**1. (1)** Assume the values in memory addresses 0x40000060, 0x40000064, 0x40000068, 0x4000006C are respectively 0xFACEBEEF, 0xBEEFFACE, 0x87654321, 0x12345678. **Write a program** in the following 8 steps

(a) Use LDMDA (b) Use LDMDB (c) Use LDMIA (d) Use LDMIB

to load the values in addresses 0x40000060, 0x40000064, 0x40000068, 0x4000006C into r4, r5, r6, r7 respectively.

(e) Use STMDA (f) Use STMDB (g) Use STMIA (h) Use STMIB

to store r4, r5, r6, r7 into memory with addresses 0x40000028, 0x4000002C, 0x40000030, 0x40000034 respectively.

**(2)** Assume r0 = 0xBEEFFACE. **Write a program** that includes subroutine **Compute** and a call to the subroutine. Subroutine **Compute** includes the 3 steps below to

1. insert the value 0x**DD** into r0 so that the final value of r0 becomes 0xBE**DD**FACE,
2. load bits 27~20 from r0 into r1 so that the final value of r1 becomes 0x000000**ED**, and
3. change bits 8, 9, and 15 of r0.

(Be sure to use an **empty descending stack** to STM and LDM in the subroutineto avoid the side effect and highlight **the stack elements with the related registers stored**. **Assume SP = 0x40000020 initially.**)

**2.** Write a programthat includes subroutine **UARTConfig** (using an **empty descending stack**, with initial stack pointer 0x40000020, to STM and LDM in the subroutine)and a call to the subroutine.

**(1)** The UART is configured as follows

(a) Pins P0.0 and P0.1 become the transmit and receive pins, Tx0 and Rx0, respectively

(b) 8 data bits, even parity, 2 stop bits

(c) a Baud rate if the UART is to generate a serial signal at a Baud rate of 9600 Baud using 32 MHz and calculate **Keil Tool LPC 2104 CPU frequency**

(d) a Baud rate if the UART is to generate a serial signal at a Baud rate of 2400 Baud using **Keil Tool LPC 2104 CPU frequency**

(Be sure to show the results in **the window of UART0** after execution.)

**(2)** include another two subroutines **Receive** and **Transmit** (both using **full ascending stacks**, with initial stack pointer 0x40000020, to STM and LDM in the subroutine)

(a) Subroutine **Receive** receives an **error-free** byte data from the receiver buffer register to R4.

(b) Subroutine **Transmit** transmits a byte data from R10 to the transmitter holding register.

**3.** Continue the program in Problem 2 to declare the string “(ID-Name)-Midterm in the spring 2023 class!” in the program as variable **StudentData**. Use calls to subroutine **Transmt** or **Receive** to do the following **4 steps**

**(1)** display **reversely the string** and continuously display the beginning part of the string “(ID-Name)”

in the **UART #1** window after program execution by using

(a) **F5 (Run)** (b) **F10 (Step over)** (c) **F11 (Step)**

(Be sure to show and **explain** **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)** to copy the string (variable **StudentData**) to memory starting from address 0x400000**30**.

**(3)** to display a sequence of **18** characters at memory address 0x400000**3C** in the **UART #1** window after program execution by using **F5 (Run)**.

**(4)** to use calls to subroutine **Receive** to receive a sequence of **error-free** **15** characters from the UART0 and put them in memory starting from address 0x400000**90**. (Show **execution results** by using **F10 (Step over) and F11 (Step)**.)

**4.** Use exception handling program in **EXAMPLE 15-1**

**(1)** to check the usage fault status register, load r10 0xFACEBEEF if a divide-by-zero **has not** taken place, and load r11 0xBEEFFACE if a divide-by-zero **has** taken place. Be sure to show **the execution results of the 3 steps** in the entry sequence upon processor exception, the final values of **r10 and r11**, and **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 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.