Experiment 3

ARM Assembly- Computations in ARM

Target:

- Understanding the architecture of ARM processor.
- Learn ARM instruction sets particularly related with computations.
- Understanding the example programs.
- Learn and write assembly programs for some computational problems.

Questions:

1. Compute the factorial of a given number using ARM processor through assembly programming

2. Combine the low four bits of each of the four consecutive bytes beginning at LIST into one 16-bit halfword. The value at LIST goes into the most signicant nibble of the result. Store the result in the 32-bit variable RESULT

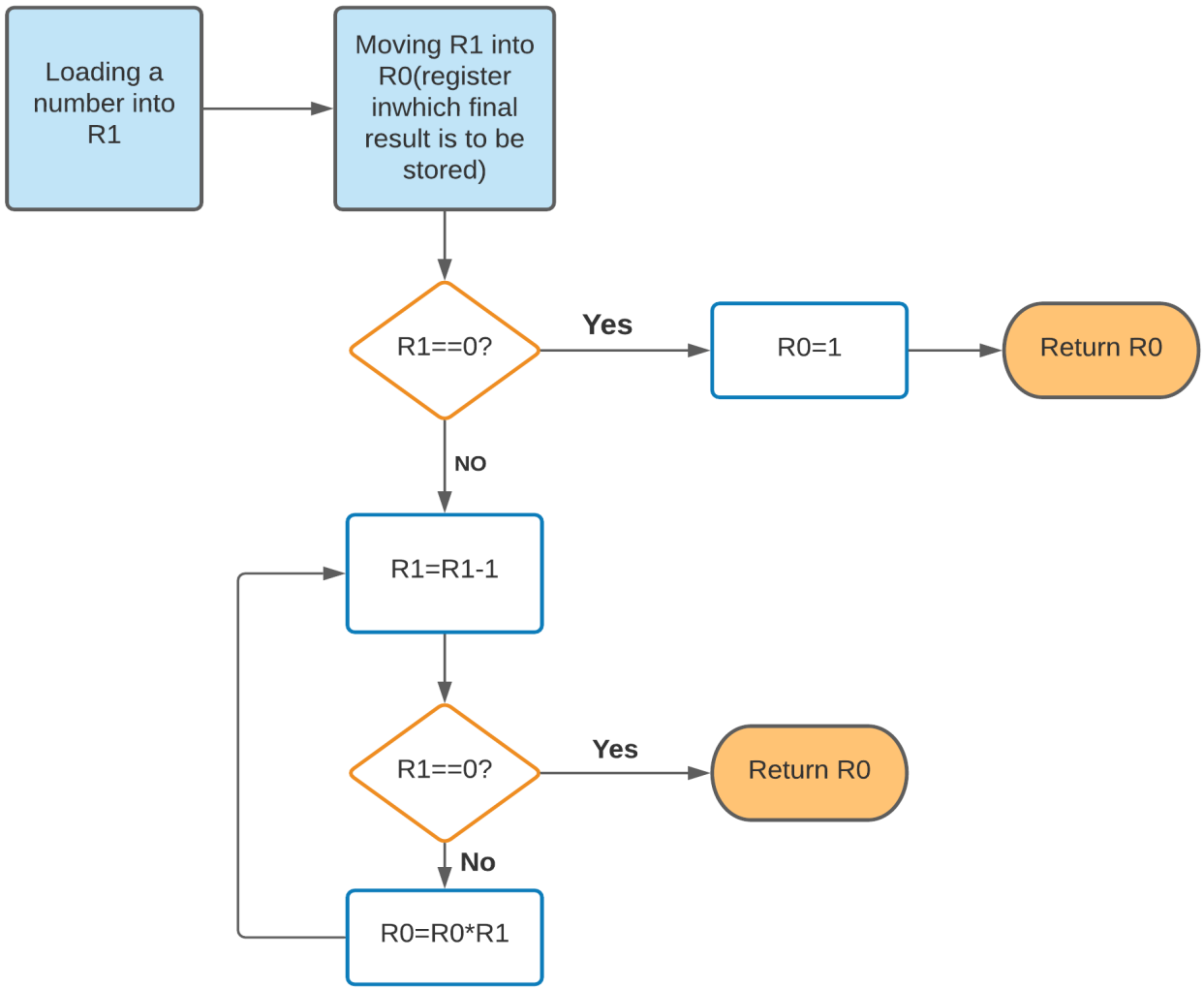
3. Given a 32 bit number, identify whether it is an even or odd. (You implementation should not involve division).

Solutions:

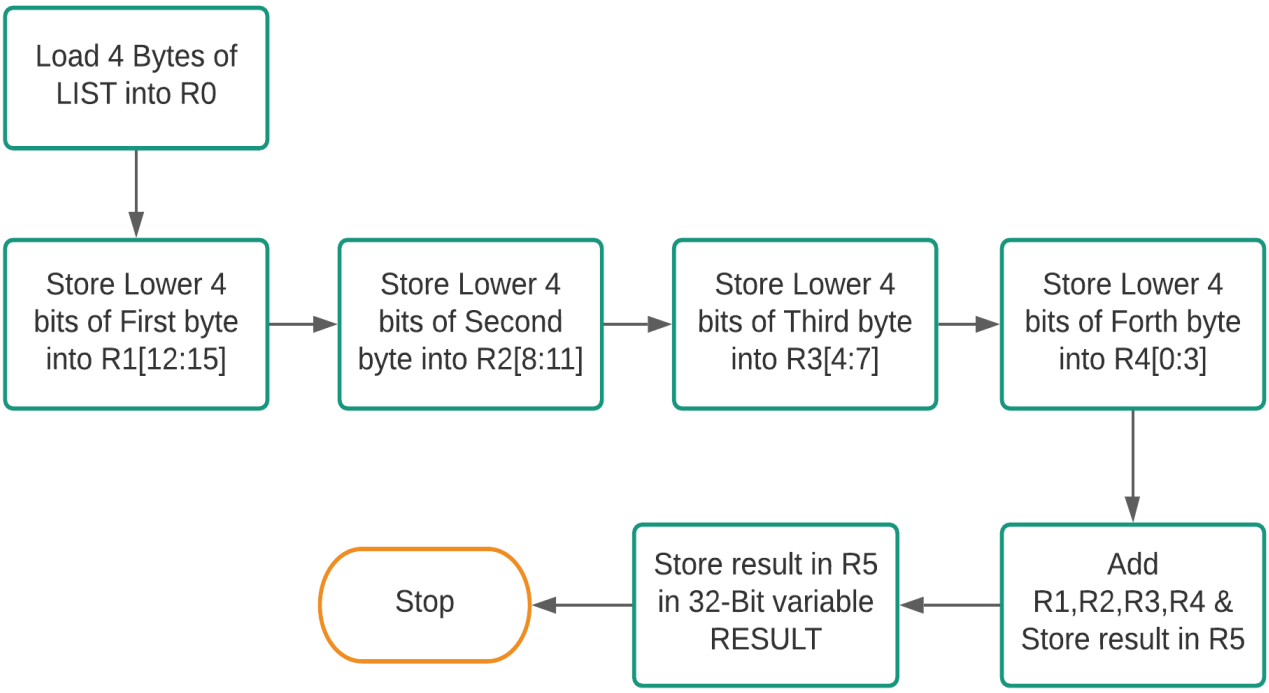
(a)FLOW CHARTS:

Question 1:

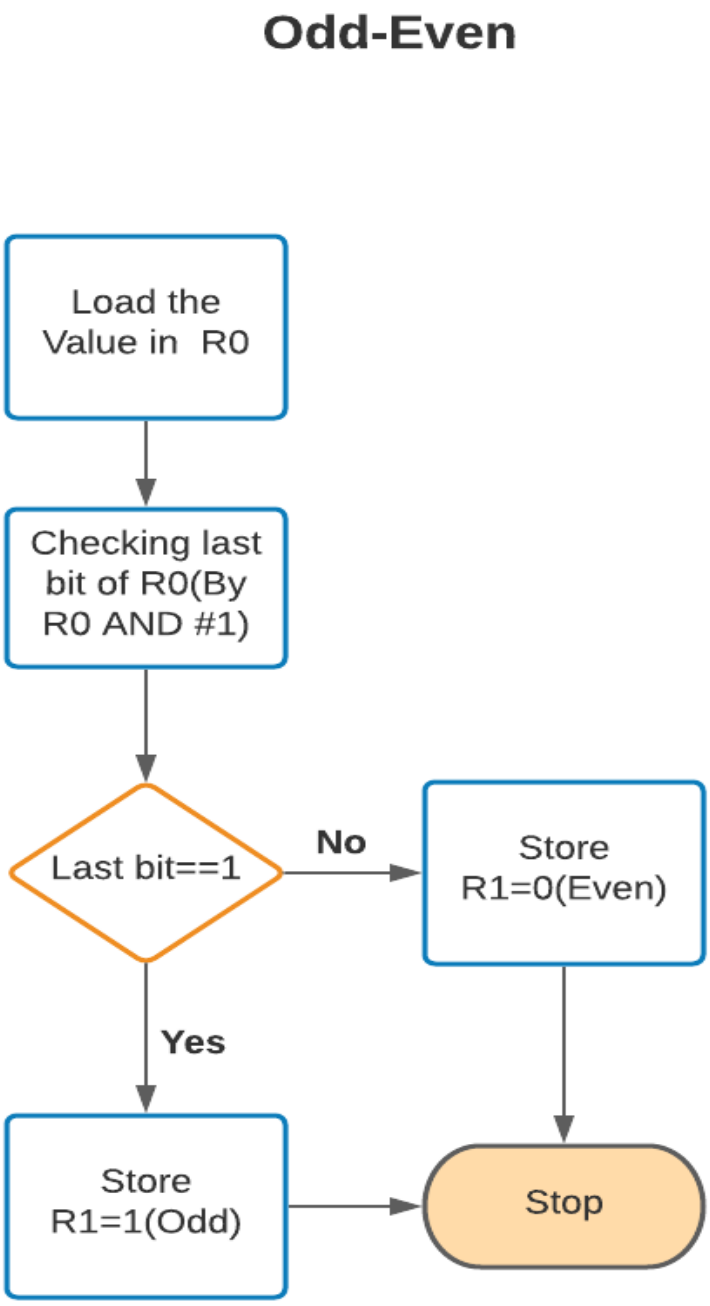
FACTORIAL



Question2:


Load 4 Bytes of LIST into R0
Store Lower 4 bits of First byte into R1[12:15]
Store Lower 4 bits of Second byte into R2[8:11]
Store Lower 4 bits of Third byte into R3[4:7]
Store Lower 4 bits of Forth byte into R4[0:3]
Add R1,R2,R3,R4 & Store result in R5
Store result in R5 in 32-Bit variable RESULT
Stop

Question3:


Odd-Even
Load the Value in R0
Checking last bit of R0(By R0 AND #1)
Last bit==1
No
Store R1=0(Even)
Yes
Store R1=1(Odd)
Stop

(b)CODES:

Question1:

```
     AREA factorial,CODE,READONLY;
     MOV R1,#5
     MOVS R0,R1
     BEQ FACT_ZERO
FACT SUBS R1,R1,#1
     CMP R1,#0
     BEQ STOP
     MUL R0,R1,R0
     BNE FACT
FACT_ZERO MOV R0,#1
STOP SWI &11
     END
```

Question2:

```
     AREA program,CODE,READONLY;
     LDR R0,LIST
     AND R1,R0,0x0F
     MOV R1,R1,LSL #12

     AND R2,R0,0xF00

     AND R3,R0,0xF0000
     MOV R3,R3,LSR #12

     AND R4,R0,0xF000000
     MOV R4,R4,LSR #24
```

```
    ADD R5,R1,R2
    ADD R5,R5,R3
    ADD R5,R5,R4

    LDR R10,RESULT
    STR R5,[R10]

LOOP B LOOP

LIST dcb 0x14
     dcb 0x2D
     dcb 0x1F
     dcb 0x4B

RESULT dcd 0x40000000

    END
```

Question3:

```
     AREA oddEven,CODE,READONLY;
START
     MOV R0,#10
     TST R0,#1
     MOV R1,#0; Stores value '0' in R1 if number is even
     BEQ STOP
     MOV R1,#1; Stores value '1' in R1 if number is odd
STOP
     SWI &11
     END
```

Inferences:

- Learnt the architecture of ARM processor
- Learnt different instruction sets of ARM
- Learnt how to use ARM instructions to perform computations
- Learnt how to do assembly program in ARM for computing factorial of a number , checking weather number is odd or even and combining bits of different words to form a word.
- Understood how to work with KEIL software.
- Understood what is happening inside registers while debugging.
- Could learn the importance of CSPR register.
- Learnt how to use branch instructions in ARM