

Lecture2

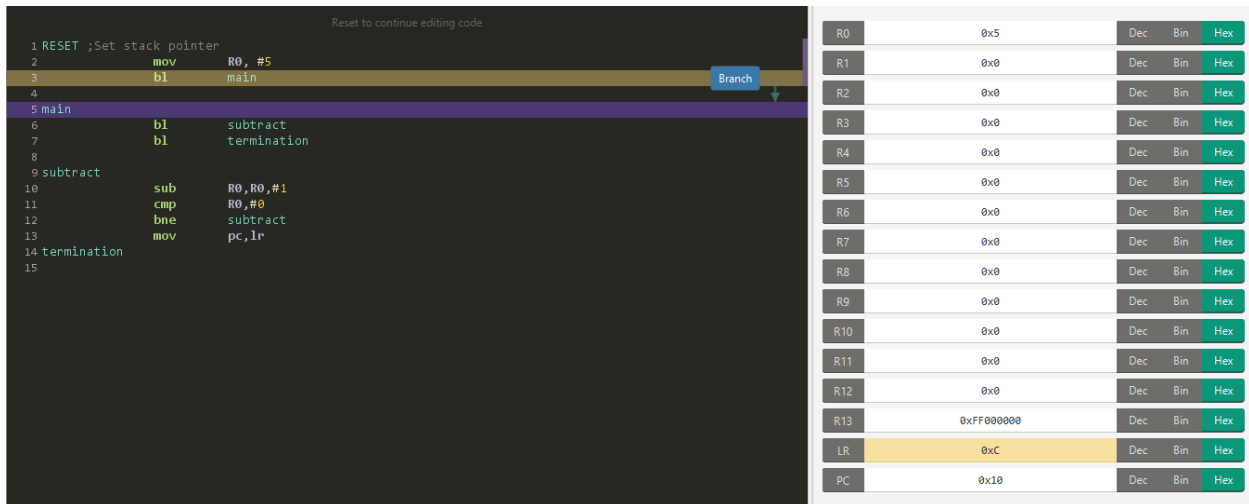
INLINE ASSEMBLY

By: Abdelrahman matarawy

ARM Assembly Basics

Lab to achieve assembly to equivalent C code:

```
register int x = 5 ;
Void main ()
{
    subtract();
}
subtract ()
{
    x -= 1 ;
    if (x != 0) subtract();
}
```



The screenshot shows an ARM assembly editor with the following code:

```
1 RESET ;Set stack pointer
2     mov     R0, #5
3     bl      main
4
5 main
6     bl      subtract
7     bl      termination
8
9 subtract
10    sub     R0, R0, #1
11    cmp     R0, #0
12    bne     subtract
13    mov     pc, lr
14 termination
15
```

On the right, a register table shows the state of the registers:

Register	Value	Dec	Bin	Hex
R0	0x5			
R1	0x0			
R2	0x0			
R3	0x0			
R4	0x0			
R5	0x0			
R6	0x0			
R7	0x0			
R8	0x0			
R9	0x0			
R10	0x0			
R11	0x0			
R12	0x0			
R13	0xFF000000			
LR	0xC			
PC	0x10			

INLINE Code:

- INLINE Function:
 - Making the Function Inline eliminates the BL and BX instructions and their execution but replace function code everywhere BL call which saving time at the possible expense of memory.

- Not inline function:

- Calling a regular function requires time to execute the BL and BX instructions but for large functions it saves memory at the expense of speed.

Using Assembly in C source: (__asm)

- Lab1:

- Simple lab without input or output parameters.

```

main.c
39
40 int main(void)
41 {
42     clk_init();
43     int val3=4;
44     /* Set Configuration of EXTI */
45     EXTI_Config_t EXTI_Conf; // Make Variable of EXTI_PinConfig
46     EXTI_Conf.EXTI_PIN = EXTI_LINE9_PB9; // Set line 9, GPIOB, P
47     EXTI_Conf.Trigger_Case = EXTI_Trigger_RASING; // Specifies R
48     EXTI_Conf.IRQ_Case = EXTI_IRQ_Enable; // Enable Interrupt Mas
49     EXTI_Conf.P_IRQ_CallBack = EXTI_CallBack; // Make Call Back
50
51     MCAL_EXTI_GPIO_Init(&EXTI_Conf);
52     IRQ_Flag = 1;
53
54     __asm("nop");
55     __asm("nop");
56     __asm("nop");
57
58     while(1)
59     {
60         if(IRQ_Flag == 1)
61         {
62             IRQ_Flag = 0;
63         }
64     }
65 }
66
Lab1.s
173 8000214: f443 0300 orr.w r3, r3, #8388608 ; 0x800000
174 8000218: 6013 str r3, [r2, #0]
175 800021a: e006 b.n 800022a <Enable_NVIC+0xba>
176 800021c: 4b06 ldr r3, [pc, #24] ; (8000238 <Enable_
177 800021e: 681b ldr r3, [r3, #0]
178 8000220: 4a05 ldr r2, [pc, #20] ; (8000238 <Enable_
179 8000222: f443 7380 orr.w r3, r3, #256 ; 0x100
180 8000226: 6013 str r3, [r2, #0]
181 8000228: bf00 nop
182 800022a: bf00 nop
183 800022c: 370c adds r7, #12
184 800022e: 46bd mov sp, r7
185 8000230: bc80 pop {r7}
186 8000232: 4770 bx lr
187 8000234: e000e100 and lr, r0, r0, lsl #2
188 8000238: e000e110 and lr, r0, r0, lsl #1
189
190 800023c <Disable_NVIC>:
191 800023c: b480 push {r7}
192 800023e: b083 sub sp, #12
193 8000240: af00 add r7, sp, #0
194 8000242: 4603 mov r3, r0
195 8000244: 80fb strh r3, [r7, #6]
196 8000246: 88fb ldrh r3, [r7, #6]
197 8000248: 2b0f cmp r3, #15
198 800024a: d854 hlt

```

- Lab2: (using output parameters)

- The %0 in the instruction template is replaced with the name of this register.

```

main.c
39
40 int main(void)
41 {
42     clk_init();
43     int val3=4;
44     /* Set Configuration of EXTI */
45     EXTI_Config_t EXTI_Conf; // Make Variable of EXTI_PinConfig
46     EXTI_Conf.EXTI_PIN = EXTI_LINE9_PB9; // Set line 9, GPIOB, P
47     EXTI_Conf.Trigger_Case = EXTI_Trigger_RASING; // Specifies R
48     EXTI_Conf.IRQ_Case = EXTI_IRQ_Enable; // Enable Interrupt Mas
49     EXTI_Conf.P_IRQ_CallBack = EXTI_CallBack; // Make Call Back
50
51     MCAL_EXTI_GPIO_Init(&EXTI_Conf);
52     IRQ_Flag = 1;
53     __asm("nop");
54     __asm("nop");
55     __asm("nop");
56
57     __asm("mov %0, #0xff"
58         : "=r" (val3));
59
60     __asm("nop");
61     __asm("nop");
62     __asm("nop");
63     while(1)
64     {
65         if(IRQ_Flag == 1)
66         {
67             IRQ_Flag = 0;
68         }
69     }
70 }
71
Lab1.s
1034 8000970: 463b mov r3, r7
1035 8000972: 4618 mov r0, r3
1036 8000974: f7ff fdb6 bl 80004e4 <MCAL_EXTI_GPIO_Init>
1037 8000978: 4b06 ldr r3, [pc, #44] ; (80009a8 <main+0>
1038 800097a: 2201 movs r2, #1
1039 800097c: 701a strb r2, [r3, #0]
1040 800097e: bf00 nop
1041 8000980: bf00 nop
1042 8000982: bf00 nop
1043 8000984: f04f 03ff mov.w r3, #255 ; 0xff
1044 8000988: 617b str r3, [r7, #20]
1045 800098a: bf00 nop
1046 800098c: bf00 nop
1047 800098e: bf00 nop
1048 8000990: 4b05 ldr r3, [pc, #20] ; (80009a8 <main+0>
1049 8000992: 781b ldrb r3, [r3, #0]
1050 8000994: 2b01 cmp r3, #1
1051 8000996: d1fb bne.n 8000990 <main+0x44>
1052 8000998: 4b03 ldr r3, [pc, #12] ; (80009a8 <main+0>
1053 800099a: 2200 movs r2, #0
1054 800099c: 701a strb r2, [r3, #0]
1055 800099e: e7f7 b.n 8000990 <main+0x44>
1056 80009a0: 08000a60 stmdaqr r0, {r5, r6, r9, fp}
1057 80009a4: 08000935 stmdaqr r0, {r0, r2, r4, r5, r8, fp}
1058 80009a8: 20000058 andcs r0, r0, r8, asr #0

```

○ Lab3: (using output parameters)

- [r7, #20] -> r7 is base of address and #20 is offset
- the value in this address save in r3
- “r” -> specifier for read and write

main.c
makefile

```

39
40 int main(void)
41 {
42     clk_init();
43     int val3=4;
44     /* Set Configuration of EXTI */
45     EXTI_Config_t EXTI_Conf; // Make Variable of EXTI_PinConfig
46     EXTI_Conf.EXTI_PIN = EXTI_LINE9_PB9; // Set line 9, GPIOB, P
47     EXTI_Conf.Trigger_Case = EXTI_Trigger_RASING; // Specifies R
48     EXTI_Conf.IRQ_Case =EXTI_IRQ_Enable; // Enable Interrupt Mas
49     EXTI_Conf.P_IRQ_CallBack = EXTI_CallBack; // Make Call Back
50
51     MCAL_EXTI_GPIO_Init(&EXTI_Conf);
52     IRQ_Flag = 1;
53     __asm("nop");
54     __asm("nop");
55     __asm("nop");
56
57     __asm("MOV R0, %0"
58         :
59         : "r" (val3));
60
61     __asm("nop");
62     __asm("nop");
63     __asm("nop");
64

```

Lab1.s

```

1035 8000972: 4618      mov r0, r3
1036 8000974: f7ff fdb6  bl 80004e4 <MCAL_EXTI_GPIO_Init>
1037 8000978: 4b0b      ldr r3, [pc, #44] ; (80009a8 <main+0>
1038 800097a: 2201      movs r2, #1
1039 800097c: 701a      strb r2, [r3, #0]
1040 800097e: bf00      nop
1041 8000980: bf00      nop
1042 8000982: bf00      nop
1043 8000984: 697b      ldr r3, [r7, #20]
1044 8000986: 4618      mov r0, r3
1045 8000988: bf00      nop
1046 800098a: bf00      nop
1047 800098c: bf00      nop
1048 800098e: 4b06      ldr r3, [pc, #24] ; (80009a8 <main+0>
1049 8000990: 781b      ldrb r3, [r3, #0]
1050 8000992: 2b01      cmp r3, #1
1051 8000994: d1fb      bne.n 800098e <main+0x42>
1052 8000996: 4b04      ldr r3, [pc, #16] ; (80009a8 <main+0>
1053 8000998: 2200      movs r2, #0
1054 800099a: 701a      strb r2, [r3, #0]
1055 800099c: e7f7      b.n 800098e <main+0x42>
1056 800099e: bf00      nop
1057 80009a0: 08000a60 stmdaeq r0, {r5, r6, r9, fp}
1058 80009a4: 08000935 stmdaeq r0, {r0, r2, r4, r5, r8, fp}
1059 80009a8: 20000058 andcs r0, r0, r8, asr r0

```

○ Lab4:

main.c
makefile

```

40 int main(void)
41 {
42     clk_init();
43     int val1, val2, val3;
44     /* Set Configuration of EXTI */
45     EXTI_Config_t EXTI_Conf; // Make Variable of EXTI_PinConfig
46     EXTI_Conf.EXTI_PIN = EXTI_LINE9_PB9; // Set line 9, GPIOB, P
47     EXTI_Conf.Trigger_Case = EXTI_Trigger_RASING; // Specifies R
48     EXTI_Conf.IRQ_Case =EXTI_IRQ_Enable; // Enable Interrupt Mas
49     EXTI_Conf.P_IRQ_CallBack = EXTI_CallBack; // Make Call Back
50
51     MCAL_EXTI_GPIO_Init(&EXTI_Conf);
52     IRQ_Flag = 1;
53     __asm("nop");
54     __asm("nop");
55     __asm("nop");
56
57     __asm("add %0, %1, %2"
58         : "=r" (val1)
59         : "r" (val2)
60         : "r" (val3) );
61
62     __asm("nop");
63     __asm("nop");
64     __asm("nop");
65     while(1)

```

Lab1.s

```

1035 8000974: 4b0c      ldr r3, [pc, #48] ; (80009a8 <main+0>
1036 8000976: 2201      movs r2, #1
1037 8000978: 701a      strb r2, [r3, #0]
1038 800097a: bf00      nop
1039 800097c: bf00      nop
1040 800097e: bf00      nop
1041 8000980: 69fb      ldr r3, [r7, #28]
1042 8000982: 69ba      ldr r2, [r7, #24]
1043 8000984: 4413      add r3, r2
1044 8000986: 617b      str r3, [r7, #20]
1045 8000988: bf00      nop
1046 800098a: bf00      nop
1047 800098c: bf00      nop
1048 800098e: 4b06      ldr r3, [pc, #24] ; (80009a8 <main+0>
1049 8000990: 781b      ldrb r3, [r3, #0]
1050 8000992: 2b01      cmp r3, #1
1051 8000994: d1fb      bne.n 800098e <main+0x42>
1052 8000996: 4b04      ldr r3, [pc, #16] ; (80009a8 <main+0>
1053 8000998: 2200      movs r2, #0
1054 800099a: 701a      strb r2, [r3, #0]
1055 800099c: e7f7      b.n 800098e <main+0x42>
1056 800099e: bf00      nop
1057 80009a0: 08000a60 stmdaeq r0, {r5, r6, r9, fp}
1058 80009a4: 08000935 stmdaeq r0, {r0, r2, r4, r5, r8, fp}
1059 80009a8: 20000058 andcs r0, r0, r8, asr r0
1060

```

○ Lab6:

main.c

makefile

```

41 {
42     clk_init();
43     int CPU_Control_Register = 0;
44     int CPU_IP_Register = 0;
45     /* Set Configuration of EXTI */
46     EXTI_Config_t EXTI_Conf; // Make Variable of EXTI_PinConfig
47     EXTI_Conf.EXTI_PIN = EXTI_LINE9_PB9; // Set line 9, GPIOB, P
48     EXTI_Conf.Trigger_Case = EXTI_Trigger_RISING; // Specifies R
49     EXTI_Conf.IRQ_Case = EXTI_IRQ_Enable; // Enable Interrupt Mas
50     EXTI_Conf.P_IRQ_CallBack = EXTI_CallBack; // Make Call Back
51
52     MCAL_EXTI_GPIO_Init(&EXTI_Conf);
53     IRQ_Flag = 1;
54     __asm("nop");
55     __asm("nop");
56     __asm("nop");
57
58     __asm("MRS %[out0], CONTROL"
59           :[out0] "=r" (CPU_Control_Register) );
60
61     __asm("nop");
62     __asm("nop");
63     __asm("nop");
64     while(1)
65     {
66         if(IRQ_Flag == 1)
67         {

```

Lab1.s

```

1034 8000970: 4b0d    ldr r3, [pc, #52] ; (80009a8 <main+0>
1035 8000972: 617b    str r3, [r7, #20]
1036 8000974: 1d3b    adds r3, r7, #4
1037 8000976: 4618    mov r0, r3
1038 8000978: f7ff fdb4 bl 80004e4 <MCAL_EXTI_GPIO_Init>
1039 800097c: 4b0b    ldr r3, [pc, #44] ; (80009ac <main+0>
1040 800097e: 2201    movs r2, #1
1041 8000980: 701a    strb r2, [r3, #0]
1042 8000982: bf00    nop
1043 8000984: bf00    nop
1044 8000986: bf00    nop
1045 8000988: f3ef 8314 mrs r3, CONTROL
1046 800098c: 61fb    str r3, [r7, #28]
1047 800098e: bf00    nop
1048 8000990: bf00    nop
1049 8000992: bf00    nop
1050 8000994: 4b05    ldr r3, [pc, #20] ; (80009ac <main+0>
1051 8000996: 781b    ldrb r3, [r3, #0]
1052 8000998: 2b01    cmp r3, #1
1053 800099a: d1fb    bne.n 8000994 <main+0x48>
1054 800099c: 4b03    ldr r3, [pc, #12] ; (80009ac <main+0>
1055 800099e: 2200    movs r2, #0
1056 80009a0: 701a    strb r2, [r3, #0]
1057 80009a2: e7f7    b.n 8000994 <main+0x48>
1058 80009a4: 08000a64 stmdbaq r0, {r2, r5, r6, r9, fp}
1059 80009a8: 08000935 stmdbaq r0, {r0, r2, r4, r5, r8, fp}
1060 80009ac: 20000000 rdcrr r0, r0, r0, r0, r0, r0, r0, r0

```

main.c

makefile

```

21 * Author: AbdelRahman Matarawy
22 */
23 #include "STM32F103C6_EXTI_Driver.h"
24 int CPU_Control_Register = 0;
25 int CPU_IP_Register = 0;
26 uint8_t IRQ_Flag;
27
28 //clk configuration
29 void clk_init()
30 {
31     RCC_GPIOB_CLK_EN();
32     RCC_AFIO_CLK_EN();
33 }
34
35 void EXTI_CallBack(void)
36 {
37     IRQ_Flag = 1;
38     __asm("MRS %[out0], IPSR"
39           :[out0] "=r" (CPU_IP_Register) );
40
41 }
42
43
44 int main(void)
45 {
46     clk_init();
47

```

Lab1.s

```

1029 8000968: e883 0007 stmia.w r3, {r0, r1, r2}
1030 800096c: 2300    movs r3, #0
1031 800096e: 743b    strb r3, [r7, #16]
1032 8000970: 2301    movs r3, #1
1033 8000972: 747b    strb r3, [r7, #17]
1034 8000974: 4b0e    ldr r3, [pc, #56] ; (80009b0 <main+0>
1035 8000976: 617b    str r3, [r7, #20]
1036 8000978: 1d3b    adds r3, r7, #4
1037 800097a: 4618    mov r0, r3
1038 800097c: f7ff fdb2 bl 80004e4 <MCAL_EXTI_GPIO_Init>
1039 8000980: 4b0c    ldr r3, [pc, #48] ; (80009b4 <main+0>
1040 8000982: 2201    movs r2, #1
1041 8000984: 701a    strb r2, [r3, #0]
1042 8000986: bf00    nop
1043 8000988: bf00    nop
1044 800098a: bf00    nop
1045 800098c: f3ef 8314 mrs r3, CONTROL
1046 8000990: 4a09    ldr r2, [pc, #36] ; (80009b8 <main+0>
1047 8000992: 6013    str r3, [r2, #0]
1048 8000994: bf00    nop
1049 8000996: bf00    nop
1050 8000998: bf00    nop
1051 800099a: 4b06    ldr r3, [pc, #24] ; (80009b4 <main+0>
1052 800099c: 781b    ldrb r3, [r3, #0]
1053 800099e: 2b01    cmp r3, #1
1054 80009a0: d1fb    bne.n 800099a <main+0x42>
1055 80009a2: 4b04    ldr r3, [pc, #16] ; (80009b4 <main+0>

```