Project Report
Assignment 2 - MIPS

Course Name: EG 212, Computer Architecture - Processor Design
Student Names: Arismita Mukherjee, Gautam Prasanna Kappagal
Student Roll Numbers: IMT2023585, IMT2023082

24 February, 2024

Description of Programs

1. **Find rank in an array**

   This program first inputs an integer n. Then we input the address where we want to start storing n integers in an array followed by the address where we want to start storing the output array. After this, we input n integers and store them at the required address. We then input the address where we want to store the rank and then input the number whose rank we want to find.

   The program uses the bubble sort algorithm to sort the integers. It then finds the rank of the number that was inputted and stores it in the rank address that was inputted.

2. **LCM GCD**

   This program inputs an address to store an integer followed by the integer itself. This is done twice as we input two numbers. It then prompts the two addresses where the GCD and LCM for both numbers will be stored.

   The program uses a recursive algorithm to find the GCD of both numbers and divides the product of both numbers by this GCD to find the LCM. It then stores these two values in their respective addresses.

3. **Binomial**

   This program inputs the address to store the exponent n of the expression we want to obtain followed by the integer n itself. It then prompts the user to enter the address from where they want to start storing the expansion coefficients in the form of an array.

   The program uses concept combinatorics to calculate the coefficients of each term in the expansion. It first uses loops to calculate the factorials of n, j, and $(n - j)$ for each j in the range of 0 to n, both inclusive. It then divides the first by the product of the latter two to obtain the coefficient of the j-th term in the expansion. Finally, it stores this value in its respective address in the array. This is repeated for all coefficients.

   The program iterates over the output array and prints the values in the form of an expanded expression of $(1 + x)^n$.

Processor

We implemented the processor using Python, coding instructions like `sw`, `lw`, `branch`, `jump`, `move`, and various other ALU instructions.

Assumptions:

- User inputs are within the range of available memory addresses.
- While using the `div` instruction, we use an empty temporary register (`$t3`) in place of `lo` to store the quotient.

- Our original MIPS codes include input and output statements that run correctly on MARS. However, to run it on the processor, we took separate inputs in the processor's main function in Python. Hence, we have removed the input and output lines from the machine code of each program for it to run on the processor albeit it is the same on MARS.

Output

1. **Find rank in an array**

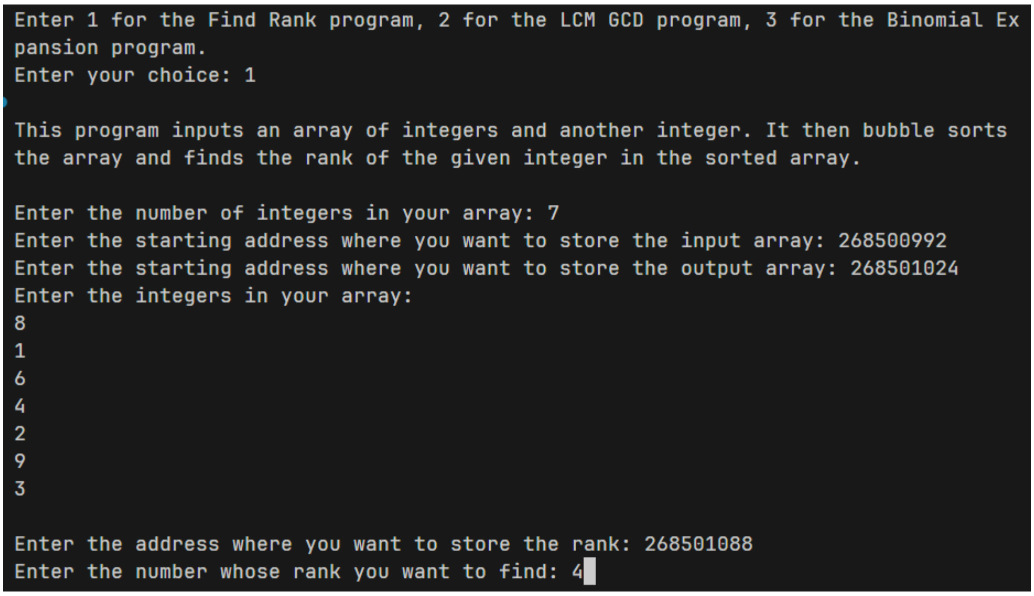

Figure 1: Input example for finding rank in an array

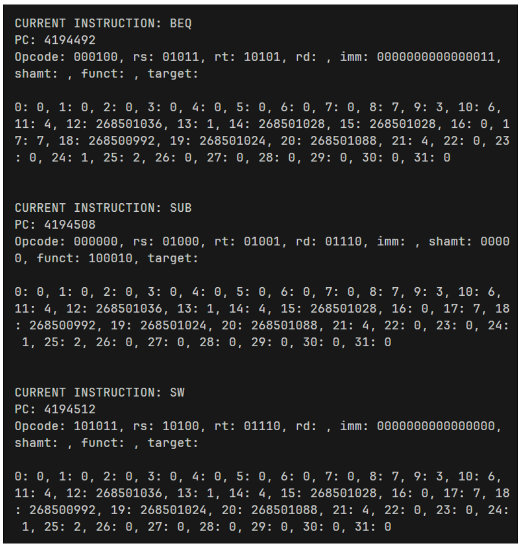

Figure 2: Some intermediate field and register example values for finding rank in an array

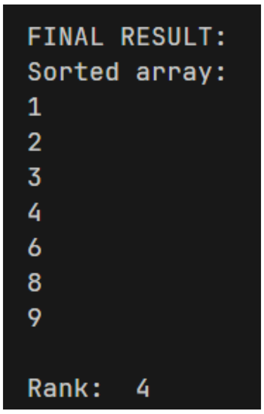

Figure 3: Final output example of finding rank in an array

2. **Calculate the GCD and LCM of two integers**

```
Enter 1 for the Find Rank program, 2 for the LCM GCD program, 3 for the Binomial
 Expansion program.
Enter your choice: 2

This program inputs two integers and finds their GCD and LCM.

Enter the address where you want to store the first number: 268500992
Enter the first number: 12
Enter the address where you want to store the second number: 268500996
Enter the second number: 18
Enter the address where you want to store the GCD: 268501000
Enter the address where you want to store the LCM: 268501004
```

Figure 4: Input example for finding the GCD and LCM of two integers

```
CURRENT INSTRUCTION: DIV
PC: 4194416
Opcode: 000000, rs: 01000, rt: 01010, rd: 01011, imm: , shamt: 0000
0, funct: 011010, target:

0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 216, 9: 0, 10: 6
, 11: 36, 12: 6, 13: 6, 14: 0, 15: 12, 16: 268500992, 17: 268500996
, 18: 268501000, 19: 268501004, 20: 12, 21: 18, 22: 0, 23: 0, 24: 1
8, 25: 12, 26: 0, 27: 0, 28: 0, 29: 0, 30: 0, 31: 0


CURRENT INSTRUCTION: MFLO
PC: 4194420
Opcode: 000000, rs: 00000, rt: 01011, rd: 01001, imm: , shamt: 0000
0, funct: 010010, target:

0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 216, 9: 36, 10:
6, 11: 36, 12: 6, 13: 6, 14: 0, 15: 12, 16: 268500992, 17: 26850099
6, 18: 268501000, 19: 268501004, 20: 12, 21: 18, 22: 0, 23: 0, 24:
18, 25: 12, 26: 0, 27: 0, 28: 0, 29: 0, 30: 0, 31: 0


CURRENT INSTRUCTION: SW
PC: 4194424
Opcode: 101011, rs: 10011, rt: 01001, rd: , imm: 0000000000000000,
shamt: , funct: , target:

0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 216, 9: 36, 10:
6, 11: 36, 12: 6, 13: 6, 14: 0, 15: 12, 16: 268500992, 17: 26850099
6, 18: 268501000, 19: 268501004, 20: 12, 21: 18, 22: 0, 23: 0, 24:
18, 25: 12, 26: 0, 27: 0, 28: 0, 29: 0, 30: 0, 31: 0
```

Figure 5: Some intermediate field and register example values for finding the GCD and LCM of two integers

```
FINAL RESULT:
GCD of 12 and 18: 6
LCM of 12 and 18: 36
```

Figure 6: Final output example of finding the GCD and LCM of two integers

3. **Find the binomial expansion of $(1 + x)^n$ for any integer n**

```
Enter 1 for the Find Rank program, 2 for the LCM GCD program, 3 for the Binomial Expansion
program.
Enter your choice: 3

This program inputs an integer n and finds the binomial expansion of (1 + x)^n.

Enter the address where you want to store n: 268500992
Enter n: 8
Enter the starting address where you want to store the coefficients: 268500996
```

Figure 7: Input example for the binomial expansion of $(1 + x)^n$

```
CURRENT INSTRUCTION: ADDI
PC: 4194508
Opcode: 001000, rs: 01110, rt: 01110, rd: , imm: 0000000000000001, shamt: , funct: , t
arget:

0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 1, 9: 1, 10: 0, 11: 1, 12: 1, 13: 1
, 14: 9, 15: 9, 16: 268500992, 17: 8, 18: 1, 19: 0, 20: 268501032, 21: 0, 22: 40320, 2
3: 40320, 24: 40320, 25: 0, 26: 0, 27: 0, 28: 0, 29: 0, 30: 0, 31: 0


CURRENT INSTRUCTION: JUMP
PC: 4194512
Opcode: 000010, rs: , rt: , rd: , imm: , shamt: , funct: , target: 0000000001000000000
0000001010100

0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 1, 9: 1, 10: 0, 11: 1, 12: 1, 13: 1
, 14: 9, 15: 9, 16: 268500992, 17: 8, 18: 1, 19: 0, 20: 268501032, 21: 0, 22: 40320, 2
3: 40320, 24: 40320, 25: 0, 26: 0, 27: 0, 28: 0, 29: 0, 30: 0, 31: 0


CURRENT INSTRUCTION: BEQ
PC: 4194388
Opcode: 000100, rs: 01110, rt: 01111, rd: , imm: 0000000000011111, shamt: , funct: , t
arget:

0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 1, 9: 1, 10: 0, 11: 1, 12: 1, 13: 1
, 14: 9, 15: 9, 16: 268500992, 17: 8, 18: 1, 19: 0, 20: 268501032, 21: 0, 22: 40320, 2
3: 40320, 24: 40320, 25: 0, 26: 0, 27: 0, 28: 0, 29: 0, 30: 0, 31: 0
```

Figure 8: Some intermediate field and register example values for the binomial expansion of $(1 + x)^n$

```
FINAL RESULT:

Binomial expansion of (1 + x)^8: 1x^0 + 8x^1 +
28x^2 + 56x^3 + 70x^4 + 56x^5 + 28x^6 + 8x^7 +
1x^8
```

Figure 9: Final output example of the binomial expansion of $(1 + x)^n$