COMP 525V: Modeling and Verifying Cache Coherence in SMV

due: Feb 25, 2001; 11:59pm (electronically, using assignment name *cache*.)

For this assignment, model and verify the cache coherence protocol described below. A cache coherence protocol maintains consistent views of a central memory across distributed processors. Such a protocol is considered correct if for each address in each cache, the cached value is the same as the central memory value. This assignment has the following goals:

- To give you experience modeling communication between entities within a distributed system.
- To explore the scalability of model checking.
- To identify useful features in design description languages for model checking.

The System Model

The distributed system model includes some number of processors, a global memory, and a shared bus. Each processor has its own cache which is connected to the global memory via a shared bus. A cache is a set of cache lines. Each cache line is associated with a memory address and contains two fields: a data value and a flag indicating whether or not the line is valid. When the flag is true, the data value should be consistent with the value in the corresponding address of the global memory.

Each processor can send read and write requests to its cache; these requests drive the protocol. The following rules govern system-wide responses to messages:

- When a cache receives a read request from its processor, it supplies the data value if the cache line for that address is valid. If the cache line is not valid, the cache requests the contents from the global memory via the bus. When the data comes back on the bus, the cache stores the value, sets the cache line to valid. and returns the value to the processor.
- When the cache receives a write request from its processor, it puts a write request out on the bus and stores the new data value.
- Every cache snoops the bus. When a cache sees a write request from another processor, it invalidates its cache line for the requested address.
- The global memory shares the bus with the caches. When the global memory sees a write request on the bus, it stores the provided data value in the provided address. When the memory sees a read request on the bus, it supplies the stored value for the requested address.

Exercises

1. Develop a model for the cache coherence protocol and its environment. Start with a small configuration containing two processors and two memory addresses.
2. Verify that the model preserves consistency between the memory and the caches.
3. Scale your model in at least two dimensions, noting how verification time, memory requirements, and bdd size vary across those dimensions. Produce a table or graph of your results. What's the largest configuration that you can verify within a reasonable amount of time and memory?
4. Critique the SMV design language relative to this project. Which constructs/features were most useful? Which did this project leave you wishing you had? For each "wish-list" item, address whether feature been implemented as a macro over the current language, or whether the expressive power of the language would need to change.

Project Timeline

Developing this model is a non-trivial task (hence the three-week period on this assignment). **Start early!** Attempt to lay out your initial system model before class on Feb 8. We will devote part of the class sessions on Feb 8th and 15th to discussing the modeling challenges that people have encountered while working on this problem.