集成电路设计仿真与验证技术
实验报告

题　　目：利用 mailbox 进行线程间通信

学　　院：电子工程学院

系　　部：微电子学系

专　　业：集成电路设计与集成系统

班　　级：电路 2101

姓名学号：赵祖康 05219038

指导教师：闫宵

一、 实验目的（重点说明本实验要学习的知识点以及要达到的目标）。

通过本次实验，利用 sv 中的 mailbox 机制实验线程间的通信

二、 实验要求：实验环境的要求（例如建议的硬件平台，软件环境及版本等）。

硬件平台：vivado2018.3

三、 实验原理（重点说明本次实验使用的知识点）

使用了 SystemVerilog 的面向对象特性，通过定义 Producer 和 Consumer 两个类来模拟生产者和消费者。这两个类通过一个共享的 mailbox 进行交易对象的发送和接收，从而实现线程间的通信。

四、 实验过程

首先定义类，并使用 rand 对其进行随机赋值并加以约束

```
class transaction;
    rand int len;
    rand int load;


    constraint c_valid {
        len > 50 && len < 200;
        load > 0 && load < 200;
    }
endclass
```

然后声明生产者并产生激励

```
// 生产者类定义
class Producer;
    mailbox #(transaction) prod2cons; // 生产者到消费者的邮箱

    function new(mailbox #(transaction) prod2cons);
        this.prod2cons = prod2cons;
    endfunction

    // 生成多笔激励
    task generate(int num);
        transaction tr;
        for (int i = 0; i < num; i++) begin
            tr = new();
            if (!tr.randomize()) $display("Randomization failed at iteration %0d", i);
            prod2cons.put(tr);  // 把产生的激励放入邮箱
            $display("Transaction put by producer: len = %d, load = %d", tr.len, tr.load);
            #1ns;
        end
    endtask
endclass
```

声明消费者

```
// 消费者类定义
class Consumer;
    mailbox #(transaction) prod2cons;

    function new(mailbox #(transaction) prod2cons);
        this.prod2cons = prod2cons;
    endfunction

    task consume();
        transaction tr;
        forever begin
            prod2cons.get(tr);  // 从邮箱中取出数据
            $display("Transaction received by consumer: len = %d, load = %d", tr.len, tr.load);
        end
    endtask
endclass
```

最后进行激励使用 `fork...join_any` 启动生产和消费进程，并在任一进程完成后停止所有并发进程确保安全。

```
mailbox #(transaction) prod2cons = new();
Consumer cons = new(prod2cons);
Producer prod = new(prod2cons);

initial begin
    fork
        prod.generate(10);
        cons.consume();
    join_any
    disable fork;
end
```

最后得到仿真结果如下

```
Time resolution is 1 ps
Transaction put by producer: len =        190, load =         24
Transaction received by consumer: len =        190, load =         24
Transaction put by producer: len =         97, load =        163
Transaction received by consumer: len =         97, load =        163
Transaction put by producer: len =        135, load =        105
Transaction received by consumer: len =        135, load =        105
Transaction put by producer: len =        132, load =         63
Transaction received by consumer: len =        132, load =         63
Transaction put by producer: len =        104, load =        180
Transaction received by consumer: len =        104, load =        180
Transaction put by producer: len =        169, load =        128
Transaction received by consumer: len =        169, load =        128
Transaction put by producer: len =        162, load =        168
Transaction received by consumer: len =        162, load =        168
Transaction put by producer: len =         74, load =        177
Transaction received by consumer: len =         74, load =        177
Transaction put by producer: len =         53, load =        131
Transaction received by consumer: len =         53, load =        131
Transaction put by producer: len =        147, load =        199
Transaction received by consumer: len =        147, load =        199
```

五、 实验总结

通过本次实验，理解了 SystemVerilog 中 mailbox 的实际应用和它在硬件验证中的重要性。同时了解到操作系统中进程/内核级线程间通信与 sv 中的异同，实验中，通过实际操作见证了异步通信模式能够如何有效地解耦生产者和消费者的操作。