COMPUTER ARCHITECTURE ASSIGNMENT – 2

Program Description: In this project, we have implemented three programs: **Fibonacci sequence, Finding Mean of 5 Numbers, and Finding the Power of a Base raised to an Exponent**. We have designed it in the MIPS ISA format.

We have designed an assembly code and a processor to run the same. The assembly code was run in the MARS simulator to generate the machine code.

Instruction Fetch:

- PC fetches the instructions from the instruction memory.
- The instruction memory contains the instructions in the form of the machine code generated by the MARS simulator.

Instruction Decode:

- In this phase, the instruction fetched gets decoded based on whether it is R format, I format or J format.
- Once this is done, it then identifies the command to be carried out based on the opcode and funct value (if it is R format).
- The instruction gets broken down into its respective components(rs,rt,etc) based on the format.

Execute Phase:

- In this phase, the ALU component receives the values from the registers after the decoding, and it carries out the operations based on the control signals it receives.

Memory Access:

The registers (in the data memory) which need to be written into are accessed in this stage.

Memory Writeback:

The values from the ALU get written into the respective registers in this phase.

C codes for the 3 programs:

1. **Fibonacci**

```c
#include <stdio.h>

int fibonacci(int n) {
    if (n <= 1)
        return n;
    return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
    int n;
    printf("Enter the value of n: ");
    scanf("%d", &n);

    printf("Fibonacci(%d) = %d\n", n, fibonacci(n));

    return 0;
}

```

2. Power

```c
#include <stdio.h>

// Function to calculate power
long long int power(int base, int exponent) {
    long long int result = 1;
   long long int i;
    // Calculate power using repeated multiplication
    for (i = 0; i < exponent; i++) {
        result *= base;
    }

    return result;
}

int main() {
    long long int base, exponent;

    // Input base and exponent
    printf("Enter the base: ");
    scanf("%lld", &base);
    printf("Enter the exponent: ");
    scanf("%lld", &exponent);

    // Calculate and print the result
    printf("%lld raised to the power %lld is: %lld\n", base, exponent, power(base, exponent));

    return 0;
}
```

3. Mean

```c
#include <stdio.h>

int main() {
    int num1, num2, num3, num4, num5, mean;

    // Input five numbers
    printf("Enter five numbers: ");
    scanf("%d %d %d %d %d", &num1, &num2, &num3, &num4, &num5);

    // Calculate the mean
    mean = (num1 + num2 + num3 + num4 + num5) / 5;

    // Print the mean
    printf("Mean of the five numbers: %d\n", mean);

    return 0;
}
```

Screenshots of the output

```
For code 1 - Finding power:
Clock cycles taken =  30
Base: 2
Exponent: 10
Reg file: [0, 0, 0, 0, 0, 0, 0, 0, 2, 1024, 10, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Result: 1024

For code 2 - Fibonacci Series:
Clock cycles taken =  50
Input: 10
Reg file: [0, 0, 0, 0, 0, 0, 0, 0, 10, 55, 89, 10, 89, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Result: 55

For code 3 - Finding mean:
Clock cycles taken =  11
Input: dict_values([1000, 2400, 2600, 5, 2000, 2000])
Reg file: [0, 0, 0, 0, 0, 0, 0, 0, 2000, 2400, 2600, 2000, 5, 2000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Result: 2000
```


Contributors:

Pravin Kumar V – IMT2023595

Vinay P Ramesh – IMT2023541

Rakshith Srinivasan- IMT2023544