

Phase 3

(Due: 11/9/2009)

Control Unit Design

In this phase of the project, you will finish the control unit design and put together your CPU core. All the control signals in your data path design will come from the control unit. Therefore, double check your data path design and make sure that all control signals are accounted for. It is also a good idea to envision how your main memory and I/O will interface with the CPU since there are some control signals there as well. Finally, some external signals to the CPU such as the main system clock and the main reset signal to the CPU are also handled in the control unit.

There are two main approaches possible here:

- **Hardwired:** The control unit is designed as a finite state machine (FSM) or algorithmic state machine (ASM). You should have a flow chart to indicate the operation sequence and then put timing reference to the micro-operations associated with each instruction (see below for details). This type of control unit is "hard" because any change in the design requires a complete makeover.
- **Micro-program:** In this approach, you will first design a FSM (the micro-sequencer) that will read and execute the "micro-instruction" in a ROM (the microprogram ROM). This type of design is "firm" since in most cases, only the microprogrammed ROM contents need to be change for any update or revision.

Suggested Procedure:

1. **Control signals:** Collect all control signals associated with the data path, main memory, I/O and external to CPU. Make sure each control signal is documented so you know exactly how they operated. Such information should include in the phase progress report and final report.
2. **Flowchart:** Draw a detailed flowchart that starts from the reset of the CPU. Typically, the first few steps are to fetch the instruction and then place the execution of the different instructions in parallel for the correct timing reference in the next step.
3. **Timing reference:** Assign the state numbers to the micro-operations, according to the flow chart. For instance, if the instruction fetch requires three cycles then it will use S1, S2 and S3. Then the first micro-operation of every instruction (macro-instruction) will be executed at S4, and so on.
4. **Micro-operations vs. Control signals:** Draw a table with control signals on the columns and all "unique" micro-operations on the rows. Checked the corresponding control signals related to the micro-operations to complete the translation from micro-operations to the control

signals. If some micro-operations are occurred at the same time, you may list them together in one entry. *It is your choice to do so or to ignore it.

5. FSM or Micro-program design: Use State diagram editor and/or VHDL editor to complete the design. I've found using the state diagram editor to complete the state transition and then use the VHDL to complete the control signal details is the most efficient approach.
6. Assembler: Create a configuration file of "caspr" for your CPU design. You will need this assembler when writing program for your CPU.
7. Verification: The more substantial verification of your CPU will occur in the next phase. Here, however, you should perform some quick check on your CPU design. Add memory and simple I/Os, e.g. LEDs and switches, to your CPU and test a simple program.

Progress Report:

The following is the "suggested" format for your phase-3 progress report:

1. Title: Title page includes course number and name, your CPU name as your project title, names and contact information (emails) of the team members, and date. Also, mark the lead person in this phase with a "*" associated with the name.
2. Flowchart: the overall operation flowchart of your CPU.
3. Control signal table: The table in which you translate micro-operations to control signals.
4. Micro-operations: Re-list micro-operations for the instruction fetch and all the instructions (macro-instructions). This time, each step should have a state number associated with it.
5. State diagram/VHDL design of the FSM or Micro-program unit.
 - If using micro-program unit, you should list all the "micro-instructions" and how they related to the micro-operations as in the table above.
6. Caspr Configuration file of your CPU.
7. Memory-map, I/O arrangement and program listing of your simple test.
8. Lesson Learned.