



# ENCORE: Efficient Architecture Verification Framework with FPGA Acceleration

## 基于FPGA加速的处理器敏捷验证框架

石侃, 徐烁翔, 刁予涵, David Boland, 张子卿, 包云岗



中国科学院计算技术研究所  
Institute of Computing Technology, Chinese Academy of Sciences



上海科技大学  
ShanghaiTech University

Imperial College  
London



THE UNIVERSITY OF  
SYDNEY



中国科学院大学  
University of Chinese Academy of Sciences

# 芯片开发的瓶颈——验证

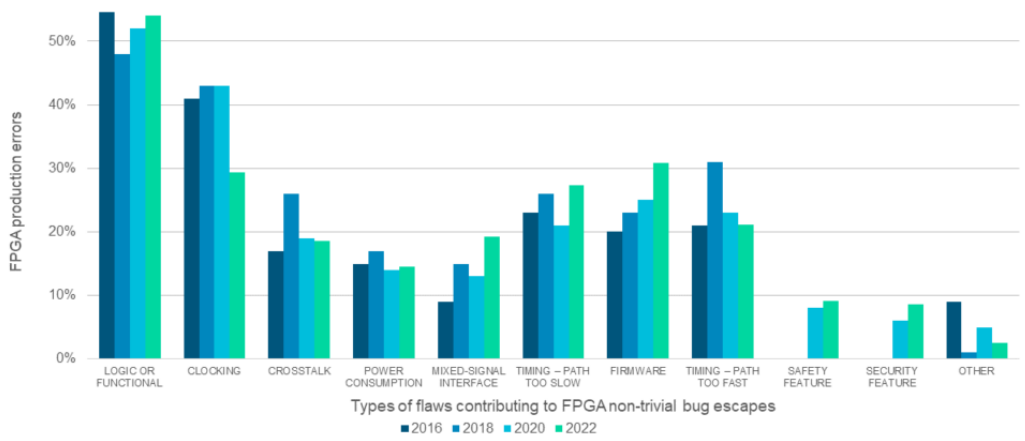
- 芯片功能验证极其重要:

- 验证过程占据了整个芯片开发周期高达70%的时间，验证投入巨大

The SoC must be verified for validation before being manufactured in a foundry, in a process known as functional verification. The verification process accounts for a large portion of the chip design life cycle —up to 70%. Once the chip is fully verified to work as intended, it is sent to the foundry for mass production.

—— Synopsys

Causes of FPGA non-trivial bug escapes into production  
Logic/functional failures consistently the top cause of FPGA non-trivial bug escapes



逻辑或功能错误是产生BUG最多的错误类型

# 芯片开发的瓶颈——验证

- 芯片验证极其重要:

- 验证过程占据了整个芯片开发周期高达70%的时间，验证投入巨大

The SoC must be verified for validation before being manufactured in a foundry, in a process known as functional verification. The verification process accounts for a large portion of the chip design life cycle —up to 70%. Once the chip is fully verified to work as intended, it is sent to the foundry for mass production.

—— Synopsys

- 芯片验证非常困难:

- 现有的验证方法:

随机约束测试,通用验证方法学UVM, 形式化验证, FPGA原型验证, 仿真加速器...

- 性能受限，成本高昂，调试能力不足

# 芯片开发的瓶颈——验证

- 芯片验证需要做出改变：

- 芯片功能验证需要一种新的非传统，更智能，更具性价比的方法

Verifying chip-level functionality can reduce verification time and identify glitches in the design that would have led to costly errors. Therefore, opting for an **alternative, smarter, and cost-effective SoC verification** strategy is essential, as last-stage testing is not a suitable option.

—— Cadence

# 验证的三角悖论

**高性能**

是否能进行高效验证



**良好的调试能力**

是否能容易定位Bug

**低成本**

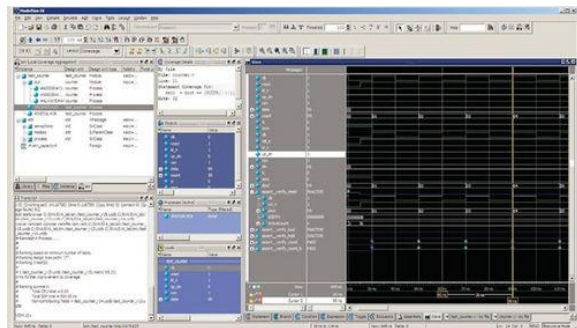
是否是在可接受范围内

# 验证的三角悖论

高性能



良好的调试能力



低成本

仿真软件：如Modelsim,  
VCS, Verilator  
通过观察波形进行调试

# 验证的三角悖论

高性能



FPGA原型验证  
借助FPGA厂商提  
供的验证工具

良好的调试能力



低成本

# 验证的三角悖论

高性能

Emulator  
硬件仿真器



良好的调试能力



低成本



# 新验证架构

性能



FPGA 原型验证



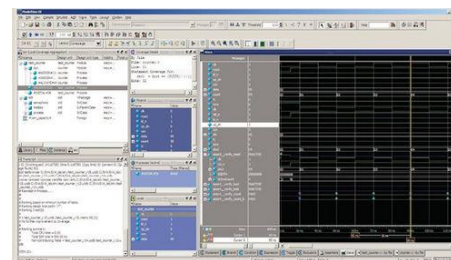
\$10k ~ \$100k

仿真加速器



\$~1M

软件仿真器

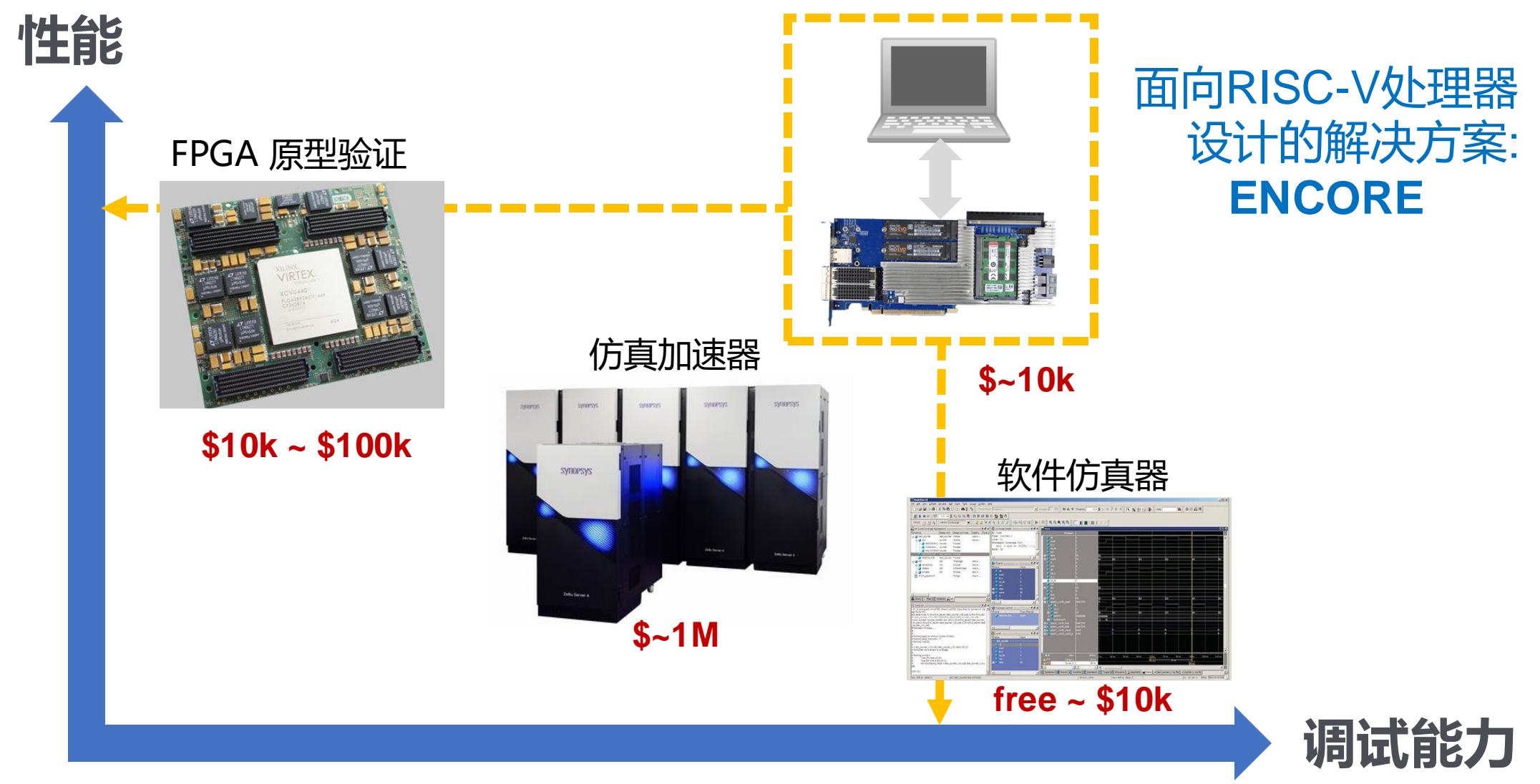


free ~ \$10k

调试能力



# 新验证架构



- **ENCORE概述:**
  - 硬件部分与软件参考模型
  - 整体架构
- **ENCORE的工作流程**
- **ENCORE调试示例**
- **性能测试**
- **小结**

RESEARCH-ARTICLE | OPEN ACCESS



## ENCORE: Efficient Architecture Verification Framework with FPGA Acceleration

Authors: [Kan Shi](#), [Shuoxiang Xu](#), [Yuhan Diao](#), [David Boland](#), [Yungang Bao](#) | [Authors Info & Claims](#)

FPGA '23: Proceedings of the 2023 ACM/SIGDA International Symposium on Field Programmable Gate Arrays • Pages 209 - 219  
<https://doi.org/10.1145/3543622.3573187>

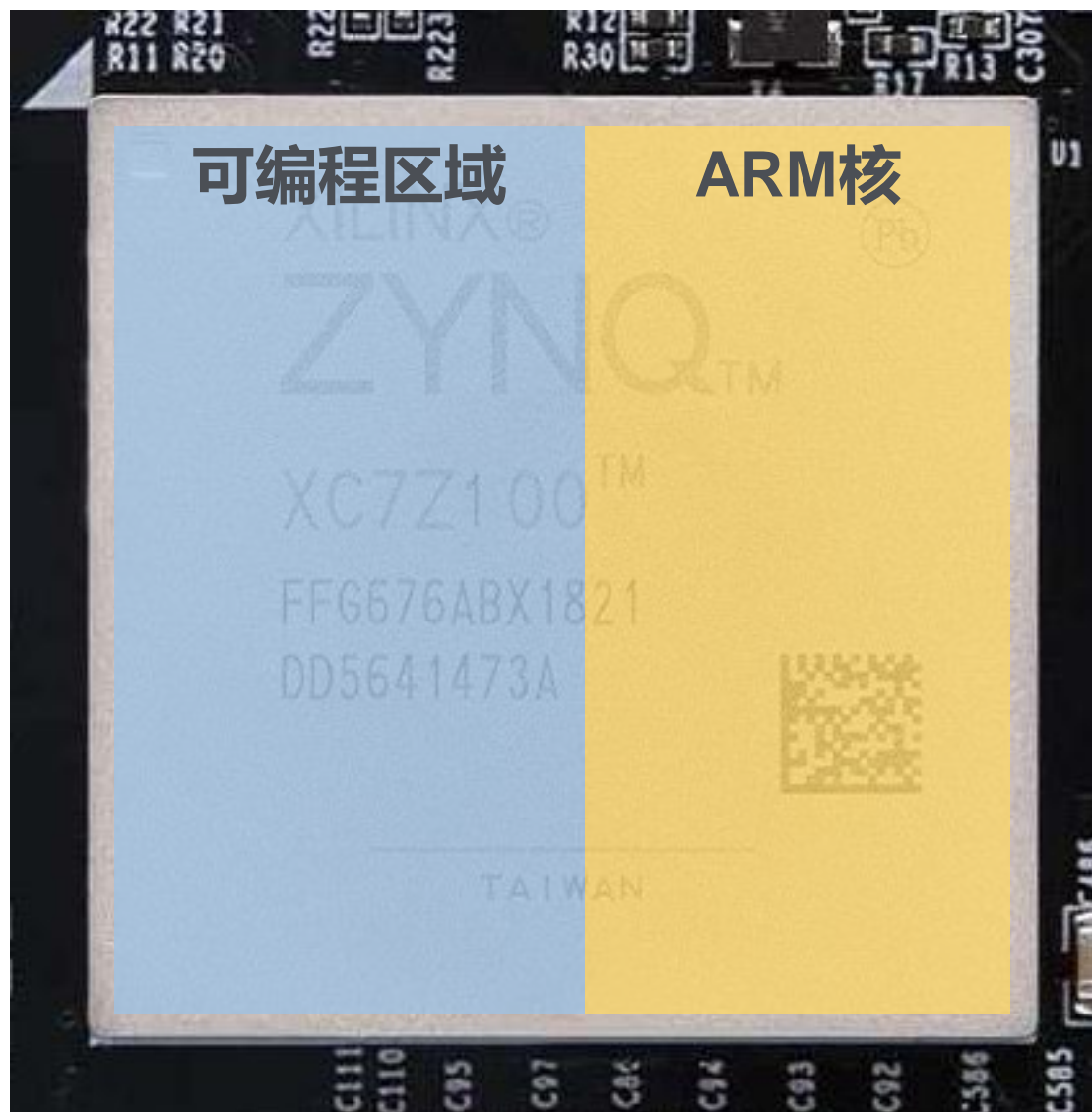
Published: 12 February 2023 [Publication History](#)



5 2,226



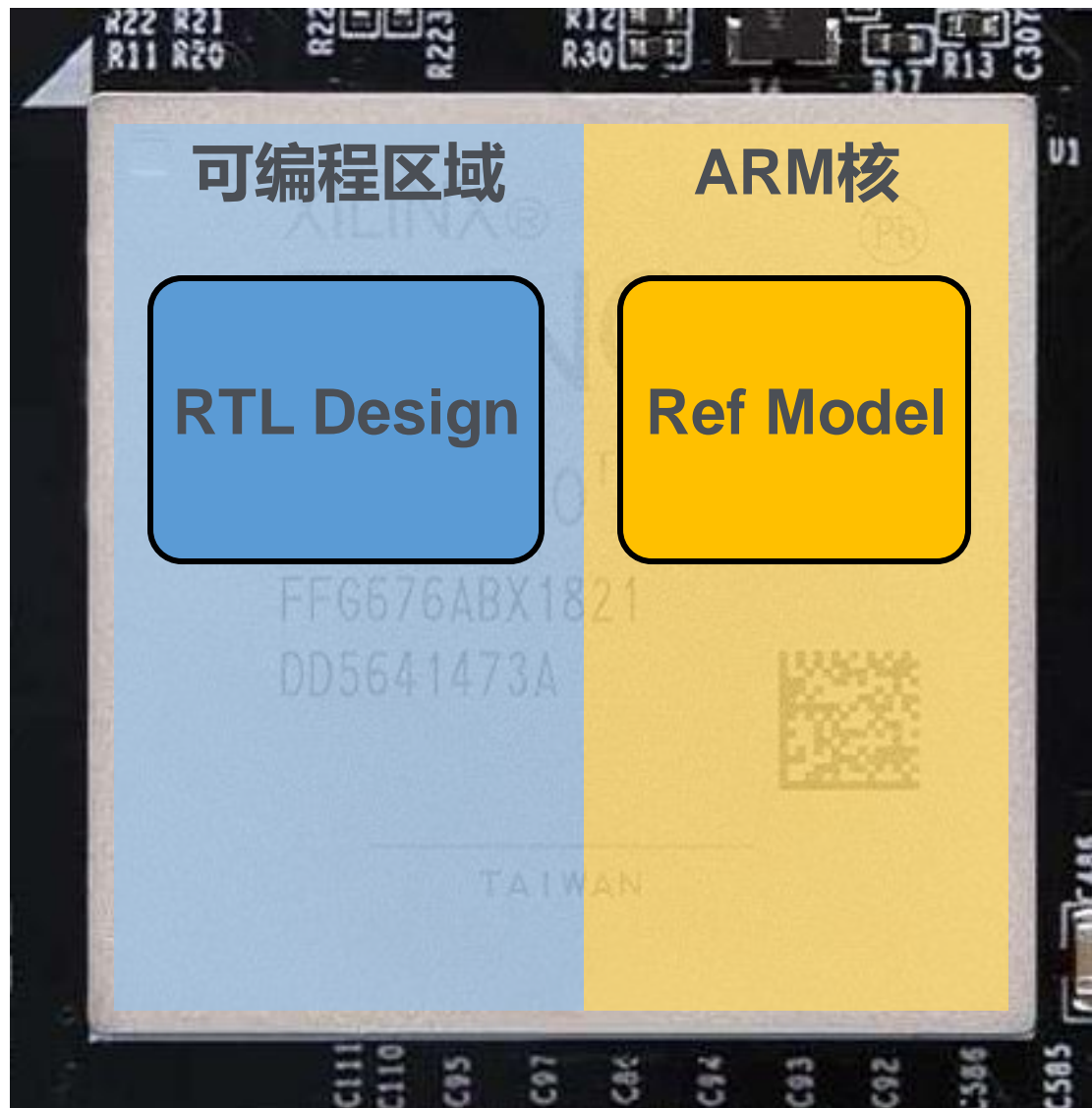
# ENCORE: 硬件架构



## • 验证框架ENCORE:

- 以处理器验证为目标
- 使用FPGA加速验证
- 有良好的调试能力

# ENCORE: 硬件架构

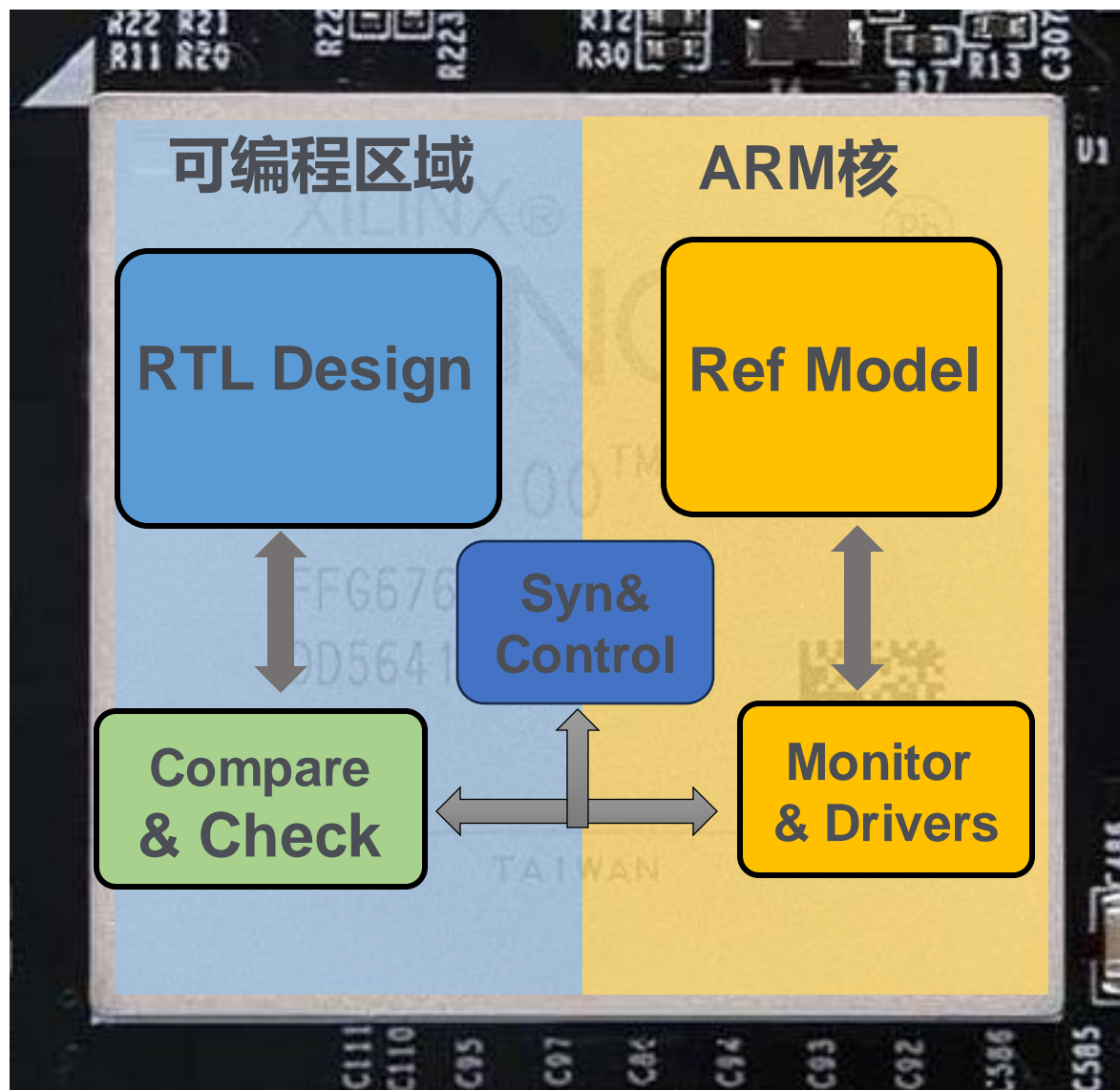


## • 验证框架ENCORE:

- 以处理器验证为目标
- 使用FPGA加速验证
- 有良好的调试能力



# ENCORE: 硬件架构



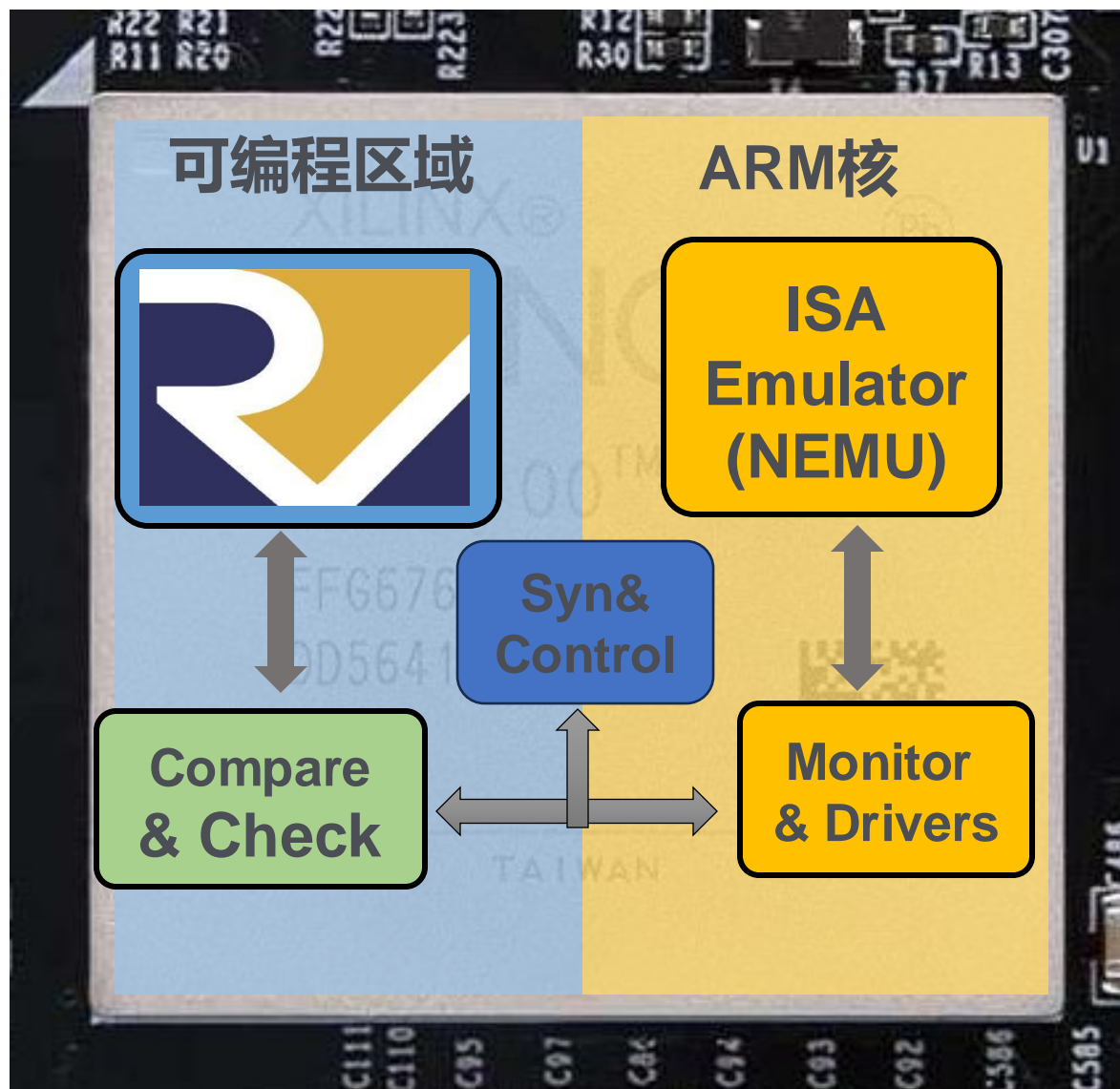
## • 验证框架ENCORE:

- 以处理器验证为目标
- 使用FPGA加速验证
- 有良好的调试能力

## • 自动化检查:

- 软硬件同步对比指令集架构 (ISA) 级信号。如PC, GPR, CSR

# ENCORE: 硬件架构



## • 验证框架ENCORE:

- 以处理器验证为目标
- 使用FPGA加速验证
- 有良好的调试能力

## • 自动化检查:

- 软硬件同步对比指令集架构 (ISA) 级信号。如PC, GPR, CSR

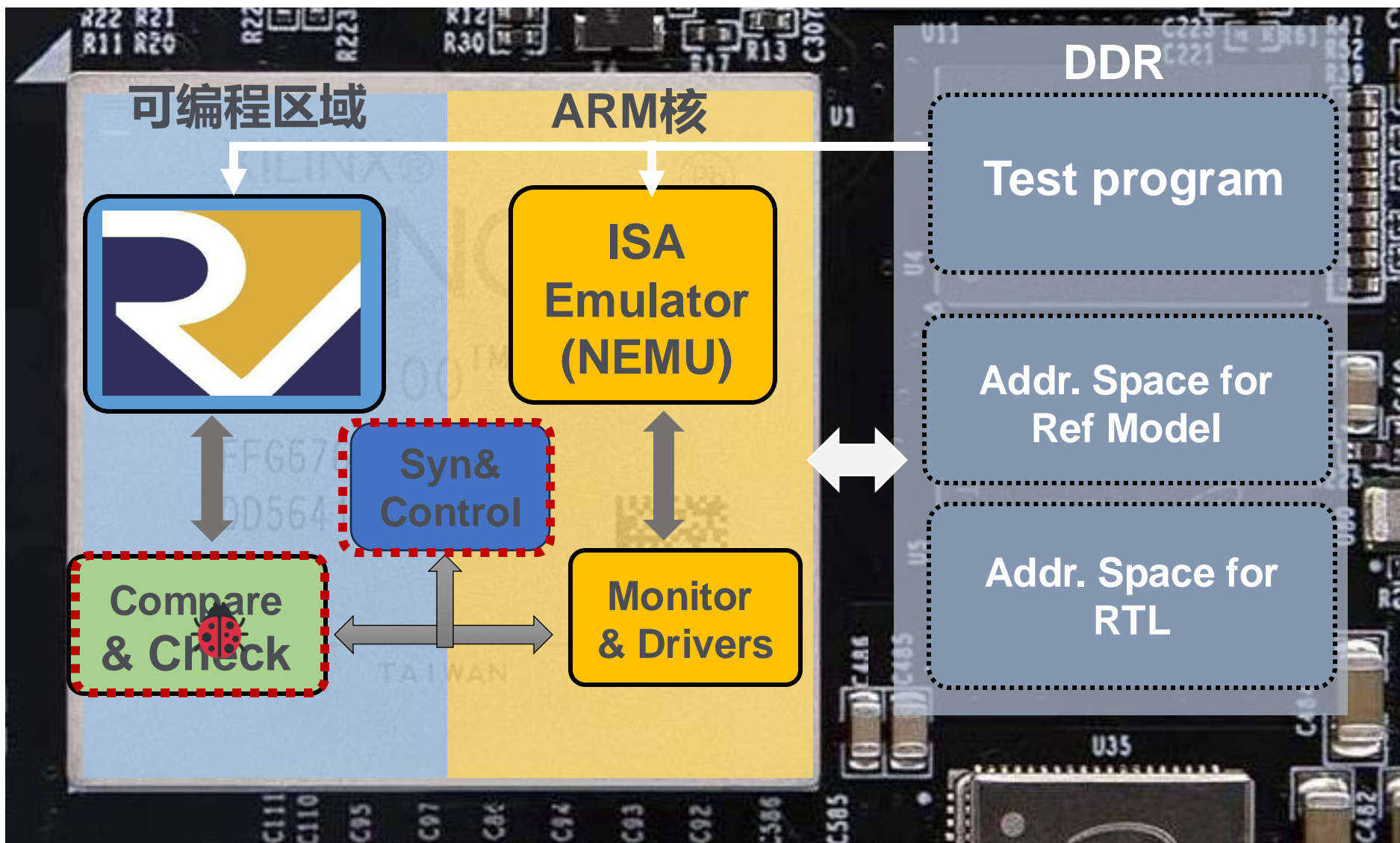
## • 示例:

- 可编程区域: RISC-V处理器核
- SoC硬核上运行: RISC-V ISA 模拟器

Nutshell: <https://github.com/OSCPU/NutShell>

NEMU: <https://github.com/NJU-ProjectN/nemu>

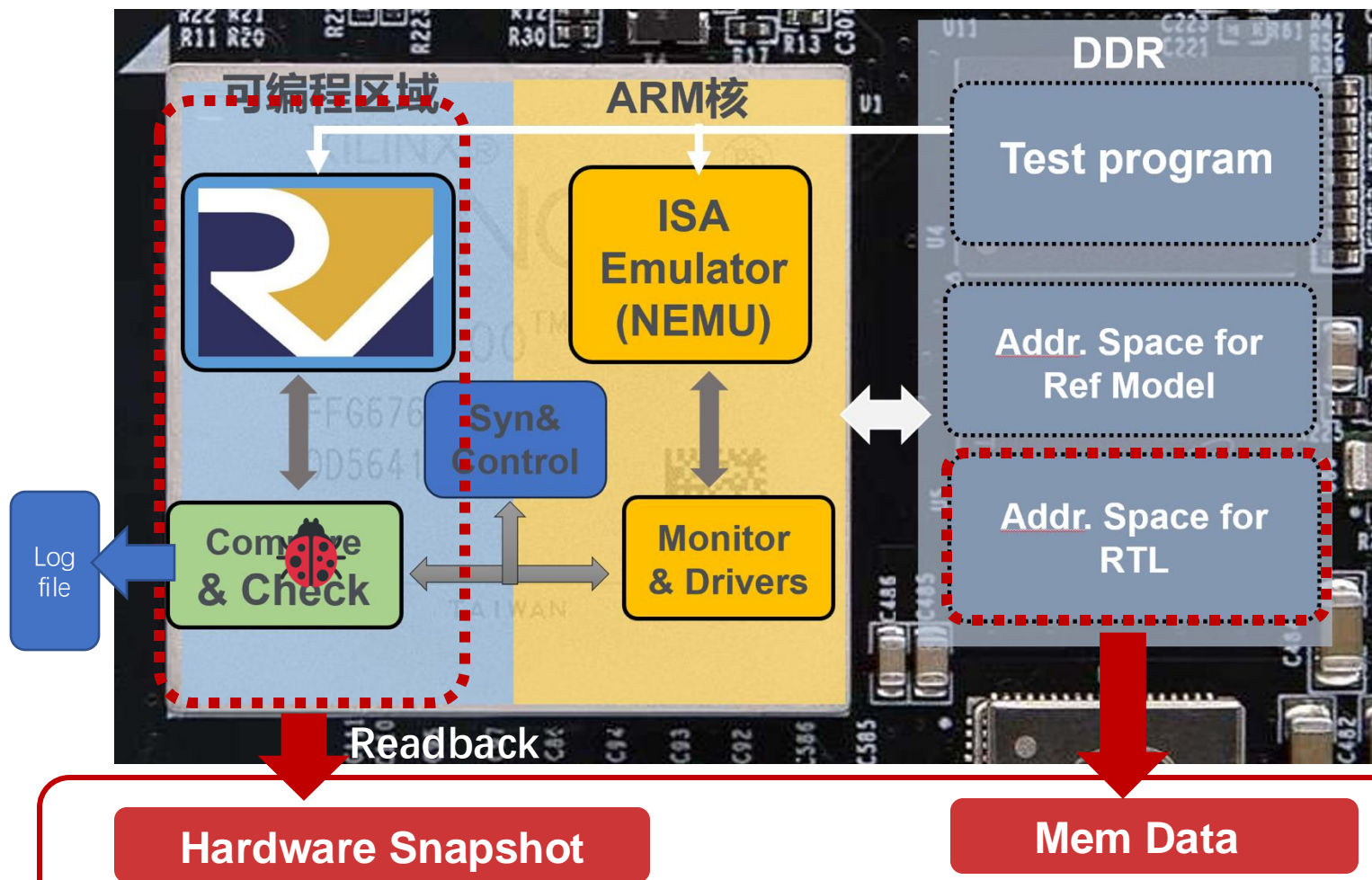
# ENCORE: 外部存储器



- **检查机制:** 对指令运行对ISA级处理器状态做出的影响进行比较。
- **同步机制:** 保证待测设计和ISA模拟器之间运行的是相同指令。

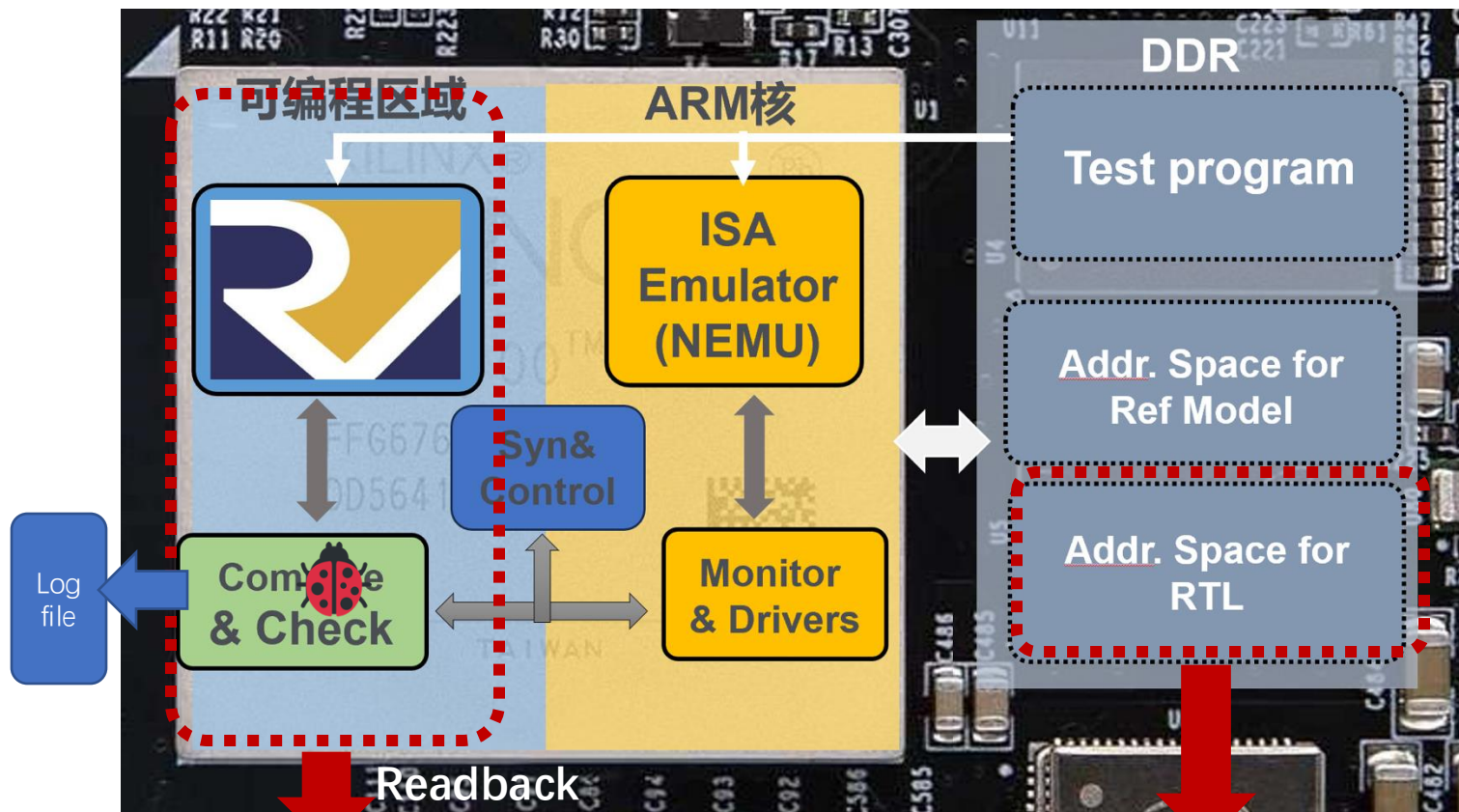


# ENCORE: 后端仿真



- ENCORE 使用适配并优化后的 StateMover 工具作为仿真后端
- 当对比过程中潜在的错误被检测到时进行硬件快照

# ENCORE: 后端仿真

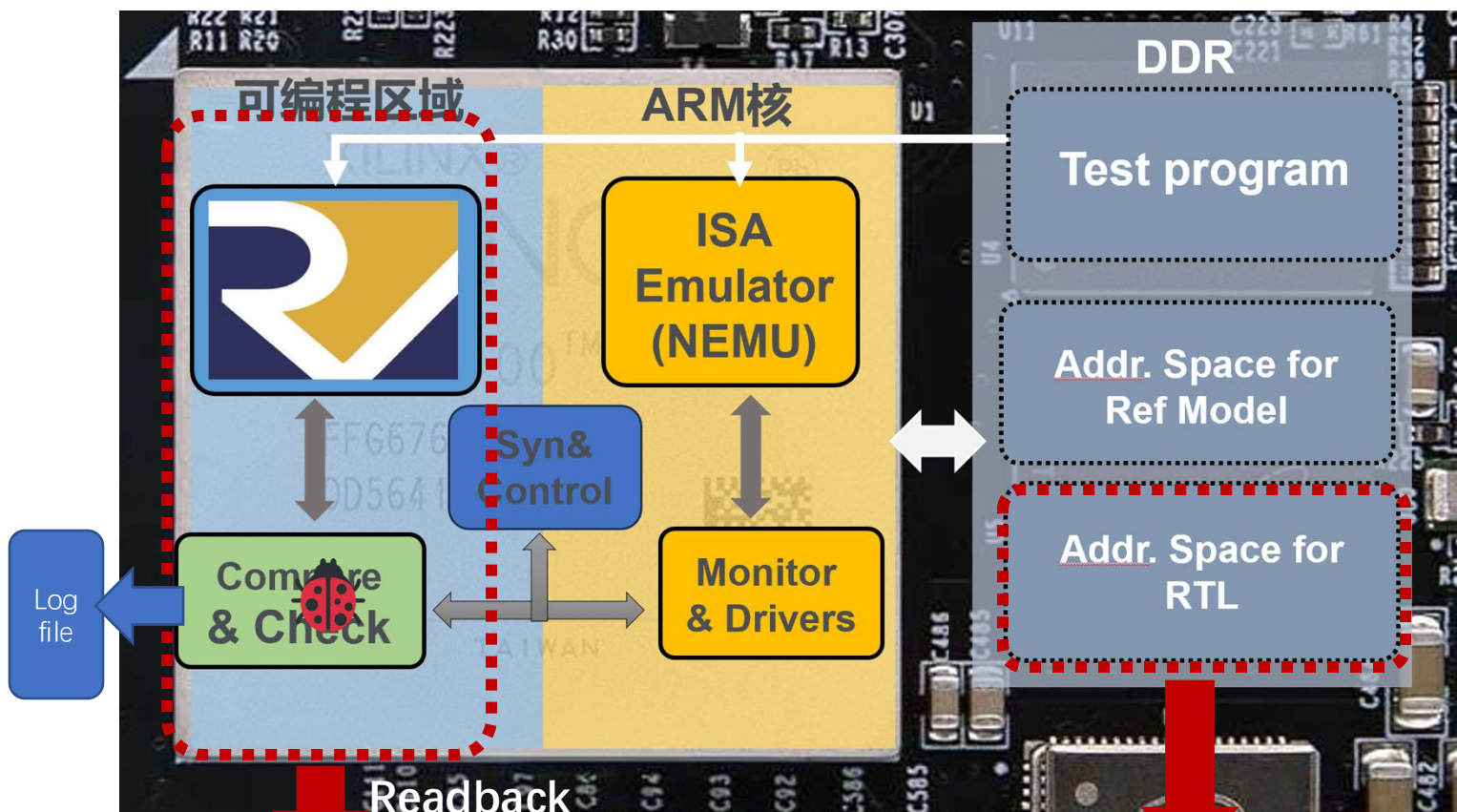


- ENCORE 使用适配并优化后的 StateMover 工具作为仿真后端
- 当对比过程中潜在的错误被检测到时进行硬件快照

硬件回读后的数据导入到后端仿真工具中并完成仿真重建



# ENCORE: 后端仿真



- ENCORE 使用适配并优化后的 StateMover 工具作为仿真后端
- 当对比过程中潜在的错误被检测到时进行硬件快照
- 扩展部分回读，减少了大量回读时间
- 扩展更多的仿真模型，如外部存储器设备

硬件回读后的数据导入到后端仿真工具中并完成仿真重建

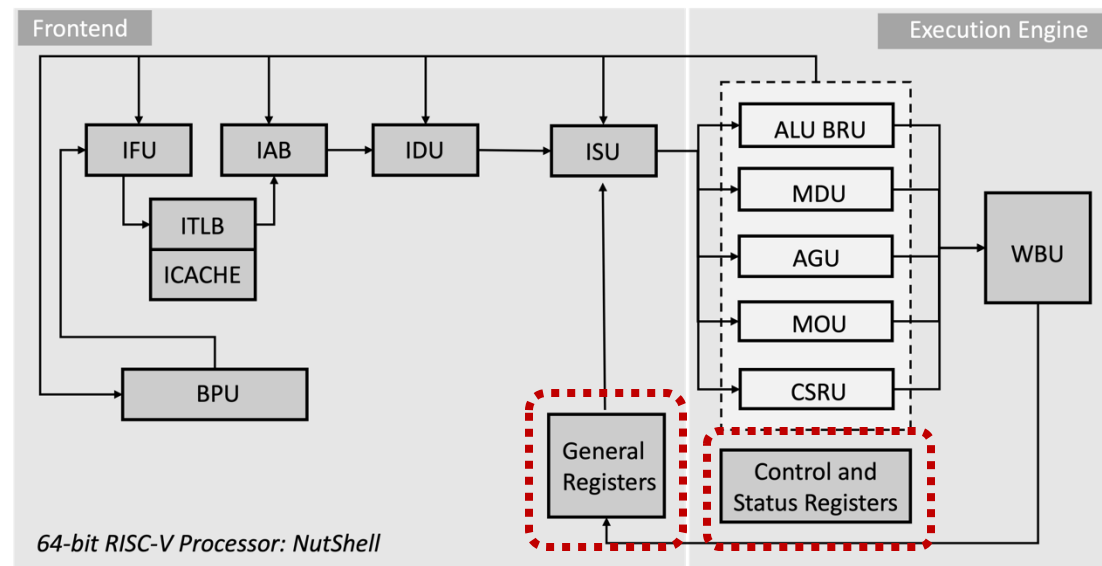
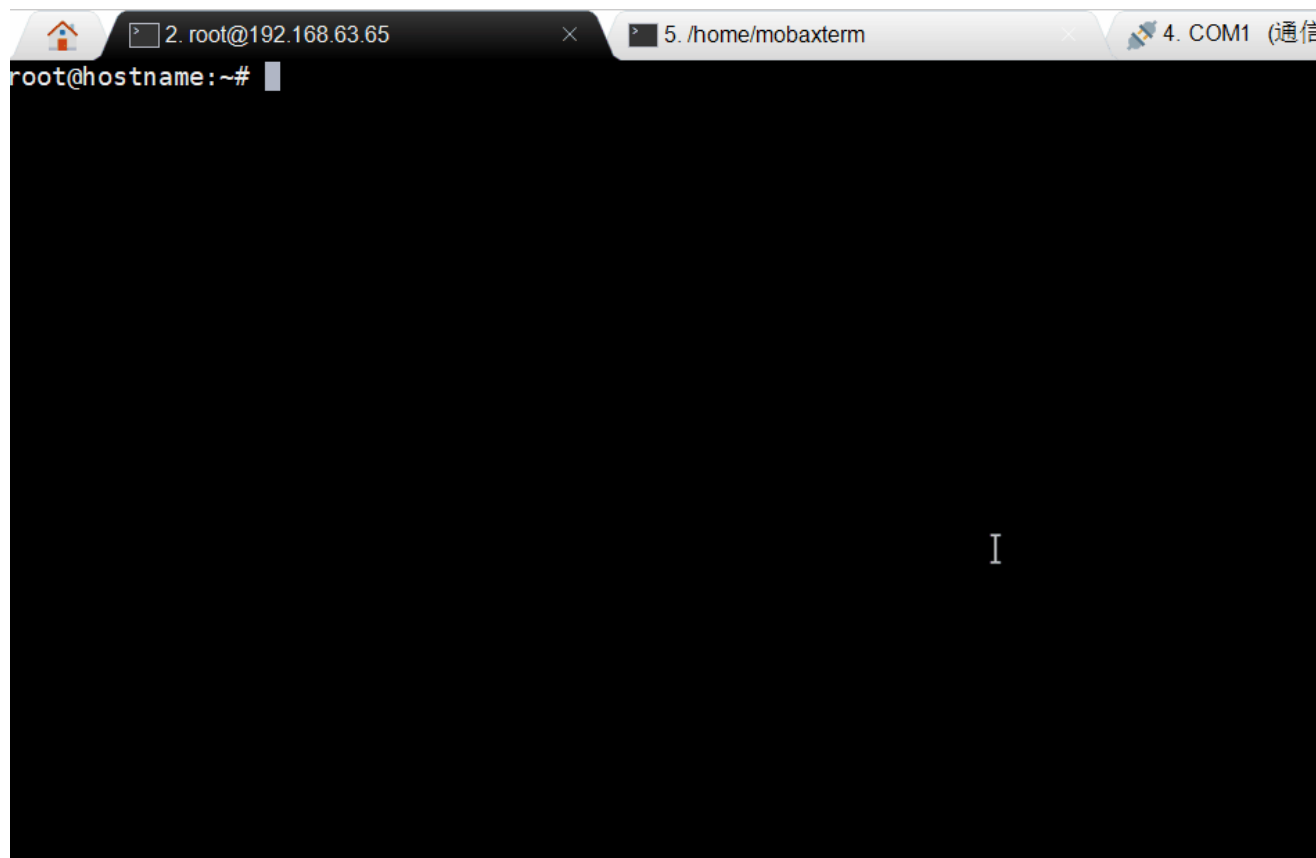


- ENCORE概述
- ENCORE的工作流程
- **ENCORE调试示例**
  - 64位RISC-V处理器设计
  - 调试过程
- 性能测试
- 小结



# ENCORE调试示例

- 使用开源64位RISC-V处理器果壳作为待测设计
- <https://github.com/OSCPU/NutShell>

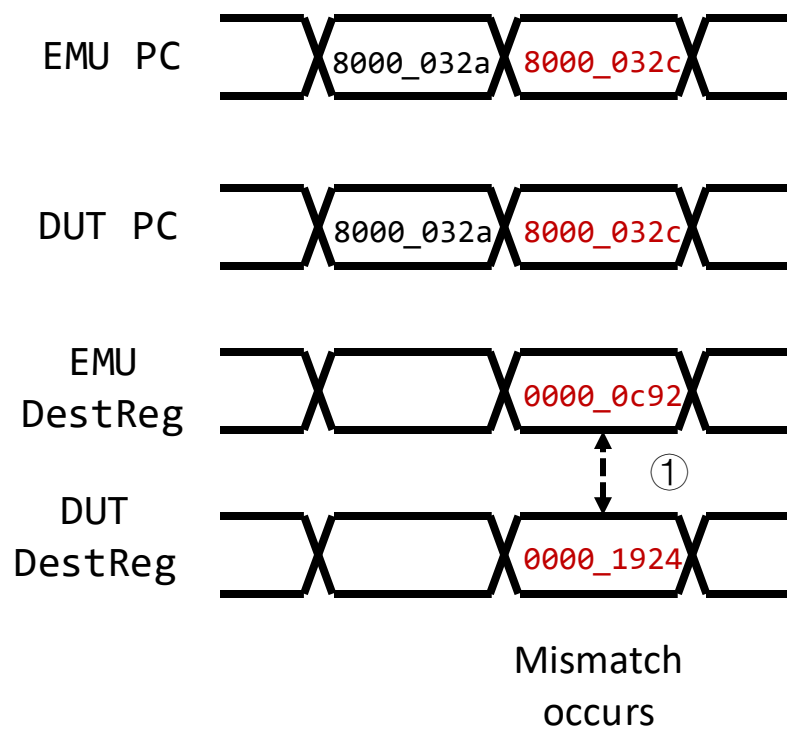


ENCORE中监测这些  
关键寄存器的值

# ENCORE调试示例

## • 比较PC和目的寄存器的值

- ENCORE可以配置为比较更多的寄存器，如浮点寄存器，CSR寄存器

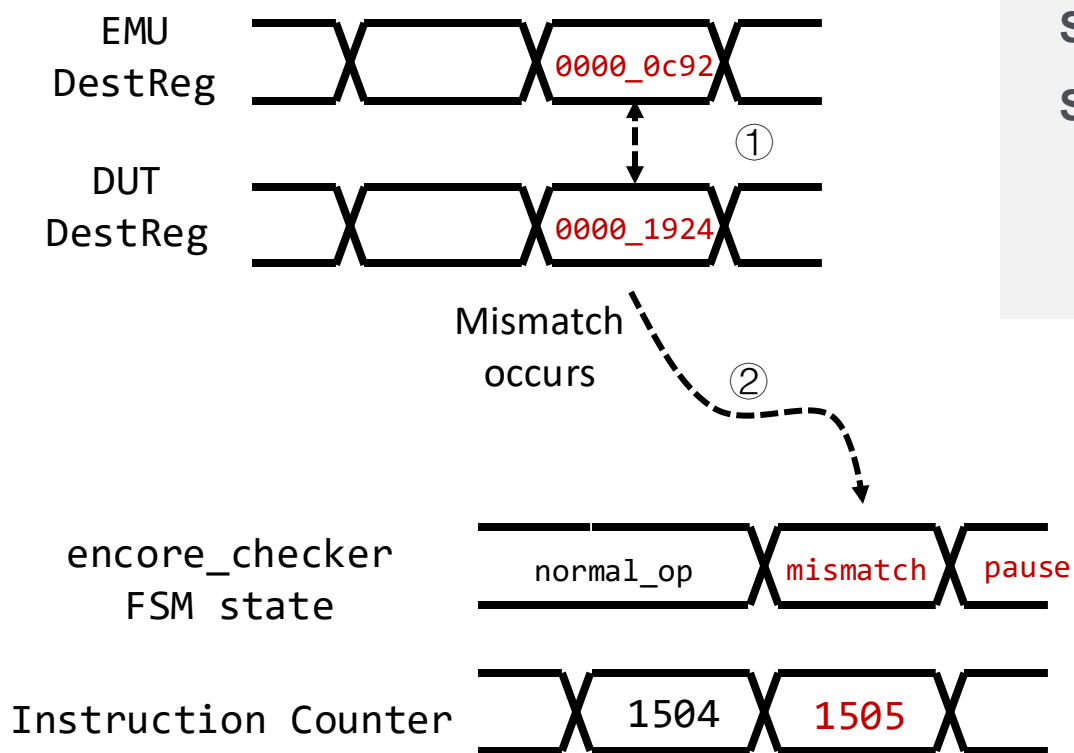


STEP ① • DUT与EMU之间出现状态不匹配的情况

# ENCORE调试示例

## • 比较PC和目的寄存器的值

- ENCORE可以配置为比较更多的寄存器，如浮点寄存器，CSR寄存器

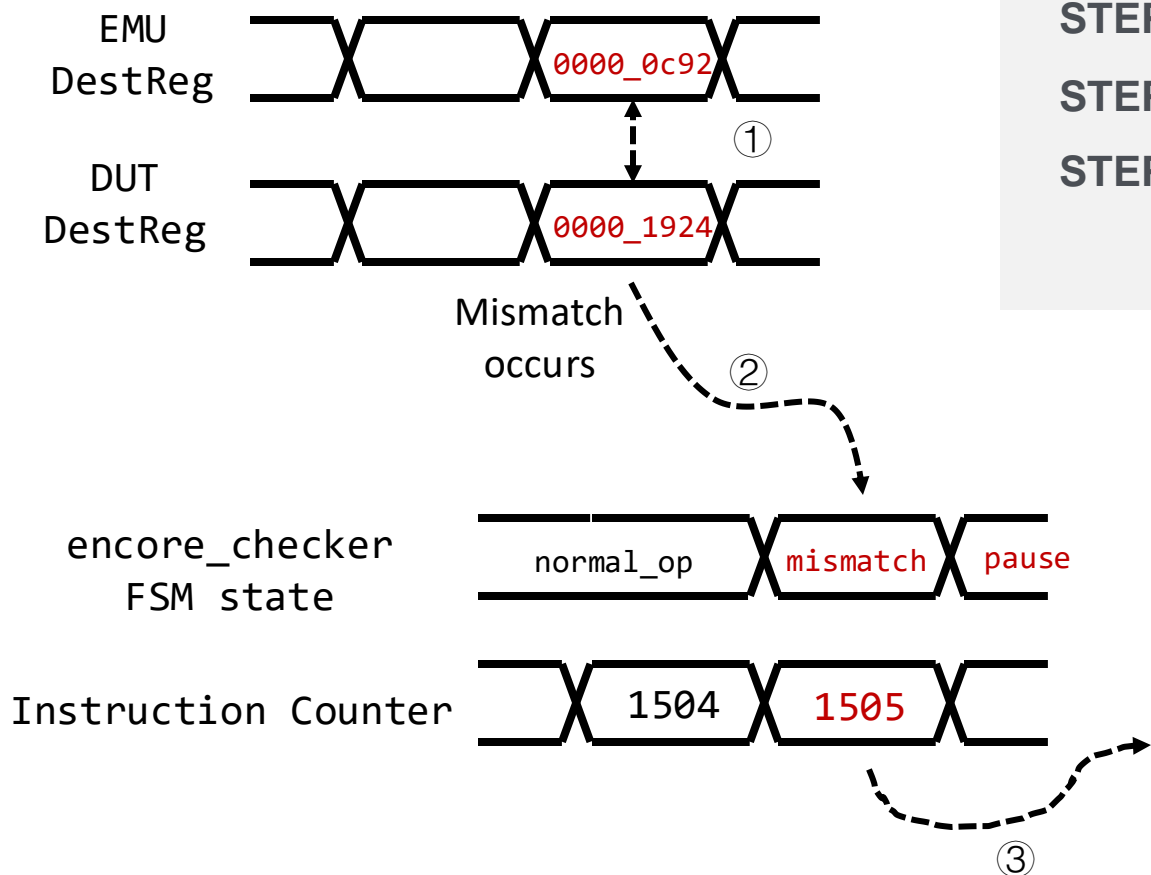


- STEP ①** • DUT与EMU之间出现状态不匹配的情况
- STEP ②** • 硬件快照触发，将指令计数器作为标识符

# ENCORE调试示例

## • 比较PC和目的寄存器的值

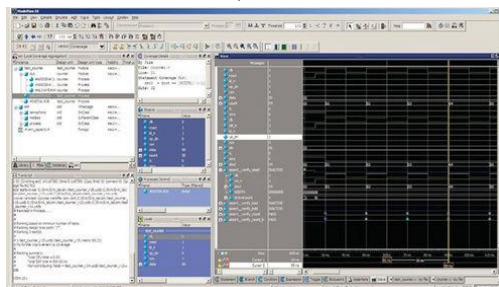
- ENCORE可以配置为比较更多的寄存器，如浮点寄存器，CSR寄存器



- STEP ①** • DUT与EMU之间出现状态不匹配的情况
- STEP ②** • 硬件快照触发，将指令计数器作为标识符
- STEP ③** • 将硬件快照加载进软件仿真器，重建仿真



硬件快照  
↓  
State.dump  
↓  
加载

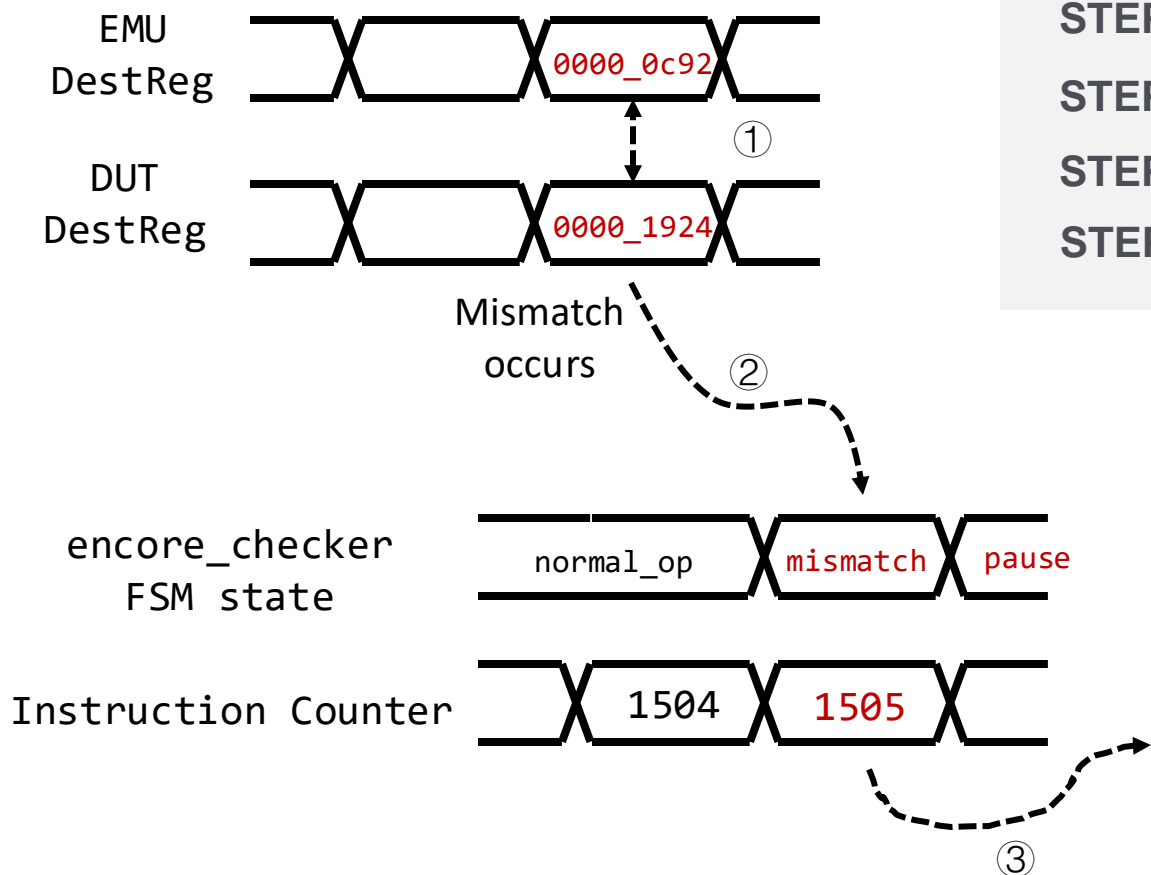




# ENCORE调试示例

## • 比较PC和目的寄存器的值

- ENCORE可以配置为比较更多的寄存器，如浮点寄存器，CSR寄存器



STEP ①

- DUT与EMU之间出现状态不匹配的情况

STEP ②

- 硬件快照触发，将指令计数器作为标识符

STEP ③

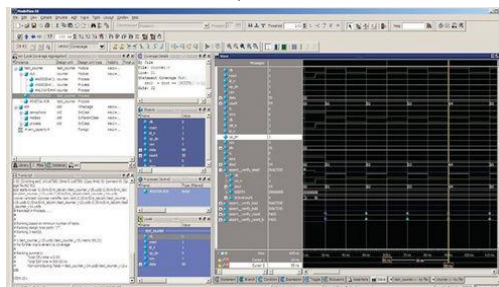
- 将硬件快照加载进软件仿真器，重建仿真

STEP ④

- 仿真器中根据指令计数器定位到出错位置  
如果需要的话根据上下文分析错误的根源



硬件快照  
State.dump  
加载



| #    | PC        | Instruction     |
|------|-----------|-----------------|
| ...  |           |                 |
| 1501 | 8000_1608 | lw a0,12(sp)    |
| 1502 | 8000_160a | addi sp,sp,32   |
| 1503 | 8000_160c | ret             |
| 1504 | 8000_032a | ld a5,0(s0)     |
| 1505 | 8000_032c | slli a4,a0,0x20 |
| ...  |           |                 |

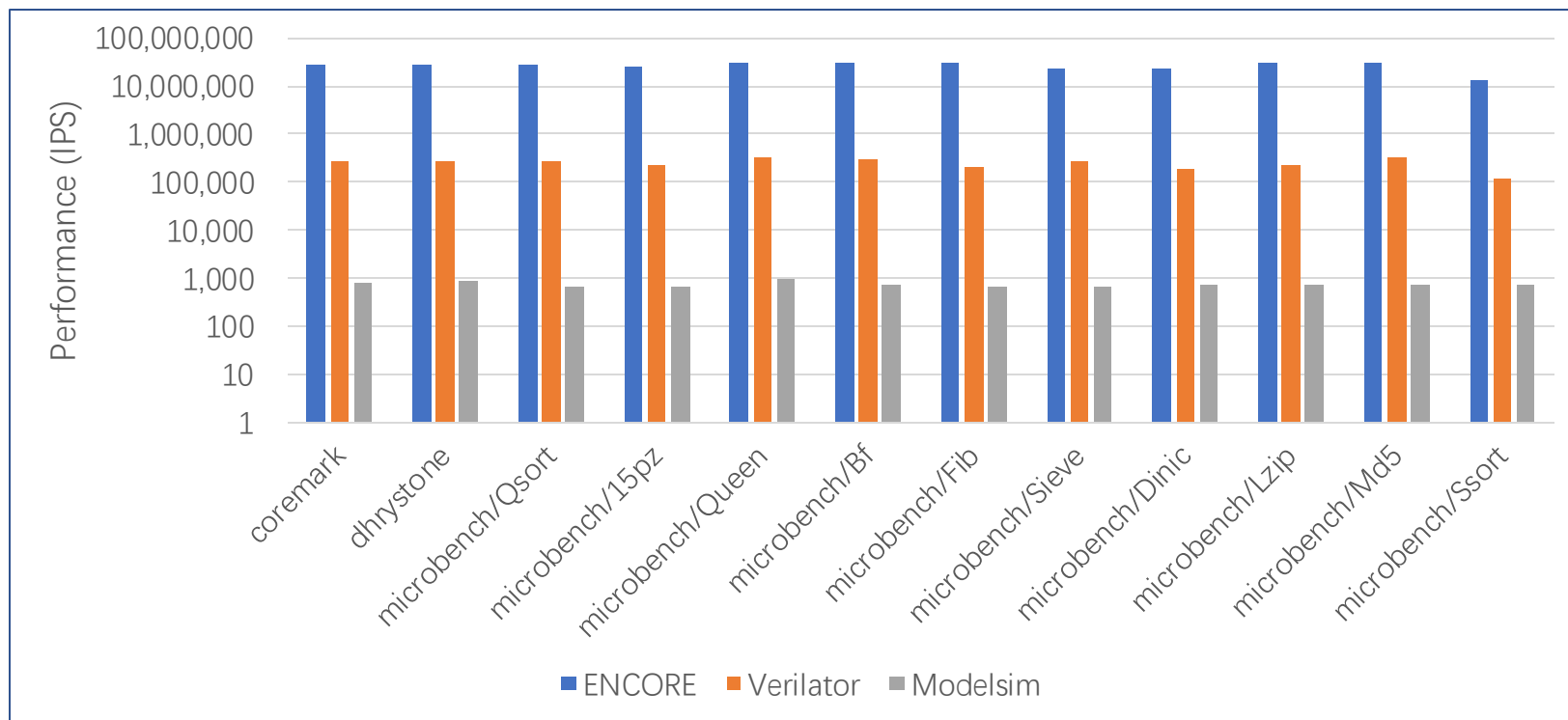
# 概要

- ENCORE概述
- ENCORE的工作流程
- ENCORE调试示例
- **性能测试**
  - **系统性能评估**
- 小结

# 性能评估

- 与软件仿真器比较

- 软件仿真器: 使用 Modelsim or Verilator
- ENCORE 相对 Verilator 加速: 84x ~ 139x
- ENCORE 相对 Modelsim 加速: 19264x ~ 44187x



# 小结

- **ENCORE 是面向处理器核的新型验证框架:**
  - 同时拥有FPGA硬件加速验证以及软件仿真的良好调试能力
- **ENCORE 能够在单个FPGA SoC芯片上进行自动检查能力:**
  - 单个FPGA同时运行硬件待测设计与软件参考模型
  - 进行在线的状态检查
  - 检查出错时自动进行硬件快照并加载进仿真器中方便进行调试
- **ENCORE 系统高性能及高可调试能力:**
  - ENCORE架构中果壳处理器仿真速度: 20MIPS
  - 具有全信号可见能力, 能进行周期精确调试
  - 较低的成本及资源开销

Paper Title: FPGA '23

**ENCORE: Efficient Architecture Verification Framework with FPGA Acceleration**

联系方式:

石侃 [shikan@ict.ac.cn](mailto:shikan@ict.ac.cn)

徐烁翔 [xushx2022@shanghaitech.edu.cn](mailto:xushx2022@shanghaitech.edu.cn)