**1.** Assume r0 = 0x12345678, r1 = 0x87654321, r2 = 0xBEEFFACE, r3 = 0xDEADBEEF. **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.** Rewrite the UART Program in Sec. 16.2.5by using **full descending stack** for subroutine **UARTconfig** and **empty descending stack** for subroutine **Transmit**

(1) (a) to configure the UART 7 data bits, odd parity, 1 stop bits, a Baud rate if the UART is to generate a serial signal at a Baud rate of 9600 Baud using 48 MHz, and show the results in the window of UART0 after execution.

(b) calculate the system clock frequency from the window of UART0 in (a).

(c) to configure the UART 8 data bits, even parity, 2 stop bits, a Baud rate if the UART is to generate a serial signal at a Baud rate of 3600 Baud **using Keil Tool LPC 2104 CPU frequency** and show the above results in the window of UART0 after execution.

(2) to include the declaration of the string **“This is a source string!”** as variable **Source**. Use calls to subroutine **Transmit** to do the following 3 steps (Hint: show the results by **F5 (Run)**)

1. display the string **twice** in the window of **UART #1** after program execution.
2. display **reversely the string**, **continuously the string**, and **reversely the string** in the window of **UART #1** after program execution.
3. display **reversely** **the string words** in the window of **UART #1** after program execution.

**3.** Continue the program in Problem 2 to declare the string “Midterm in the spring 2023 class (ID-Name)!” in the program as variable **StudentData**.

(1) to include subroutines **Receive** (using **full ascending stack**) to receive an **error-free** byte data from the receiver buffer register to R0.

(2) (a) to copy the string (variable **StudentData**) **reversely** to memory starting from address 0x400000**60**.

(b) to use calls to subroutine **Transmit** to display a sequence of **10** characters at memory address 0x400000**66** in the **UART #1** window after program execution by using **F5 (Run)**.

(c) to use calls to subroutine **Receive** to receive a sequence of **error-free** **20** characters from the UART0 and put them in memory starting from address 0x400000**40**. (Show **execution results** by using **F10 (Step over) and F11 (Step)**.)

4. Rewrite Program 15-1 to include the following 2 declarations and

**Error DCB “divide-by-0 happened!”, 0**

**Message DCB “divide-by-0 not happened!”, 0**

1. check the usage fault status register, **write string Error to memory with starting address 0x20000020** if a divide-by-zero **has taken place**, and **write string Message to memory with starting address 0x20000080** if a divide-by-zero **has not taken place**.

(Be sure to show the related **memory change** by **setting** or **not setting** **DIV\_0\_TRP**, the **3 steps** in the **entry** sequence upon processor exception, the **2 steps** in the **exit** sequence, and give the **type of the stack** used here.)

(2) to switch modes and show the mode changes

(a)from **privileged thread mode** to **unprivileged thread mode**

(b)from **privileged thread mode** to **privileged handler mode**

(c) from **privileged handler mode** to **privileged thread mode**

(d) from **privileged handler mode** to **unprivileged thread mode**

**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, register and stack 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.