

**ESE 345 - Computer Architecture**

**Prof. Mikhail Dorojevets**

***Pipelined Multimedia Unit***

**By**

**Wilmer Suarez**

**&**

**Himanshu Goel**

**Procedure:**

**I.**                   It is suggested to**read** the Intel MMX and Sony Cell SPU[[1]](http://www.ece.sunysb.edu/~midor/ESE345/project.html" \l "_ftn1" \o ") papers below and understand the original concept of multimedia processing introduced as MMX architecture for Intel processors in the 1990s.

**II.**                **Refresh your knowledge** of VHDL/Verilog in the HDL design of digital circuits based on the Web-based tutorial and the examples in the companion CD-ROM for the text book. �

**III.**             **Develop**a detailed block diagram and the HDL model of the three-stage

multimedia unit and its modules.

**IV.**                        **Verify** individual modules of your design with their testbenches before instantiating them in higher order modules. Verify the final model with a testbench module and generate file **Results** showing the status of each stage of the unit during execution.

**Requirements:**

1. **Multimedia ALU**

The ALU must be implemented as **behavioral models in VHDL or continuous assignment (dataflow)** **models in Verilog.**

1. **Register file**

The register file has 32 64-bit registers. On any cycle, there can be 3 reads and 1 write. Each cycle two/three 64-bit register values are read, and one 64-bit value can be written if a write signal is valid. This ***register write*** signal must be explicitly declared so it can be checked during simulation and demonstration of your design. A technique of **data forwarding** is to be used so that a write and read to the same register will return the new value for the read.

The register module must be implemented as **a behavioral model in VHDL (a (dataflow/RTL model in Verilog).**

1. **Instruction buffer**

The instruction buffer can store 32 24-bit instructions. The contents of the buffer should be loaded by the testbench instructions from a test file at the start of simulations. �Each cycle one instruction specified by the Program Counter (PC) is fetched, and the value of PC is incremented by 1.

The instruction buffer module must be implemented as **a behavioral model in VHDL (a (dataflow/RTL model in Verilog).**

1. **Three-stage pipelined multimedia unit**

**Clock edge-sensitive** pipeline registers separate the IF, ID, and EXE stages.

The EXE stage of the pipeline is responsible for calculating the result and writing it to the register file.

All instructions (including **li**) take three cycles to complete. This pipeline must be implemented as **a behavioral model in VHDL (a (dataflow/RTL model in Verilog).**Three instructions can be at different stages of the pipeline at every cycle.

1. **Testbench**

This module supplies an instruction code to be loaded from a file to the instruction buffer and, when it is finished, **checks the contents of the register file**.

It must be implemented as a **behavioral model.**

It is up to each team to choose how the assembly test code for the unit is converted to binary and saved in a file from which is to be loaded into the instruction buffer at the start of simulation.

1. **Results**

This file must show status of the pipeline with the opcodes, input operands, and results of execution of instructions in the pipeline for each cycle.

**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.**