



# System Verilog Final Project

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)

2025年1月9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\_finalProject 单元下下载以下文件


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


(2) 参考文档  doc\_helps


(3) uvm示范工程  uvm\_demo

(4) vivado综合示例  vivado\_helps

 constraint 约束文件

 source 存放dut

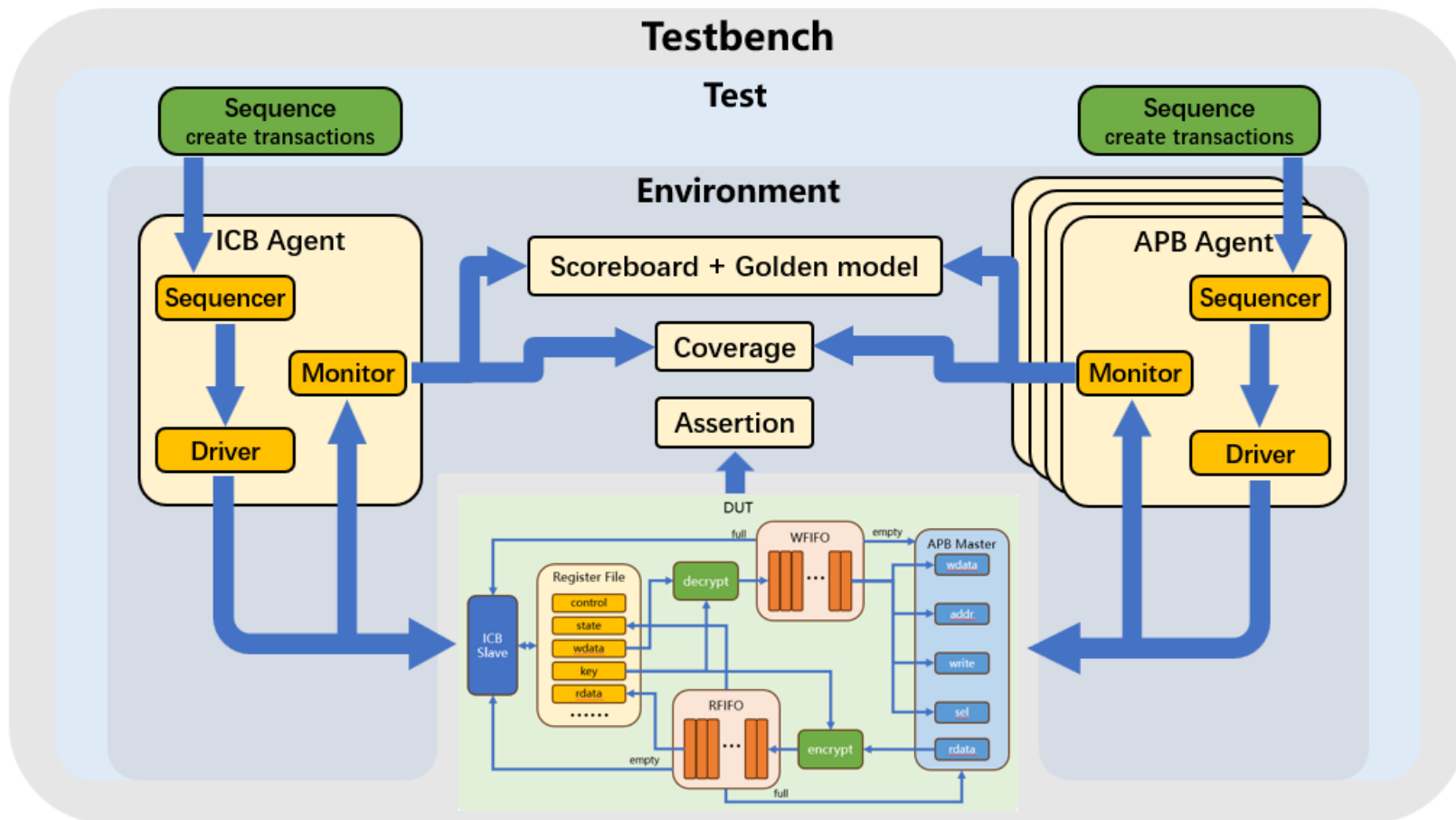
 lab1\_project.tcl 运行的脚本

 readme.txt 操作指南

## 二、实验内容



实现**完整的**仿真平台，验证使用**DES算法**实现加解密功能的DUT





## 二、实验内容



要求	类型	项目
必做	验证计划制定	参考doc_helps中给出的验证计划示例，制定自己的验证计划，包括功能覆盖及代码覆盖目标
必做	ICB驱动设计	完成验证平台设计，该平台至少能够通过interface连接DUT，并拥有ICB Agent来驱动DUT进行工作： (1) 完成ICB读操作 (2) 完成ICB写操作
必做	APB驱动设计	完成验证平台设计，该平台至少能够通过interface链接DUT，并拥有APB Agent来响应DUT的指令，进行工作： (1) 完成APB读操作 (2) 完成APB写操作
必做	功能验证设计	(1) 完成ICB Agent和APB Agent的monitor模块，采集数据，包括ICB总线和APB总线上的读写数据等。 (2) 设计Scoreboard和golden model模块，结合monitor采集的数据，完成功能的验证

## 二、实验内容



要求	类型	项目
必做	断言设计	<p>根据验证计划，设计适当的assertion来检查ICB总线时序的正确性：</p> <ul style="list-style-type: none"><li>(1) 信号的X态检查</li><li>(2) 信号传输时的稳定性检查</li><li>(3) 信号的时序关系检查，如未握手的保持，握手的出现，CMD通道发送指令则RSP通道必会返回等</li></ul>
必做	断言设计	<p>根据验证计划，设计适当的assertion来检查APB总线时序的正确性：</p> <ul style="list-style-type: none"><li>(1) 信号的X态检查</li><li>(2) 信号传输时的稳定性检查</li><li>(3) 信号的时序关系检查，如PSEL，PENABLE的拉高，PENABLE握手后拉低等</li></ul>



## 二、实验内容



要求	类型	项目
必做	覆盖率设计	<p>(1) 对DUT中所有有效寄存器进行功能覆盖率统计：无论寄存器能否读写，均对其进行读写检查</p> <p>(2) 对DUT中wfifo和rfifo的空满状态进行功能覆盖率统计</p> <p>(3) 对APB总线、ICB总线的读写操作进行功能覆盖率统计</p> <p>(4) 对DES加解密功能正确性的覆盖</p> <p>(5) 给出DUT的代码覆盖率报告并分析（Statement、Branch、Condition、Expression Coverage等）</p>
必做	随机化设计	<p>采用随机化完成测试，要求如下：</p> <p>(1) 配置随机化</p> <p>(2) 操作发起时间随机化</p> <p>(3) 读写操作随机化</p> <p>(4) 数据随机化：读写数据、读写地址等</p>



## 二、实验内容



要求	类型	项目
必做	DUT设计	实现DES加解密算法
必做	DUT综合	DUT在vivado内完成综合（在报告中说明消耗的资源、时序等）

**请注意，为了统一评判标准，统一使用提供的脚本文件生成工程脚本文件中定义的板卡芯片（xc7k70tfbg484-1）不可更改**



## 二、实验内容



推荐	类型	项目
增益	优秀工程	(1) 详实的报告内容和良好的代码风格。 (2) 使用UVM实现仿真平台
减益	规避行为	(1) 迟到的提交，将在提交项上丧失少量分数。 (2) 弄虚作假报告与实际工程不符，将丧失对应单项的所有分数。 (3) 抄袭，期末作业分数将折扣处理。



## 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) DUT设计与功能的简单说明（本次DUT**必须使用DES**实现加解密功能）
- (2) UVM验证平台的说明，包含**验证计划**、基本的框架图、对象要素列表、对象功能列表、断言、覆盖率等
- (3) 验证的时序图及分析。
- (4) 总结，概述完成的内容（对照前面列出的表格，**打分关注**）

**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：1月28日23:59

文件名更改：学号\_姓名\_SV2024LAB\_finalProject

```
文件名：学号_姓名_SV2024LAB_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。**

# X、Additional Support

## DES加解密算法

- DES算法的加密、解密流程如右图，其中加密过程与解密过程的唯一区别是16轮中使用的密钥是相反的。即加密时第1轮的密钥是解密时第16轮的密钥，加密时第2轮的密钥是解密时第15轮的密钥……以此类推。
- DES的具体实现，网上有很多参考资料，在此推荐一个解说得较为详细的：[算法科普：神秘的DES加密算法-腾讯云开发者社区-腾讯云](#)
- 有许多在线网址可实现DES的加解密功能，同学们可以此作为golden model，确认自己的功能是否正确。[DES加密/解密 - 锤子在线工具](#)

DES 加密/解密

运算模式: CBC (密码块链)

填充模式: None

密钥长度: 64 bits

密钥: Hex 2222777733330000

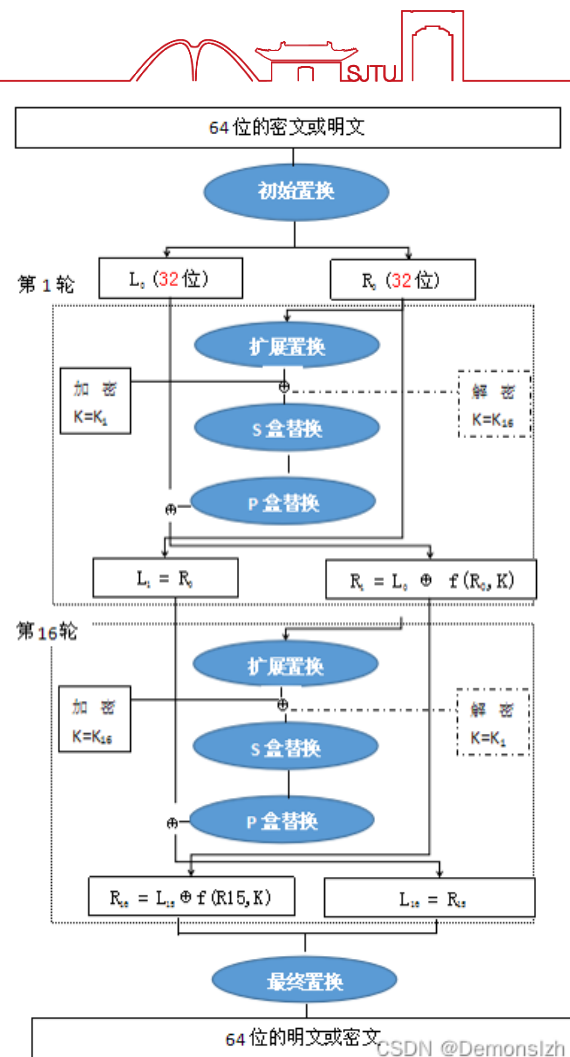
偏移: Text null or 64 bits

字符编码: Hex (16 进制)

格式: Hex

CDEF0123456789AB

E5C5E465F03198E0

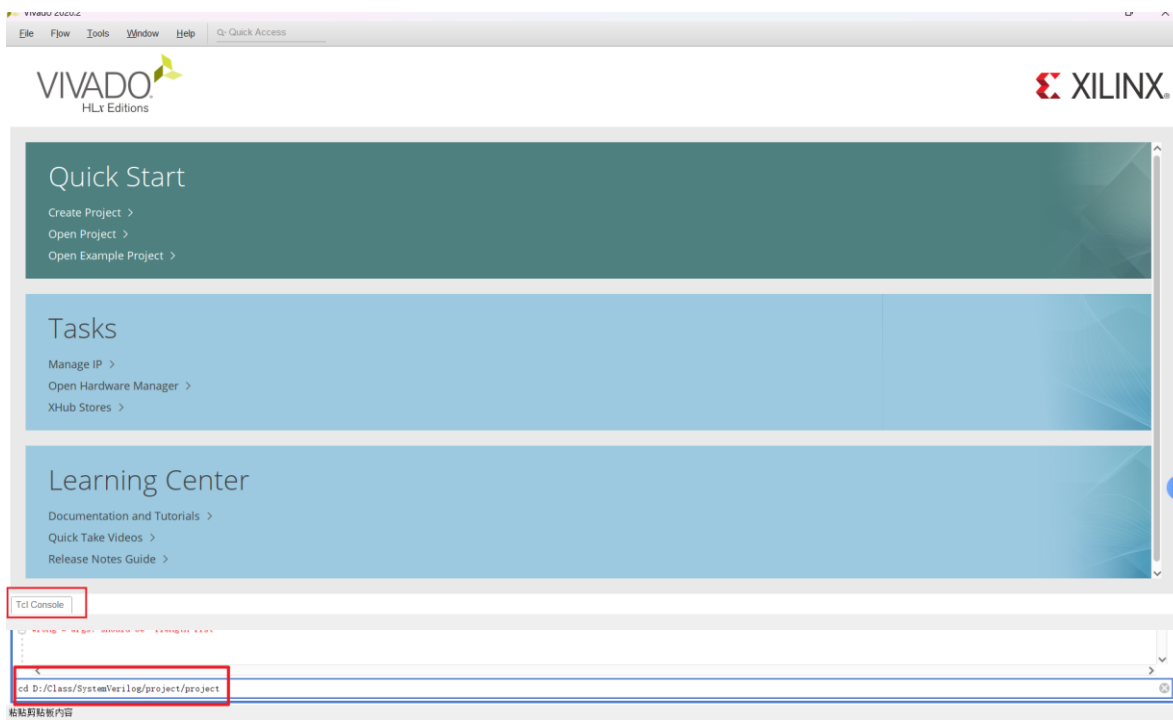


# X、Additional Support



## ■ VIVADO使用

- 复制vivado\_helps文件夹内内容到空白文件夹（路径不可有中文、空格）
- 将经过验证完善的DUT复制到文件夹/source/dut下
- 打开安装好的vivado，并将命令行路径指引到文件夹下



图形化界面

# X、Additional Support



## ■ VIVADO使用

- 复制vivado\_helps文件夹内内容到空白文件夹（路径不可有中文、空格）
- 将经过验证完善的DUT复制到文件夹/source/dut下
- 打开安装好的vivado，并将命令行路径指引到文件夹下
- 输入命令：source project.tcl



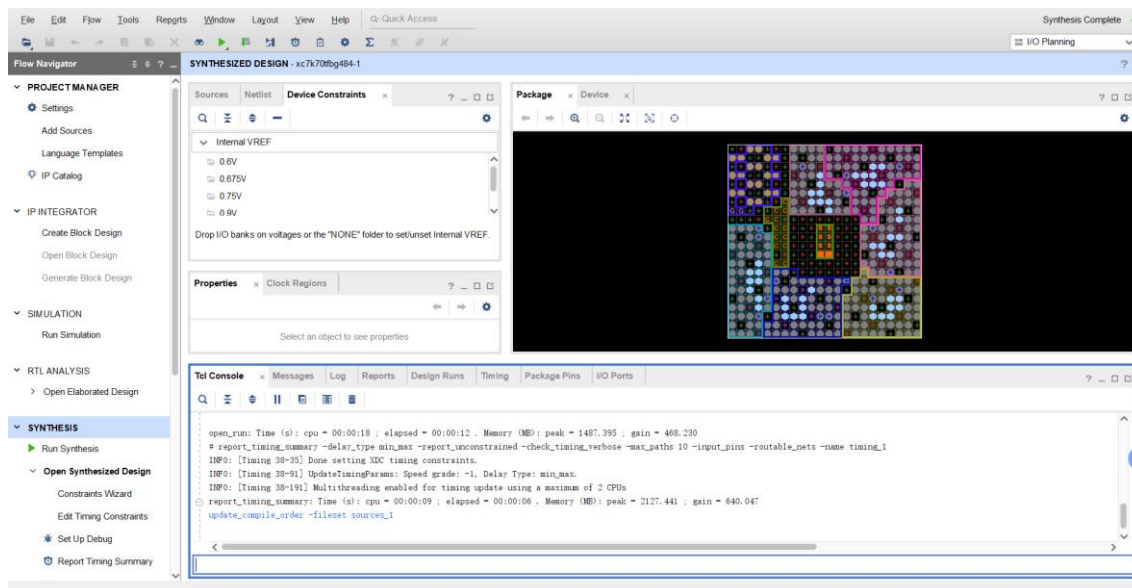
```
Tcl Console
[Runs 36-547] User Strategy 'Flow_RuntimeOptimized' from file 'C:\Users\sjl\AppData\Roaming\Xilinx\Vivado\2020.2\strategies\Flow_RuntimeOptimized_Vivado Implementation 2019.psg' discarded because strategy with same name already parsed from 'D:/work/Vivado/2020.2/strategies/VDI2019.psg'
[Runs 36-547] User Strategy 'Flow_RuntimeOptimized' from file 'C:\Users\sjl\AppData\Roaming\Xilinx\Vivado\2020.2\strategies\Flow_RuntimeOptimized_2_Vivado Implementation 2019.psg' discarded because strategy with same name already parsed from 'D:/work/Vivado/2020.2/strategies/VDI2019.psg'
[Runs 36-547] User Strategy 'Flow_RuntimeOptimized' from file 'C:\Users\sjl\AppData\Roaming\Xilinx\Vivado\2020.2\strategies\Flow_RuntimeOptimized_3_Vivado Implementation 2019.psg' discarded because strategy with same name already parsed from 'D:/work/Vivado/2020.2/strategies/VDI2019.psg'
start_gui
cd D:/work/vivado_project/projects
|
<
source project.tcl
```

# X、Additional Support



## ■ VIVADO使用

- 复制vivado\_helps文件夹内内容到空白文件夹（路径不可有中文、空格）
- 将经过验证完善的DUT复制到文件夹/source/dut下
- 打开安装好的vivado，并将命令行路径指引到文件夹下
- 输入命令：source project.tcl
- 等待综合完成





# X、Additional Support



## ■ VIVADO使用

- 复制vivado\_helps文件夹内内容到空白文件夹（路径不可有中文、空格）
- 将经过验证完善的DUT复制到文件夹/source/dut下
- 打开安装好的vivado，并将命令行路径指引到文件夹下
- 输入命令：source project.tcl
- 等待综合完成

### 命令行界面

#### Vivado 2020.2 Tcl Shell

命令行启动图形化界面：  
start\_gui

```
Vivado 2020.2 Tcl Shell - C:\So  ×  +  v

**** Vivado v2020.2 (64-bit)
**** SW Build 3064766 on Wed Nov 18 09:12:45 MST 2020
**** IP Build 3064653 on Wed Nov 18 14:17:31 MST 2020
** Copyright 1986-2020 Xilinx, Inc. All Rights Reserved.

vado% cd D:\\Class\\SystemVerilog\\project\\project
vado% source project.tcl
```