**PRE-LAB:**

Write the first draft of the program and determine the expected results (in memory or register) after each instruction by hand. **This must be completed before coming to the lab.**

**PROGRAMMING ASSIGNMENT:**

Write **ONE assembly program** to perform the following tasks. **Include brief comments explaining the code.** Your code should do the following: -

**In the data section**

1. Define an array of bytes that has four elements called Array1\_bytes and initialize it with 2, 3, 0xFE, and 0xFF.
2. Define an array of words that has four elements called Array1\_Signed\_Words and initialize it with zeros.
3. Define an array of words that has four elements called Array1\_unsigned\_Words and initialize it with zeros.
4. Define an array of words that has four elements called Array1\_Add\_Words and initialize it with zeros.
5. Define an array of words that has four elements called Array1\_Mul\_Words and initialize it with zeros.
6. Define an array of words that has four elements called Array1\_Div\_Words and initialize it with zeros.

**In the Code region**

Write a code to do the following:-

1. Extend every element in Array1\_bytes to a word using signed extension and store the result in Array1\_Signed\_Words.
2. Extend every element in Array1\_bytes to a word using unsigned extension and store the result in Array1\_unsigned\_Words.
3. Add every corresponding elements in the arrays Array1\_unsigned\_Words and Array1\_Signed\_Words and store the result in Array1\_Add\_Words.
4. Multiply every element in Array1\_unsigned\_Words by 10 and store the result in Array1\_Mul\_Words.
5. Divide every element in Array1\_unsigned\_Words by 2 and store the result in Array1\_Div\_Words using unsigned division.

You can use the following sequence

1. Initialize the registers R0, R1, R2, R3, R4, and R5 to point at the beginning addresses of the arrays Array1\_bytes, Array1\_Signed\_Words, Array1\_unsigned\_Words, Array1\_Add\_Words, Array1\_Mul\_Unsigned\_Words, and Array1\_Div\_Unsigned\_Words, respectively.
2. Extend the byte pointed by R0 using unsigned extension and store the result in R6.
3. Extend the byte pointed by R0 using signed extension and store the result in R7.
4. Store R6 in the word pointed by R2.
5. Store R7 in the word pointed by R1.
6. Add R6 and R7 and store the result in the word pointed by R3.
7. Multiply R6 by 10 and store the result in the word pointed by R4.
8. Divide R6 by 2 using unsigned division and store the result in the word pointed by R5.
9. Repeat the steps 2-8 three more times – each time you work on one element in the arrays – DO NOT FORGET TO UPDATE THE POINTERS.

Run the code and record the results in the following table – verify the results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Array | Element 0 | Element 1 | Element 2 | Element 3 |
| Array1\_bytes | 02 | 03 | FE | FF |
| Array1\_Signed\_Words | 00000002 | 00000003 | 000000FE | 000000FF |
| Array1\_unsigned\_Words | 00000004 | 00000006 | 000000FC | 000000FE |
| Array1\_Add\_Words | 00000014 | 0000001E | 000009EC | 000009F6 |
| Array1\_Mul\_Words | 00000001 | 00000001 | 0000007F | 0000007F |
| Array1\_Div\_Words | 00000000 | 00000000 | 00000000 | 00000000 |

**Things to turn in the lab report**

A. Your Code. **[25 marks]**

B. Table 1 showing the results of your program. **[10 marks]**