Indian Institute of Technology Bombay

Microprocessors Lab
EE 337

Lab 1
January 14, 2025

Mridul Choudhary
23B3933

Contents

1 Debugging GCD Calculator in ASM

1.1 Aim of the experiment

The primary aim of the experiment is to debug an assembly program provided in the lab folder and ensure its successful execution. Additionally, it seeks to enhance the students' skills in handling memory manipulations, debugging using breakpoints, analyzing program flow, and utilizing tools like the Logic Analyzer to determine waveform frequencies.

1.2 Design

1.2.1 Initial Setup and Debugging:

1. Students are instructed to create a new project in Keil µVision using the provided ”Error code.asm” program.
2. Debugging the program involves identifying syntax errors, fixing them step by step, and ensuring successful compilation.

1.2.2 Memory and Execution Steps:

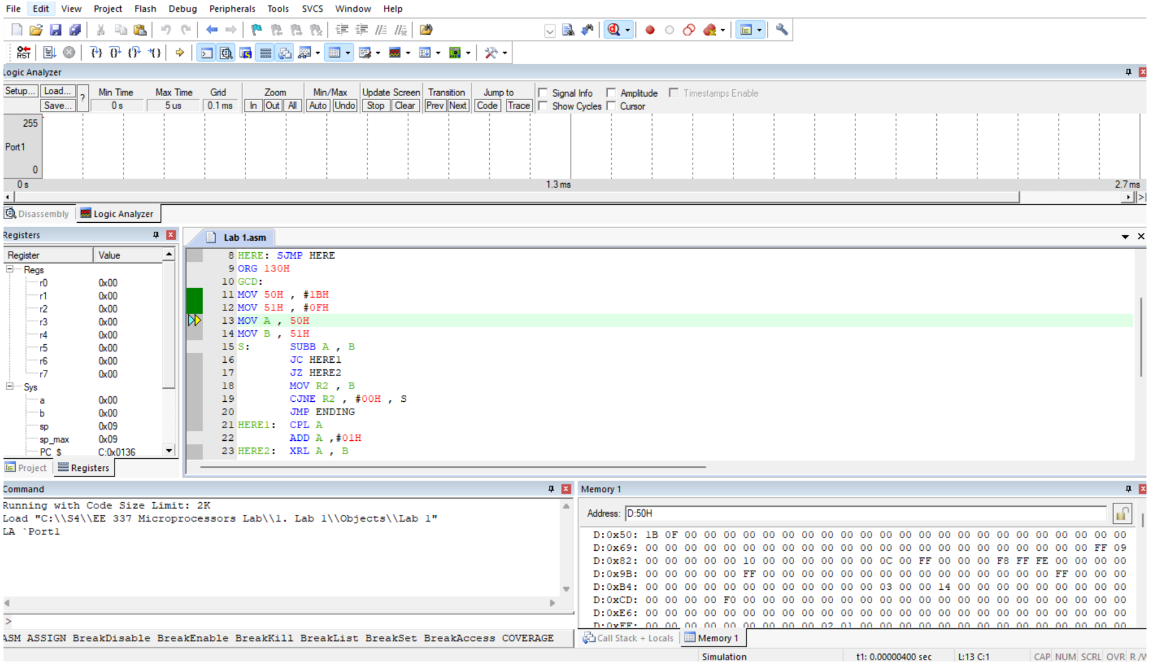
1. Post-correction, the program is further debugged with specific instructions which includes modifying the memory location 50h to 1E and modifying the memory location 51h to 0C.
2. Execution progresses step-by-step until reaching the first instruction of the label HERE2 via breakpoints.

1.2.3 Waveform Frequency Analysis:

1. The Logic Analyzer is used to observe Port 1 and compute its waveform frequency by running the corrected program.

1.3 Experimental results

Sr. No.	Error Code	Corrected Code
1	SUB A , B	SUBB A , B
2	CJNZ R2 , #00H , S	CJNE R2 , #00H , S
3	INC A , #01H	ADD A , #01H



```
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
  PC $      C:0x0136
```

```
HERE: SJMP HERE
ORG 130H
GCD:
MOV 50H , #1BH
MOV 51H , #0FH
MOV A , 50H
MOV B , 51H
S:      SUBB A , B
        JC HERE1
        JZ HERE2
        MOV R2 , B
        CJNE R2 , #00H , S
        JMP ENDING
HERE1:  CPL A
        ADD A ,#01H
HERE2:  XRL A , B
```

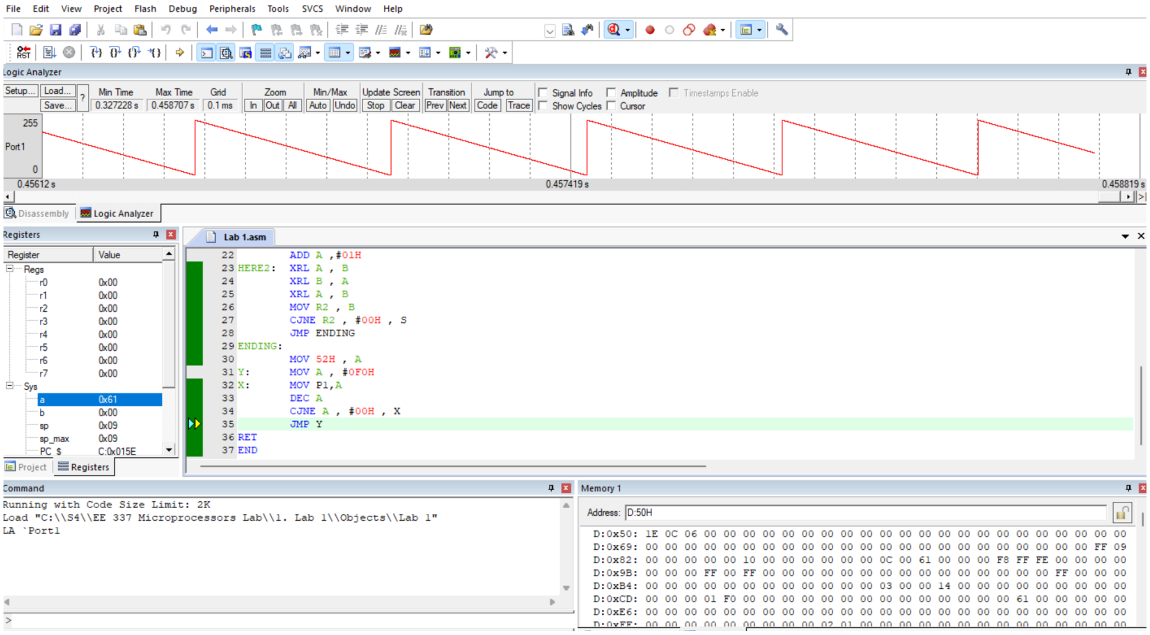
```
Running with Code Size Limit: 2K
Load "C:\\S4\\EE 337 Microprocessors Lab\\1. Lab 1\\Objects\\Lab 1"
LA `Port1
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Address: D:50H

```
D:0x50: 1B 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x69: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 09
D:0x82: 00 00 00 00 00 10 00 00 00 00 00 00 0C 00 FF 00 00 00 F8 FF FE 00 00 00 00
D:0x9B: 00 00 00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00
D:0xB4: 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 14 00 00 00 00 00 00 00 00 00
D:0xCD: 00 00 00 00 F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0xE6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0xFF: 00 00 00 00 00 00 00 00 02 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation t1: 0.00000400 sec L:13 C:1 CAP NUM SCRL OVR R/W

Figure 1: Values being registered at the memory locations



```
Register    Value
Regs
  r0        0x00
  r1        0x00
  r2        0x00
  r3        0x00
  r4        0x00
  r5        0x00
  r6        0x00
  r7        0x00
Sys
  a         0x61
  b         0x00
  sp        0x09
  sp_max    0x09
  PC $      C:0x015E
```

```
        ADD A ,#01H
HERE2:  XRL A , B
        XRL B , A
        XRL A , B
        MOV R2 , B
        CJNE R2 , #00H , S
        JMP ENDING
ENDING:
        MOV 52H , A
Y:      MOV A , #0F0H
X:      MOV P1,A
        DEC A
        CJNE A , #00H , X
        JMP Y
RET
END
```

```
Running with Code Size Limit: 2K
Load "C:\\S4\\EE 337 Microprocessors Lab\\1. Lab 1\\Objects\\Lab 1"
LA `Port1
>
```

Address: D:50H

```
D:0x50: 1E 0C 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x69: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 09
D:0x82: 00 00 00 00 00 10 00 00 00 00 00 00 0C 00 61 00 00 00 F8 FF FE 00 00 00 00
D:0x9B: 00 00 00 FF 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00
D:0xB4: 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 14 00 00 00 00 00 00 00 00 00
D:0xCD: 00 00 00 01 F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 00 00
D:0xE6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0xFF: 00 00 00 00 00 00 00 00 02 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 2: After the values at the memory locations are changed

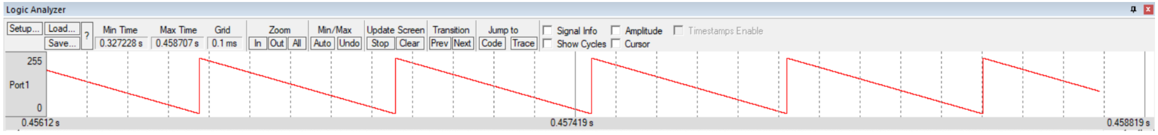
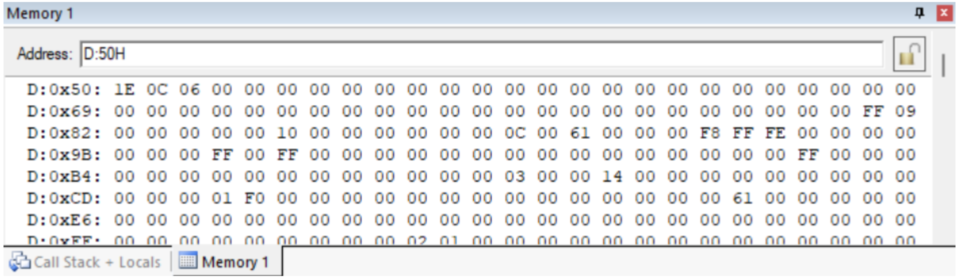


Figure 3: Logic Analyzer during the runtime



```
Address: D:50H
D:0x50: 1E 0C 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0x69: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 09
D:0x82: 00 00 00 00 00 10 00 00 00 00 00 00 0C 00 61 00 00 00 F8 FF FE 00 00 00 00
D:0x9B: 00 00 00 FF 00 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF 00 00 00
D:0xB4: 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 14 00 00 00 00 00 00 00 00 00
D:0xCD: 00 00 00 01 F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 00 00 00 00 00
D:0xE6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D:0xFF: 00 00 00 00 00 00 00 00 00 02 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 4: Memory during the runtime

1.4 Conclusion and Inference

The experiment provided a comprehensive understanding of debugging and running assembly programs in Keil µVision. By following the prescribed steps, students were able to:

1. Correct syntax and logical errors in the given code.

2. Modify memory content at specific addresses effectively.

3. Gain proficiency in stepping through programs, using breakpoints, and analyzing hardware registers.

4. Accurately determine the waveform frequency of Port 1, demonstrating their ability to integrate software tools with hardware functionality.