CSE 306
Computer Architecture Sessional

Assignment 04: 8 bit MIPS Pipelined Execution
Report of Group 01 Section B1

Prepared by:
1705061 - Maksudur Rahaman Rana
1705064 - Tanvir Raihan
1705081 - Md. Kamrujjaman
1705083 - Hozifa Rahman Hamim
1705090 - Ashrafi Zannat Ankon
1305050 - Zaki Tahmeed

Introduction

In this assignment, we need to design an 8-bit MIPS with pipelined datapath.Pipelining is an implementation technique in which multiple instructions are overlapped in execution for better throughput. In our design, each instruction is divided into five stages:

- **Instruction fetch (IF)**
- **Instruction decode (ID)**
- **Execution and address calculation (EX)**
- **data memory access (MEM)**
- **Write back (WB)**

The length of the clock cycle equals the maximum time to execute any single stage. Therefore, each instruction takes up to **five** clock cycles to be executed.The main components of the processor are as follows:

- **Instruction Memory**
- **Register File**
- **ALU**
- **Control Unit**
- **Data Memory**
- **Four Pipeline Registers** and
- **Forwarding Unit**.

Additional components such as comparators, adders, mux etc can be added as required for our design.The main objective of this assignment is to get acquainted with the concept of pipelining and implementing it for designing MIPS Pipelined Datapath. Also, learning the concept of hazard and resolving it using forwarding technique is also a learning outcome of this assignment.

Complete Block Diagram of the Pipelined Datapath

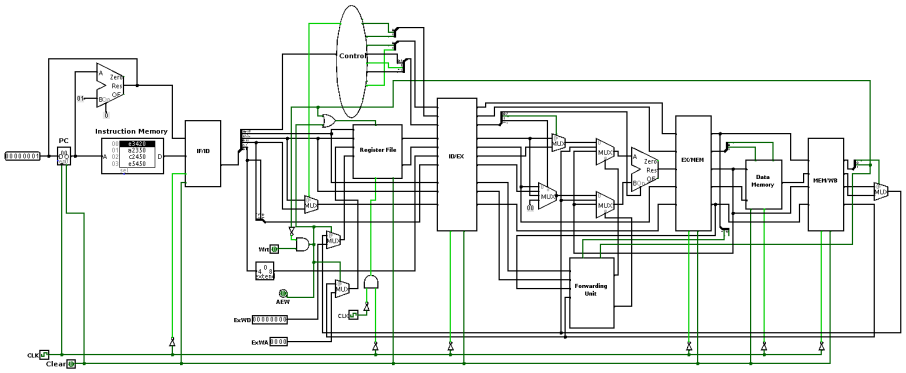

Figure 1: Block Diagram of the 8-bit Pipelined Datapath

Block diagrams and size of pipeline registers

In our design, we have used **four** pipelined registers namely

- **IF/ID**
- **ID/EX**
- **EX**
- **MEM/WB**

IF/ID Pipeline Register

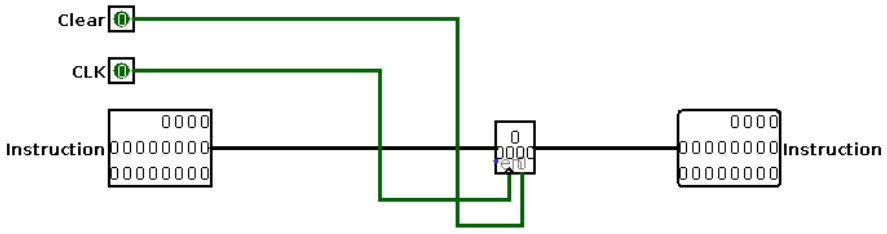

Figure 2: IF/ID Pipeline Register

- This register connects the Instruction fetch(IF) and Instruction Decode(ID) stages .
- . Size of this register is **20 bits**.

ID/EX Pipeline Register

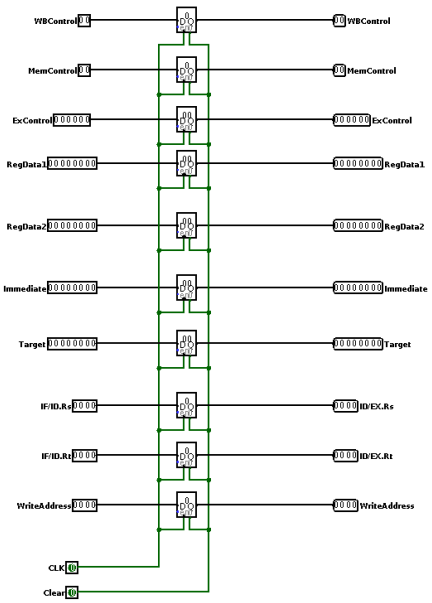

Figure 3: ID/EX Pipeline Register

- This register connects the Instruction Decode(ID) and Execution and Address Calculation (EX) stages.
- Size of this register is **54 bits**.

EX/MEM Pipeline Register

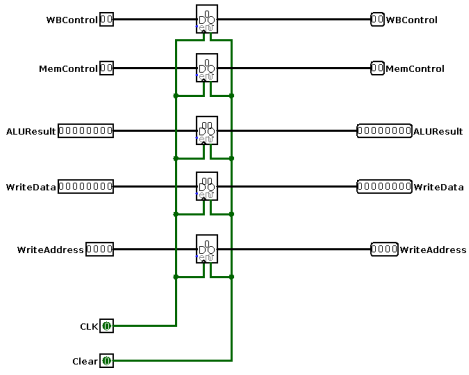

Figure 4: EX/MEM Pipeline Register

- This register connects the Execution and Address Calculation(EX) and Data Memory Access(MEM) stages.
- Size of this register is **24 bits**.

MEM/WB Pipeline Register

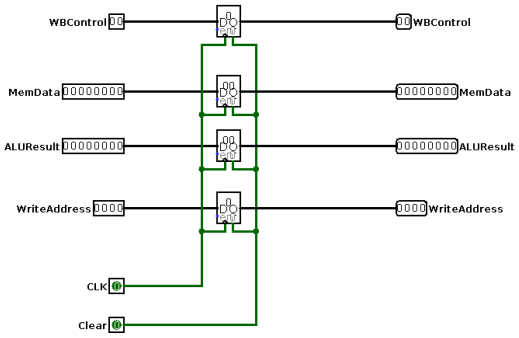

Figure 5: MEM/WB Pipeline Register

- This register connects the Data Memory Access(MEM) and Write Back(WB) stages.
- Size of this register is **22 bits**.

Mechanism and block diagram of forwarding unit

Mechanism

The forwarding unit is used to detect hazards and resolve them using the Forwarding technique.For our implementation, we have considered three types of hazards: **EX Hazard**,**MEM Hazard** and **Double Data Hazard**.

- In case of EX hazard, the forwarding unit forwards the calculated value from the ALU of EX/MEM stage to ID/Ex stage by generating proper forwarding signal.This signal controls a mux which selects between the register data and forwarded data.
- The same technique is used for MEM hazard except that data is forwarded from the MEM/WB stage in this case.
- In case of double data hazard, the most recent value is forwarded which is the data from the EX/MEM stage to ensure data consistency.

Block Diagram

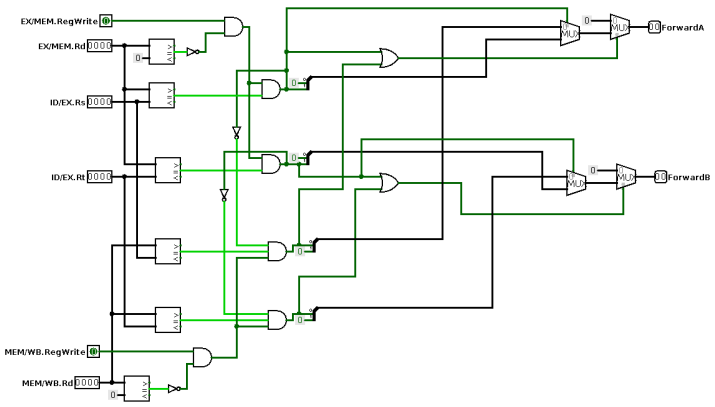

Figure 6: Forwarding Unit

Simulator Used along with version number

Name : Logisim **Version** : 2.7.1

Discussion

The main objective of this assignment is to design an 8-bit MIPS pipelined datapath using four pipelined registers between five stages. The purpose of using the pipelined registers was to increase throughput by executing multiple sub-instructions in a clock cycle. Using this datapath,we could evaluate multiple instructions at the same time and we came to experience the improvement.We have understood the concept of Pipelining and Data Hazards after completing this assignment. We have also understood the forwarding technique which is used for resolving data hazard.