

CPE 325: Embedded Systems Laboratory

Laboratory Assignment #5

Assignment

[100 pts]

1. Write an assembly program that multiplies the corresponding elements of two arrays and stores the results in a third array. The arrays need to have at least 6 elements of varying numbers. Your program should basically implement the loop below:

```
for (k = 0; k < array_size; k++)
```

```
    array_3[k] = array_1[k] * array_2[k];
```

This program is responsible for initializing both input arrays, calling the two subroutines – SW_product, HW_product, and allocating the space to store the results calculated by the subroutines. This program is also responsible for passing the required input parameters and address of the memory location reserved to store the result to the subroutines using the program stack.

2. SW_product uses Shift-and-Add multiplication algorithm and HW_product uses Hardware Multiplier to multiply the elements.
3. Determine the number of clock cycles required to evaluate the equation by each subroutine. How many elements can be evaluated using SW_product and HW_product in a sec? Explain which one is more efficient and why?
4. **Bonus (Each technique 8 pts – Total 16pts)** Write an assembly program that passes a base number (other than 0 or 1) to a subroutine which populates an array with the first 5 powers of that base number. For instance, if you pass the number 6 to the subroutine, the result array should be populated with 6^1 , 6^2 , 6^3 , 6^4 and 6^5 . Create subroutines to calculate the results using software and hardware. Use Shift-and-add multiplication algorithm from part 1 and for the hardware use hardware multiplier. The parameters and results should be passed between subroutines and main program using either registers or the program stack.

Deliverables

1. Lab report with screenshots of final outputs and the answers to the questions in part 3
2. Source files (.asm files)

Notes:

1. Do not hard code numbers or arrays in the sub routines. You have to pass the required parameters by using the specified method.
2. Do not allocate space to store the result in subroutines. It should be done using **.data** section and passed to the subroutines the way described above.
3. Try different inputs before you conclude which method is efficient. In your explanation, include the inputs as well.
4. Assume that any of the results does not exceed 16-bits.