COA Lab [CS39001]
Assignment-4

July 31, 2024

You have to write MIPS programs for the following tasks:

1. Write a MIPS program that takes 3 integers as input and performs modular exponentiation of the numbers. Say the numbers are M, N and d(M, N, d are all 32 bit positive integers). So the result would be M^d modulo N

   (a) First write a function **DecimalToBinary** with appropriate parameters, which takes an integer d as input and convert it into corresponding binary representation $(d_0||d_1||d_2||\cdots||d_i||\cdots||d_{n-1})$, where d_0 is the most significant bit and d_{n-1} is the least significant bit of d. Store the binary representation in separate locations in memory. From the **main** function call **DecimalToBinary**. Print the binary equivalent of d from the **main** function.

   (b) Write two functions that can respectively perform squaring and multiplications. Name these functions as **Square** and **Multiply**. The **Square** function takes one input, returns its squared value, and the **Multiply** function takes two values and returns their multiplication value.

   (c) Write a function **ModExp** that implements the following exponentiation algorithm:

   The square and multiply algorithm performs a squaring at each step, while the multiplication operation is performed only for the exponent bits that are set to one. Algorithm 1 has the details. The function returns $S \leftarrow M^d \mod N$ where the exponent,

   d is $(d_0||d_1||d_2||\cdots||d_i||\cdots||d_{n-1})$ in binary, where d_0 is the most significant bit and d_{n-1} is the least significant bit of d.

   ”mod” stands for modulo operation, often denoted by the symbol ”%”. When we say ”a mod b”, we’re referring to the remainder when integer ”a” is divided by integer ”b”. [For example, 13 mod 5 equals 3.]

Algorithm 1 Square and multiply exponentiation algorithm

```
1: procedure SQUARE AND MULTIPLY(M, d, N)
2:     S ← M^d mod N
3:     S ← M
4:     for ( do i from 1 to n − 1)
5:         S ← S ∗ S mod N
6:         if d_i = 1 then
7:             S ← S ∗ M mod N
8:         return S
9:         =0
```

Sample Output: Here is one sample output:

Enter M, d, N: 3 3 1024
The exponent in binary is 11.
The exponentiation value of $3^3 \mod 1024$ is 27.

Another sample output:

Enter M, d, N: 11 7 13
The exponent in binary is 111.
The exponentiation value of $11^7 \mod 13$ is 2.