**Assignment #01**

**REPORT TITLE:** RISC-V (RV32I) SINGLE CYCLE

**Name:** Muhammad Hussain

**Roll no**: 21B-111-CE

**Section:** EL-B

**Course:** Computer architecture and organization (CE-201)

**INSTRUCTOR:** DR ALI AHMED

REPORT TITLE: RISC-V (RV32I) SINGLE CYCLE

❖TABLE OF CONTENTS:

Introduction---------------------------------------------------------------------------1

Methodology -------------------------------------------------------------------------2

Fetch -------------------------------------------------------------------------3

Decode ----------------------------------------------------------------------4

Execute ---------------------------------------------------------------------5

Memory --------------------------------------------------------------------6

Write back -----------------------------------------------------------------7

Circuit Diagram --------------------------------------------------------------------8

Result -------------------------------------------------------------------------------9

Conclusion --------------------------------------------------------------------------10

Title: Understanding the Working of a 32-bit Processor: A Comprehensive Report

**Introduction**

A 32-bit processor is a central processing unit (CPU) that operates on data and instructions encoded in 32-bit units. This report aims to provide a detailed explanation of the working of a 32-bit processor, focusing on the stages involved in executing instructions and the techniques employed to achieve its functionality.

**METHODOLOGY:**

The methodology that is going to discuss next is based on following a sequence of stages that facilitate the design process. As we know, every processor has 5 stages. These stages are: 1) Fetch 2) Decode; 3) Execute; 4) Memory; 5) Write Back.

**Instruction Fetch Stage**

The first stage of a 32-bit processor's pipeline is the instruction fetch stage. Here, the CPU fetches the next instruction from memory using the program counter (PC) register. The PC holds the memory address of the next instruction to be fetched. The CPU then increments the PC, ensuring that subsequent instructions are fetched sequentially.

**Instruction Decode Stage**

In the instruction decode stage, the fetched instruction is decoded to determine the operation to be performed and the operands involved. The instruction decoder identifies the opcode (operation code) and other fields within the instruction, such as registers or memory addresses. This information is then passed to the subsequent stages for further processing.

**Execution Stage**

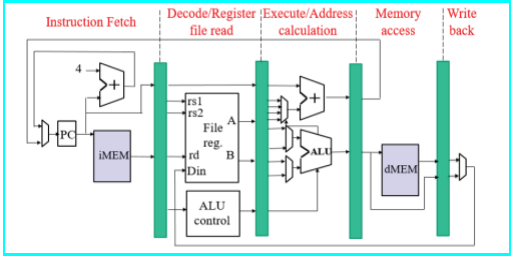
The execution stage performs the actual computation or operation specified by the instruction. This stage can vary depending on the specific operation being performed. Arithmetic and logical operations are typically executed in this stage. The operands fetched earlier are used as inputs, and the result is stored in a register.

**Memory Access Stage**

If the instruction involves accessing memory, such as loading or storing data, the memory access stage is activated. The processor calculates the memory address based on the operands and accesses the memory subsystem to read from or write to the specified location. The data retrieved from memory is often stored in registers for further processing.

**Write Back Stage**

In the final stage of the pipeline, the write back stage, the results of the computation or the data retrieved from memory are written back to the appropriate destination. This could be a register or a memory location, depending on the specific instruction. The updated value is stored for future use or made available to subsequent instructions.



**Achieving Functionality**

To achieve the functionality of a 32-bit processor, various techniques are employed:

a. Instruction pipelining: The processor divides the instruction execution into multiple stages, allowing multiple instructions to be executed concurrently. This improves overall throughput and performance.

b. Register file: A register file consists of a set of registers that store data within the processor. These registers provide fast access to operands, enabling efficient data manipulation and computation.

c. Arithmetic and logic units (ALUs): The ALU performs arithmetic and logical operations, such as addition, subtraction, AND, OR, etc. These units are crucial for executing instructions that involve computation.

d. Control unit: The control unit coordinates the flow of instructions and data within the processor. It controls the sequencing of stages, ensures proper synchronization, and manages the overall execution of instructions.

e. Memory management: A 32-bit processor typically employs memory management techniques, such as virtual memory and memory caches, to efficiently access and manage large amounts of data.

f. Instruction set architecture (ISA): The ISA defines the set of instructions and their encoding formats supported by the processor. It provides the interface between software and hardware, enabling software developers to write programs that can run on the processor.

**Conclusion**

In conclusion, a 32-bit processor executes instructions by breaking them down into multiple stages, including instruction fetch, decode, operand fetch, execution, memory access

