. ABOUT THE OUTPUT VALUES IN REGISTERS – registers need to be change for delay, the final

Result is in ports which is toggling

Result: the 8051 ALP to perform toggling of ports is executed using Keil software and the results are verified Manually.

TASK 2) -

Aim: To write an 8051 ALP to Get data from Port 1 and send to port 2 using keil software and to verify the result manually.

Tools Required : Keil Microvision Software

Algorithm:

1. make port 1 as input by sending FF (1111 1111)
2. load FF to Accumulator and then to port 1
3. now, move data from port 1 to accumulator
4. move data from accumulator to port 2
5. repeat from step 3.

Program:

Label	**Mnemonics**	**Operands**	**addressing mode used**	**Machine cycle Required**	**Memory Byte Required**	**Type of Instruction**	**Comments**	**Flags getting affected by the Instruction.**
	MOV	A, #0FFH	Immediate	1	2	Data transfer	Load FF to accumulator	NONE
	MOV	P1, A	Register	1	1	Data transfer	Load data from A to port 1 so that it can be used as input	NONE
HERE	MOV	A, P1	Register	1	1	Data transfer	Move data form port	NONE

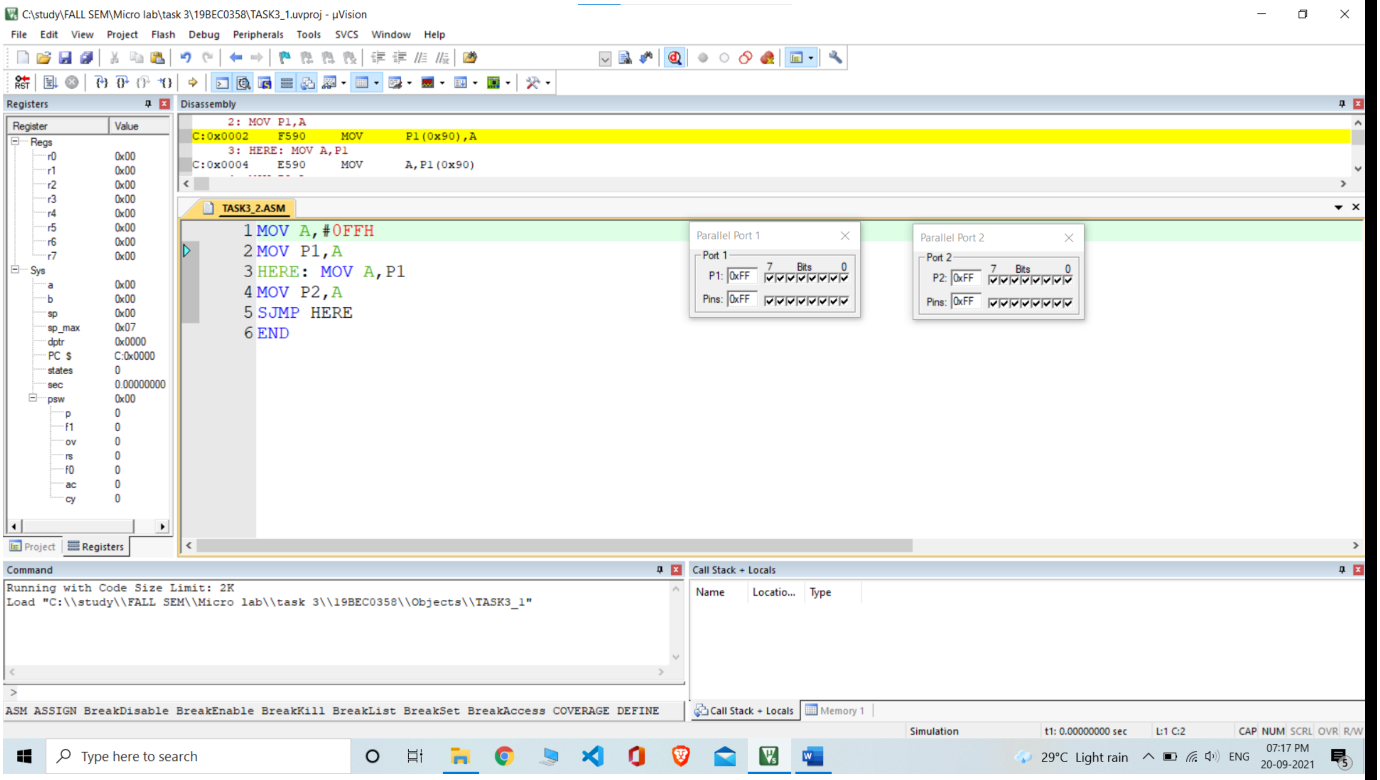
							1 to A	
	MOV	P2, A	Register	1	1	Data transfer	Move data from A to port 2	NONE
	SJMP	HERE		2	2	Program branching	Make unconditional jump to HERE label	NONE
	END						End of the program	NONE

Output : Ports containing the Result: P0 & P1

Manual Calculation : NONE

Results and Observations

Program and registers before execution:



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

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
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Disassembly

```
     2: MOV P1,A
C:0x0002    F590     MOV      P1(0x90),A
     3: HERE: MOV A,P1
C:0x0004    E590     MOV      A,P1(0x90)
```

TASK3_2.ASM

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

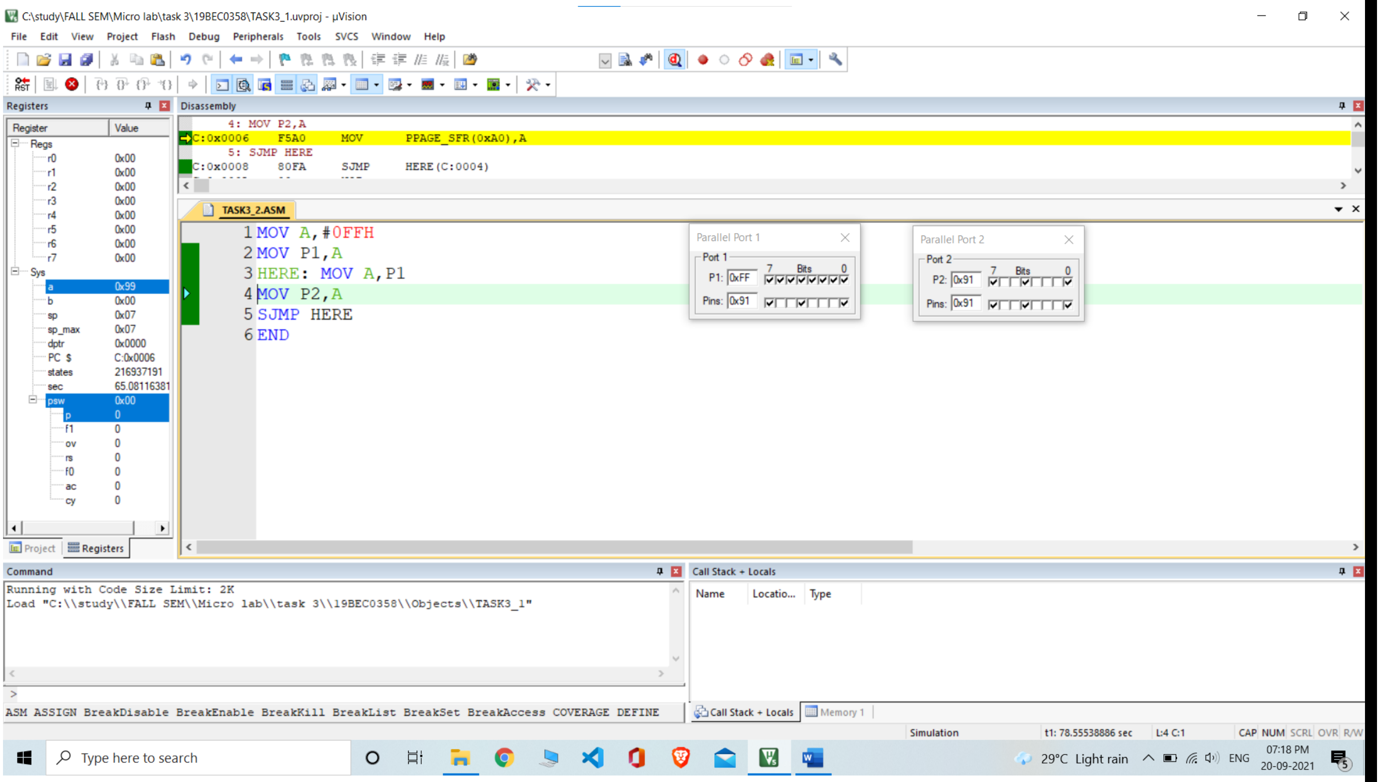
Parallel Port 1 — Port 1: P1: 0xFF, Pins: 0xFF

Parallel Port 2 — Port 2: P2: 0xFF, Pins: 0xFF

Command

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
```

Program and registers after execution: FINAL STEP



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x99
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x0006
states	216937191
sec	65.08116381
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

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

TASK3_2.ASM

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

Parallel Port 1
Port 1
P1: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑
Pins: 0x91 ☑□□☑□□□☑

Parallel Port 2
Port 2
P2: 0x91 7 Bits 0 ☑□□☑□□□☑
Pins: 0x91 ☑□□☑□□□☑

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
```

```
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
```

Name | Locatio... | Type

t1: 78.55538886 sec L:4 C:1

Inferences:

1. About PSW VALUES – they remain unchanged

2. ABOUT THE OUTPUT VALUES IN REGISTERS the final Result is in port 2 which is changed when port 1 is changed

Result: the 8051 ALP to data transfer from ports is executed using Keil software and the results are verified Manually.

TASK 3) -

Aim: To write an 8051 ALP using timer 1 mode 1 to generate a 1000Hz square wave frequency on one of the pins of P1 using keil software and to verify the result manually.

Tools Required : Keil Microvision Software

Algorithm:

1.calculate the time period from given frequency

2.using time period, calculate the value of count

3. subtract the count value from FFFF and load into timer 1

4. now compliment port 1 by using timer 1

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	000H					Starting of program	none
	MOV	TMOD, #10H	Immediate	1	2	Data transfer	Timer 1 mode 1 is selected (0001 0000)	None
HERE	MOV	TL1, #33H	Immediate	1	2	Data transfer	Lower bit of timer = 33	None
	MOV	TH1, #0FEH	Immediate	1	2	Data transfer	Higher bit of timer = FE	None
	CPL P1.0		Direct addressing	1	1	logical	Compliment value of port 1	none
	ACALL	DELAY		2	2	branching	Call delay subroutine	None

	SJMP	HERE		2	2	branching	Short jump to here label	None
DELAY	SETB	TR1	Direct	1	2	Boolean	Start timer	none
AGAIN	JNB	TF1, AGAIN	Direct	1	2	Boolean	Monitor timer flag, if it is 0, then go to again label	Timer flag
	CLR	TR1	Direct	1	1	Boolean	Stop timer	None
	CLR	TF1	Direct	1	1	Boolean	Reset timer flag	None
	RET		Assembler directive	1	1	Program Branching	Return	None
	END		Assembler directive				end	None

Output : Ports containing the Result: P1, in the form of waveform

Manual Calculation : Frequency = 1000Hz

T = 1/F = 1mS

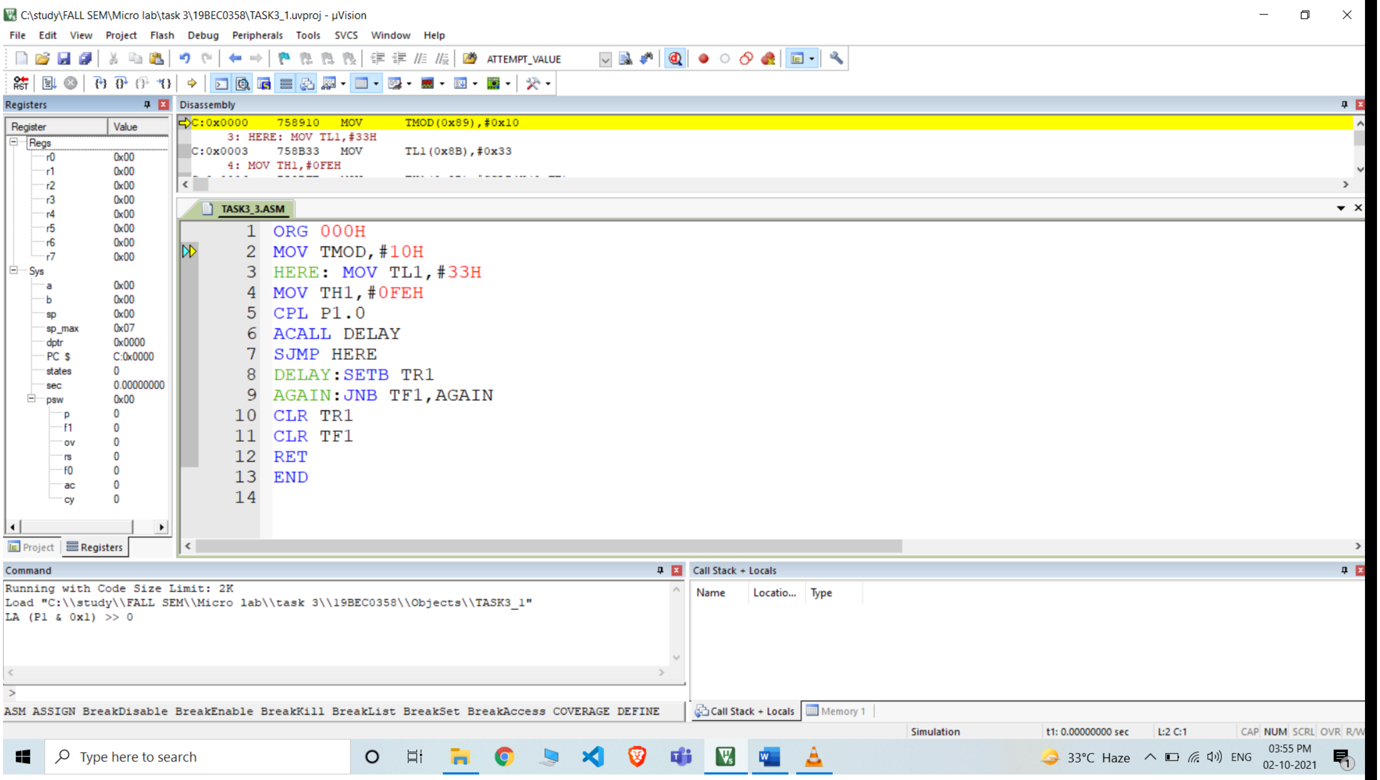
For half cycle = T/2 = 0.5mS

Count = 0.5mS / 1.085uS = 460.82 → 461

Timer bits = 65536- 461 = FE33H

Results and Observations

Program and registers before execution:



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

Disassembly:

```
C:0x0000    758910   MOV      TMOD(0x89),#0x10
     3: HERE: MOV TL1,#33H
C:0x0003    758B33   MOV      TL1(0x8B),#0x33
     4: MOV TH1,#0FEH
```

TASK3_3.ASM:

```
ORG 000H
MOV TMOD,#10H
HERE: MOV TL1,#33H
MOV TH1,#0FEH
CPL P1.0
ACALL DELAY
SJMP HERE
DELAY:SETB TR1
AGAIN:JNB TF1,AGAIN
CLR TR1
CLR TF1
RET
END
```

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
  dptr        0x0000
  PC $        C:0x0000
  states      0
  sec         0.00000000
  psw         0x00
    p         0
    f1        0
    ov        0
    rs        0
    f0        0
    ac        0
    cy        0
```

Command:

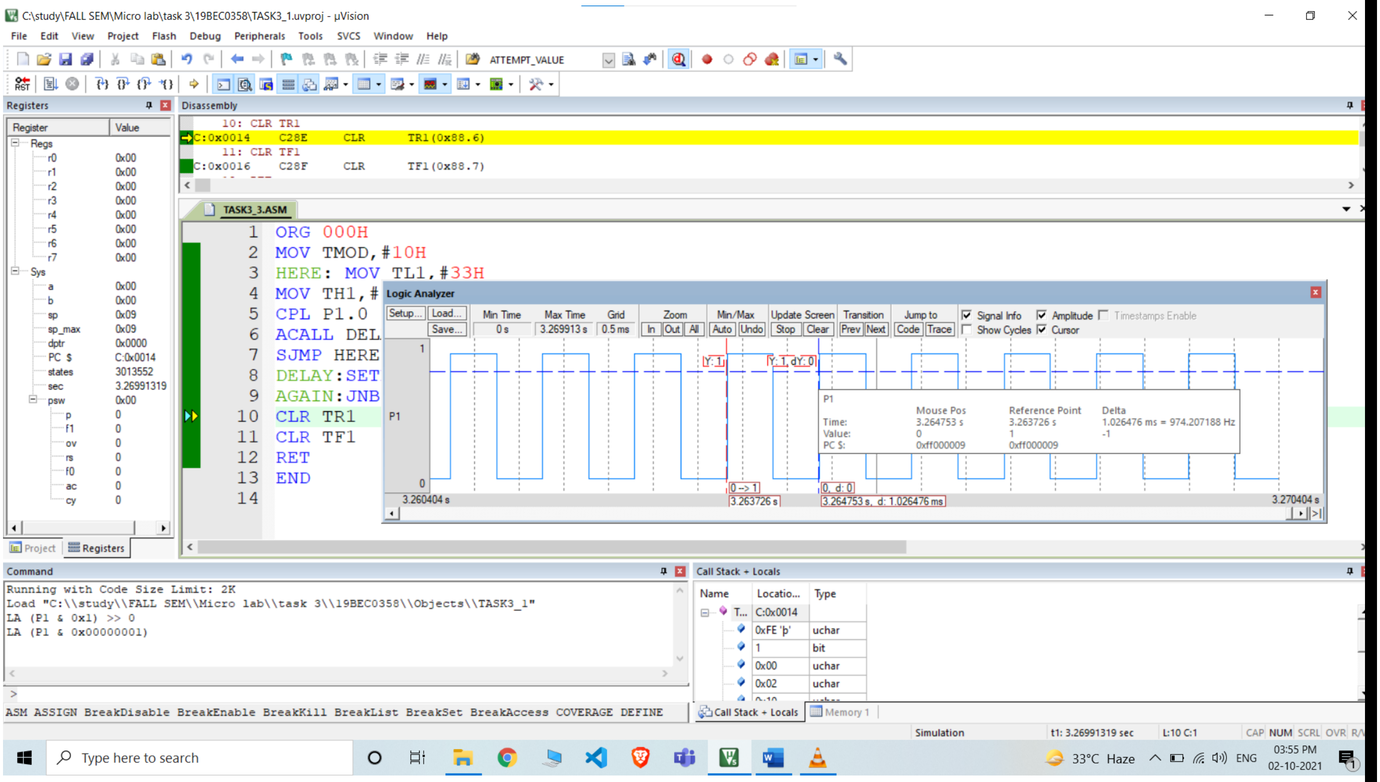
```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
LA (P1 & 0x1) >> 0
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
```

Call Stack + Locals:

Name	Locatio...	Type

```
t1: 0.00000000 sec    L:2 C:1
```

Program and registers after execution: FINAL STEP



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

Disassembly:

```
    10: CLR TR1
C:0x0014    C28E     CLR      TR1(0x88.6)
    11: CLR TF1
C:0x0016    C28F     CLR      TF1(0x88.7)
```

TASK3_3.ASM:

```
ORG 000H
MOV TMOD,#10H
HERE: MOV TL1,#33H
MOV TH1,#
CPL P1.0
ACALL DEL
SJMP HERE
DELAY:SET
AGAIN:JNB
CLR TR1
CLR TF1
RET
END
```

Logic Analyzer signal info:

```
P1
         Mouse Pos       Reference Point    Delta
Time:    3.264753 s      3.263726 s         1.026476 ms = 974.207188 Hz
Value:   0               1                  -1
PC $:    0xff000009      0xff000009
```

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          0x09
  sp_max      0x09
  dptr        0x0000
  PC $        C:0x0014
  states      3013552
  sec         3.26991319
  psw         0x00
    p         0
    f1        0
    ov        0
    rs        0
    f0        0
    ac        0
    cy        0
```

Command:

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
LA (P1 & 0x1) >> 0
LA (P1 & 0x00000001)
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
```

Call Stack + Locals:

Name	Locatio...	Type
T...	C:0x0014	
	0xFE 'þ'	uchar
	1	bit
	0x00	uchar
	0x02	uchar

```
t1: 3.26991319 sec    L:10 C:1
```

Inferences:

1. About PSW VALUES – they remain unchanged

2. ABOUT THE OUTPUT VALUES IN REGISTERS the final Result is in port1 which is complimented

Continuously with frequency = 974.20Hz

Result: the 8051 ALP to to generate a 1000Hz square wave frequency on one of the pins of P1 is executed using Keil software and the results are verified Manually.

TASK 4) -

Aim: To write an 8051 ALP using timer 1 mode 1 to generate a 500Hz square wave frequency on one of the pins of P1 using keil software and to verify the result manually.

Tools Required : Keil Microvision Software

Algorithm:

1.calculate the time period from given frequency

2.using time period, calculate the value of count

3. subtract the count value from FFFF and load into timer 1

4. now compliment port 1 by using timer 1

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	ORG	000H					Starting of program	none
	MOV	TMOD, #10H	Immediate	1	2	Data transfer	Timer 1 mode 1 is selected	None

HERE	MOV	TL1, #66H	Immediate	1	2	Data transfer	Lower bit of timer = 66	None
	MOV	TH1, #0FCH	Immediate	1	2	Data transfer	Higher bit of timer = FC	None
	CPL P1.0		Direct addressing	1	1	logical	Compliment value of port 1	none
	ACALL	DELAY		2	2	branching	Call delay subroutine	None
	SJMP	HERE		2	2	branching	Short jump to here label	None
DELAY	SETB	TR1	Direct	1	2	Boolean	Start timer	none
AGAIN	JNB	TF1, AGAIN	Direct	1	2	Boolean	Monitor timer flag, if it is 0, then go to again loop	Timer flag
	CLR	TR1	Direct	1	1	Boolean	Stop timer	None
	CLR	TF1	Direct	1	1	Boolean	Reset timer flag	None
	RET		Assembler directive	1	1	Program branching	Return	None
	END		Assembler directive				end	None

Output : Ports containing the Result: P1, in the form of waveform

Manual Calculation : Frequency = 500Hz

T = 1/F = 2mS

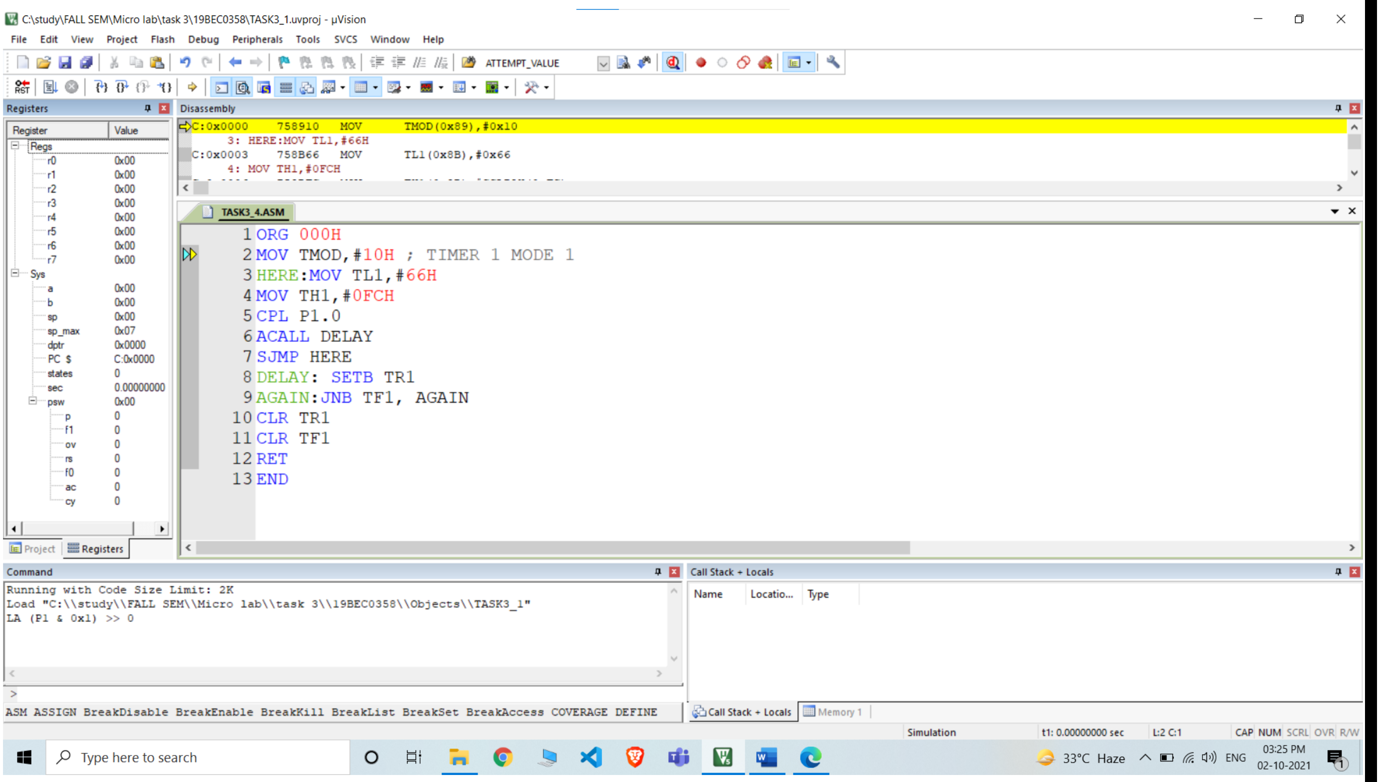
For half cycle = T/2 = 1mS

Count = 1mS / 1.085uS = 921.86 → 922

Timer bits = 65536- 922 = FC66H

Results and Observations

Program and registers before execution:



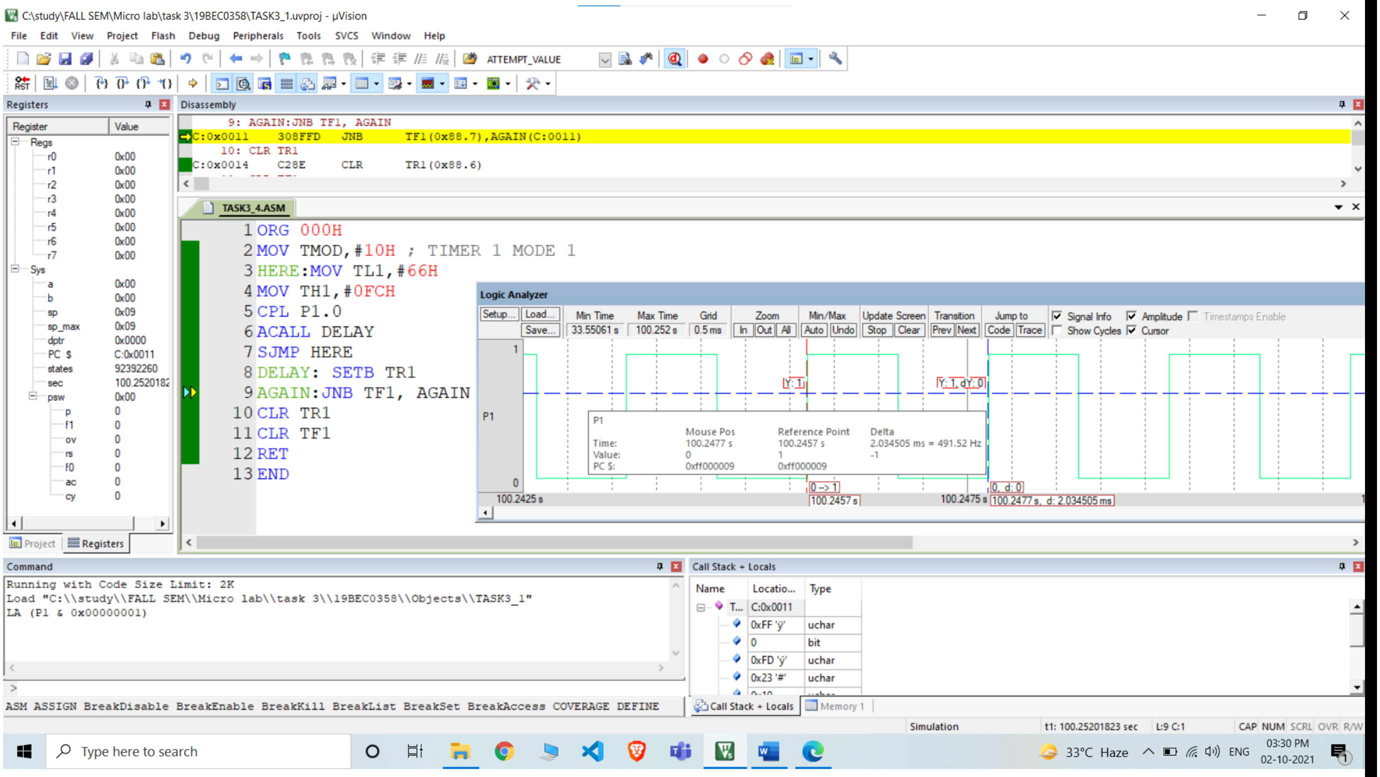
Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

```
C:0x0000    758910   MOV      TMOD(0x89),#0x10
     3: HERE:MOV TL1,#66H
C:0x0003    758B66   MOV      TL1(0x8B),#0x66
     4: MOV TH1,#0FCH
```

```
ORG 000H
MOV TMOD,#10H ; TIMER 1 MODE 1
HERE:MOV TL1,#66H
MOV TH1,#0FCH
CPL P1.0
ACALL DELAY
SJMP HERE
DELAY: SETB TR1
AGAIN:JNB TF1, AGAIN
CLR TR1
CLR TF1
RET
END
```

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
LA (P1 & 0x1) >> 0
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
```

Program and registers after execution: FINAL STEP



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

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	0x09
sp_max	0x09
dptr	0x0000
PC $	C:0x0011
states	92392260
sec	100.2520182
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

```
     9: AGAIN:JNB TF1, AGAIN
C:0x0011    308FFD   JNB      TF1(0x88.7),AGAIN(C:0011)
    10: CLR TR1
C:0x0014    C28E     CLR      TR1(0x88.6)
```

```
ORG 000H
MOV TMOD,#10H ; TIMER 1 MODE 1
HERE:MOV TL1,#66H
MOV TH1,#0FCH
CPL P1.0
ACALL DELAY
SJMP HERE
DELAY: SETB TR1
AGAIN:JNB TF1, AGAIN
CLR TR1
CLR TF1
RET
END
```

Min Time 33.55061 s Max Time 100.252 s Grid 0.5 ms

☑ Signal Info ☑ Amplitude □ Timestamps Enable
□ Show Cycles ☑ Cursor

```
P1
          Mouse Pos       Reference Point    Delta
Time:     100.2477 s      100.2457 s         2.034505 ms = 491.52 Hz
Value:    0               1                  -1
PC $:     0xff000009      0xff000009
```

Y: 1 | Y: 1, dY: 0 | 0 --> 1 | 0, d: 0
100.2425 s | 100.2457 s | 100.2475 s | 100.2477 s, d: 2.034505 ms

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
LA (P1 & 0x00000001)
```

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE

Name	Locatio...	Type
T...	C:0x0011	
	0xFF 'ÿ'	uchar
	0	bit
	0xFD 'ý'	uchar
	0x23 '#'	uchar

Simulation t1: 100.25201823 sec L:9 C:1 CAP NUM SCRL OVR R/W

33°C Haze ENG 03:30 PM 02-10-2021

Inferences:

1. About PSW VALUES – they remain unchanged

2. ABOUT THE OUTPUT VALUES IN REGISTERS the final Result is in port1 which is changed

Continuously and the frequency = 491.52Hz

Result: the 8051 ALP to to generate a 500Hz square wave frequency on one of the pins of P1 is executed using Keil software and the results are verified Manually.

TASK 5) -

Aim: To write an 8051 ALP to count the pulses and display the state of the TL1 count on P2 using keil software and to verify the result manually.

Tools Required : Keil Microvision Software

Algorithm:

1.Select couter 1 mode 2 in TMOD register

2.send high to low signal to port 3.5

3.whenever we send a signal , copy the count value to accumulator and load it to port 2

Program:

Label	Mnemonics	Operands	addressing mode used	Machine cycle Required	Memory Byte Required	Type of Instruction	Comments	Flags getting affected by the Instruction.
	MOV	TMOD, #01100000B	Immediate	1	2	Data transfer	counter 1 mode 2 is selected	None
	MOV	TH1, #0H	Immediate	1	2	Data transfer	Higher bit of timer = 0	None
	SETB	P3.5	Direct	1	1	Boolean	Make P 3.5 input	None
AGAIN	SETB	TR1	Direct	1	1	Boolean	Start Counter	none
BACK	MOV	A, TL1	Direct	1	2	Data transfer	Move data from TL1 to A	P flag
	MOV	P2, A	Direct	1	2	Data transfer	Move dat from A to port 2	None
	JNB	TF1, BACK	Direct	2	1	Branching	Monitor the flag	None

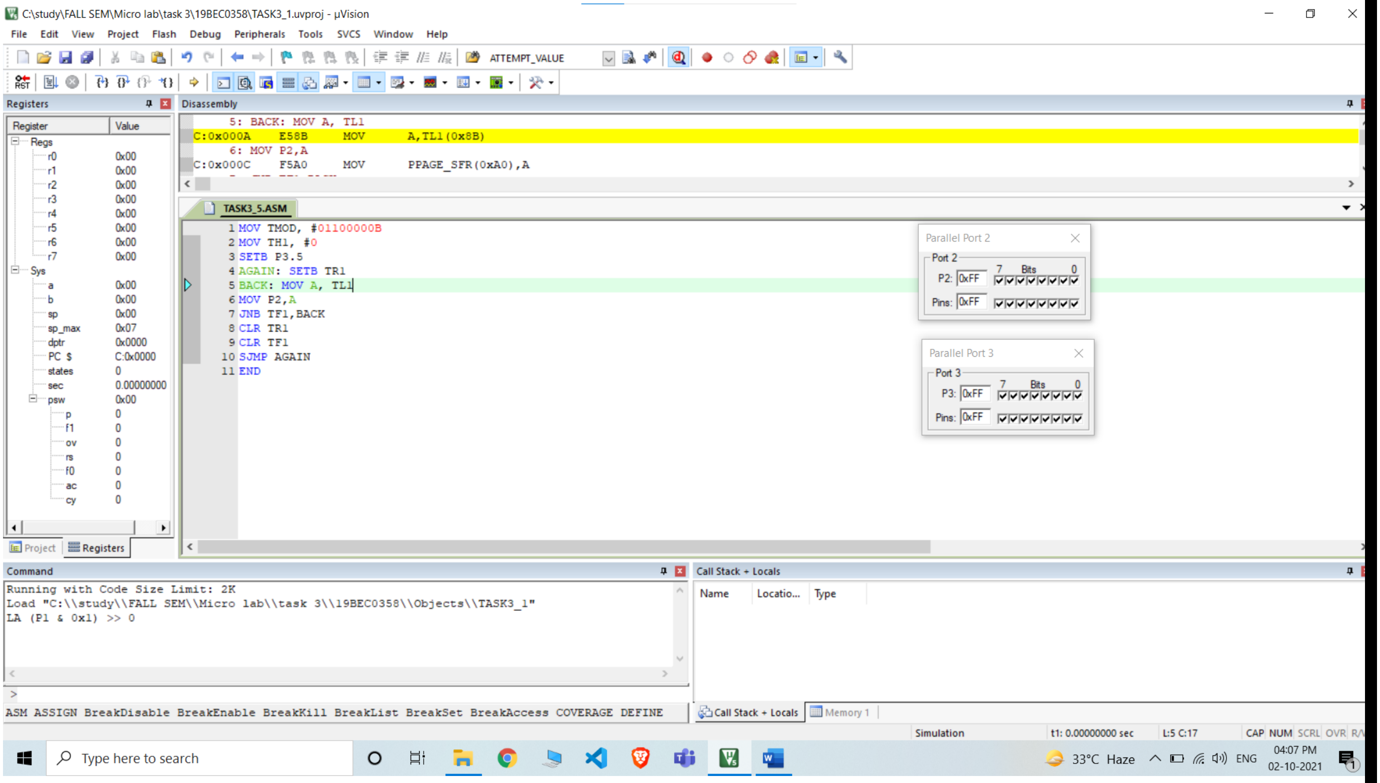
	CLR	TR1	Direct	1	1	Boolean	Stop counter	None
	CLR	TF1	Direct	1	1	Boolean	Clear flag	None
	SJMP	AGAIN		2	2	Branching	Move to again label	none
	END							

Output : Ports containing the Result: P2, tells about count value

Manual Calculation : None

Results and Observations

Program and registers before execution:



C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision

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
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

Disassembly

```
     5: BACK: MOV A, TL1
C:0x000A    E58B     MOV      A,TL1(0x8B)
     6: MOV P2,A
C:0x000C    F5A0     MOV      PPAGE_SFR(0xA0),A
```

TASK3_5.ASM

```
MOV TMOD, #01100000B
MOV TH1, #0
SETB P3.5
AGAIN: SETB TR1
BACK: MOV A, TL1
MOV P2,A
JNB TF1,BACK
CLR TR1
CLR TF1
SJMP AGAIN
END
```

Parallel Port 2

Port 2 P2: 0xFF Pins: 0xFF

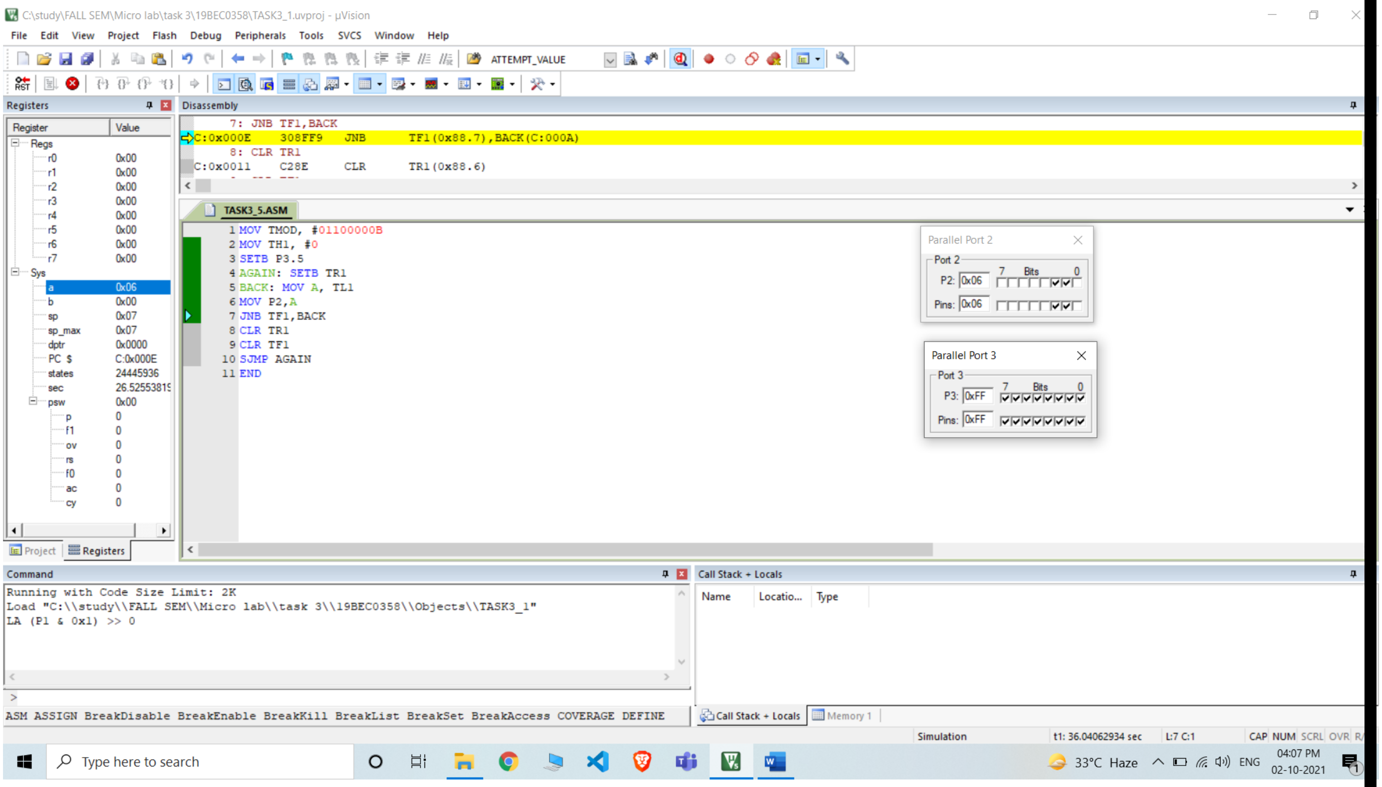
Parallel Port 3

Port 3 P3: 0xFF Pins: 0xFF

Command

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
LA (P1 & 0x1) >> 0
```

Program and registers after execution: FINAL STEP


C:\study\FALL SEM\Micro lab\task 3\19BEC0358\TASK3_1.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
ATTEMPT_VALUE
Registers
Disassembly
TASK3_5.ASM
Project
Registers
Command
Call Stack + Locals
Name
Locatio...
Type
Memory 1
Type here to search

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x06
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x000E
states	24445936
sec	26.5255381$
psw	0x00
p	0
f1	0
ov	0
rs	0
f0	0
ac	0
cy	0

```
     7: JNB TF1,BACK
C:0x000E    308FF9   JNB      TF1(0x88.7),BACK(C:000A)
     8: CLR TR1
C:0x0011    C28E     CLR      TR1(0x88.6)
```

```
MOV TMOD, #01100000B
MOV TH1, #0
SETB P3.5
AGAIN: SETB TR1
BACK: MOV A, TL1
MOV P2,A
JNB TF1,BACK
CLR TR1
CLR TF1
SJMP AGAIN
END
```

Parallel Port 2
Port 2
P2: 0x06 7 Bits 0 ☐☐☐☐☐☑☑☐
Pins: 0x06 ☐☐☐☐☐☑☑☐

Parallel Port 3
Port 3
P3: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑
Pins: 0xFF ☑☑☑☑☑☑☑☑

```
Running with Code Size Limit: 2K
Load "C:\\study\\FALL SEM\\Micro lab\\task 3\\19BEC0358\\Objects\\TASK3_1"
LA (P1 & 0x1) >> 0
```

```
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE
```

Simulation t1: 36.04062934 sec L:7 C:1 CAP NUM SCRL OVR R/

33°C Haze ENG 04:07 PM 02-10-2021

Inferences:

1. About PSW VALUES – they remain unchanged

2. ABOUT THE OUTPUT VALUES IN REGISTERS the final Result is in port 2 which is changed

Continuously as we give a signal to port 3.5

Result: the 8051 ALP to count the pulses and display the state of the TL1 count on P2 is executed using Keil software and the results are verified Manually.

---XXXXXXX---