

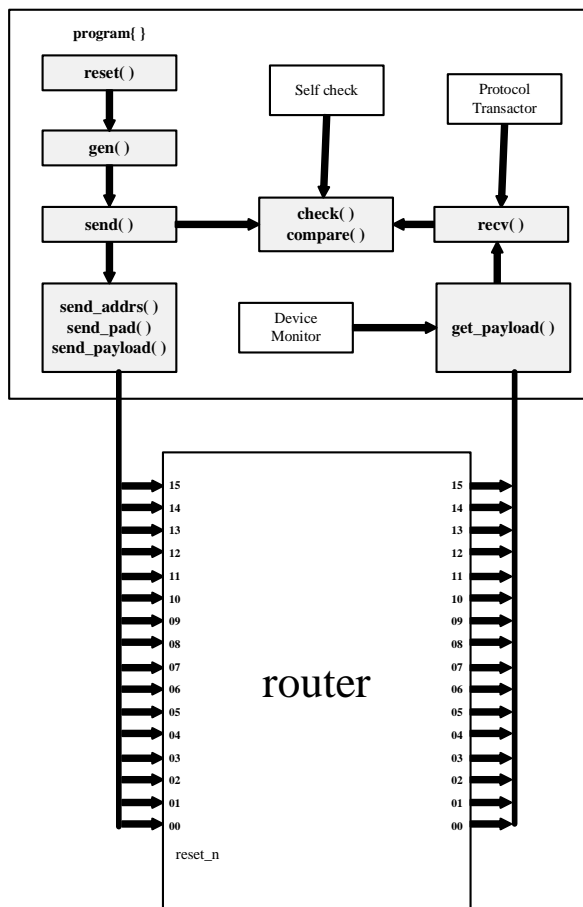
LAB2 说明

一、实验介绍

通过 Lab1 的实验，相信大家已经对通过使用 SystemVerilog 语言编写验证文件有了一个比较深入的了解，在本次实验中，我们为同学们提供一些带有一些基本测试功能的测试平台，内部包含有面向对象的发送与接收机制。

首先，在 rtl 文件夹中，我们为大家提供的 dut 代码是属于加密性质的，其中 router_questa.vp 适用于 modelsim 和 questa sim 平台，router_vcs.vp 适用于 vcs 平台仿真，我们一般选用 router_questa.vp 即可。

在 class exercise1 中，我们为同学们提供了一个基本的验证平台，该平台未使用面向对象的语法，譬如 class。该验证平台可实现如下图中的所有功能，基本上每段代码我都给出了注释，便于同学们理解，基本写法和 lab1 中的提出的要求是一致的，因此也是对 lab1 实验的一种自我检查，本 Testbench 为大家提供了 2000 个数据包，代码中可能会有些许错误，因而不能直接正确实现全部数量的数据包验证，请同学们认真理解验证代码，更改少量错误即可。

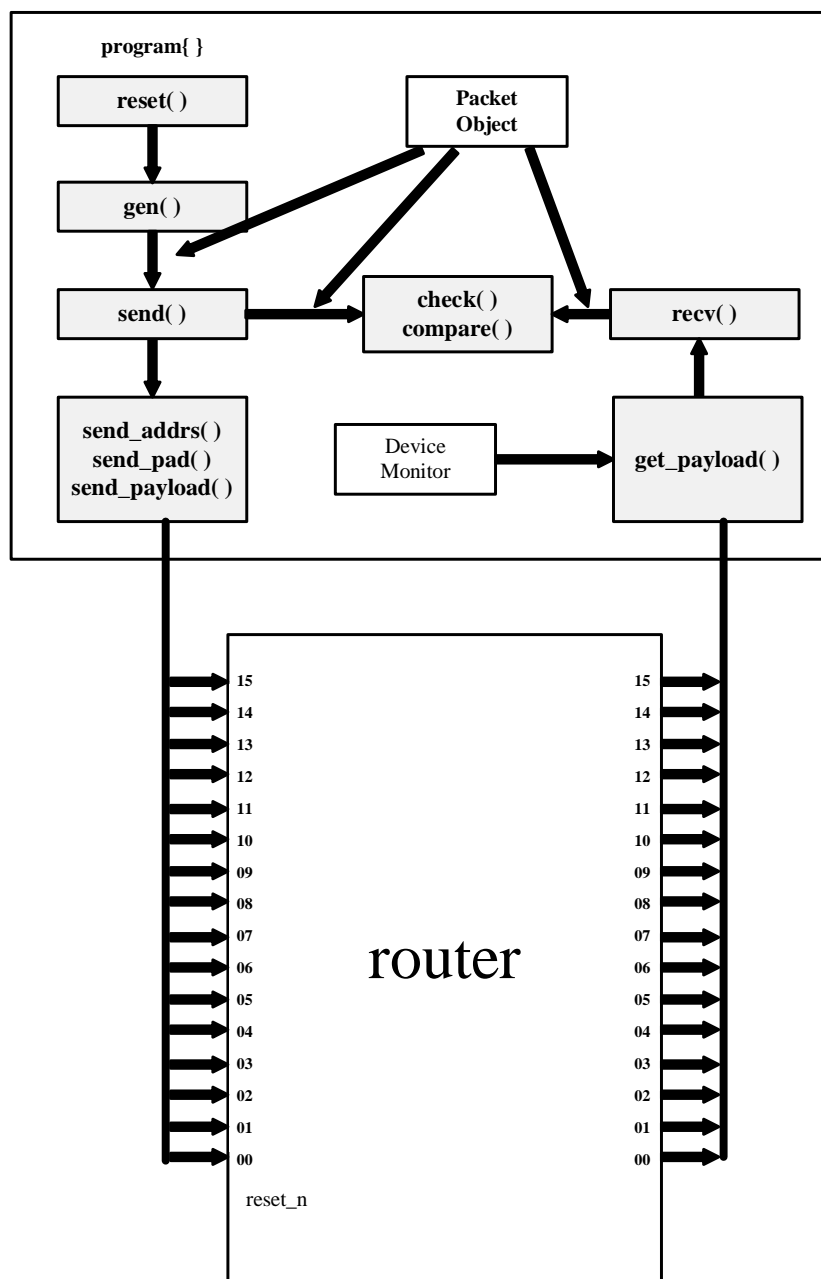


在 class exercise2 文件夹中，我们为同学们提供了一个面向对象的基本的验证平台，该平台应该是没有问题的。

为此，我们创建了一个为 DUT 提供基本必要输入的“Packet”类，并在代码中通过\$urandom()函数来创建数据包的有效载荷等，随后在 test.sv 中实例化 packet 类，把每个 packet 包发送到 dut 中，同时 Testbench 会接收来自 DUT 的输出并且进行比较，针对每个批次的数据包进行 DUT 功能正确性检查。

注意：关于 DUT 模块的基本功能及有效指令请参考 router 说明.pdf。

本验证平台实现功能如下图所示：



二、RTL 代码辅助理解

Exercise1 中的所有代码均有注释，同学们可以自行好好体会，我们主要讲解下 exercise2 中的内容，首先看下 Pack.v 中的内容，其中包含一个 class，其中包含作为输入发送给 DUT 的字段。其中还包括对这些信号的适当约束，以将它们保持在有效的范围内。

```
class Packet;
  rand bit[3:0] sa, da;           // random port selection
  rand logic[7:0] payload[$];    // random payload array
  string name;                   // unique identifier

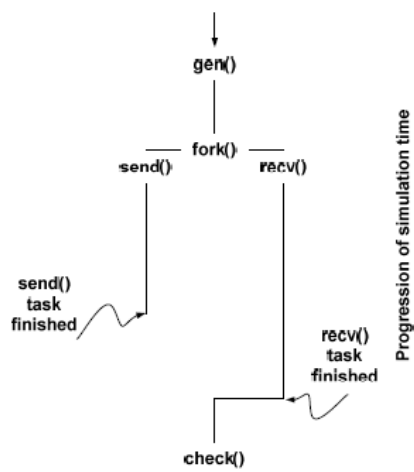
  constraint Limit {
    sa inside {[0:15]};          // 范围限制
    da inside {[0:15]};
    payload.size() inside {[2:4]};
  }
endclass
```

随机化的前提

范围限制

如果您想更关注某部分的传输，可以更改限制范围。

本实验所采用的测试方法如下：1) 创建一个带有范围限制的随机数据包。2) 把所有的数据包发送给 DUT 模块。3) 从 DUT 输出端口接收数据与之比较，检查正确性，如下图所示：



```
repeat(run_for_n_packets) begin
  gen();
  fork
    send();
    recv();
  join
  check();
end
```

更多理解请参考 exercise1 中的注释

同学们可尝试使用命令行的模式编译仿真：

- 1) vlib work
- 2) vlog 绝对路径/router_questa.vp
- 3) vlog 绝对路径/exercise1/testbench/*.sv
- 4) vism -novopt router_test_top

三、实验报告要求

1、回答下列问题：

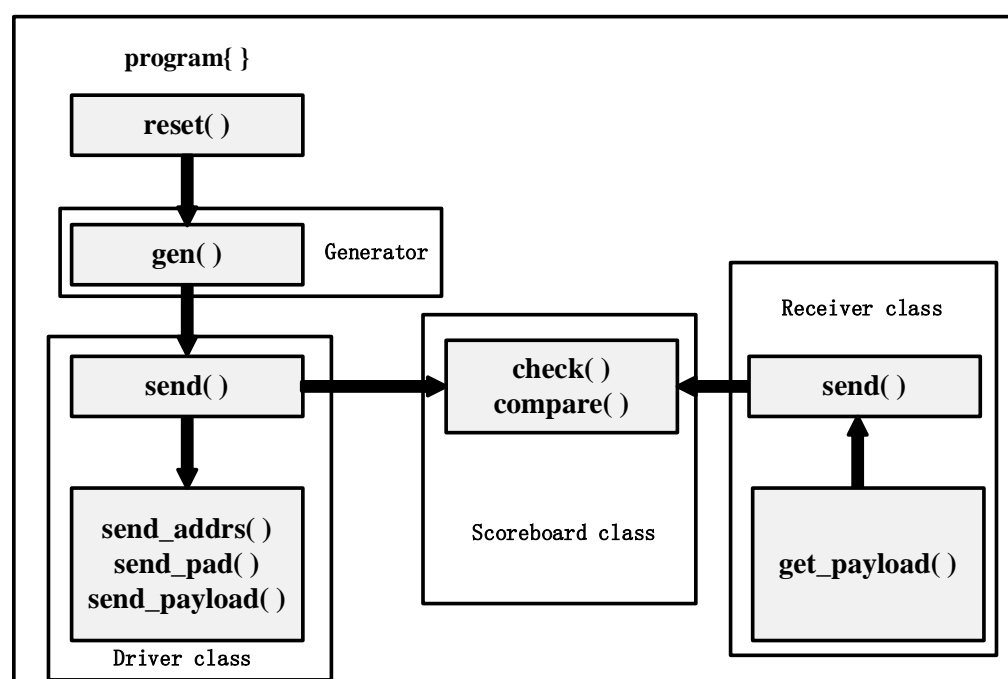
1) 这条语句的意义是什么？

```
repeat(5)@(router.cb);
```

2) 描述下 reset 任务的时序和功能特征

2、本验证平台只实现了 Packet 和 Generator 类，我们在 example 文件夹中给大家提供基本的模板，同学们不一定要完全按这个模板中给出的 Task 编写，请同学们根据类的写法，参考 exercise1 中的内容，把 exercise1 中的所有模块更改为类对象，主要更改如下：

编写一个 Scoreboard 类，该类用来比对数据包；编写一个 Driver 类，该类用来驱动 Packet 给 DUT 模块；创建一个 Receiver，该类用来接收 DUT 的输出，进行正确性检查。扩展 testbench 去驱动所有输入端口和监测所有输出端口。



请同学们将实验报告和代码打包发送到 chenyangbing@sjtu.edu.cn 邮箱中，文件名为 LAB2_学号_姓名（如 LAB2_119039910110_马梅），截止日期：2019 年 12 月 12 日。实验报告是评定实验成绩的主要依据，请同学们尽量做到书写规范，思路清晰易懂，添加适当的注释，切勿互相抄袭，一经发现必定给予严厉的处罚。