

**ESE 345 - Computer Architecture**

**Prof. Mikhail Dorojevets**

***Pipelined Multimedia Unit***

**By**

**Wilmer Suarez**

**&**

**Himanshu Goel**

**Expected Results**

A full project report including the goals, multimedia unit block diagram, design procedure, all testbenches, conclusions, **the VHDL/Verilog source code** of the multimedia unit, and simulations results **in printed form** must be presented **at the start** of your 20-minute demonstration during a time slot assigned to your team by TA.

The **electronic version of the report** must be also sent to the TA & Instructor before the start of the presentation.

**Project presentation will not start without these printed and electronic documents submitted.**

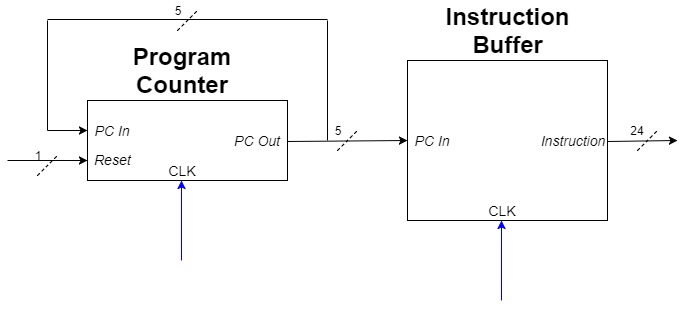
**Goals**

The primary goal of this project was to demonstrate an understanding of the material covered in this class through implementation of a 3-stage pipelined multimedia processing unit for Single-Instruction Multiple-Data operations.

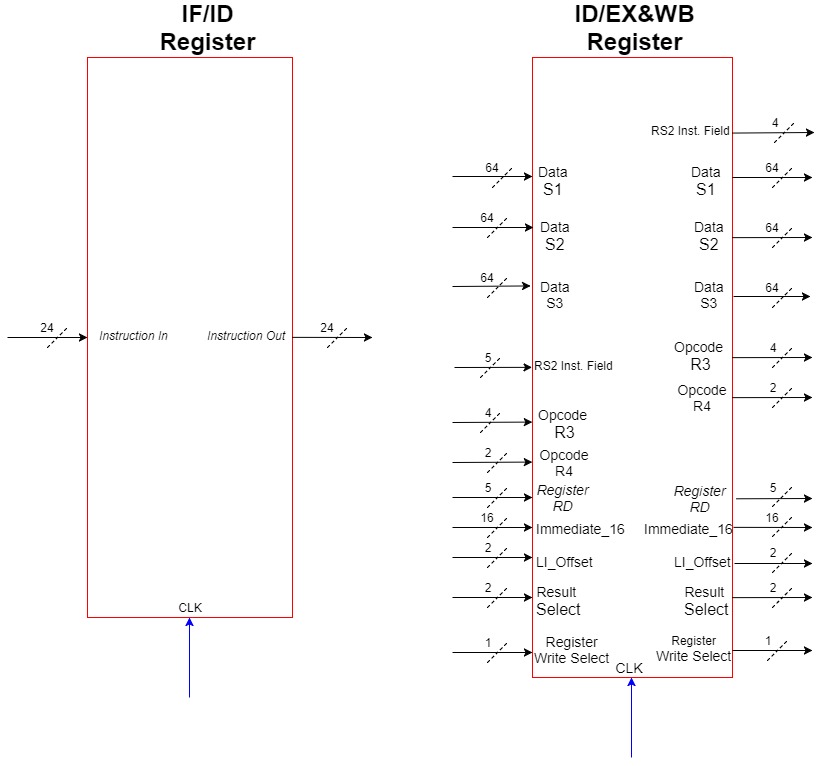
**Multimedia Unit Block Diagram**

**Design Process**

**Program Counter and Instruction Buffer**

­­­­

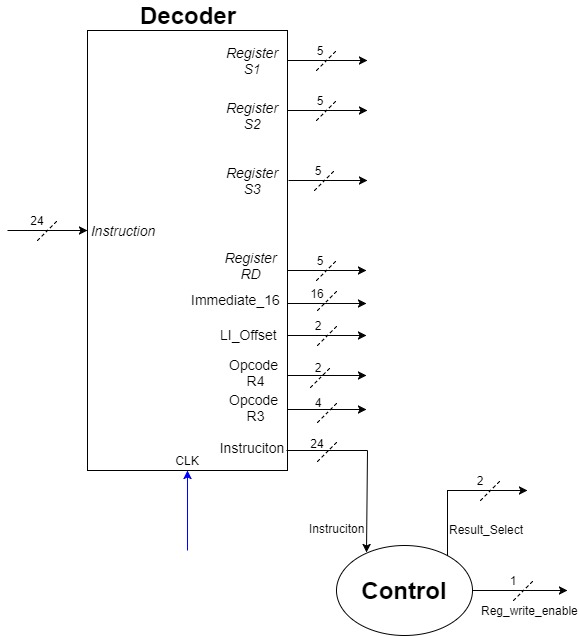
**Pipeline Registers**



The pipeline registers, IF/ID and ID/ED&WB, are used to separate the 3 stages of the datapath. The registers are clock edge-sensitive (positive edge) to allow one instruction to be processed at every stage, and one instruction to be completed every cycle (after the initial 3 cycles).

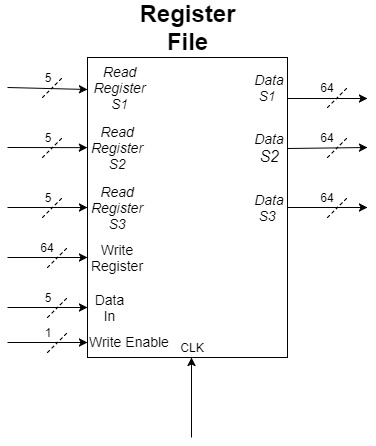
The first register receives instructions from the instruction buffer in the *Instruction Fetch (IF) Stage* and outputs an instruction after every positive edge of the clock. The instruction output is sent to the Decoder in the *Instruction Decode (ID) Stage*.

**Instruction Decoder and Control Unit**

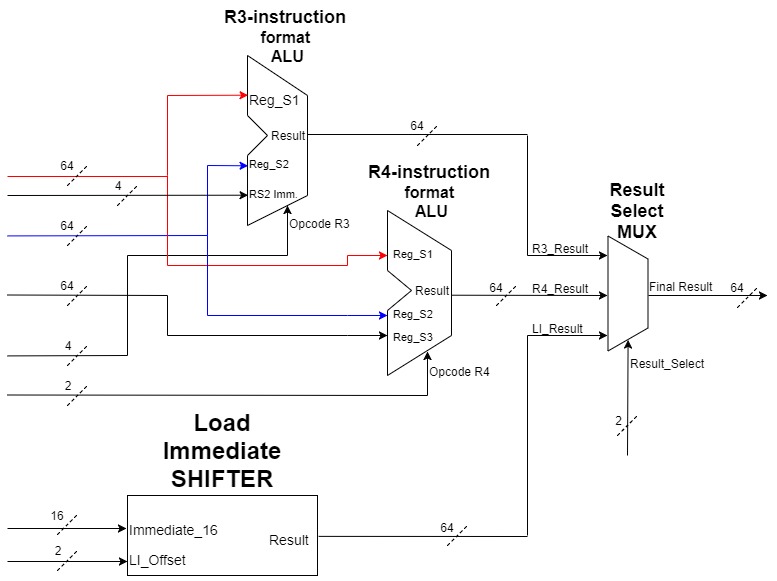


The Decoder (*in the Instruction Decode (ID) Stage*)

**Register File**



**ALU 1, 2 and Load Immediate Shifter**



**Multimedia Unit**

**Test Conditions**

**Instruction Buffer**

**Register File**

**Program Counter**

**Pipeline Registers**

**ALU 1**

**ALU 2**

**Multiplexer**

**Load Immediate Shifter**

**Control Unit**

**Instruction Decoder**

**Multimedia Unit**

**Conclusions and Discussion**

**VHDL Code**

**Instruction Buffer**

**Register File**

**Program Counter**

**Pipeline Registers**

**ALU 1**

**ALU 2**

**Multiplexer**

**Load Immediate Shifter**

**Control Unit**

**Instruction Decoder**

**Multimedia Unit**

**VHDL Testbenches**

**Instruction Buffer**

**Register File**

**Program Counter**

**Pipeline Registers**

**ALU 1**

**ALU 2**

**Multiplexer**

**Load Immediate Shifter**

**Control Unit**

**Instruction Decoder**

**Multimedia Unit**

**Simulation Results**

**Instruction Buffer**

**Register File**

**Program Counter**

**Pipeline Registers**

**ALU 1**

**ALU 2**

**Multiplexer**

**Load Immediate Shifter**

**Control Unit**

**Instruction Decoder**

**Multimedia Unit**

**RTL Design/Schematic**

**Multimedia Unit Test Programs**

**2x2 Matrix Multiply**