

12/14/2021



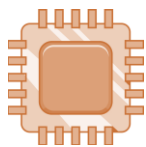
Homework Solution 5

Lec 19-21



MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Fall 2021



۱) برنامه‌ای بنویسید که با استفاده از آن بتوان تشخیص داد که مقدار قرار گرفته در رجیستر R0 پالیندروم است یا خیر. (برای مثال، ۰۱۱۰ یک پالیندروم ۴ بیتی است).
ابتدا بایستی مقدار قرار گرفته در R0 را reverse و سپس آن را با مقدار اولیه داده شده در ثبات R0 مقایسه کنیم. اگر دو مقدار برابر باشند، پالیندروم داریم.

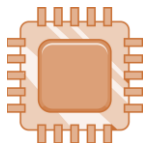
```
AREA my_data, DATA
INPUT EQU 0x80000001 ;change value for different inputs

EXPORT __main
AREA palindrome, CODE, READONLY
ENTRY

__main
    LDR r0,=INPUT
    MOV r1,r0 ; make copy of input
    MOV r2,#0 ;used for storing inverted number
    MOV r3,#32 ; loop counter
    MOV r4,#0 ; is the result

loop
    AND r5,r1,#1 ;take only the first bit
    LSL r2,r2,#1 ; shift one to left
    ADD r2,r2,r5
    LSR r1,r1,#1
    SUBS r3,r3,#1
    BNE loop

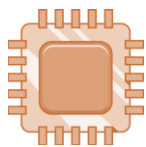
    CMP r0,r2
    BNE here
    MOV r4,#1
here b here
END
```



۲) هنگامی که بر روی کیبورد، دو کاراکتر ۴ و ۶ را تایپ می‌کنیم، 0x34 و 0x36 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.

```
EXPORT __main
AREA bcd, CODE, READONLY
ENTRY

__main
MOV R0, 0x34 ; input 4
MOV R1, 0x36 ; input 6
MOV R3, 0xF ; used for AND to make unpacked BCD
;turn r0 and r1 to bcd of each
AND R0, R0, R3
AND R1, R1, R3
;combine the two BCD values in one register
LSL R2, R0, #4
ADD R2, R1
here b here
END
```

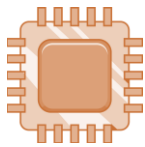


۳) الف) کد اسمبلی معادل قطعه کد زیر را بنویسید. (مقادیر متغیرهای استفاده شده در ثبات‌ها طبق جدول زیر ذخیره شده‌است).

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;  
    }  
}
```

```
export __main  
area start ,code, readonly  
entry  
  
__main  
    b while  
while  
    neg r7,r5  
    cmp r4,r5  
    bhi ifelse  
  
ifelse  
    cmp r4,r7  
    bhi iff  
    mul r5,r5,r6  
    rsb r5,r5,#2  
iff  
    sub r6,r6,r4  
    neg r4,r4  
    b while  
  
end
```



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

```
AREA myData, DATA
```

```
EXPORT __main
```

```
AREA myCode, CODE, READONLY
```

```
ENTRY
```

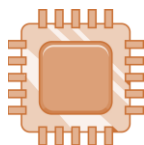
```
__main
```

```
    CMP r0, r1;
```

```
    CMPEQ r2, r3;
```

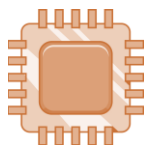
```
    ADD r4, #1;
```

```
HERE B HERE; stay here forever
```



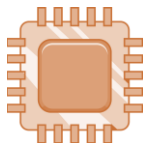
۴) برنامه‌ای بنویسید که مقدار ب م و ک م دو مقدار ذخیره شده در R0 و R1 را محاسبه کرده و به ترتیب در R2 و R3 ذخیره کند.

```
1  AREA myData, DATA
2  CONST_1 EQU 0x34
3
4  num1 RN r0;
5  num2 RN r1;
6
7  n1 RN r2; temp numbers that we work them for gcd
8  n2 RN r3;
9
10
11  GCD RN r4;
12  LCM RN r5;
13
14  TMP RN r6;
15
16
17  EXPORT __main
18  AREA myCode, CODE, READONLY
19  ENTRY
20
21  __main
22
23  LDR num1, =54;
24  LDR num2, =13;
25  MOV n1, num1;
26  MOV n2, num2;
27
28  loop
29
30  CMP n1,n2;
31  BNE inside_if_else;
32  MOV GCD, n1;
33  B inside_lcm; end the loop if n1 == n2
34
35  inside_if_else
36  CMP n1, n2;
37  BLS inside_else;
38
39  SUB n1, n1, n2;inside if
40  B loop;
41  inside_else
42  SUB n2, n2, n1;inside else
43  B loop;
44
45  inside_lcm ;this line calculate the lcm
46  MUL TMP, num1, num2;
47  UDIV LCM, TMP, GCD;
48
49  HERE B HERE;
50  END
```



۵) برنامه‌ای بنویسید که مقدار ذخیره شده در ثبات R0 را در یک آرایه ۱۰ عضوی به روش دودویی، جستجو کند (binary search) (فرض کنید که آرایه از قبل به صورت صعودی، مرتب شده است).

```
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
```



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

```
1      AREA myData, DATA, READWRITE
2
3      Nth RN r0;
4      fib_val RN r1;
5
6      TMP_1 RN r2;
7      TMP_2 RN r3;
8      TMP_3 RN r4;
9
10
11     EXPORT __main
12     AREA myCode, CODE, READONLY
13     ENTRY
14
15     __main
16     LDR Nth, =10; 5th member of fib
17     LDR TMP_1, =0;
18     LDR TMP_2, =1;
19     LDR fib_val, =0;
20     CMP Nth, #1;
21     BHI loop; if higher than one
22
23     BNE here;
24     LDR fib_val, = 1;
25     B here;
26 loop
27     SUB Nth, Nth, #2;
28
29     inside_loop
30     MOV TMP_3, TMP_2;
31     ADD TMP_2, TMP_1, TMP_2;
32     MOV TMP_1, TMP_3;
33
34     SUBS Nth, #1;
35     BNE inside_loop;
36
37     MOV fib_val, TMP_2;
38     here B here;
39
40     END
41
```