

**ELEC 5200 - FINAL PART OF PROJECT**  
**Final Project Due: Monday, May 1 (last class day)**

1. Design and test a VHDL “behavioral” model of the **control unit** to realize the behavior described in your Control Signal Table from the previous project.  
(Refer to the *Control Unit* VHDL example reviewed in class.)
  - I suggest a *Moore model* format, separating the state transitions and output equations, although you may have one or more control signals that are functions of both state and one or more inputs, such as an ALU “Zero” flag, etc.
  - Each control signal should be driven by a single signal-assignment statement.
  - In some special cases you may need to synchronize a control signal with the CPU clock.
  - ***Thoroughly simulate the control unit before attaching it to the data path.***
2. A sample VHDL **Memory** model (mips\_memory\_bin.vhd) is available on the class web page for you to adapt to your model. This model will automatically load a test program, at the start of simulation, from object-code file “testprog”. This object-code file is simply a list of 32-bit binary numbers, one per line, to be loaded into contiguous memory locations beginning at address 0.
3. Create a “**System component**” by instantiating and connecting your control unit, data path, and memory components. A clock input should be the only external port of this "System". If you wish, you may build a “clock generator” into the system and omit the clock port.  
Example:     `clk <= not clk after 50ns; -- Clock with period 100ns`
4. A test program is supplied for your final simulation (the one to be submitted). There are 4 files:
  - mips\_test\_prog.asm   - the source file
  - mips\_test\_prog.lst   - the assembler listing file
  - testprog             - the object file in binary formatFeel free to create other test programs to exercise your model. Note that the memory assumes the existence of a file named “testprog”, which must contain 32-bit binary codes, listed one per line as in the provided test program.
5. For the **final simulation**, to minimize the size of the listing, display only **one line per clock transition** (i.e. trigger only on clock signal transitions). Show a sufficient set of signals to demonstrate correct operation of each instruction (control unit state, address bus, data bus, ALU output, register file outputs, register file input, memory control signals, etc.) On the simulation listing, annotate by writing the corresponding assembly language instruction next to each execute cycle and highlighting the “significant” result register or bus value.