EE 337: Microprocessors Laboratory (Spring 2022)
Indian Institute of Technology Bombay

Assignment 2: 20 points Date: January 18, 2022

1. [5 points] Write an assembly language program to add two 16-bit numbers **x** and **y**. Use the following program as a starting point. Add your code in the `ADD16` subroutine.

```
// -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL ADD16
HERE: SJMP HERE
ORG 130H
// ****************

ADD16:

// ADD YOUR CODE HERE

RET
END
```

   - The number **x** is stored at locations `70H` and `71H`, with its most significant byte (MSB) in `70H` and the least significant byte in `71H`.
   - The number **y** is stored at locations `72H` (MSB) and `73H` (LSB).
   - Since the result $\mathbf{z} = \mathbf{x} + \mathbf{y}$ can be 17 bits long, store the result in memory locations `74H`, `75H`, `76H`.
   - For $\mathbf{z} = z_{16}z_{15}z_{14}\ldots z_3z_2z_1z_0$ where z_0 is the least significant bit and z_{16} is the most significant bit, the memory location `74H` should have $0000000z_{16}$, the memory location `75H` should have the bits $z_{15}z_{14}\ldots z_8$, and the memory location `76H` should have the bits $z_7z_6\ldots z_0$.

2. [15 points] In a fully connected neural network, forward propagation involves the *Multiply and Accumulate (MAC)* operation. The neuron input is multiplied with the weight attributed to the path between that neuron and the neuron to the next layer. Then, all such products coming from different source neurons to the same destination neuron are added.

   Let us consider the neuron inputs to be $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$. Let the corresponding weights be $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$. The result after the MAC operation will be

$$\mathbf{x} = \mathbf{a}_1\mathbf{b}_1 + \mathbf{a}_2\mathbf{b}_2 + \mathbf{a}_3\mathbf{b}_3.$$

   Write an assembly language program to generate the result **x**, when $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are three 8-bit neuron inputs present in the memory locations `70H` to `72H` and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are 8-bit weights present in the memory locations `73H` to `75H`.

   Since the result **x** can be 18 bits long, store the result in memory locations `50H`, `51H`, `52H`. For $\mathbf{x} = x_{17}x_{16}x_{15}\ldots x_3x_2x_1x_0$, the memory location `50H` should have $000000x_{17}x_{16}$, the memory location `51H` should have the bits $x_{15}x_{14}\ldots x_8$, and the memory location `52H` should have the bits $x_7x_6\ldots x_0$.

Use the following program as a starting point. Use the `ADD16` subroutine written in the previous question to implement the MAC operation.

```
 // -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL MAC
HERE: SJMP HERE
ORG 130H
// *****************

MAC:

// MAC OPERATION, CALL THE ADDITION SUBROUTINE USING "CALL ADD16"

RET


ADD16:

//16 BIT ADDITION CODE

RET
END
```

TA Checkpoints

1. For question 1, check the following two cases:
   - $\mathbf{x} = \texttt{1234H}, \mathbf{y} = \texttt{DCBAH}$.
   - $\mathbf{x} = \texttt{FFFFH}, \mathbf{y} = \texttt{FFFFH}$.
2. For question 2, check the following three cases:
   - $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 = \texttt{18H}, \texttt{24H}, \texttt{0CH},\ \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 = \texttt{03H}, \texttt{02H}, \texttt{06H}$.
   - $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 = \texttt{FFH}, \texttt{FFH}, \texttt{01H},\ \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 = \texttt{FFH}, \texttt{FFH}, \texttt{FFH}$.
   - $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 = \texttt{FFH}, \texttt{FFH}, \texttt{FFH},\ \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3 = \texttt{FFH}, \texttt{FFH}, \texttt{FFH}$.
3. For question 2, as the student to explain the code in the `MAC` subroutine.