

EEE 416: Microprocessor and Embedded System Laboratory

Laboratory Task

Date: 29 Nov, 2023

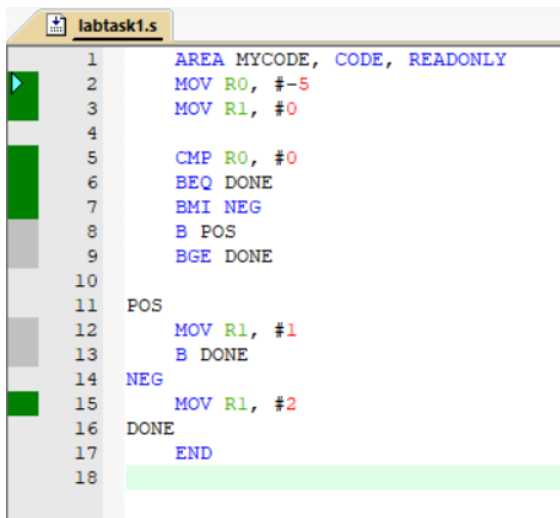
Experiment no: 06

Experiment name: Assembly Language Programming for ARM architecture

Student no: 1906055

1. Task 1: Toggle an LED every 1 second using SysTick.

Code:



```
1  AREA MYCODE, CODE, READONLY
2  MOV R0, #-5
3  MOV R1, #0
4
5  CMP R0, #0
6  BEQ DONE
7  BMI NEG
8  B POS
9  BGE DONE
10
11 POS
12  MOV R1, #1
13  B DONE
14 NEG
15  MOV R1, #2
16 DONE
17  END
18
```

Register	Value
Current	
R0	0x00000005
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000004C
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000004C
Mode	Supervisor

Address	Disassembly
0x0000004C	ANDEQ R0, R0, R0
0x00000050	ANDEQ R0, R0, R0
0x00000054	ANDEQ R0, R0, R0
0x00000058	ANDEQ R0, R0, R0
0x0000005C	ANDEQ R0, R0, R0

labtask1.s

```

1  AREA MYCODE, CODE, READONLY
2  MOV R0, #0
3  MOV R1, #0
4
5  CMP R0, #0
6  BEQ DONE
7  BMI NEG
8  B POS
9  BGE DONE
10
11 POS
12  MOV R1, #1
13  B DONE
14  NEG
15  MOV R1, #2
16  DONE
17  END
18

```

Register	Value
Current	
R0	0x00000005
R1	0x00000001
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000028
CPSR	0x200000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000028
Mode	Supervisor

Address	Disassembly
0x00000028	ANDEQ R0, R0, R0
0x0000002C	ANDEQ R0, R0, R0
0x00000030	ANDEQ R0, R0, R0
0x00000034	ANDEQ R0, R0, R0
0x00000038	ANDEQ R0, R0, R0

labtask1.s

```

1  AREA MYCODE, CODE, READONLY
2  MOV R0, #5
3  MOV R1, #0
4
5  CMP R0, #0
6  BEQ DONE
7  BMI NEG
8  B POS
9  BGE DONE
10
11 POS
12  MOV R1, #1
13  B DONE
14  NEG
15  MOV R1, #2
16  DONE
17  END
18

```

Registers		Disassembly	
Register	Value		
Current			
R0	0xFFFFFFFF	0x00000028 00000000 ANDEQ R0,R0,R0	
R1	0x00000002	0x0000002C 00000000 ANDEQ R0,R0,R0	
R2	0x00000000	0x00000030 00000000 ANDEQ R0,R0,R0	
R3	0x00000000	0x00000034 00000000 ANDEQ R0,R0,R0	
R4	0x00000000	0x00000038 00000000 ANDEQ R0,R0,R0	
R5	0x00000000		
R6	0x00000000		
R7	0x00000000		
R8	0x00000000		
R9	0x00000000		
R10	0x00000000		
R11	0x00000000		
R12	0x00000000		
R13 (SP)	0x00000000		
R14 (LR)	0x00000000		
R15 (PC)	0x00000028		
CPSR	0xA00000D3		
SPSR	0x00000000		
User/System			
Fast Interrupt			
Interrupt			
Supervisor			
Abort			
Undefined			
Internal			
PC \$	0x00000028		
Mode	Supervisor		
States	8		

labtask1.s	
1	AREA MYCODE, CODE, READONLY
2	MOV R0, #-5
3	MOV R1, #0
4	
5	CMP R0, #0
6	BEQ DONE
7	BMI NEG
8	B POS
9	BGE DONE
10	
11	POS
12	MOV R1, #1
13	B DONE
14	NEG
15	MOV R1, #2
16	DONE
17	END
18	

Output is shown in R1 for input R0 for zero, pos and then neg values

Task 2:

Create a system that has 2 interrupts – an external interrupt connected to PC 13 (where a Push button is connected), and SysTick. The **handler** of the interrupts performs the following functions –

- SysTick: Toggle the LED connected to PA5 every 0.25 sec
- EXTI: creates a delay greater than 0.25 sec using for loop.

Now observe what happens when the Pushbutton is pressed if –

- EXTI has higher priority than SysTick (say, 0 and 1 respectively)
- SysTick has higher priority

Code:

The screenshot displays the Keil IDE interface. The top pane shows the assembly source code for `labtask1.s`:

```
1 AREA MYCODE, CODE, READONLY
2 MOV R0, #3
3 MOV R1, #1
4 MOV R2, R1
5
6 FOR
7   CMP R2, R0
8   BGT DONE
9   MUL R3, R1, R2
10  MOV R1, R3
11  ADD R2, R2, #1
12  B FOR
13 DONE
14 END
```

The bottom-left pane shows the **Registers** window with the following values:

Register	Value
R0	0x00000003
R1	0x00000006
R2	0x00000004
R3	0x00000006
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000024
CPSR	0x200000D3
SPSR	0x00000000

The bottom-right pane shows the **Disassembly** window, listing the instructions and their addresses:

Address	Disassembly
0x00000024	ANDEQ R0, R0, R0
0x00000028	ANDEQ R0, R0, R0
0x0000002C	ANDEQ R0, R0, R0
0x00000030	ANDEQ R0, R0, R0
0x00000034	ANDEQ R0, R0, R0

The top-right pane shows the assembly source code for `labtask1.s` again, with the execution pointer at line 2:

```
1 AREA MYCODE, CODE, READONLY
2 MOV R0, #3
3 MOV R1, #1
4 MOV R2, R1
5
6 FOR
7   CMP R2, R0
8   BGT DONE
9   MUL R3, R1, R2
10  MOV R1, R3
11  ADD R2, R2, #1
12  B FOR
13 DONE
14 END
```

R1 is our result here

Task 3

Load and observe the SONAR code.

- Trim and modify the code and the setup so that you can measure the Pulse with when a push button is pressed. (Remove the sonar sensor. In place of the Echo pin, connect a pushbutton with pull down resistor.)
- Change the code so that we can measure the delay between 2 consecutive button presses.

The screenshot displays an IDE with two main panels. The top panel shows the assembly code for a file named 'labtask1.s'. The code is as follows:

```
1 AREA MYCODE, CODE, READONLY
2 MOV R0, #7
3 MOV R1, #3
4 MOV R4, #0
5
6 FOR
7   CMP R0, R1
8   BLT DONE
9   BEQ DONE
10  ADD R4, R4, #1
11  SUB R0, R0, R1
12  B FOR
13
14 DONE
15 END
```

The bottom panel is divided into two sections. On the left is the 'registers' window, which lists registers R0 through R15, CPSR, and SPSR. The 'Current' register is R15 (PC), with a value of 0x00000024. On the right is the 'Disassembly' window, which shows the disassembled code for the same file. The disassembly is as follows:

```
0x00000024 00000000 ANDEQ R0,R0,R0
0x00000028 00000000 ANDEQ R0,R0,R0
0x0000002C 00000000 ANDEQ R0,R0,R0
0x00000030 00000000 ANDEQ R0,R0,R0
0x00000034 00000000 ANDEQ R0,R0,R0
```

Here the result of division is stored in R4 and the remainder is stored in R1