Laboratory 3: Stack Operations and Subroutine

Experiment Sheet

Purpose
The purpose of this laboratory is to learn how to implement stack operations and subroutines in ARM assembly language programming. This will involve pushing and popping values from the stack and using the `ADDS` instruction to observe how the carry flag works during arithmetic operations.

Essential Knowledge

1. **Stack Operations**
   - The stack is a memory structure that follows the Last In, First Out (LIFO) principle. It is commonly used for temporary storage of data and return addresses in subroutine calls.
   - The `PUSH` instruction is used to place values onto the stack, while the `POP` instruction retrieves values from the stack.
2. **Subroutines**
   - Subroutines allow for modular programming by encapsulating a set of operations that can be reused.
   - The `BL` (Branch with Link) instruction is used to call a subroutine, and `BX LR` is used to return from the subroutine.
3. **ADDS Instruction**
   - The `ADDS` instruction performs addition and updates the condition flags, including the carry flag, based on the result.
   - It is essential to understand how to handle overflow when using this instruction.
4. **Maximum Signed Integer**
   - The maximum positive signed 32-bit integer value is `0x7FFFFFFF`. Adding 1 to this value will cause an overflow.