

گزارش پیاده‌سازی تمرین پنجم

متین زیودار - 97243037

سوال 1


- a. درست
- b. نادرست - مقدار immediate بایستی دارای # باشد.
- c. نادرست - مقصد این دستور نامعتبر است و نیز مقدار immediate بایستی دارای # باشد.
- d. درست
- e. نادرست - در ARM تنها 16 رجیستر داریم (R0 - R15).
- f. درست
- g. نادرست - در ARM تنها 16 رجیستر داریم (R0 - R15).

سوال 2

در معماری ARM، رجیستر C در رجیسترهای flag بیانگر بیت Carry می‌باشد. در نتیجه باید این بیت را در محاسبه‌های صورت گرفته بدست آوریم.


a. این مقدار برابر با 1 می‌شود.

$$\begin{array}{r} \text{FFFFFF54} \\ + \text{FFFFFFC4} \\ \hline \text{1FFFFFF18} \end{array}$$

 Carry = 1

b. این مقدار برابر با 0 می‌شود.

$$\begin{array}{r} \text{00000000} \\ + \text{FFFFFFFF} \\ \hline \text{0FFFFFFFF} \end{array}$$

 Carry = 0

سوال 3

برای محاسبه ب.م.م از تابع بازگشتی استفاده کردیم که سودو کد آن به صورت زیر است. برای پیاده‌سازی کد را تبدیل به کد ARM کردیم.

```
int gcd (int a, int b){
    if (a == 0)
        return b;
    return gcd(b % a, a);
}
```

نتیجه در Keil:

The screenshot displays the Keil uVision IDE interface during the assembly of the GCD function. The Registers window on the left lists registers R0 through R15 and the Program Status Register (PSR). The Disassembly window shows the assembly code for the GCD function, with a red arrow pointing to the instruction `MOV R1, #12`. The Memory window on the right shows the memory dump. The Command window at the bottom shows an error message: `*** error 65: access violation at 0x08080000 : no 'execute/read' permission`.

Registers:

Register	Value
R0	0x00000004
R1	0x0000000C
R2	0x00000008
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000600
R14 (LR)	0x080001F1
R15 (PC)	0x08080000
PSR	0x21000000

Disassembly:

```

1  EXPORT SystemInit
2  EXPORT __main
3
4  AREA MYPROG, CODE, READONLY
5
6  SystemInit FUNCTION
7
8  MOV R1, #12 ;a = 12
9  MOV R2, #8 ;b = 8
10 ENDFUNC
11
12 ; add numbers from 1 to 9
13
14 ; main logic of code
15
16 __main FUNCTION
17 STR R1, [SP, #-4]! ;Caller-Saved register
18 STR R2, [SP, #-4]! ;Caller-Saved register
19 BL GCD
20 LDR R2, [SP], #4 ;Caller-Saved register
21 LDR R1, [SP], #4 ;Caller-Saved register
22
23 B END_F
24
25 GCD
26 STR LR, [SP, #-4]! ;save LR on Stack (Callee-saved)
27 CMP R1, #0

```

Memory:

Address	Value
0x20000500	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000510	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000520	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000530	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000540	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000550	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000560	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000570	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000580	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000590	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200005A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200005B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200005C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200005D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200005E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200005F0	29 02 00 08 F1 01 00 08 08 00 00 00 0C 00 00 00
0x20000600	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000610	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000620	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000630	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000640	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000650	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000660	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000670	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Command:

```

Load "D:\university\term_6\riz\Homeworks\src\008 GCD\Objects\HW5_Q3.axf"
*** error 65: access violation at 0x08080000 : no 'execute/read' permission

```

Call Stack + Locals:

Name	Location/Value	Type
Reset_Handler	0x00000000	void f()

Simulation: t1: 0.02183267 sec L9 C:1 CAP: NUM SCRL: OVR: R/W

سوال 4

برای محاسبه *reverse* ابتدا بیت آخر را بدست می‌آوریم (شیفت به راست به اندازه 31 بیت) و سپس آن را به مکان مورد نظر خود می‌بریم (شیفت به راست با توجه به این که کدام مکان عدد جدید قرار می‌گیرد) و در نهایت با رجیستر R1 که مقدار نهایی را نگه می‌دارد جمع می‌کنیم.

منطق *factorial* این سوال همانند تمرین x86 می‌باشد. در اینجا هم از استک استفاده شده است. در هر مرحله برای آنکه مقدار i را نگه داریم که در $factorial(i - 1)$ ضرب کنیم، مقادیر رجیستر R1 را روی استک پوش می‌کنیم و سپس پس از فراخوانی بازگشتی، آن را پاپ می‌کنیم. بیشترین عدد قابل محاسبه 12! است زیرا نتیجه 13! در 32 بیت جا نمی‌شود. نتیجه در Keil:

The screenshot shows the Keil uVision IDE with the following components:

- Registers Window:** Displays the state of various registers. R10 is highlighted with a red arrow and contains the value 0x1C8CFC00. Other registers like R0, R1, R2, etc., contain zero or specific values.
- Disassembly Window:** Shows the assembly code for the `SystemInit` and `main` functions. The `REVERSE` function is defined, which calculates the factorial of a number by shifting bits. The code includes comments like `;n = 30000000` and `;reversed(n) = 0000000B = 12`.
- Command Window:** Shows the load command and an error message: `*** error 65: access violation at 0x08080000 : no 'execute/read' permission`.
- Call Stack + Locals Window:** Shows the current call stack with a single entry for `0x00000000`.
- Simulation Status:** At the bottom, it indicates the simulation is running with a time of 0.02186825 sec.

A red text annotation in the disassembly window states: $factorial(12) = 479001600 = 0x1C8CFC00$.

سوال 5

منطق حل این سوال اینگونه است که از 3 بیت پایین شروع کرده و در هر دور حلقه این مقدار را که در R0 ذخیره شده است را با $5 = (101)_2$ مقایسه می‌کنیم و اگر برابر باشد، مقدار counter (رجیستر R1) را 1 واحد زیاد می‌کنیم. تعداد الگوی "101" در شماره دانشجویی من ($101110010111100111110011101 = 97243037$) برابر با 3 است. نتیجه در Keil:

The screenshot displays the Keil uVision IDE interface during a simulation. The **Registers** window on the left shows the state of the Cortex-M3 registers. Register R10 is highlighted with a red arrow and contains the value 0x05C8CF9D. The **Disassembly** window shows the assembly code for the `main` function, with the instruction `MOV R10, #0x50000000` highlighted. The **Source** window shows the corresponding C code, with the line `R10 = 0x05C8CF9D = 97243037` highlighted. The **Command** window at the bottom displays an error message: "Load "D:\\university\\term_6\\riz\\Homeworks\\src\\010 Sequence Detector\\Objects\\HW5_Q5.axf" *** error 65: access violation at 0x08080000: no 'execute/read' permission". The **Call Stack + Locals** window shows the current call stack, with the `Reset_Handler` function at the bottom.

سوال 6

در این سوال ابتدا not رجیستر R1 را در R0 میریزیم سپس در یک حلقه بیت‌ها را متناظراً در R5 میریزیم. اگر شماره بیت در بازه‌ی i تا j باشد، از رجیستر R0 (مقدار toggle شده) و در غیر این صورت از رجیستر R1 (مقدار اصلی) بر میداریم. در نهایت نتیجه نهایی را در R1 می‌نویسیم. نتیجه در Keil:

The screenshot displays the Keil uVision IDE interface with the following components:

- Registers Window:** Shows the state of various registers. R1 is highlighted with a red arrow and contains the value 0x12345678. R5 contains 0x12345978.
- Disassembly Window:** Shows assembly instructions. The first instruction is MOV r0, r0 at address 0x08080000.
- C Code Window:** Shows the implementation of the toggle function. The function SystemInit is defined, and the main logic is in the __main function. The code uses MOV and ORR instructions to toggle bits in R1 based on the value in R5.


```

      EXPORT SystemInit
      EXPORT __main

      AREA MYPROG, CODE, READONLY

      SystemInit FUNCTION
      ; MOV R1, #0x12345678
      MOV R1, #0x12000000
      ORR R1, #0x340000
      ORR R1, #0x5600
      ORR R1, #0x78
      MOV R2, #8 ; i = 8
      MOV R3, #12 ; j = 12
      ENDFUNC

      ; main logic of code
      __main FUNCTION
      MVN R0, R1 ; R0 = (not)R1
      MOV R4, #32 ; loop counter
      MOV R5, #0 ; result
      
```
- Command Window:** Shows an error message: "Load 'D:\university\term_6\riz\Homeworks\src\011 Toggle i-j\Objects\HW5_Q6.axf' *** error 65: access violation at 0x08080000 : no 'execute/read' permission".
- Call Stack + Locals Window:** Shows the current function being executed, Reset_Handler, at location 0x00000000.