**1.** Rewrite the UART Program in Sec. 16.2.5by using **full descending stack** for subroutine **UARTconfig** and **empty descending stack** for subroutine **Transmit** (both with initial stack pointer 0x40000020, to STM and LDM in the subroutine)

**(1)** (a) to configure the UART 5 data bits, even parity, 2 stop bits, a Baud rate if the UART is to generate a serial signal at a Baud rate of 12800 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, odd parity, 1 stop bits, a Baud rate if the UART is to generate a serial signal at a Baud rate of 12800 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 “**(ID-Name)-Midterm Exam in Spring 2024!**” as variable **StudentData**. Use calls to subroutine **Transmit** to do the following 3 steps

1. display **reversely the string** and **continuously the string**
2. display **reversely the string characters in multiple-of-3 positions** (assuming the last character starts from position 0)
3. display **reversely** **the string “words”**

in the **UART #1** window after program executions (a) by using **F5 (Run),** **F10 (Step over)** and **F11 (Step)** respectively (b&c) by using **F5 (Run)** only.

**(3)** to include subroutines **Receive** (using **empty ascending stack** with initial stack pointer 0x40000020, to STM and LDM in the subroutine) to receive an **error-free** byte data from the receiver buffer register to R1.

(a) to copy the string (variable **StudentData**) **reversely** to memory starting from address 0x400000**70**.

(b) to use calls to subroutine **Transmit** to display a sequence of **10** characters at memory address 0x400000**78** 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**A0**. (Show **execution results** by using **F10 (Step over) and F11 (Step)**.)

(Be sure to show and **explain** in Problem (2)&(3) **the Line Status results in the UART0 window and the output in the UART #1 window after execution.** Also,be sure to **highlight** **the stack elements with the related registers stored for subroutines Receive and Transmit**.)

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

**MSG\_with\_Error DCB “DIVIDE-BY-0 Happened!”, 0**

**MSG\_without\_Error DCB “DIVIDE-BY-0 Not Happened!”, 0**

**(1)** check the usage fault status register, **write string MSG\_with\_Error to memory with starting address 0x20000030** if a divide-by-zero **has taken place**, and **write string MSG\_without\_Error to memory with starting address 0x20000090** if a divide-by-zero **has not taken place**.

(Be sure to show the **first step (stacking)** in the **entry** sequence upon processor exception 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.