Assignment No: 03

Department of Computer Science

Iqra University Islamabad

Computer Organization and Assembly Language

Maqsood Ahmed

ID: 38186

Problem # 1: [CLO2]

Compare and contrast the performance of floating-point calculations performed by an FPU versus software emulation on a CPU without a dedicated FPU.

Answer:

FPU vs. Software Emulation Performance

FPU (Floating-Point Unit):

- **Speed**: Much faster for floating-point calculations because it's specifically designed for these operations.
- **Efficiency**: Uses specialized hardware that executes floating-point instructions in fewer cycles.

Software Emulation:

- **Speed**: Slower because the CPU has to simulate floating-point operations using basic integer instructions.
- **Efficiency**: Consumes more CPU cycles and can lead to slower overall performance.

Problem # 2: [CLO2]

How do different data formats used by FPUs (e.g., single-precision vs. double-precision) affect the accuracy and performance of calculations.

Answer:

Data Formats (Single-Precision vs. Double-Precision)

Single-Precision:

- **Accuracy**: Lower precision with approximately 7 decimal digits of accuracy.
- **Performance**: Faster because it uses less memory (32 bits) and bandwidth.

Double-Precision:

- **Accuracy**: Higher precision with approximately 15-16 decimal digits of accuracy.
- **Performance**: Slower compared to single-precision due to larger size (64 bits) and increased computational load.

Problem # 3: [CLO2]

Explain the concept of floating-point normalization and its role in FPU operations

Answer:

Floating-Point Normalization

- **Normalization**: Adjusts the floating-point number so that the leading digit is non-zero, maximizing precision.

- **Role in FPU Operations**: Ensures consistent representation of numbers, improves accuracy, and simplifies comparison and arithmetic operations.

Problem # 4: [CLO2]

Describe pipelining techniques used in FPUs to improve performance.

Answer:

Pipelining in FPUs

- **Instruction Pipelining**: Divides the processing of instructions into several stages (fetch, decode, execute) allowing multiple instructions to be processed simultaneously.

- **Parallel Execution**: Allows different floating-point operations to overlap, improving throughput and reducing latency.

Problem # 5: [CLO2]

How can assembly language programmers interact with the FPU to perform specific floating-point operations? (This could involve specific instructions or function calls).

Answer:

Assembly Language Interaction with FPU

- **Specific Instructions**: Use of floating-point instructions like `FADD` (floating-point addition), `FMUL` (floating-point multiplication), etc.

- **Function Calls**: Calling library functions that perform complex floating-point operations, utilizing the FPU.

The End