**1.** Assume r0 = 0xBBBBBBBB, r1 = 0xFFFFFFFF, r2 = 0xEEEEEEEE, r3 = 0xDDDDDDDD. **Write 4 programs** respectively using

1. STMIA (2) STMIB (3) STMDA (4) STMDB

to store r0, r1, r2, r3 into memory with addresses 0x40000020, 0x40000024, 0x40000028, 0x4000002C respectively. **Then**, use **a corresponding LDM instruction**

(1) LDMDA (2) LDMDB (3) LDMIA (4) LDMIB

to load the values in addresses 0x40000020, 0x40000024, 0x40000028, 0x4000002C respectively into r4, r5, r6, r7. (**Note:** **be sure not to use !** in STM and LDM instructions.)

2. Assume r0 = 0xDEADBEEF, r1 = 0xFACEABCD. **Write a program** that includes **5 subroutines** **func1**, **func2**, **func3**, **func4**, **func5** and calls to the 5 subroutines one by one.

(1) **func1** inserts the value 0x**FF** into r0 so that the final value of r0 becomes

0xDE**FF**BEEF,

(2) **func2** loads bits 12~19 from r0 into r1 so that the final value of r1 becomes 0xFACEAB**FB**,

1. **func3** puts the number of different bits between r0 and r1 into r3 **using TST**.
2. **func4** puts the number of different bits between r0 and r1 into r4 **without using TST**.
3. **func5** puts the different bit numbers between r0 and r1 one by one in the words started from memory address 0x40000050.

(Do not forget **STM** and **LDM** in the subroutine to avoid the side effect, be sure to use **STMDA** with **SP!** and the corresponding **LDM**, and highlight **the memory locations with the related registers stored**. **Assume SP = 0x40000020 initially.**)

3. **Assume** the word at memory address 0x40000000 = 0xCDEF5678, the word at memory address 0x40000004 = 0x5678CDEF, the word at memory address 0x40000008 = 0xBEEFFACE, the word at memory address 0x4000000C = 0xFACEBEEF initially.

(1)Write **a subroutine CheckParity** **to check parity**. **Subroutine CheckParity** gets the input word in R10, check the parity of the word in R10 and write the value 0xFACEFACE/0xBEEFBEEF if odd/even into R11.

(2) **Call subroutine CheckParity to check what parity (odd or even)** the 4 words at addresses 0x40000000, 0x40000004, 0x40000008 and 0x4000000C have and write the value 0xFACEFACE/0xBEEFBEEF if odd/even into the word at addresses 0x40000030, 0x40000034, 0x40000038 and 0x4000003C respectively.

(Do not forget **STM** and **LDM** in the subroutine to avoid the side effect, be sure to use **STMDB** with **SP!** and the corresponding **LDM**, and highlight **the memory locations with the related registers stored**. **Assume SP = 0x40000020 initially.**)

**4. Assume** the word at memory address 0x40000000 = 0x8765ABCD, the word at memory address 0x40000004 = 0xCDEF6543, the word at memory address 0x40000008 = 0xFACEBEEF, the word at memory address 0x4000000C = 0x87654321. **Write a program** that includes **2 subroutines sub1** and **sub2**, and calls to the 2 subroutines one by one.

(1) **sub1** puts **even** parity for bits 3, 4, 5, 9 and 11 of the word at memory address 0x40000000 into bit 8 of the word at memory address 0x40000008 (**Note:** **Be sure to use** **TST** **and do not use EOR.)**

(2) **sub2** puts **odd** parity for bits 6, 9, 13. 16 and 19 of the word at memory address

0x40000000 into bit 20 of the word at memory address 0x40000008. **(Note: Do not use TST and EOR.)**

(Do not forget **STM** and **LDM** in the subroutine to avoid the side effect, be sure to use **STMIB** with **SP!** and the corresponding **LDM**, and highlight **the memory locations with the related registers stored**. **Assume SP = 0x40000020 initially.**)

**Note:** Please

1. put necessary **Keil Tool DEBUG window screenshots** to show your **program** and **execution results** including **highlighted necessary initial assumptions and subsequent memory and register changes**,
2. **comment student ID+your English name in every screenshots**, and
3. put reports into one word file named by student\_ID+your\_name.