Objective: In this assignment the student will:

- Exercise modeling high-level constructs for describing data.
- Exercise random creation of data including describing data relationships and controlling the data outlook.
- Building of procedural code.
- Using Specman for test creation.
- Connecting to the simulator and accessing simulator signals.
- Writing drivers to the DUT

Procedure:

1. Create a packet struct:
    a. Create fields that represent the actual fields of the packet
    b. Create a method that accepts the length, address and payload of the packet, calculate the parity of the packet and return it.

Implementation notes:

- Call the struct you define *packet_s*.
- Call the file you write *router_env.e*.

Help Tips: (use Specman Advisor for online help)

- modeling data items, specification based verification, defining struct, scalar types, defining fields, physical fields
- method definition, parameter passing,for each in, passing information to method extensions, invoking methods, for each with and without

2. Add support for error creation:
    a. Add to the packet the ability to control error creation. There can be two kinds of errors, a parity error and an address error.
    b. Use the environment provided to write the following tests:
        I. Create 100 packets with 10% parity error
        II. Create 100 packets with 10% address error
        III. Create 100 packets with 10% errors (both parity and address)

Implementation notes:

- Implement first test in the provided *ex2test1.e* file. This file add a list of packets to sys and print them when the test starts. Implement the second in file *ex2test2.e* and the third in file *ex2test3.e*.
- Instructions how to run the test can be found in *ex2README* file.

Help Tips: select, handling corner cases, importing e files, output routines.

3. Connect to the DUT:

Write a unit called env_u, put it under sys and define an event called clock that triggers each time the DUT clock happen. Write a test file called *ex3test.e* that run the DUT for 1000 clock cycles.

Help Tips: units overview, is instance, unit, defining clocks, declare time consuming method, wait, run(), start TCM(), stop_run(), hdl_path().

Note: Use the provided script *ex3run.sh* run both Specman and the simulator for this exercise.

Due: October 27, 2003