Verilog LCD Module Simulator – Design Requirements

ECE 4890: Senior Seminar; Spring 2010
7 May 2010
Robert Ewing
Steven Wilson
**Background**
The customer is the instructor for a university engineering course (“Rapid Prototyping with FPGAs” at the University of Colorado, Colorado Springs) for the rapid-prototyping of digital systems using an FPGA evaluation board. The board used for the course, the “Spartan-3E Starter Kit”, features a character-based LCD module with controller [1, 2]. To use the LCD, students of the course need to be able to design and debug LCD controller interfaces written in the Verilog HDL. Although designs can be tested on the board itself, the LCD controller provides virtually no feedback until after an initialization sequence is completed. Because the initialization sequence incorporates most of the complexity of the overall interface, it is likely that many design problems in the controller will preclude receiving any visible feedback from the LCD controller and thus frustrate debugging efforts.

The preferred method of verifying digital logic operation, in both this course and in industry, is to use a Verilog simulator to simulate the user design’s behavior in conjunction with simulation models to simulate the behavior of the other components in the system and automatically perform validity checks on input and output signals, reporting timing and protocol violations to the user. Such simulation models are frequently provided by vendors of digital components. However, the vendor of the LCD controller does not provide a suitable simulation model, so it is impossible for students to verify their LCD interface designs using the preferred automatic methods.

The proposed solution is to develop the desired simulation model as a Verilog module. The model shall simulate the LCD controller, validate the basic protocol and initialization sequence, and accept characters to be displayed via a GUI element. The simulation model shall report timing violations (both basic waveform violations as well as post-instruction delay violations) in order to indicate the nature of the error to the designer.

Since the goal is to validate the low-level behavior rather than to provide a 100% complete simulation, the crucial elements are the waveform sequences and timings for reads/writes and the instructions required for the initialization sequence. It is expected that the simulation will allow students to identify interface problems whose nature would not be obvious by observing the actual hardware. The interface between the LCD controller and the LCD matrix itself shall not be simulated, as this aspect of the controller's operation is primarily of concern when designing the LCD module itself. Since students are using the LCD module rather than designing it, this part of the controller's functionality shall be simulated at a higher level of abstraction in order to simplify the implementation.

**Input**
The input will be activity on the LCD interface generated by the user’s Verilog HDL design.
Output
The output will consist of the following:

- Verilog HDL signal activity on the simulated LCD interface
- Human-readable error messages for each type of error
- Verilog HDL counters for each type of error
- A Tk GUI element simulating the visible state of the LCD module

Requirements Specification
1: The simulation model shall be written in Verilog, because that is the language used to teach the course and thus the primary language for modules making use of the simulation model, as well as the language for which expertise will be readily available on campus for future maintenance.

2: The simulation model shall be compatible with ModelSim, because that is the primary simulator used at UCCS.

3: The simulation model shall check for timing/sequence violations on the microcontroller/LCD controller interface. This ensures that the basic read/write transactions are valid. A confirmation message shall be printed upon a successful bus transaction.

3a: The simulation model shall report an error, including information on the type of error that occurred and the simulation time at which it occurred, on any of the following timing violations:

1. Elapsed time of less than "T_C" (enable cycle time) between rising edges of E.
2. Elapsed time of less than "T_PWR" (enable pulse width for reads) between a rising edge of E and the following falling edge during a read operation.
3. Elapsed time of less than "T_PWW" (enable pulse width for writes) between a rising edge of E and the following falling edge during a write operation.
4. Elapsed time of less than T_AS (address setup time) between any edge on RS or RW and a rising edge on E.
5. Elapsed time of less than T_AH (address hold time) between a falling edge of E and any transition on RS or RW.
6. Elapsed time of less than T_DSW (data write setup time) between any transition on DB4-DB7 and a falling edge on E.
7. Elapsed time of less than T_H (data hold time) between a falling edge on E and any transition on DB4-DB7.
8. Elapsed time of less than $t_{PW}$ (enable pulse width) between a rising edge on $E$ and any transition on $RS$ or $RW$.

9. Elapsed time of less than $t_{PO}$ (power-on wait) between the start of simulation and the first rising edge of $E$.

3b: The delays associated with each timing constraint in requirement 3a shall be configurable by editing a configuration file. This is needed because there is no single authoritative source for LCD timings (as the Spartan-3E Starter Kit manual and the LCD controller datasheet provide conflicting information), as well as to accommodate discrepancies between the datasheet and the observed behavior of the actual hardware. Delays between 0 nanoseconds and 1200 nanoseconds (inclusive) shall be handled correctly.

4: The simulation model shall support the 4-bit interface mode of the LCD controller, because that is how the LCD controller is connected on the Spartan-3E Starter Kit board.

5: The simulation model shall simulate at least the instructions used in the initialization sequence, because a successful initialization will use all of those instructions.

5a: The simulation model shall simulate the following instructions: "Function Set", "Display ON/OFF Control", "Display Clear", "Entry Mode Set", and "Write data to RAM". A confirmation message shall be printed when an instruction is successfully issued.

5b: Behavior shall be simulated for at least the following configuration:

- I/D = 1 for the Entry Mode Set instruction (increment DD RAM address)
- S = 0 for the Entry Mode Set instruction (display shift disabled)
- D = 1 for the Display On/Off instruction (display enabled)
- C = 0 for Display On/Off (cursor not displayed)
- B = 0 for Display On/Off (cursor not blinking)
- DL = 0 for Entry Mode Set (4-bit interface)
- N = 1 for Entry Mode Set (2-line mode)

5c: The simulation model shall indicate an error when a new cycle is initiated before the delay of the previous instruction has elapsed.

5d: The post-instruction delays shall be configurable for each supported instruction by editing the configuration file of requirement 3b. Delays between 37 microseconds and 2 milliseconds (inclusive) shall operate correctly.

5e: The simulation model shall simulate the writing and display of ROM-based characters to the LCD (i.e. characters written to "DD RAM").
The simulation model shall report a confirmation when a correct instruction is sent in the initialization sequence, and shall report an error when an incorrect instruction is sent in the initialization sequence. The error message shall indicate both the incorrect instruction and the instruction that was expected at that point in the initialization sequence.

The simulation model shall be accompanied by a self-checking Verilog testbench, which shall test all of the above requirements without manual operation of the testing sequence.

The simulation model shall be accompanied by a GUI written in Tcl/Tk (the language and GUI toolkit integrated with ModelSim). The GUI shall reflect a simulated LCD screen consistent with the behavior supported by the simulation. It shall be possible to disable the GUI function – preferably automatically – such that the non-GUI portions of the simulation may run on versions of ModelSim, such as the student version, that do not allow use of Tk.

Acceptance Testing
The acceptance testing shall be performed by the automated testbench of requirement 5g, which shall test both valid and invalid sequences to verify that each type of error is correctly detected. The testbench shall verify that errors are appropriately generated for violations of each of the timing constraints of requirement 3a, for violations of the post-instruction delay of requirement 5c, and violations of the initialization sequence of requirement 5f. It shall also test valid sequences, verifying that the appropriate confirmation messages of requirements 3, 5a, and 5f are generated.

Additional Design Considerations
It is important that the simulation run quickly enough to support an expedient test/debug cycle. This is a qualitative judgment, but its importance will inform the design process. In general, this consideration is expected to lead to a relatively high level of abstraction in the simulation of the LCD controller’s internal state. A detailed simulation of the controller’s internal operation would require extensive reverse engineering, and is not necessary to satisfy the customer’s requirements.

Design Deliverables
- Verilog source file(s) implementing the LCD simulation module.
- Verilog source file(s) implementing the testbench.
- Tcl/Tk source file(s) implementing the GUI display and any necessary post-processing of simulation data.
- Design document to support future maintenance.
**References**
