Microcontroller and Microprocessors
Experiment - 6
Arithmetic Operations, Square Wave Generation and Buzzer Using ARM Processor

Varun Singh Inda
16BEE0023

Arithmetic Operations

Aim :- To perform the arithmetic operations using the ARM processor.

Procedure :-
1. Click on START.
2. Open Keil µvision5.
3. Open PROJECT. Create New µvision project.
4. Open Legacy Device Database.
5. Click on + next to NXP in the application box that opens.
6. Select LPC2148 from the list under NXP.
7. Click OK.
8. Proceed further clicking Yes.
9. In the main screen, select blank page icon present under File.
10. A new Text Window opens, where we are to write the program to be executed.
11. Go to File in the menu bar and save the program with the extension .asm.
12. In project window, select target on clicking + and chose Add existing file to source group1 and chose the program to be executed.
13. On clicking + next to SOURCE GROUP, right click, build target.
14. Click on Debug icon.
15. Start debug session.
16. Click on OK when the window pops in.
17. Press RUN or F5.

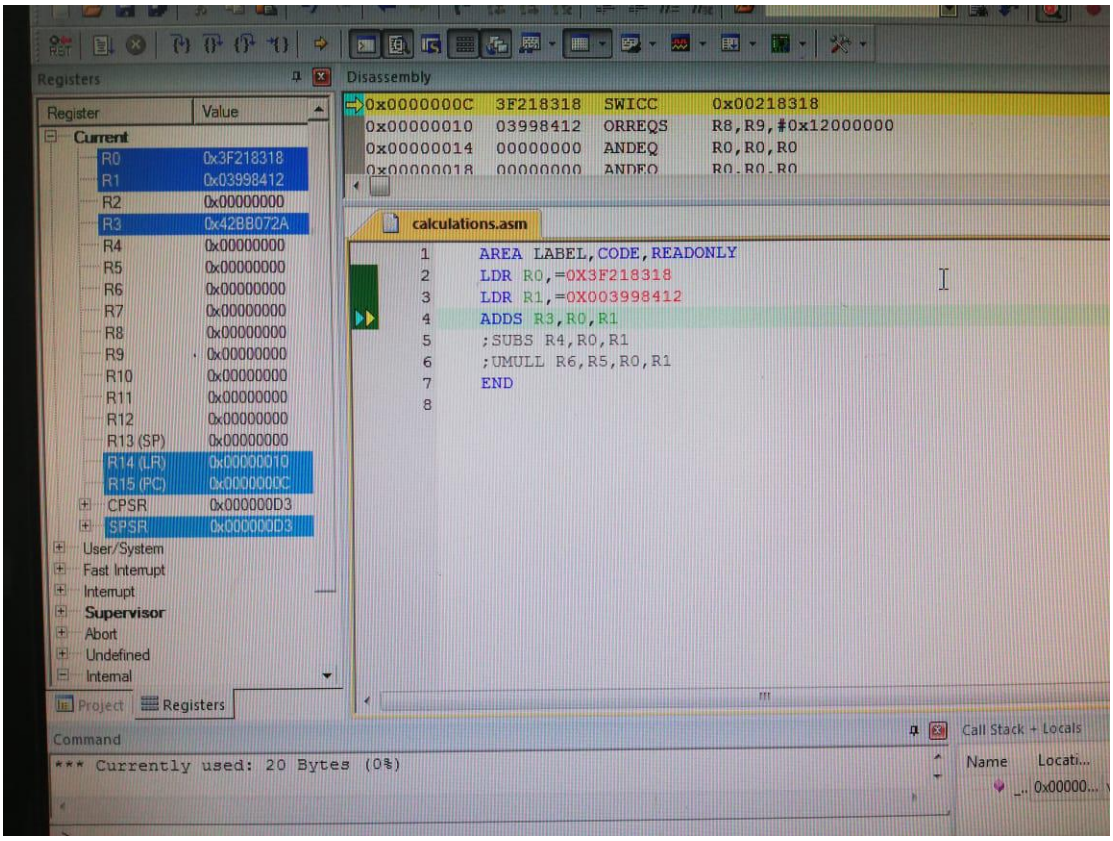
1. Addition :-

ALGORITHM:
1. START.
2. DECLARE REGISTER R0 WITH A SPECIFIC MEMORY LOCATION.
3. LOAD THE DATA AT R0 INTO R1.
4. INCREMENT VALUE OF R0 BY 4.
5. LOAD THIS INCREENTED DATA INTO R2.
6. ADD THE VALUES OF R1 AND R2 AND STORE THE RESULT IN R3.

7. INCREMENT THE VALUE OF R0 BY 4 AGAIN.
8. STORE TE RESULT IN R3.
9. RUN THE LOOP INDEFINITELY.
10. END.

CODE :-

```
AREA LABEL, CODE, READONLY
 LDR R0, =0X3F218318
 LDR R1, =0X003998412
 ADDS R3,R0,R1
  END
```



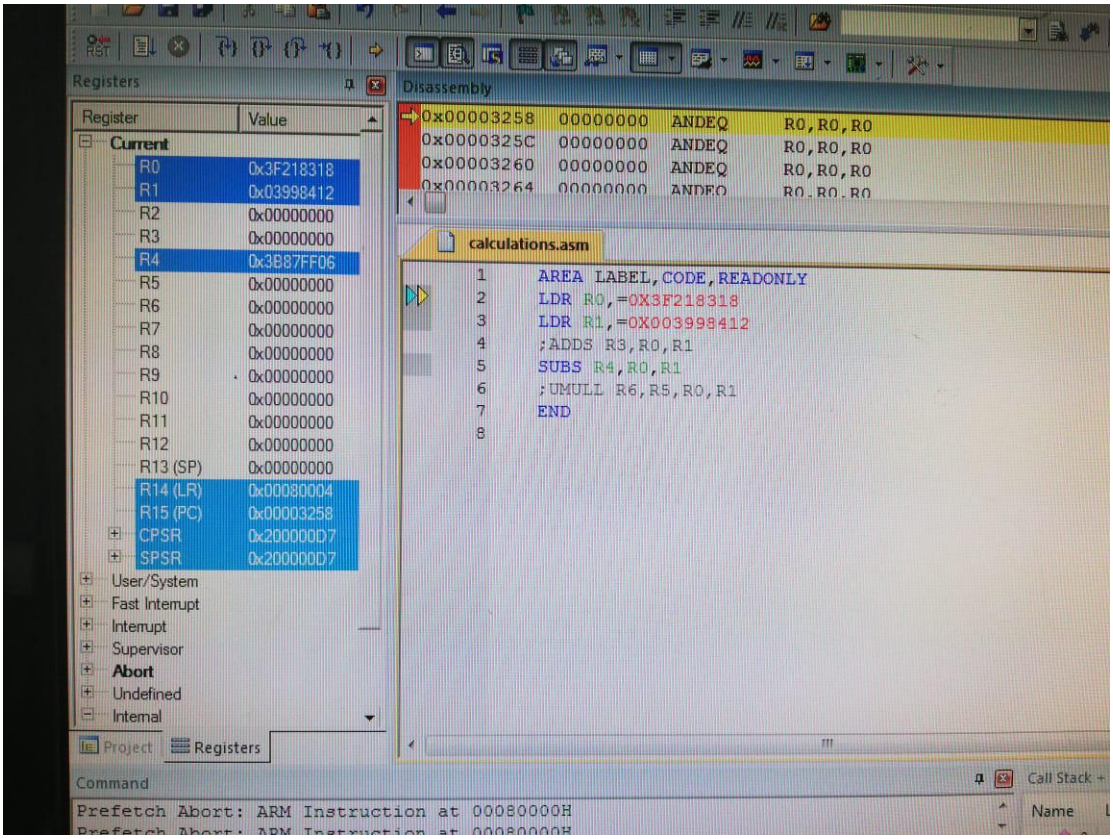
2. Subtraction :-

ALGORITHM:

1. START.
2. DECLARE REGISTER R0 WITH A SPECIFIC MEMORY LOCATION.
3. LOAD THE DATA AT R0 INTO R1.
4. INCREMENT VALUE OF R0 BY 4.
5. LOAD THIS INCREENTED DATA INTO R2.
6. SUBTRACT THE VALUE OF R1 FROM R2 AND STORE THE RESULT IN R3.
7. INCREMENT THE VALUE OF R0 BY 4 AGAIN.
8. STORE TE RESULT IN R3.
9. RUN THE LOOP INDEFINITELY.
10. END.

CODE :-

```
AREA LABEL, CODE, READONLY
LDR R0, =0X3F218318
LDR R1, =0X003998412
SUBS R4,R0,R1
 END
```



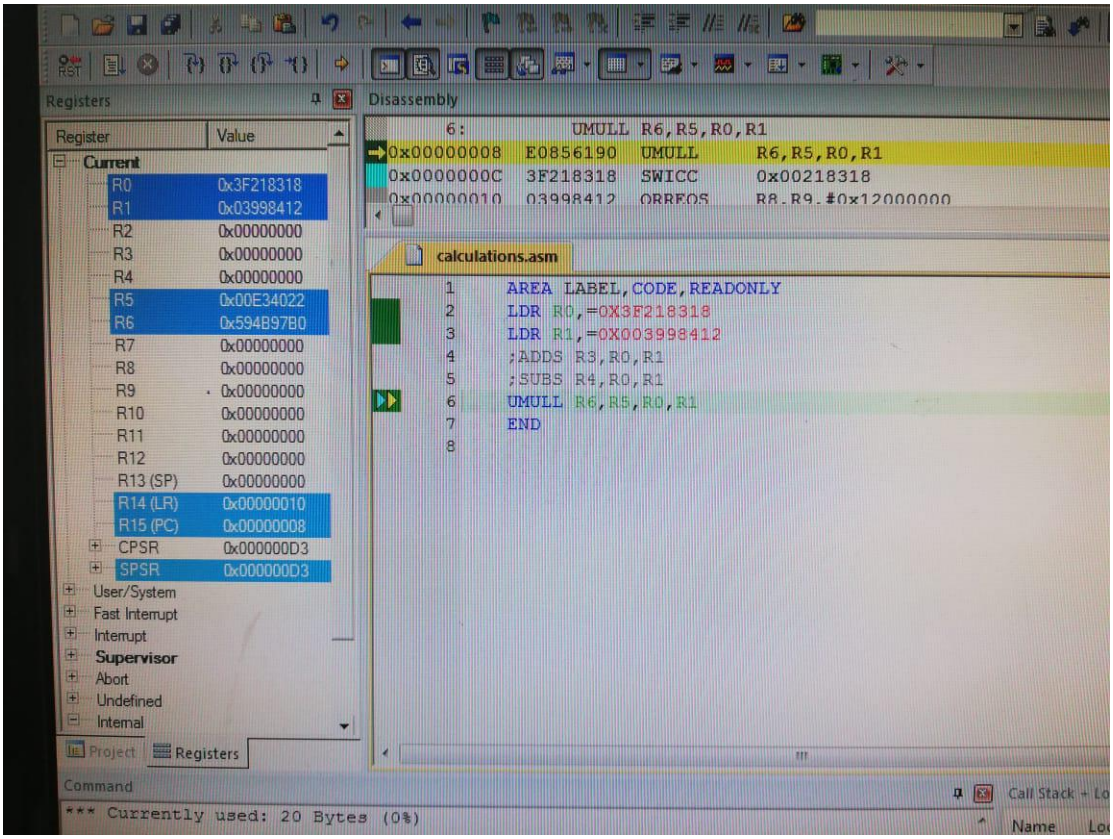
3. Multiplication :-

ALGORITHM:

1. START.
2. DECLARE REGISTER R0 WITH A SPECIFIC MEMORY LOCATION.
3. LOAD THE DATA AT R0 INTO R1.
4. INCREMENT VALUE OF R0 BY 4.
5. LOAD THIS INCREENTED DATA INTO R2.
6. MULTIPY THE VALUES OF R1 AND R2 AND STORE THE RESULT IN R3 AND R4.
7. INCREMENT THE VALUE OF R0 BY 4 AGAIN.
8. STORE TE RESULT IN R3.
9. RUN THE LOOP INDEFINITELY.
10. END.

CODE :-

```
AREA LABEL, CODE, READONLY
LDR R0, =0X3F218318
LDR R1, =0X003998412
UMULL R6,R5,R0,R1
END
```



4. Division :-

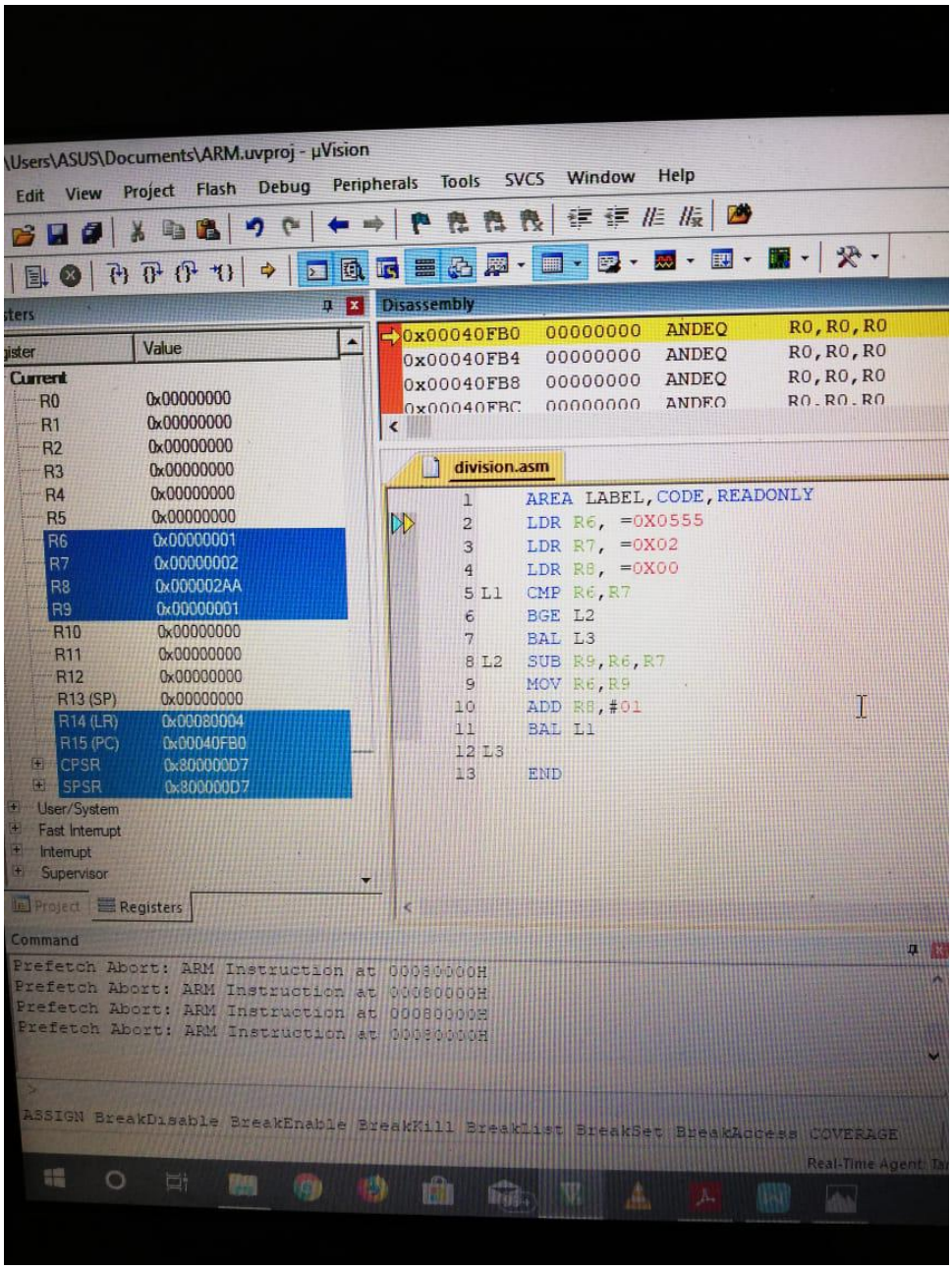
ALGORITHM:

1. START.
2. LOAD THE REGISTERS R0, R1 WITH VALUES.
3. CLEAR REGISTER R2.
4. IF THE DESTINATION VALUE IS GREATER THAN OR EQUAL TO THE SOURCE VALUE, IT BRANCHES TO THE LOOP L.
5. SUBTRACT R1 FROM R0, INCREMENT REGISTER R2 EACH TIME THE LOOP RUNS.
6. BRANCH ALWAYS LOOP L1.
7. END.

CODE :-

```
AREA LABEL,CODE,READONLY
LDR R6, =0X0555
LDR R7, =0X02
```

```
      LDR R8, =0X00
L1    CMP R6,R7
      BGE L2
      BAL L3
L2     SUB R9,R6,R7
      MOV R6,R9
      ADD R8,#01
      BAL L1
L3
      END
```



Conclusion :- All the arithmetic operations were done properly and the results were matching with the theoretical results.

Square Wave Generation

Aim :- To generate a square wave using ARM processor

PROCEDURE:
1. Click on START.
2. Open Keil µvision5.
3. Open PROJECT. Create New µvision project.
4. Open Legacy Device Database.
5. Click on + next to NXP in the application box that opens.
6. Select LPC2148 from the list under NXP.
7. Click OK.
8. Proceed further clicking Yes.
9. In the main screen, select blank page icon present under File.
10. A new Text Window opens, where we are to write the program to be executed.
11. Go to File in the menu bar and save the program with the extension .asm.
12. In project window, select target on clicking + and chose Add existing file to source group1 and chose the program to be executed.
13. On clicking + next to SOURCE GROUP, right click, build target.
14. Click on Debug icon.
15. Start debug session.
16. Click on OK when the window pops in.
17. Press RUN or F5.

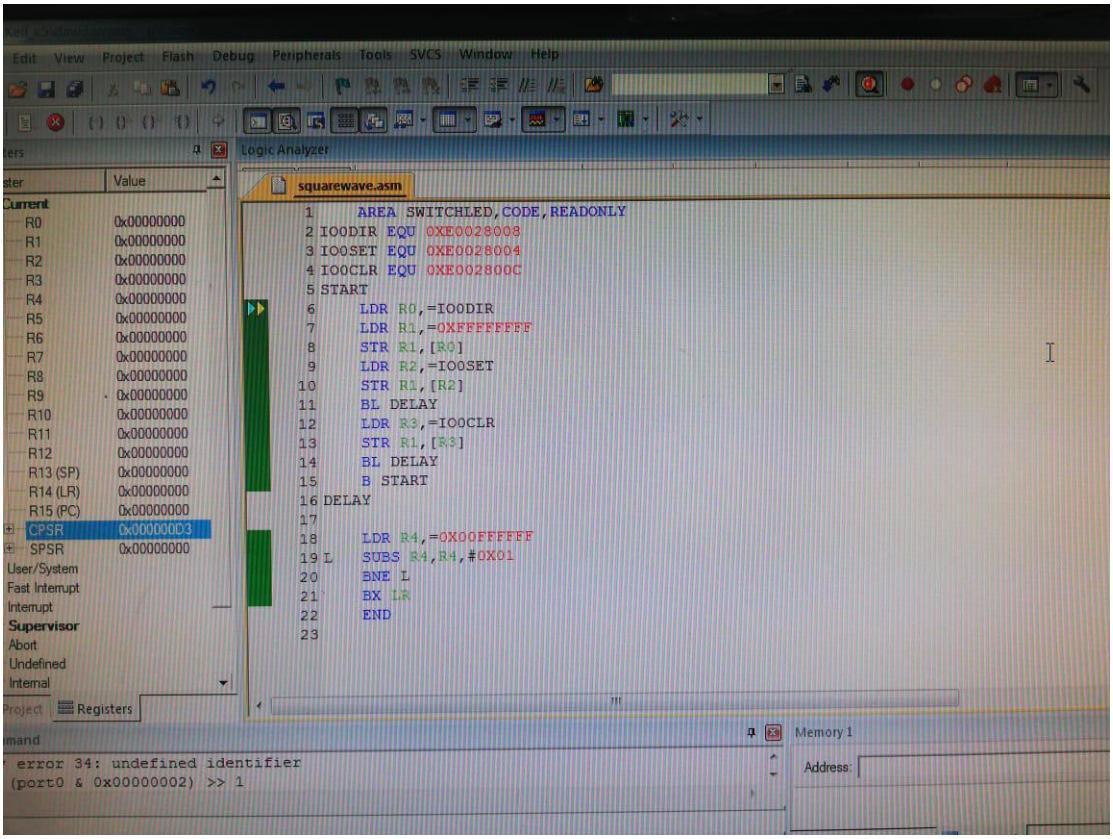
ALGORITHM:
1. Start.
2. Define io0dir and io0pin with constant given values.
3. Store the value in R0, in register R1.
4. Define R2 with the value of io0pin.
5. Store the value at R2 in register R0.
6. Call branch loop delay.
7. Go back to start once over, hence run the loop again and again.
8. Define the delay function.
9. End.

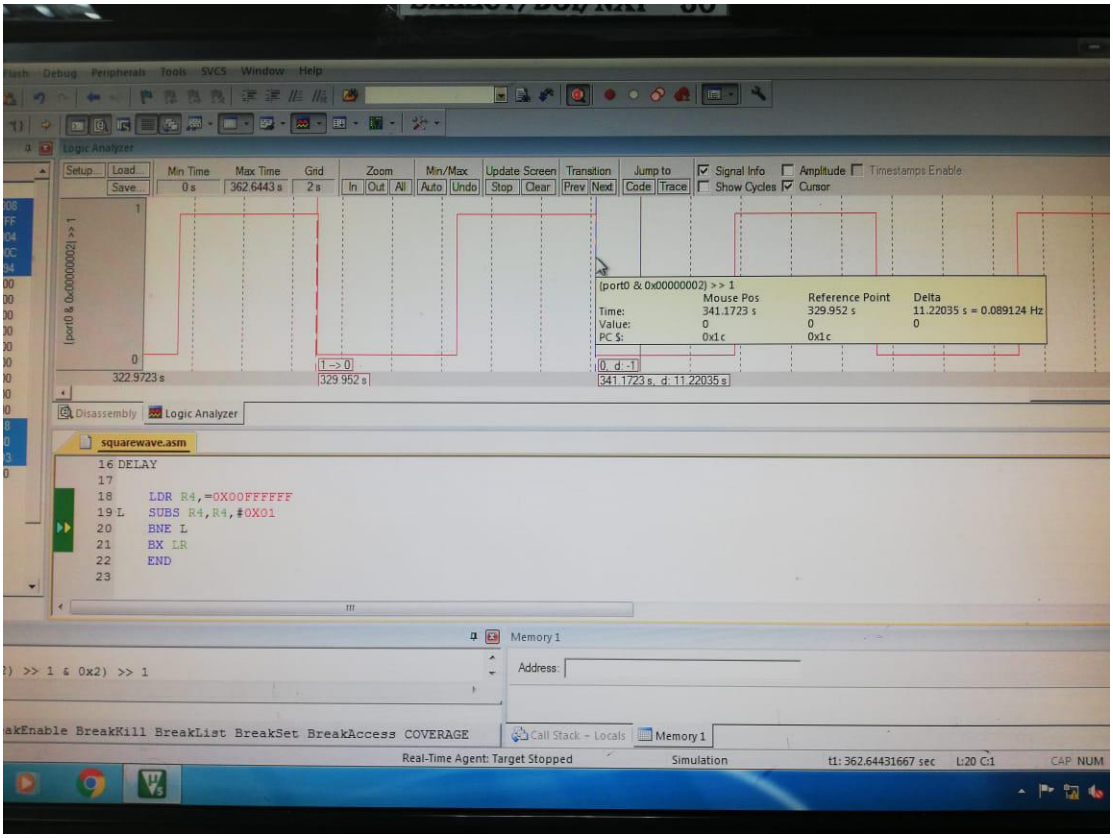
CODE :-

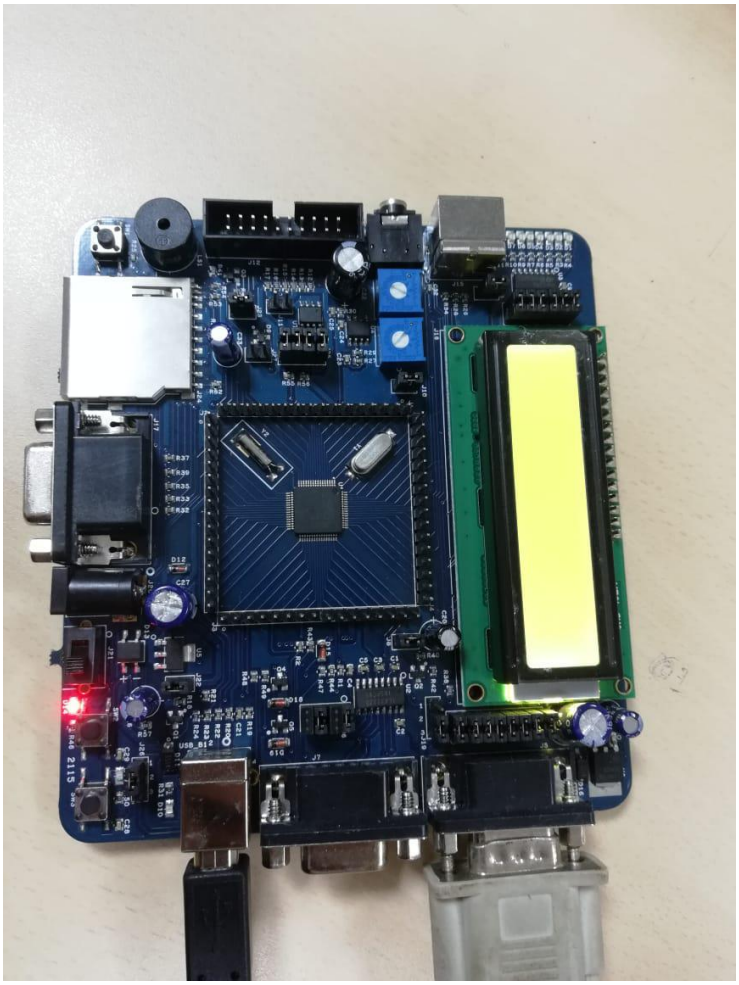
```
    AREA SWITCHLED,CODE,READONLY
IO0DIR EQU 0XE0028008
IO0SET EQU 0XE0028004
```

```
IO0CLR EQU 0XE002800C
START
    LDR R0,=IO0DIR
    LDR R1,=0XFFFFFFFF
    STR R1,[R0]
    LDR R2,=IO0SET
    STR R1,[R2]
    BL DELAY
    LDR R3,=IO0CLR
    STR R1,[R3]
    BL DELAY
    B START
DELAY

    LDR R4,=0X00FFFFFF
L   SUBS R4,R4,#0X01
    BNE L
    BX LR
    END
```




Logic Analyzer
Setup...
Load...
Save
Min Time
0 s
Max Time
362.6443 s
Grid
2 s
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Stop
Clear
Transition
Prev
Next
Jump to
Code
Trace
Signal Info
Amplitude
Timestamps Enable
Show Cycles
Cursor
(port0 & 0x00000002) >> 1
Mouse Pos
Reference Point
Delta
Time:
341.1723 s
329.952 s
11.22035 s = 0.089124 Hz
Value:
0
0
0
PC $:
0x1c
0x1c
322.9723 s
329.952 s
341.1723 s, d: 11.22035 s
Disassembly
Logic Analyzer
squarewave.asm
16 DELAY
17
18 LDR R4,=0X00FFFFFF
19 L SUBS R4,R4,#0X01
20 BNE L
21 BX LR
22 END
23
Memory 1
Address:
) >> 1 & 0x2) >> 1
akEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
Call Stack + Locals
Memory 1
Real-Time Agent: Target Stopped
Simulation
t1: 362.64431667 sec
L:20 C:1
CAP NUM



Conclusion :- The square wave was obtained properly using the ARM processor and the results were same as the theoretical results.

Buzzer Interface

Aim:- To make ARM processor work like a buzzer.

PROCEDURE:

1. Click on START.
2. Open Keil µvision5.
3. Open PROJECT. Create New µvision project.
4. Open Legacy Device Database.
5. Click on + next to NXP in the application box that opens.
6. Select LPC2148 from the list under NXP.
7. Click OK.
8. Proceed further clicking Yes.
9. In the main screen, select blank page icon present under File.
10. A new Text Window opens, where we are to write the program to be executed.
11. Go to File in the menu bar and save the program with the extension .asm.
12. In project window, select target on clicking + and chose Add existing file to source group1 and chose the program to be executed.
13. On clicking + next to SOURCE GROUP, right click, build target.
14. Click on Debug icon.
15. Start debug session.
16. Click on OK when the window pops in.
17. Press RUN or F5.

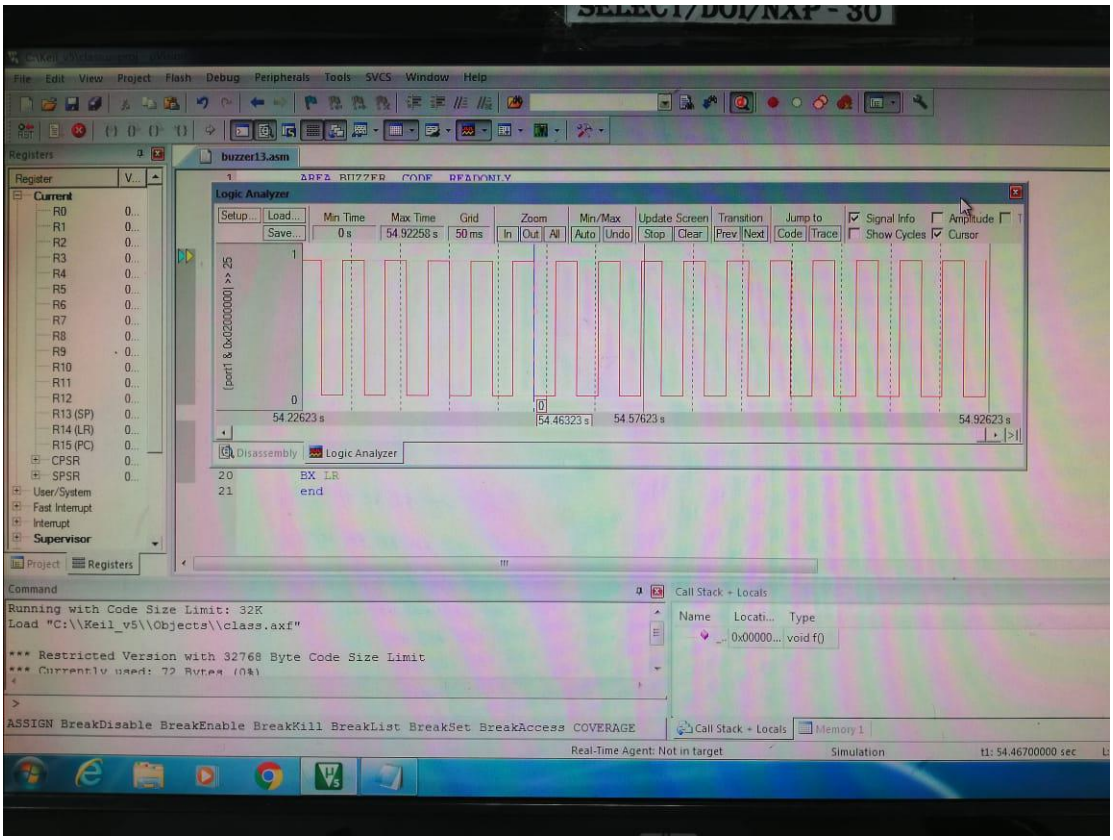
ALGORITHM:

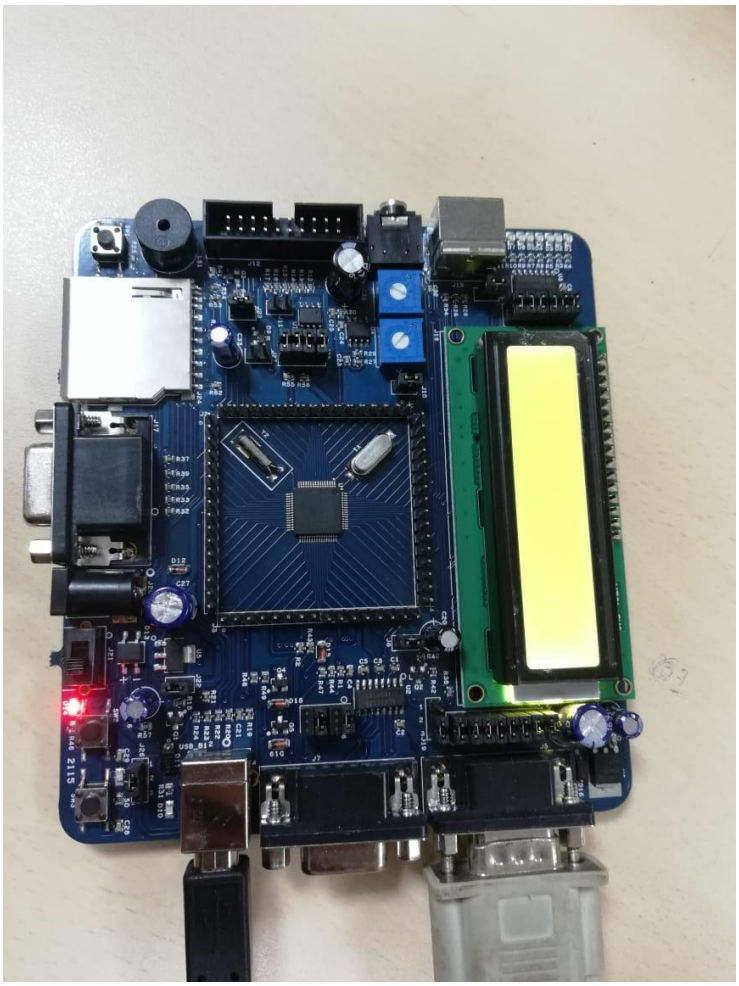
1. Start.
2. Define io0dir and io0pin with constant given values.
3. Store the value in R0, in register R1.
4. Define R2 with the value of io0pin.
5. Store the value at R2 in register R0.
6. Call branch loop delay.
7. Go back to start once over, hence run the loop again and again.
8. Define the delay function.
9. End.

CODE :-

```
    AREA BUZZER,CODE,READONLY
IO1DIR EQU 0XE0028018
IO1SET EQU 0XE0028014
IO1CLR EQU 0XE002801C
START
    LDR R0,=IO1DIR
    LDR R1,=0X02000000
    STR R1,[R0]
    LDR R2,=IO1SET
    STR R1,[R2]
    BL DELAY
    LDR R3,=IO1CLR
    STR R1,[R3]
    BL DELAY
    B START
DELAY

    LDR R4,=0X00FFFF
L   SUBS R4,R4,#0X01
    BNE L
    BX LR
    END
```





Conclusion:- The ARM processor was working as a buzzer on the port1.25 as read in the theoretical analysis.