Lab 4: Simple Four-function Calculator

Introduction

A four-function calculator performs the basic arithmetic operations: add (+), subtract (−), multiply (×), and divide (÷). The objective of this laboratory exercise is to design and implement a modular program that performs these operations on two unsigned 8-bit numbers, using subroutines.

Program Design & Implementation

Since we are working with unsigned 8-bit numbers, we must think about the range of the output for the arithmetic operations. Consider, for example, multiplying the two 8-bit hexadecimal numbers $1C and $0C; their product is $150 (decimal 336). Since this is larger than 255 (decimal), the product will not fit into a single 8-bit byte. Hence, we need to use a 16-bit, or "double byte", result for all of our operations. Remember that we have three 16-bit registers available to us in the 68HC11: accumulator D (ACCD), and the X and Y index registers (IX and IY). These registers can be used in the subroutines to hold a 16-bit result.

Your calculator program will use the two 8-bit accumulators to pass the two arithmetic operands to the subroutines. If *A* and *B* are the two operands, your calculator must compute the following:

addition:	$A + B$
subtraction:	$A - B$
multiplication:	$A \times B$
division:	$A \div B$

The operand *A* will be passed to the subroutines in accumulator A (ACCA); operand *B* will be passed in accumulator B (ACCB).

One other important point deserves mention: although the 68HC11 has a powerful instruction set, it does not have explicit instructions to carry out the four operations we need for our calculator. For example, there is no instruction to subtract an 8-bit number from a 16-bit register. We need to design an algorithm to do this. One possible algorithm is:

```
GET_DIFF:           ; Subroutine to compute A-B
     LDX #$0000     ; clear the IX register
     ABX            ; add B to IX
     STX Temp       ; store B in a 16-bit variable, Temp
     TAB            ; copy A into ACCB
     LDX #$0000     ; clear the IX register
     ABX            ; add A to IX
     XGDX           ; swap ACCD and IX, now ACCD has A
     SUBD Temp      ; subtract B from A
     RTS            ; return with A-B in ACCD
```

Note how the comments in the above program describe what each line does. Not only is this helpful for debugging the program, but it also documents the code. Be sure to carfeully comment all of your programs.

The Four-function Calculator

Write a program to add, subtract, multiply, and divide two unsigned 8-bit numbers. The two numbers must be passed to the subroutines using ACCA and ACCB. The 16-bit result must be returned to the main program using ACCD. After completing each arithmetic operation, your main program must store the result in a 16-bit memory location. When the main program ends, you can examine these four 16-bit memory locations to verify the result of each operation.

The addition operation will need a special algorithm, similar to the subtraction algorithm above. For the division operation, you should use the `IDIV` instruction (not the `FDIV` instruction). Be sure return the quotient, and not the remainder, in accumulator D.

Test your program with the following inputs:

1. A = $1C B = $0C
2. A = $FF B = $FF
3. A = $2B B = $2D

Demonstrate your working program to the teaching assistants using inputs of their choice. In your report, list the inputs and output of each arithmetic operation.