

ECE 322 EMBEDDED SYSTEM

Lab :- 6

Name :- ARVIND KUMAR

Roll N:- 2021BEC0035

DATE : 13/02/2024

Submitted By- ARVIND KUMAR

Code:

The screenshot displays an embedded system development environment. On the left, a 'Registers' window shows the state of various registers. The 'Core' registers (R0-R15) are listed with their current values. R14 (LR) is 0x00000029, and R15 (PC) is 0x0000003C. The 'xPSR' register is 0x61000000. Below the registers, a tree view shows 'Banked', 'System', and 'Internal' components, with 'Thread' set to 'Privileged', 'MSP' at 52187997, 'States' at 52187997, and 'Sec' at 4.34899975.

The main window shows assembly code for 'lab_6.Subroutinecall.s'. The code is as follows:

```

57 ENDP
58
59 ;;users main program;;;
60
61 Reset_Handler
62
63
64 LDR R1, N ;Load count into R1
65
66 MOV R0, #0 ;Clear accumulator R0
67
68 BL SUMUP
69
70 LDR R3, SUMP ;Load address of SUM to R3
71 STR R0, [R3] ;Store SUM
72
73 LDR R4, [R3]
74
75 MOV R7, #8
76
77 LDR R5, SUMP2 ;Load address of SUM2 to R5
78 STR R7, [R5] ;Store SUM2
79
80 LDR R6, [R5]
81
82
83 STOP
84 B STOP
85
86 END

```

The assembly code is displayed in a window titled 'lab_6.Subroutinecall.s'. The code includes a reset handler that loads a count into R1, clears the accumulator R0, branches to a subroutine SUMUP, loads the address of SUM into R3, stores R0 into [R3], loads the value from [R3] into R4, moves the value 8 into R7, loads the address of SUM2 into R5, stores R7 into [R5], loads the value from [R5] into R6, and then branches to a STOP instruction.



Registers

Register	Value
Core	
R0	0x00000080
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)	0x20000FE0
R14 (LR)	0xFFFFFFFF9
R15 (PC)	0x1E6D0508
xPSR	0x60000003
Banked	
System	
Internal	
Mode	Handler
Privilege	Privileged
Stack	MSP
States	702
Sec	0.00005850

Disassembly

```
0x1E6D0508 0000 MOVs r0,r0
0x1E6D050A 0000 MOVs r0,r0
0x1E6D050C 0000 MOVs r0,r0
0x1E6D050E 0000 MOVs r0,r0
0x1E6D0510 0000 MOVs r0,r0
0x1E6D0512 0000 MOVs r0,r0
0x1E6D0514 0000 MOVs r0,r0
0x1E6D0516 0000 MOVs r0,r0
0x1E6D0518 0000 MOVs r0,r0
0x1E6D051A 0000 MOVs r0,r0
0x1E6D051C 0000 MOVs r0,r0
```

Lab_6_subrotandstack.s

```
62  PUSH  {R0, R3}      ;Notice the stack address is 0x200000FF8
63      MOV   R0, #6
64      MOV   R3, #7
65  POP  {R0, R3}      ;Should be able to see R0 = #0x75 and R3 = #5 after pop
66
67
68
69  Loop
70
71      ADD R0, R0, #1
72      CMP R0, #0x80
73      BNE Loop
74
75
76      MOV   R5, #9 ;; prepare for function call
77
78
79      BL   function1
80
81      MOV   R3, #12
82
83  stop
84      B     stop
85
86
87      END
```