

一种用于 SOC 芯片的联合 验证方法

2023. 9



白盒子（上海）微电子科技有限公司

目录

CONTENTS

1

相关背景&出发点

2

实施方案简介

3

成果简介



白盒子（上海）微电子科技有限公司，是在上海市政府的大力支持下，由IC行业资深人士领军组建的高端芯片公司，致力于SDH（软件定义硬件）先进芯片设计技术研究，通过架构创新，成为高性能大算力基带SoC领域的创新者和领航者。白盒寓意开放协同，助力生态建设，赋能智慧通信。

1

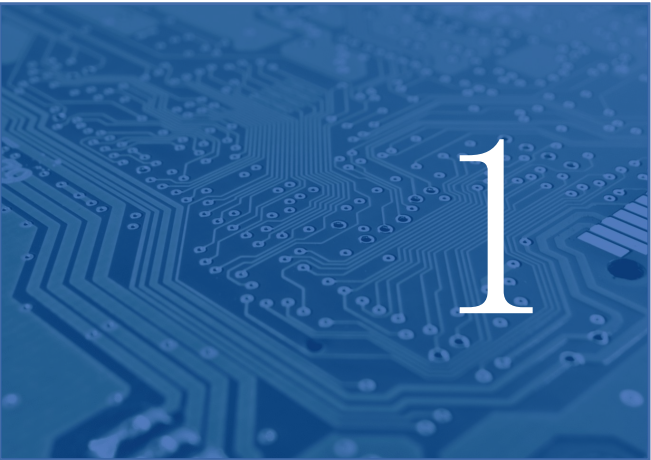
通信SOC芯片+模组+系统解决方案

以芯片为核心、模组为主体、解决方案为基石，构建完备且竞争力强的产品。“交钥匙”方案，降低通信产业门槛

2

全维度综合优势资深团队

来自国内一流通信及芯片设计公司、上市公司的资深人才组成了覆盖研发、管理、市场等全维度、综合优势显著的人才团队



1



相关背景和出发点简介



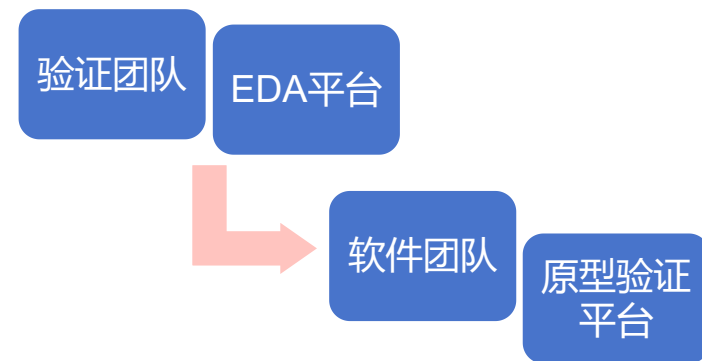
通信芯片：由多核SoC、高速外设及一系列复杂通信算法模块组成。

通常做法：

- EDA验证通常覆盖一些短的系统case；
- 复杂case从验证团队剥离，由解决方案团队在原型验证平台上测试。

问题：

- 软件团队介入时间晚；
- 对芯片及case不够熟悉，进度慢；
- 可能导致T0时间延迟；
- 芯片规模较大时，映射到原型验证平台较困难；
- Debug过程痛苦和漫长。



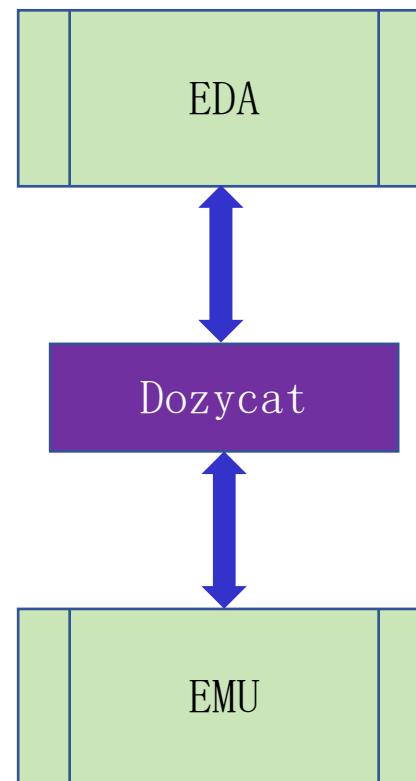


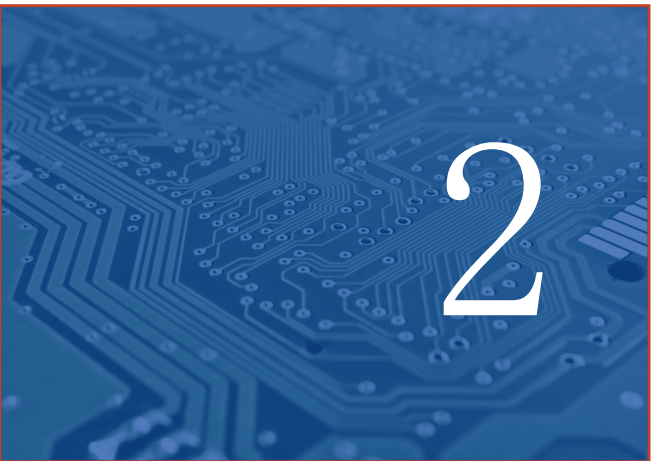
解决方案：

- 自研工具链Dozycat；
- 屏蔽EDA、EMU验证的差异。

收益：

- 解决了传统emulator问题；
- 验证的case可以无缝切换到emulator；
- 极大提高验证效率；
- 提高验证完备性；
- 减少总的验证人力投入。





2



实施方案介绍

工具链使用的大体流程



- 用例描述表格的填写对象是EDA验证工程师或者软件开发工程师;
- 具备开发工作量小、贴近业务、易理解、易维护等特点;
- 可选的EDA或EMU运行选项;
- 用例包含4大流程:
 - ✓ 系统初始化流程;
 - ✓ 参数配置流程;
 - ✓ 激励配置流程;
 - ✓ 数据检查流程;

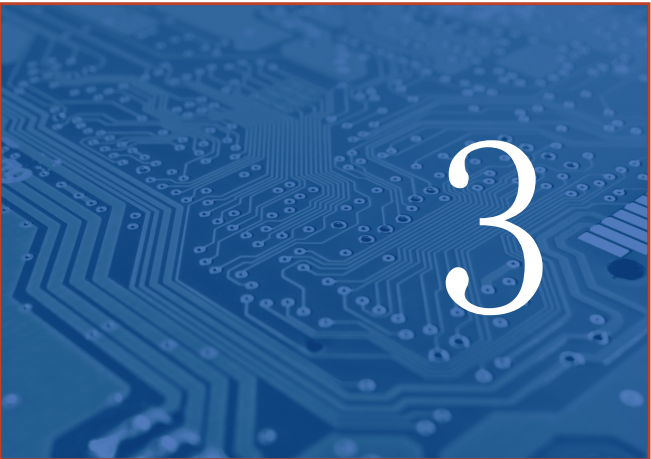
- Dozycat工具是轻量编译器;
- 编译器需求来源主要来自于用例梳理阶段提炼的自定义高阶语言;
- 三大功能特性:
 - ✓ C语言原义操作;
 - ✓ 自定义的INLINE函数;
 - ✓ 显式的直接函数调用;

必须解决的关键问题



case_name	case_test_features	case_configuration
spi_quar_std_test	1, We must <u>init</u> the soc clock and reset. Please use function : <u>clock_init()</u>. 2, Second step we need to <u>init</u> the <u>ddr</u> subsys, please use function: <u>ddr_init()</u>. 3, Please use function <u>spi_system_config()</u> to configure <u>spi</u> to <u>QSPI</u> mode. 4, Configure the feature-related parameters and then <u>enable SPI</u>. 5, You can use <u>send_tx_data()</u> to send data to SPI VIP. 6, Check RX data is done and RX <u>fifo</u> is not empty. 7, Finally, we call function <u>check_rece_data()</u> to check the return value	<pre>printf("SPI TEST CASE START!\n"); clock_init(); //CRG configuration process ddr_init(); //DDR configuration process qspi_system_config(); //Config SPI DUT to QSPI Mode SPI0_REG.CTRLR0.BIT.DFS_32 = 0x1f; //Config SPI Data Frame Size SPI0_REG.CTRLR0.BIT.SCPH = 1; //Config Serial Clock Phase SPI0_REG.CTRLR0.BIT.SCPOL = 0; //Config Serial Clock Polarity SPI0_REG.BAUDR.ALL = 2; //Config Baud Rate SPI0_REG.SSIENR.ALL = 1; //Enable SPI send_tx_data(0x11221122); //Start to send TX DATAS delay(100); status_bit_check(SPI_SR,0,0); //Check SPI IDLE status_bit_check(SPI_SR,2,1); //Check rx fifo is not empty reg_data_check(SPI_DRX,0x22221111,16); //Check rx data printf("SPI TEST CASE END!\n");</pre>

SPI测试用例描述示意

A blue background with a white circuit board pattern, featuring various lines, dots, and geometric shapes.

3

A blue background with a white circuit board pattern, featuring various lines, dots, and geometric shapes.

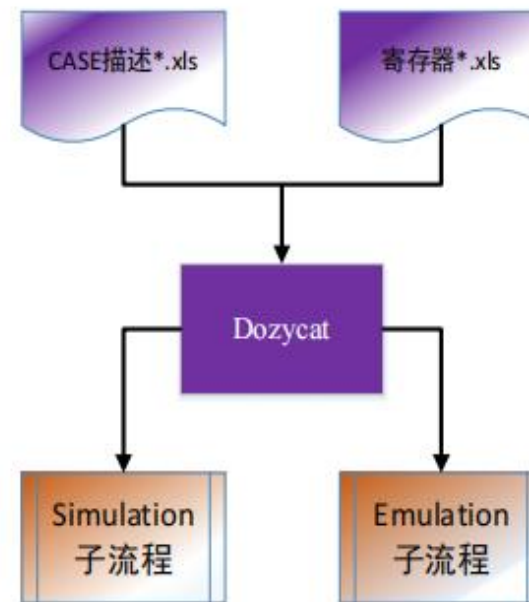
成果简介

用于两款芯片验证流程



- ❑ Soc相关的case都用excel文件形式呈现；
- ❑ 结合寄存器列表；
- ❑ 用Dozycat转换生成执行文件；
- ❑ 加载到simu或emu中运行。

1. 该方法使验证人员对case的运行时间不感知，而专注于case本身，对于多模块共同协作的场景有很大帮助。
2. 验证团队内部可解决，不需要软件团队参与。
3. 兼顾低功耗验证、门级网表后仿、功耗评估、性能测试等场景。





谢谢

白盒子（上海）微电子科技有限公司
