



# System Verilog Lab2

TA: 孙佳磊

Mail: [sjl\\_519021910940@sjtu.edu.cn](mailto:sjl_519021910940@sjtu.edu.cn)

TA: 谢琳

Mail: [lynnxie@sjtu.edu.cn](mailto:lynnxie@sjtu.edu.cn)

2024年11月21日



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

# 一、实验准备



## ■ 实验材料

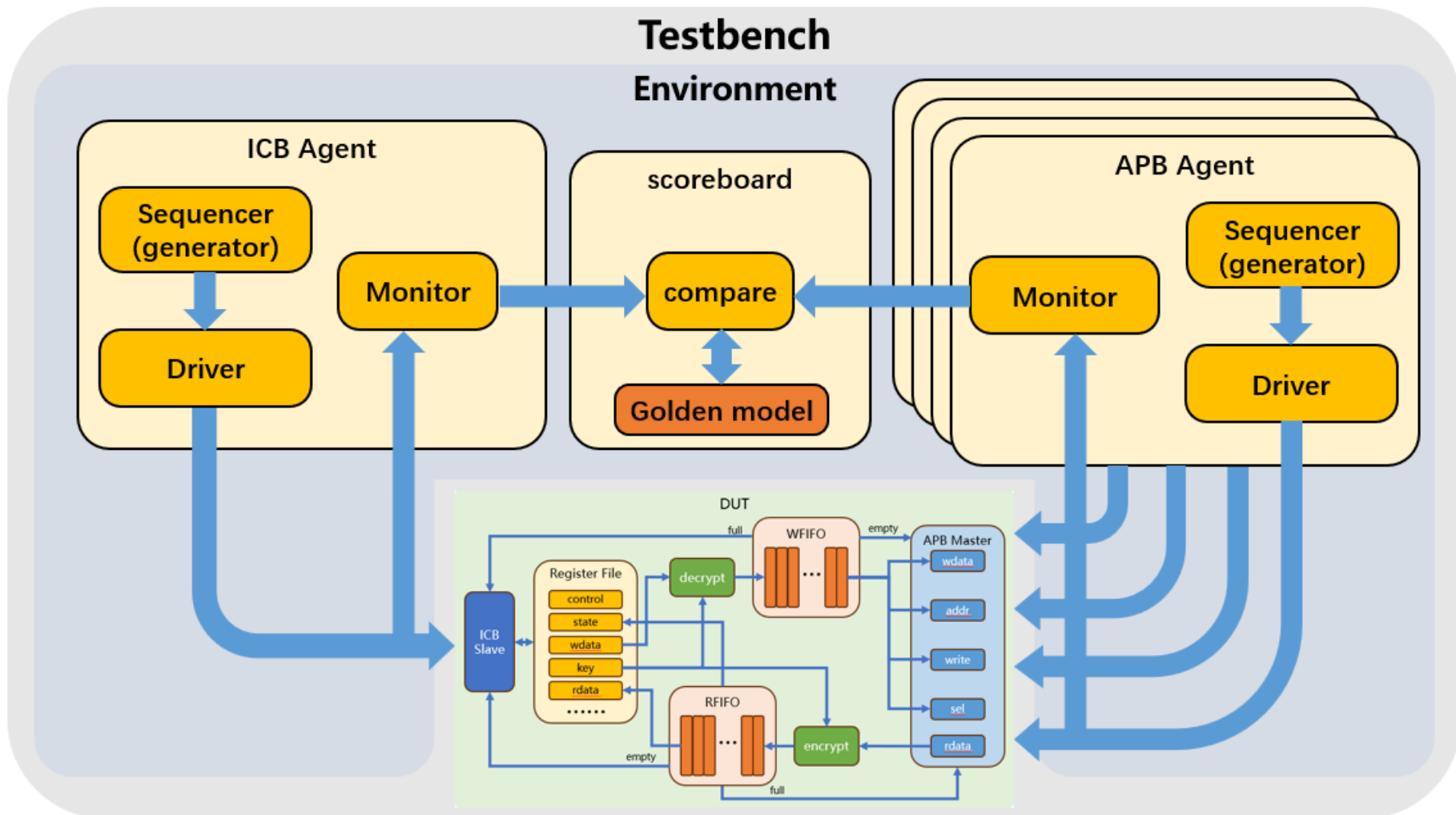
登录canvas, 在  ▼ SystemVerilog\_2024\_lab2 单元下下载以下文件

(1) 实验PPT (本文档)  SystemVerilog\_2024\_lab2.pptx

(2) sv文件模板  lab2\_svfile

(3) quiz文件  lab2\_quizpart

## 二、实验内容



## 二、实验内容



要求	项目
必做	ICB端： (1) ICB总线时序正确性检查 (2) 所有寄存器的读写测试 (3) 随机地址读写测试
必做	APB端： (1) APB总线时序正确性检查 (2) 任意地址的读写测试 (3) 数据包解码正确性检查 (4) 4个通道读写覆盖
必做	WFIFO与RFIFO： 配合ICB端和APB端读写完成两个FIFO空满状态测试覆盖
必做	Encrypt/Decrypt： 完成加/解密正确性检查
必做	数据流：ICB端与APB端数据正确传输检查
必做	Interface中加入clock_block来对时序进行调整



## 二、实验内容



要求	项目
选做	<p>随机化测试：</p> <ul style="list-style-type: none"><li>(1) 测试数据，如ICB的wdata、address、请求类型，APB的rdata等</li><li>(2) 测试请求的随机化，即在随机的时刻驱动ICB master发起指令，并据情况接收APB端的结果</li></ul>
选做	<p>完成一个完整的testbench结构：</p> <ul style="list-style-type: none"><li>(1) 搭建monitor对象：采集ICB端和APB端的数据</li><li>(2) 搭建golden model对象：根据ICB和APB端输入输出及编解码结果判断加解密和传输结果的正确性</li><li>(3) 搭建scoreboard对象（golden model可以内嵌在这个对象里），对上述判断结果进行错误率的统计，统计结果在测试结束时进行打印。</li></ul>
选做	实现DUT使用DES进行加解密的功能并完成相关测试



## 二、实验内容

- 实验提供仿真平台的基本架构，需要同学们根据备注及要求填充内容
- 仿真平台的基本架构不可更改，起码要完成提供的基本对象的功能实现，可以进一步增添内容
- 本次实验不可使用UVM实现



```

=====
// Description:
// This file realize the ICB AGENT, includes data generator, driver and
// monitor.
// Designer : lynnxie@sjtu.edu.cn
// Revision History
// V0 date:2024/11/07 Initial version, lynnxie@sjtu.edu.cn
=====

`timescale 1ns/1ps

package icb_agent_pkg;
    import objects_pkg::*;

    // Generator: Generate data for driver to transfer
    class icb_generator;...
endclass //icb_generator

    // Driver: Converts the received packets to the format of the ICB protocol
    class icb_driver;...
endclass //icb_driver

    // **Optional** Monitor: Collect ICB data and convert it to data package for
    // scoreboard to compare result.
    class icb_monitor;...
endclass //icb_monitor

    // Agent: The top class that connects generator, driver and monitor
    class icb_agent;...
endclass //icb_agent
endpackage
    
```

```

=====
// Description:
// This file realize the APB AGENT, includes data generator, driver and
// monitor.
// Designer : lynnxie@sjtu.edu.cn
// Revision History
// V0 date:2024/11/11 Initial version, lynnxie@sjtu.edu.cn
=====

`timescale 1ns/1ps

package apb_agent_pkg;
    import objects_pkg::*;

    // Generator: Generate data for driver to transfer
    class apb_generator;...
endclass //apb_generator

    // Driver: Converts the received packets to the format of the APB protocol
    class apb_driver;...
endclass //apb_driver

    // **Optional** Monitor: Collect APB data and convert it to data package for
    // scoreboard to compare result.
    class apb_monitor;...
endclass //icb_monitor

    // Agent: The top class that connects generator, driver and monitor
    class apb_agent;...
endclass //apb_agent
endpackage
    
```

```

=====
// Description:
// This file is the top testbench file. In the testbench_top module, build
// the instance of the module of dut and program of testbench, even the
// interface and some global signals. 'testbench' file control the total
// sim process, here add your command to ENV object.
// Designer : lynnxie@sjtu.edu.cn
// Revision History
// V0 date:2024/11/07 Initial version, lynnxie@sjtu.edu.cn
=====

`timescale 1ns/1ps

module testbench_top ();

    // Parameters
    //===== ...

    // Signals Declaration
    //===== ...

    // Signals' Function
    //===== ...

    // Connections between DUT And Testbench
    //===== ...
endmodule

program testbench(...
);...
endprogram
    
```

```

=====
// Description:
// This file build the environment for the whole test environment
// Designer : lynnxie@sjtu.edu.cn
// Revision History
// V0 date:2024/11/11 Initial version, lynnxie@sjtu.edu.cn
=====

`timescale 1ns/1ps

package env;

    import icb_agent_pkg::*;
    import apb_agent_pkg::*;
    import objects_pkg::*;
    import scoreboard_pkg::*;
    import coverage_pkg::*;

    class env_ctrl;...
endclass //env_ctrl
endpackage
    
```

## 三、报告与提交



### ▪ 报告要求

实验报告需要包括详细的设计思路描述和时序描述，为了方便阅读和评估，我们强烈建议包括如下部分：

- (1) 模块设计简单说明（基于LAB1报告进行简述）
- (2) 验证平台的详细说明，包含基本的框架图、对象要素列表、对象功能列表
- (3) DUT功能验证结果及分析
- (4) 总结，概述完成的内容（对照前面列出的表格，**打分关注**）
- (5) quiz问答【详见下一个part】

**请注意，实验报告将是判定实验成绩的主要依据，请尽量做到书写规范，思路清晰易懂，切勿互相抄袭，发现抄袭一律不给分。**

## 三、报告与提交



### ▪ 文件提交

文件提交方式：Canvas

文件提交格式：压缩包

截止日期：12月11日23:59

```
文件名：学号_姓名_SV202XLAB_0X  
示例   ：124039910123_小明_SV2024LAB_01  
  
|----->design_files  
|           设计文件夹：存放设计相关的SystemVerilog、verilog文件  
|----->testbench  
|           测试平台文件夹：存放测试平台搭建相关文件  
|----->report  
|           报告文件夹：存放编写的报告  
|----->README (可选)  
|           若文件结构迫不得已需要更改或出现其他需要说明的情况，可在该文件中说明
```

**请注意，由于提交的文件需要被运行检查，这是一个繁重的工作，所以我们同样强烈建议严格按照如上文件格式提交，以方便我们运行你的设计。**





# System Verilog Lab2

## Quiz Part



上海交通大學  
SHANGHAI JIAO TONG UNIVERSITY

# To Do



- 在Canvas的下载资料中打开quizpart文件夹

 `interprocess_sync.sv`

 `package_usage.sv`

 `thread_control.sv`

- 请结合思考指导仿真运行这些文件，文件运行的**仿真波形或输出截图**和**对思考题的解答**请附在实验报告的末尾。



# Package的使用



- 编译package\_usage.sv, 并思考问题
  - 1. 38行处sun类对象s并未被识别, 如何改正
  - 2. 40行处sun state状态并未打印, 如何改正
  - 思考: Package与Class的关系

# 进程间通信



- 编译并运行interprocess\_sync.sv，并思考问题
  - 1. 将b\_event\_use设为1，其余为0，运行并观察结果。
    - 如何触发事件？@e1 与 e1.triggered() 有什么区别？
  - 2. 将b\_mailbox\_use设为1，其余为0，运行并观察结果。
    - 如何向信箱里放入指定类型？如何使用信箱？
  - 3. 将b\_mailbox\_user\_define设为1，其余为0，运行并观察结果。

# 进程的控制



- 编译thread\_control.sv, 并思考问题
  - 1. 直接运行代码, 代码依次执行了三个进程, 每个进程中有三个子进程
  - 2. 观察打印结果, 掌握fork join, fork join\_any, fork join\_none的区别
  - 3. disable fork 和 wait fork 的作用