# CS 401 NAME(s): Due May 1

## FINAL GROUP PROJECT: CPU DESIGN PART III: CONTROL UNIT DESIGN

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY** | **Beginning**  **70% – 79%** | **Satisfactory**  **80% – 89%** | **Excellent**  **90% – 100%** |
| 70 pts possible  **CONTROL UNIT DESIGN** | Rudimentary decoder tables (single cycle) or rudimentary finite state machine design (multi-cycle) for 5-7 non mips instructions or 11-12 MIPS instructions. | Complete set of decoder tables (single cycle) or complete finite state machine design for the decoder (multi-cycle) for 8-9 non-MIPS or 13-14 MIPS instructions. | Neat, well commented, complete set of decoder tables (single cycle) or neat, well commented complete finite state machine design (multi-cycle) for 10 or more non-MIPS instructions or 15 or more MIPS instructions. |
| 20 pts possible  **CONTROL UNIT VHDL** | VHDL code for the control unit | Partially working VHDL code for the control unit. | Working VHDL code for the control unit. |
| 10 pts possible  **CONTROL UNIT TEST** | Simulation testbench created but not documented well or does not work properly | At least one working test program (either ad-hoc or well-know algorithm) runs correctly in simulation. | Both test programs (the ad-hoc and the well-known algorithm) runs correctly in simulation. |

3.1 Control Unit Design

In this project your group will design the control unit. The control unit interprets your machine code and provides the control signals that move data through the data path unit to/from memory and registers.

Before Designing the Control Unit

* You must know whether your control unit is a single-cycle or multi-cycle design. Which type of design you pick will determine how the Finite State Machine which forms the heart of the control unit operates.
* What type main memory will your CPU require? Are you interfacing to the SDRAM or are you using an on-chip FPGA memory like the single-cycle mips CPU? If you are using the SDRAM, you will need a multi-cycle CPU design.

3.2 Control Unit Design Process  
Review your Assembly, ALU, and DPU design.

1. Make a list of all the op-codes for each of the instructions supported.
2. Make a list of all control signals required by the DPU and the ALU. Keep the control signals for the ALU separate from the control signals needed by the components in the DPU.
3. Design the Finite State Machine controller for your CPU.
4. If you are doing a single-cycle CPU, you will need to include copies of all the decoder tables for your CPU (e.g. see pages 383-387 of your textbook)
5. If you are doing a multi-cycle CPU draw out the entire FSM for your design. (e.g. see page 408 in your textbook for an example of the finite state machine for the multi-cycle mips processor). NOTE: You may find it helpful to place the FSM into tabular form (e.g. excel spreadsheet), remember, every FSM can be drawn as a table or as a graph diagram.)
6. Create a test-bench to test your entire CPU. Verify that both the ad-hoc program and the well-known algorithm you designed for FP1 run correctly.

### What to Hand In:

Control Unit Design

* Multi-Cycle or Single-Cycle CPU design.
* Neatly drawn decoder tables for the single-cycle design, neatly drawn FSM diagram for the multi-cycle design.
* VHDL code for your completed design in each of your group member’s folder on CS1
* Simulation testbench trace tables showing your processor running correctly for both of your test programs.