Computer Structure Lab - Fall 1999
Dr. Guy Even

Handout #4: A Read Machine and A Write Machine

In this handout you will design bus masters capable of initiating bus transactions.

1 A Read Machine

The Read Machine is an application that reads the contents of a fixed memory address (0x800000) and stores the value in a register. The Read Machine is connected as a bus master to the I/O Control Logic.

The state diagram of the Read Machine is depicted in Figure 1. The functionality of the Read Machine is as follows:

1. The machine exits the "wait" state when the STEP_EN signal is active.
2. The machine initiates a read transaction in the "fetch" state.
3. The machine waits for an `ack` signal during the "wait4ack" state.
4. The machine writes the fetched value in its register when entering the "load" state. Note that the machine keeps the address on the Address bus valid for half a cycle during the "load" state.
5. The `reset` signal causes the machine to transition to the "wait" state.

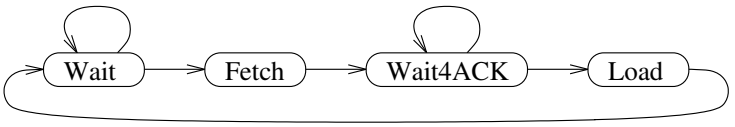

Figure 1: State diagram of the Read Machine

2 A Write Machine

The Write Machine is an application that writes your favorite value (i.e. 17) to a fixed memory address (0x800000). The Write Machine is connected as a bus master to the I/O Control Logic.

The state diagram of the Write Machine is depicted in Figure 2. The functionality of the Write Machine is as follows:

1. The machine exits the "wait" state when the STEP_EN signal is active.
2. The machine initiates a write transaction in the "store" state.

3. The machine waits for an `ack` signal during the "wait4ack" state.

4. Note that the machine keeps the address and the data (on the Address and Data busses) valid for half a cycle during the "terminate" state.

5. The `reset` signal causes the machine to transition to the "wait" state.

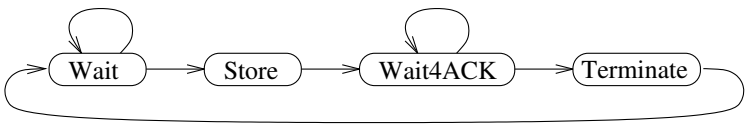

Figure 2: State diagram of the Write Machine

3 Pre-lab Assignment

(submit in the beginning of your lab meeting)

1. (20 pts) Design a Read Machine and a Write Machine.

2. (10 pts) Suppose that we wish to monitor the activity of a Read Machine using the Logic Analyzer module. We would like to start sampling when the `STEP_EN` signal rises, and end the sampling two cycles after the Read Machine returns to the "wait" state. Write the equations for the following signals:

   (a) Write enable (`WE`) signal of Logic Analyzer's RAM.

   (b) Count enable (`CE`) signal of the Logic Analyzer's counter.

   (c) The reset signal of the Logic Analyzer's counter.

   Remark: Make sure that, after an active `reset` signal generated by the I/O Control Logic, the Logic Analyzer is properly initialized. This means that the first sampling session that follows a reset functions correctly.

3. (5 pts) Same question for the Write Machine.

4. (5 pts) Is it possible to design a modified Read Machine that has the same functionality but does not have the "load" state? Can you also get rid of the "terminate" state in the Write Machine?

4 Post-lab Assignment

(submit one week after your lab meeting)
(60 pts) Prepare two designs:

1. A design with the I/O control logic, the Monitor Slave (with the Logic Analyzer), and the Read Machine.

2. A design with the I/O control logic, the Monitor Slave (with the Logic Analyzer), and the Write Machine.

Implement your designs on the RESA (submit schematics/ABEL). Write a configuration file for the PC monitor program (submit configuration file). Submit monitoring results of: (a) the state transitions; (b) the bus activity of the machines; and (c) the accessed memory address (In the Read Machine, this means that the register of the Read Machine is monitored as well as the main memory address. In the Write Machine, only the main memory address is monitored).