دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

12/14/2021

Homework 5

Lec 19-21

MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2021

توجه: با توجه به اینکه با paste کردن کد دندانه گذاری آن خراب میشـد و نامرتب میشـد، عکس هایی از کد رو گذاشتم. تکست کد نیز در فایل موجود میباشد

1) برنامه‌ای بنویسید که با استفاده از آن بتوان تشخیص داد که مقدار قرارگرفته در رجیستر R0 پالیندروم است یا خیر. (برای مثال، 0110 یک پالیندروم 4 بیتی است.)

```
    AREA myData, DATA
CONST EQU 0x14411441;

NUMBER RN r0;
REVERSE RN r1;

NUM RN r2;

TMP RN r3;
TMP_2 RN r4;

LAST_DIGIT RN r5;


    EXPORT __main
    AREA myCode, CODE, READONLY
    ENTRY

__main

    LDR NUMBER, =CONST;
    MOV NUM, NUMBER;
    LDR REVERSE, =0; reverse
    LDR TMP, =32;
    ;LDR TMP_2, =1;
loop

    MOVS NUMBER, NUMBER, LSL #1; shift to left and mov

    MOVS REVERSE, REVERSE, RRX; shift to right and mov

    SUBS TMP, #1;
    BNE loop;

    CMP NUM, REVERSE; compare number with its reverse.
HERE B HERE
    END
```

2) هنگامی که بر روی کیبورد، دو کاراکتر 4 و 6 را تایپ می‌کنیم، 0x34 و0x36 در واقع به ما داده می‌شود.
برنامه‌ای بنویسید که 0x34 و 0x36 را به packed BCD تبدیل کرده و نتیجه را در رجیستر R2 ذخیره نماید.

```
1       AREA myData, DATA
2  NUM1 RN r0;
3  NUM2 RN r1;
4  RESULT RN R2;
5
6       EXPORT __main
7       AREA myCode, CODE, READONLY
8       ENTRY
9
10 __main
11
12      LDR NUM1, =0x34;
13      LDR NUM2, =0x36;
14
15      SUB NUM1, NUM1, #48;
16      SUB NUM2, NUM2, #48;
17
18      LSL NUM1, NUM1, #4;
19      ADD RESULT, NUM1,NUM2;
20
21 HERE B HERE;
22      END
```

3) الف) کد اسمبلی معادل قطعه کد زیر را بنویسید. (مقادیر متغیرهای استفاده شده در ثبات‌ها طبق جدول زیر ذخیره شده‌است.)

a	R4
b	R5
c	R6

```
while (a - b > 0) {
    if (a > -b) {
        c = c - a;
        a = -a;
    }
    else {
        b = c * b;
        a = 2 - b;
    }
}
```

```
1     AREA myData, DATA
2  TMP RN R0;
3
4     EXPORT __main
5     AREA myCode, CODE, READONLY
6     ENTRY
7
8  __main
9
10 loop
11    CMP r4, r5;
12    BLS HERE;
13    CMN r4, r5;
14    BLS INSIDE_ELSE;
15    SUB r6, r6, r4;
16    LDR TMP, =0;
17    SUB r4, TMP, R4;
18    B loop;
19
20 INSIDE_ELSE
21    MUL r5, r6, r5;
22    LDR TMP, =2;
23    SUB r4, TMP, r5;
24    B loop;
25
26 HERE B HERE; stay here forever
27
```

ب) اگر قصد داشته باشیم کد if ((R0==R1) && (R2==R3)): R4++ را به زبان اسمبلی بنویسیم، کد مناسب را فقط با سه دستور پیاده کنید.

```
1     AREA myData, DATA
2
3     EXPORT __main
4     AREA myCode, CODE, READONLY
5     ENTRY
6
7  __main
8     CMP r0, r1;
9     CMPEQ r2, r3;
10    ADD r4, #1;
11
12 HERE B HERE; stay here forever
13
14
15
```

4) برنامه‌ای بنویسید که مقدار ب م م و ک م م دو مقدار ذخیره شده در R0 و R1 را محاسبه کرده و به ترتیب در R2و R3 ذخیره کند.

```
    AREA myData, DATA
CONST_1 EQU 0x34

num1 RN r0;
num2 RN r1;

n1 RN r2; temp numbers that we work them for gcd
n2 RN r3;


GCD RN r4;
LCM RN r5;

TMP RN r6;


    EXPORT __main
    AREA myCode, CODE, READONLY
    ENTRY

__main

    LDR num1, =54;
    LDR num2, =13;
    MOV n1, num1;
    MOV n2, num2;

loop

    CMP n1,n2;
    BNE inside_if_else;
    MOV GCD, n1;
    B inside_lcm; end the loop if n1 == n2

inside_if_else
    CMP n1, n2;
    BLS inside_else;

    SUB n1, n1, n2;inside if
    B loop;
inside_else
    SUB n2, n2, n1;inside else
    B loop;

inside_lcm ;this line calculate the lcm
    MUL TMP, num1, num2;
    UDIV LCM, TMP, GCD;

HERE B HERE;
    END
```

4) برنامه‌ای بنویسید که مقدار ذخیره شده در ثبات R0 را در یک آرایه 10 عضوی به روش دودویی، جستجو کند(binary search) (فرض کنید که آرایه از قبل به صورت صعودی، مرتب شده است.)

(در صفحه ی بعد)

```
1     AREA myData, DATA, READWRITE
2   value_to_find_const EQU 9;
3   mem_addr RN r4;
4   mem_val RN r5;
5   left RN r6;
6   right RN r7;
7   mid RN r8;
8   tmp RN r9;
9   value_to_find RN r10;
10  found RN r11;
11  arr_size EQU 10;
12      EXPORT __main
13      AREA myCode, CODE, READONLY
14      ENTRY
15
16  __main
17      LDR mem_addr, =ARR;
18      LDR left, =0;
19      LDR right , =arr_size;
20      SUB right,right,#1;
21      LDR mid, =0;
22      LDR value_to_find, =value_to_find_const;
23      LDR found, =0;
24
25  loop
26      CMP left, right; check to loop condition here
27      BLS inside_loop;
28      B here;
29  inside_loop
30      ADD mid, left, right;
31      LDR tmp, =2;
32      UDIV mid, mid, tmp; calculating the mid -> mid=(left+right)/2
33
34      LDR tmp,=ARR;
35      ADD tmp, mid, tmp;
36      LDRB mem_val, [tmp]; load just one byte
37
38      CMP mem_val, value_to_find;
39      BNE else_if;
40      LDR found, =1;
41      B here;
42
43  else_if
44      BHI inside_else;
45      ADD left, mid, #1;
46      B loop;
47
48  inside_else
49      SUB right, mid, #1;
50      B loop;
51
52      ALIGN 4;
53  ARR DCB 1,2,2,4,5,6,7,9,11,13;
54
55  here B here;
56
57      END
58
```

6) برنامه‌ای بنویسید که جمله‌n ام دنباله ی فیبوناچی را در ثبات R1 قرار دهد(مقدار n در ثبات R0 قرار گرفته است.)

```
    AREA myData, DATA, READWRITE

Nth RN r0;
fib_val RN r1;

TMP_1 RN r2;
TMP_2 RN r3;
TMP_3 RN r4;


    EXPORT __main
    AREA myCode, CODE, READONLY
    ENTRY

__main
    LDR Nth,=10; 5th member of fib
    LDR TMP_1, =0;
    LDR TMP_2, =1;
    LDR fib_val, =0;
    CMP Nth,#1;
    BHI loop;if higher than one

    BNE here;
    LDR fib_val, = 1;
    B here;
loop
    SUB Nth, Nth, #2;

inside_loop
    MOV TMP_3, TMP_2;
    ADD TMP_2, TMP_1, TMP_2;
    MOV TMP_1, TMP_3;

    SUBS Nth, #1;
    BNE inside_loop;

    MOV fib_val,TMP_2;
here B here;

    END
```

- مهلت ارسال تمرین ساعت 23.55 روز **جمعه سوم دی‌ماه** می‌باشد.
- سوالات خود را می‌توانید از طریق تلگرام از تدریسیارهای گروه خود بپرسید.
- کدهای اسمبلی را با استفاده از keil انجام دهید.
- ارائه پاسخ تمرین به بهتر است به روش‌های زیر باشد:
  1) ارائه اسکرین شات از کد و نتیجه اجرای آن در یک فایل pdf
  2) قرار دادن فایل کد و اسکرین شات از نتیجه اجرای کد. در صورت استفاده از این روش حتما هر سوال را در پوشه جداگانه قرار دهید.
- فایل پاسخ تمرین را تنها با قالب **HW5 -9731***.zip** یا **HW5 -9731***.pdf** در مودل بارگذاری کنید.
  - نمونه: HW5-9731097