EngrECE 298
System-on-Chip Description and Modeling
Spring 2004

Assignment 3

Posted: May 7, 2004 (week 5)
Due: May 14, 2004 (week 6)

Task: Complete the MPEG-Audio Decoder Specification Model

Instructions:

The goal of this third assignment is to complete the Specification Model for the MPEG-Audio decoder that has been prepared in the previous assignment. As a result, the model should fully comply with the guidelines discussed in the lectures and also simulate successfully.

Task 1: Complete the tasks listed in the previous assignment.

In particular, complete the Specification Model of the MPEG-Audio Example such that the model satisfies the following conditions:

1. The model contains all necessary code for the decoder.
2. The model contains a test bench with stimulator, DUT, and monitor.
3. The model compiles successfully with the SpecC compiler.
4. The model simulates successfully.

Task 2: Create a well-specified behavioral hierarchy for the decoder.

The decoder is the design under test (DUT). As such, it has to comply with the modeling guidelines for a well-specified specification model. Otherwise, no system exploration and implementation is possible.

In particular, a clean and well-organized behavioral hierarchy is necessary for the design under test. Create such a hierarchy and keep the following issues in mind:

- Granularity:
  As a rule of thumb, each C function should become a separate SpecC behavior.
- Hierarchy:
  Try to mimic the given functional hierarchy. Use the SourceNavigator to browse the function call tree of the C code and replicate a similar hierarchy by using appropriate SpecC behaviors.

- Concurrency:
  Add explicit concurrency wherever possible. For every composite behavior, decide if a **`par`** or **`pipe`** construct could be used to specify any potential concurrency.
- Communication:
  Use channels from the standard library or local variables. Avoid global variables and pointers at all costs.

For the modeling task, you may want to consult the following Technical Report that contains detailed rules and guidelines for the specification model, as well as examples that show how C code can be converted to SpecC:

- CECS Technical Report 03-21:
  “System-on-Chip Specification Style Guide”
  by A. Gerstlauer, K. Ramineni, R. Doemer, D. Gajski.
- http://www.ics.uci.edu/~doemer/publications/CECS_TR_03_21.pdf

--
Rainer Doemer (ET 444C, x4-9007, doemer@uci.edu)