

Embedded Systems

Lab Assignment - 1

SHOBHIT KUMAR - 2021MT12072

Answer 1

Code-1: Folder Name Q1-Code-1

Code-2: Folder Name Q1-Code-2

Answers

- a) On reset the mode of operation is **Supervisor**.
- b) States taken by Arithmetic, Load and Store instruction is as follows:
1. Arithmetic Instruction: **1 State**
 2. Load Instruction: **3 States**
 3. Store Instruction: **2 States**
- c) The number of states taken by BGE instruction is not same in case of branch is taken and branch not taken. The following is the number of states taken in each case

1. Branch is not taken: **1 State**

2. Branch is taken: **3 States**

- d) States taken for both programs excluding last

B STOP instruction:

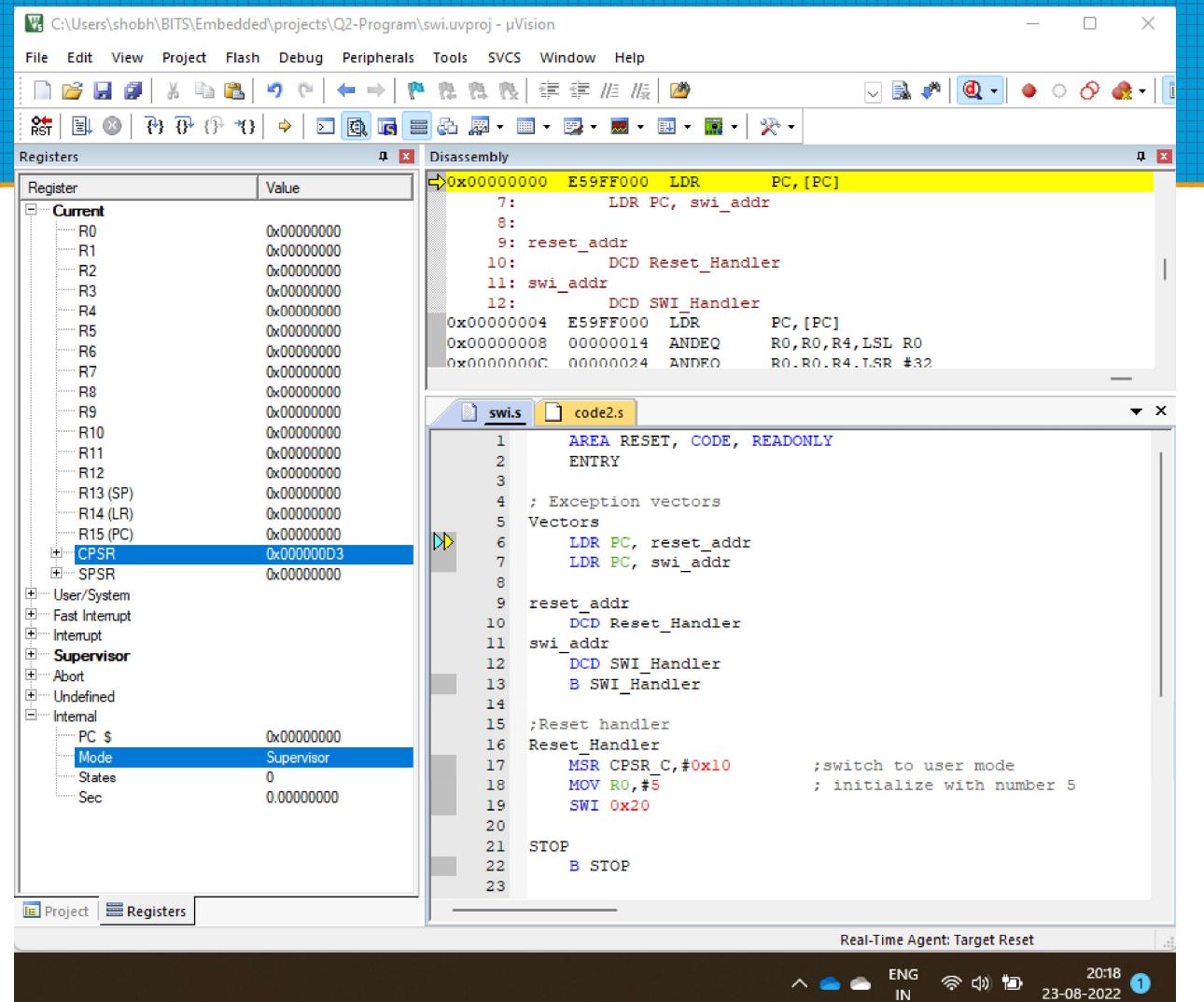
Condition	Code-1 States	Code-2 States
$a < b$	33	30
$a > b$	29	29
$a = b$	29	29

Answer 2

Code: Folder Name : SWI

Step 1:

After reset device is in supervisor mode



Answer 2

Code: Folder Name : SWI

Step 2:

Device is in user mode
after updating CPSR

The screenshot shows the µVision IDE interface with several windows open:

- Registers Window:** Shows the current register values. The PC register is highlighted with a blue selection bar.
- Disassembly Window:** Displays assembly code. The instruction at address 0x0000001C is highlighted with a yellow selection bar. The assembly code for this instruction is: `EF000020 SWI 0x00000020`.
- Source Code Window:** Shows the source code for the file `swi.s`. The assembly code corresponds to the highlighted instruction in the disassembly window.
- Status Bar:** Shows the message "Real-Time Agent: Target Stopped".
- System Tray:** Shows icons for battery, signal strength, and system status, along with the date and time: 23-08-2022 20:20.

Registers Window Data:

Register	Value
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)	0x0000001C
CPSR	0x00000010
SPSR	0x00000000

Disassembly Window Data:

Address	Instruction	Description
0x00000010	EA000003 B	0x00000024
17:	MSR CPSR_C, #0x10	;switch to user mode
0x00000014	E321F010 MSR CPSR_c, #0x00000010	
18:	MOV R0, #5	; initialize with
0x00000018	E3A00005 MOV R0, #0x00000005	
19:	SWI 0x20	
20:	STOP	
21:	21: STOP	
0x0000001C	EF000020 SWI 0x00000020	
22:	B STOP	

Source Code Window Data:

```
12 DCD SWI_Handler
13 B SWI_Handler
14
15 ;Reset handler
16 Reset_Handler
17 MSR CPSR_C, #0x10 ;switch to user mode
18 MOV R0, #5 ; initialize with number 5
19 SWI 0x20
20
21 STOP
22 B STOP
23
24 SWI_Handler
25 LDR SP, =0x40000100
26 STMFD SP!, {R1-R12} ;Store the registers on the stack
27
28 ; write code logic
29 LDR R1, [LR, #-4] ;Since on execution of SWI, LR is
30 ;R0 now contains the address of S
31 BIC R1, R1, #0xFF000000 ;Clear top 8 bits as lower commen
32 CMP R1, #0x20 ;Compare R1(passwd exception num
33 MOVEQ R2, R0 ;if Zero flag set, then load R0 t
34 MOVEQ R3, #10 ;Load number 10 in R3 is zero fla
```

Answer 2

Code: Folder Name : SWI

Step 3:

Device switches to supervisor mode on executing SWI instruction

The screenshot shows the uVision IDE interface with the following windows:

- Registers Window:** Shows the current register values. The CPSR register is highlighted in blue, showing its value as 0x00000093.
- Disassembly Window:** Displays assembly code for the program. The CPSR register is modified at address 0x00000010. The PC is at address 0x0000001C, and the CPSR is at address 0x00000014.
- Code Window:** Shows the source code for the project. It includes a Reset Handler and an SWI Handler. The CPSR is modified in both handlers.

```
Registers window content (highlighted):
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)    0x00000020
  R15 (PC)    0x00000010
  CPSR        0x00000093
  SPSR        0x00000010

User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC $       0x00000010
  Mode       Supervisor
  States     12
  Sec        0.00000300
```

```
Disassembly window content:
0x00000010 EA000003 B 0x00000024
  17: MSR CPSR_C,#0x10 ;switch to user mode
0x00000014 E321F010 MSR CPSR_c,#0x00000010
  18: MOV R0,#$5 ; initialize with
0x00000018 E3A00005 MOV R0,#0x00000005
  19: SWI 0x20
  20:
  21: STOP
0x0000001C EF000020 SWI 0x00000020
  22: B STOP
```

```
Code window content (swi.s):
6   LDR PC, reset_addr
7   LDR PC, swi_addr
8
9   reset_addr
10  DCD Reset_Handler
11  swi_addr
12  DCD SWI_Handler
13  B SWI_Handler
14
15 ;Reset handler
16 Reset_Handler
17  MSR CPSR_C,#0x10 ;switch to user mode
18  MOV R0,#$5 ; initialize with number 5
19  SWI 0x20
20
21 STOP
22 B STOP
23
24 SWI_Handler
25  LDR SP,=0x40000100
26  STMFD SP!,{R1-R12} ;Store the registers on the stack
27
28 ; write code logic
```

Answer 2

Code: Folder Name : SWI

Step 4:

SWI exception code is extracted in R1 register after decrementing LR by 4 as the address of SWI instruction. Clearing top 8 bits we get the exception code

The screenshot shows the uVision IDE interface with the following details:

- Registers Window:** Shows the current state of various registers. The R1 register is highlighted with its value as 0xEF000020.
- Disassembly Window:** Displays assembly code for two files: `swi.s` and `code2.s`.
 - `swi.s` contains instructions to initialize R0 with 5, perform a SWI 0x20, and handle the SWI exception.
 - `code2.s` contains the main logic, including writing code logic, saving registers to the stack, clearing top 8 bits of R1, comparing R1 with 0x20, and performing various arithmetic operations like MOVEQ and MUL.
- Code View:** The assembly code is annotated with comments explaining the purpose of each step, such as "Since on execution of SWI, LR is now contains the address of S" and "Clear top 8 bits as lower commen".

Answer 2

Code: Folder Name : SWI

Step 5:

Multiplication is done
and R0 has the result

The screenshot shows the uVision IDE interface with the following details:

- Registers Window:** Shows the current values of various registers. The R0 register is highlighted with a blue selection bar, containing the value `0x00000032`. Other registers like R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13 (SP), R14 (LR), and R15 (PC) also have their current values displayed.
- Disassembly Window:** Displays the assembly code for the project. The assembly code includes:
 - Line 36: `MUL R0,R2,R3`
 - Line 37: `LDMFD SP!, {R1-R12}` ; Restore the register base
 - Line 38: `E8BD1FFE LDMIA R13!, {R1-R12}`
 - Line 39: `MOVS PC,LR`
 - Line 40: `E1B0F00E MOVS PC,R14`
 - Line 41: `0x0000004C 40000100 ANDMI R0,R0,R0,LSL #2`
 - Line 42: `0x00000050 00000000 ANDEQ R0,R0,R0`
 - Line 43: `0x00000054 00000000 ANDEQ R0,R0,R0`
 - Line 44: `0x00000058 00000000 ANDEQ R0,R0,R0`
- Code Editor:** Shows two files: `swi.s` and `code2.s`.
 - `swi.s` contains:
 - Line 18: `MOV R0,#5` ; initialize with number 5
 - Line 19: `SWI 0x20`
 - Line 20: `STOP`
 - Line 21: `B STOP`
 - Line 24: `SWI_Handler`
 - Line 25: `LDR SP,=0x40000100`
 - Line 26: `STMFD SP!, {R1-R12}` ; Store the registers on the stack
 - Line 28: `; write code logic`
 - Line 29: `LDR R1,[LR,-4]` ; Since on execution of SWI, LR is
 - Line 30: `BIC R1,R1,#0xFF000000` ; R0 now contains the address of S
 - Line 31: `CMP R1,#0x20` ; Clear top 8 bits as lower common
 - Line 32: `MOVEQ R2,R0` ; Compare R1(passwd exception num
 - Line 33: `MOVEQ R3,#10` ; if Zero flag set, then load R0 t
 - Line 34: `MUL R0,R2,R3` ; Load number 10 in R3 is zero fla
 - Line 35: `LDMFD SP!, {R1-R12}` ; Restore the register back on sta
 - Line 36: `MOVS PC,LR`
 - `code2.s` contains:
 - Line 18: `MOV R0,#5` ; initialize with number 5
 - Line 19: `SWI 0x20`
 - Line 20: `STOP`
 - Line 21: `B STOP`
 - Line 24: `SWI_Handler`
 - Line 25: `LDR SP,=0x40000100`
 - Line 26: `STMFD SP!, {R1-R12}` ; Store the registers on the stack
 - Line 28: `; write code logic`
 - Line 29: `LDR R1,[LR,-4]` ; Since on execution of SWI, LR is
 - Line 30: `BIC R1,R1,#0xFF000000` ; R0 now contains the address of S
 - Line 31: `CMP R1,#0x20` ; Clear top 8 bits as lower common
 - Line 32: `MOVEQ R2,R0` ; Compare R1(passwd exception num
 - Line 33: `MOVEQ R3,#10` ; if Zero flag set, then load R0 t
 - Line 34: `MUL R0,R2,R3` ; Load number 10 in R3 is zero fla
 - Line 35: `LDMFD SP!, {R1-R12}` ; Restore the register back on sta
 - Line 36: `MOVS PC,LR`

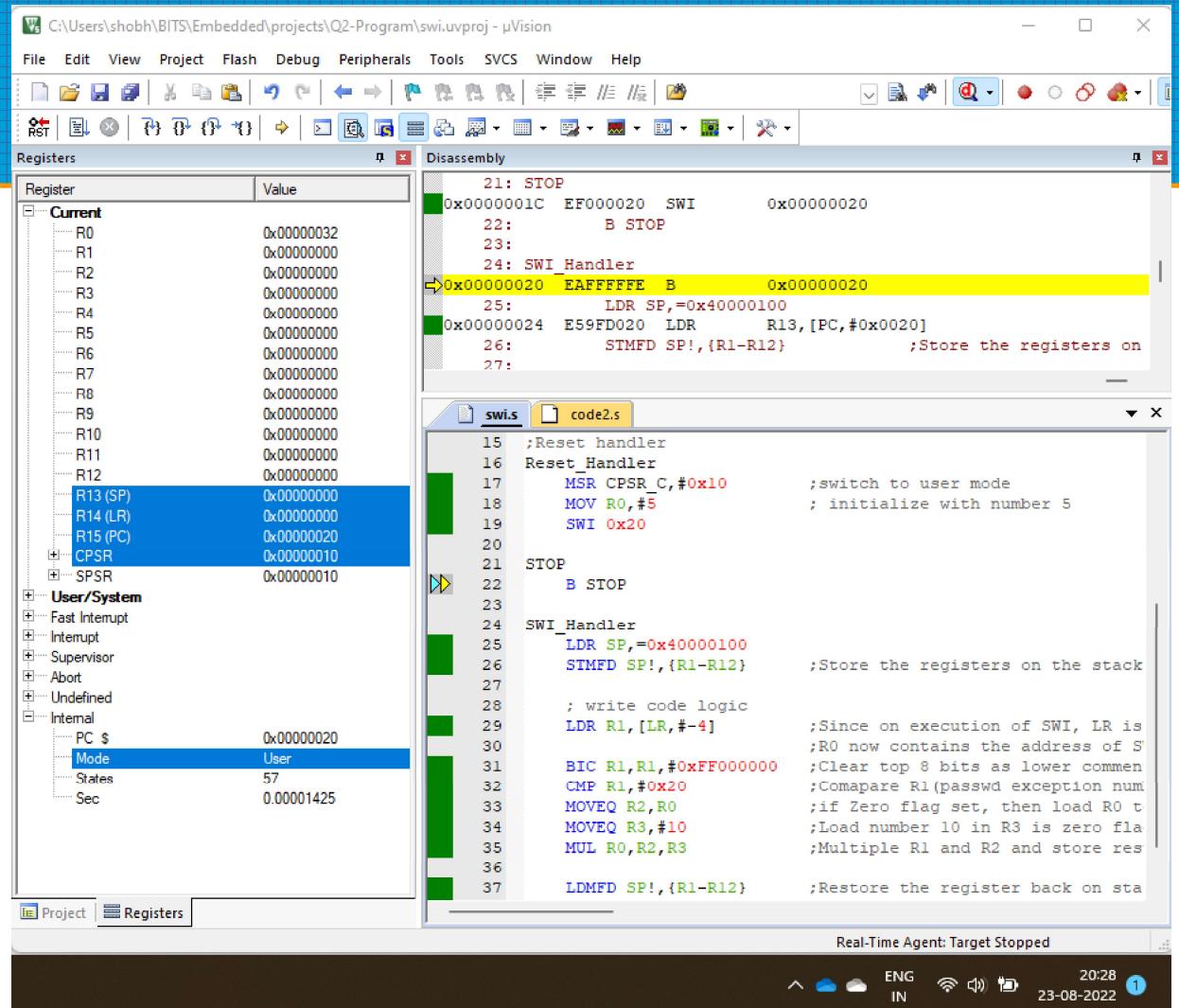
At the bottom of the interface, there is a status bar with the message "Real-Time Agent: Target Stopped".

Answer 2

Code: Folder Name : SWI

Step 6:

Return to user mode. R0 is maintained as multiplication result as exception code passed was 0x20



Answer 2

Code: Folder Name : SWI

Step 7:

Trying with SWI 0x25. After executing BIC which clears top 8 bits R1 will have 0x25

The screenshot shows the µVision IDE interface with the following details:

- Registers Pane:** Shows the current register values. R1 is highlighted with a blue selection bar, showing its value as 0xEF000025.
- Disassembly Pane:** Displays assembly code with comments. The highlighted instruction is at address 0x00000030: `E3C114FF BIC R1,R1,#0xFF000000`. This instruction is described as "Clear top 8 bits as lower commen".
- Code Editor:** Shows two files: `swi.s` and `code2.s`.
 - `swi.s` contains the assembly code for the SWI handler, including the BIC instruction at address 0x00000030.
 - `code2.s` contains the main program logic, including the initialization of R0 to 5 and the execution of the SWI instruction.
- Status Bar:** Shows "Real-Time Agent: Target Reset" and the system status icons.

Answer 2

Code: Folder Name : SWI

Step 8:

When we return to usermode R0 is maintained as 0x00000000

The screenshot shows the µVision IDE interface with the following details:

Registers Pane:

Register	Value
R0	0x00000000
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)	0x00000020
CPSR	0x00000010
SPSR	0x00000010

User/System pane:

- Fast Interrupt
- Interrupt
- Supervisor
- Abort
- Undefined
- Internal
 - PC \$ 0x00000020
 - Mode User
 - States 57
 - Sec 0.00001425

Disassembly Pane:

```
21: STOP
0x0000001C EF000025 SWI      0x00000025
22: B STOP
23:
24: SWI_Handler
→ 0x00000020 EAFFFFFE B      0x00000020
25: LDR SP,=0x40000100
0x00000024 E59FD020 LDR      R13,[PC,#0x0020]
26: STMFD SP!,{R1-R12}        ;Store the registers on
27:

15: ;Reset handler
16: Reset_Handler
17: MSR CPSR_C,#0x10          ;switch to user mode
18: MOV R0,#5                 ; initialize with number 5
19: SWI 0x25
20:
21: STOP
22: B STOP
23:
24: SWI_Handler
25: LDR SP,=0x40000100
26: STMFD SP!,{R1-R12}        ;Store the registers on the stack
27:
28: ; write code logic
29: LDR R1,[LR,-4]             ;Since on execution of SWI, LR is
30: ;R0 now contains the address of S'
31: BIC R1,R1,#0xFF000000     ;Clear top 8 bits as lower commen
32: CMP R1,#0X20               ;Comapare R1(passwd exception num
33: MOVEQ R2,R0                ;if Zero flag set, then load R0 t
34: MOVEQ R3,#10               ;Load number 10 in R3 is zero fla
35: MUL R0,R2,R3              ;Multiple R1 and R2 and store res
36:
37: LDMFD SP!,{R1-R12}         ;Restore the register back on sta
```

Code Editor:

```
Real-Time Agent: Target Reset
```

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

swi.s code2.s

2022-08-23 20:30 1