Objective: In this assignment the student will:

- Complete the verification environment
- Exercise building an input injector
- Exercise building output collector.
- Exercise building checkers.
- Exercise simple test writing.

Procedure:

Complete the verification environment:

1. Create a channel unit called channel_u and add 3 of those to env_u (each for each output port).

Help Tips:

- list of instances

2. Create a packet injector.
   a. Add a method to env_u that receives a packet and inject it to the DUT.
   b. Define a field under env_u that determine the delay between two consecutive packets that are injected to the DUT.
   c. Define a field under env_u that defines how many packets are generated and a loop that creates those packets and send them to the injector method.
   d. Create a method that loops on the number of packets, generate each packet and send it to be injected. Keep in mind that we will control the packet creation from the test, therefore the test file needs access to the generated packet.

Help Tips:

- injecting, injecting and collecting
- tick_max, configure run
- keep soft
- pack, a simple example of packing
- reading HDL, sync, wait, true
- starting the main TCM, starting multiple TCMs, stopping by main TCM

3. Create a collector method for the channel_u. Note that the collector has to respond to the DUT according to the output protocol.

Help Tips:

- injecting and collecting
- unpack, a simple example of unpacking
- get_enclosing_unit

4. Create a checker (scoreboard), you can define either a global checker under env_u or a checker under each channel (recommended).

Help Tips:

- Scoreboard
- check, check method, dut_error
- compare

Due: December 3, 2003