



# Heterogeneous Verification to Accelerate the Innovative SoC Design

S2C | Jimmy & Alex

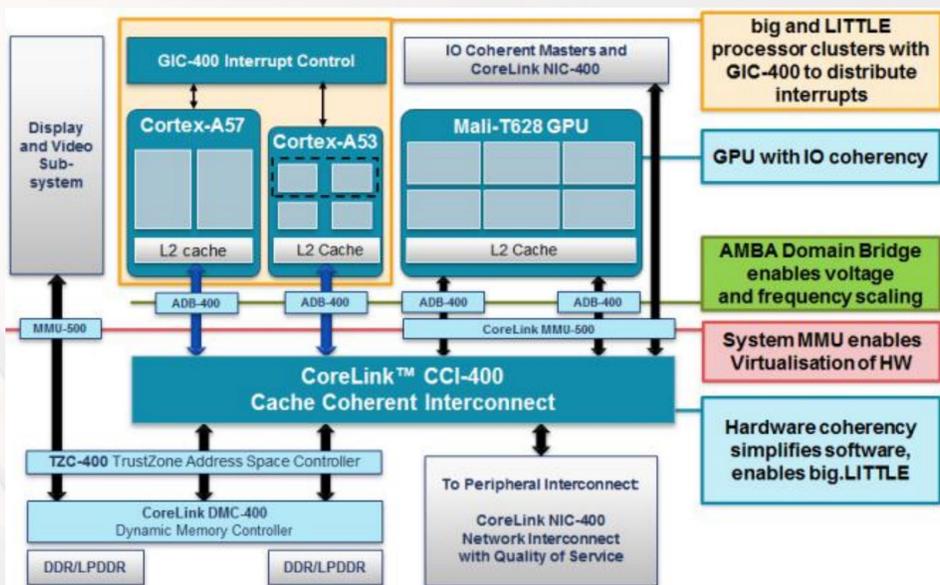
2023.09



# 先进节点SoC的特点

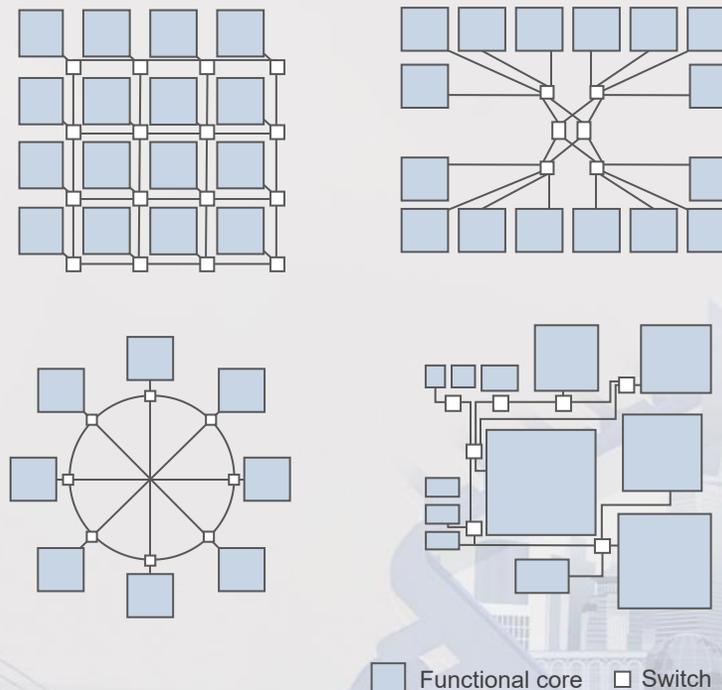
## 复杂的多核CPU

- CPU/GPU/NPU
- 多核CPU调试
- 多种CPU Spec测试

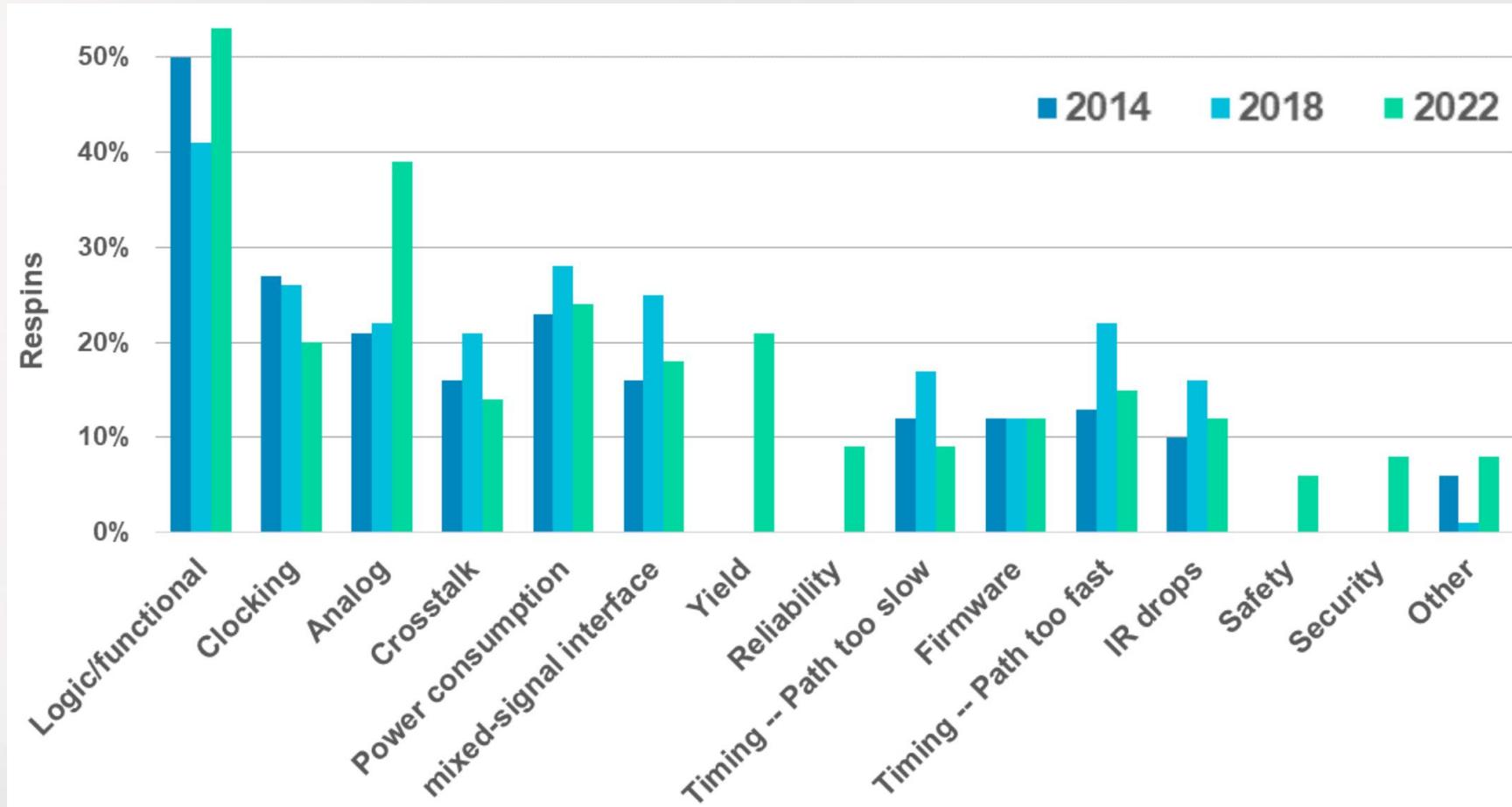


## 复杂的系统拓扑

- 互连线多而复杂
- 高速接口/高吞吐量
- NoC

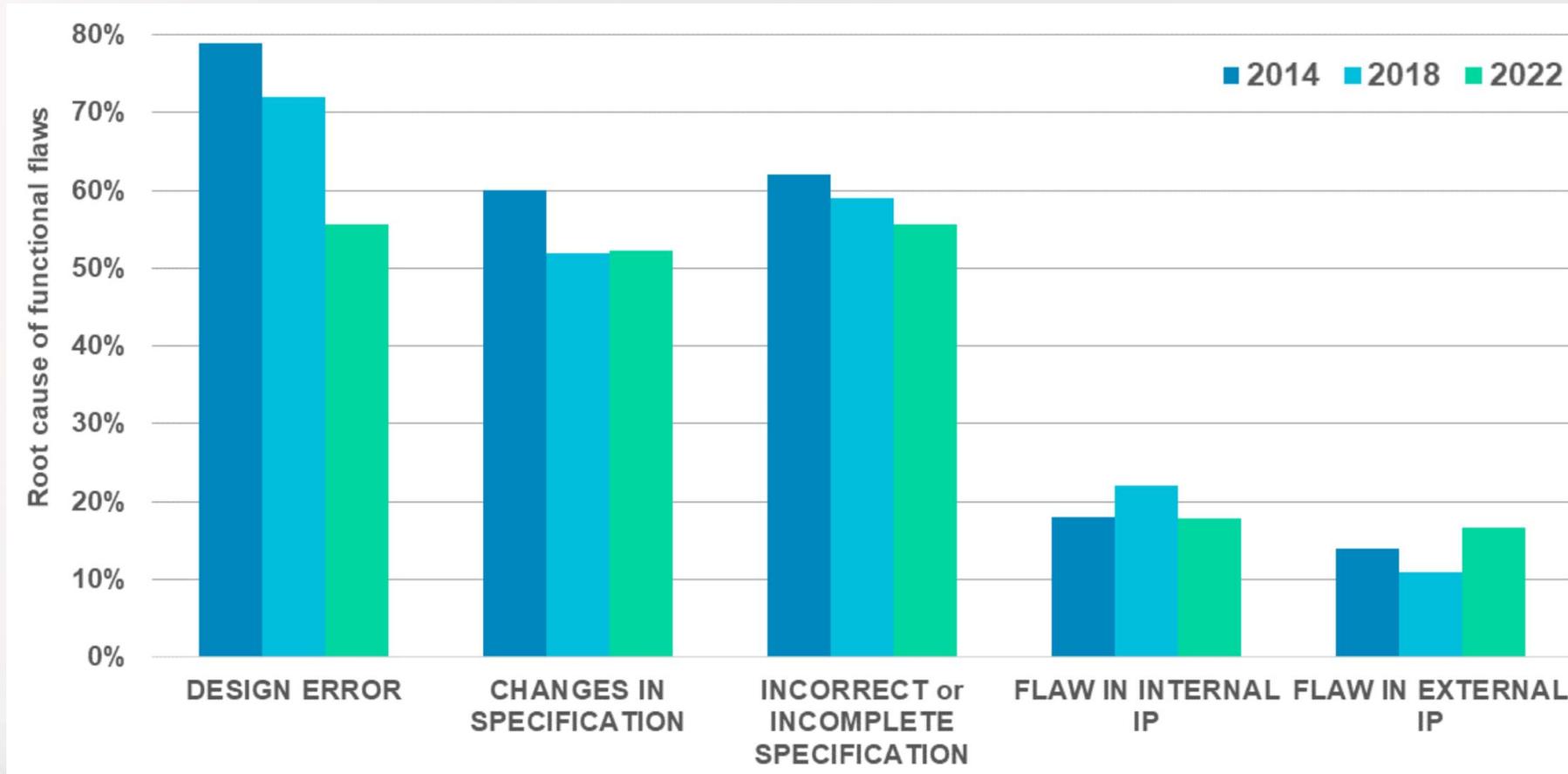


# 流片失败的主要因素



Source: 2022 Wilson Research Group IC/ASIC Functional Verification Trends

# 功能缺陷的主要因素



Source: 2022 Wilson Research Group IC/ASIC Functional Verification Trends

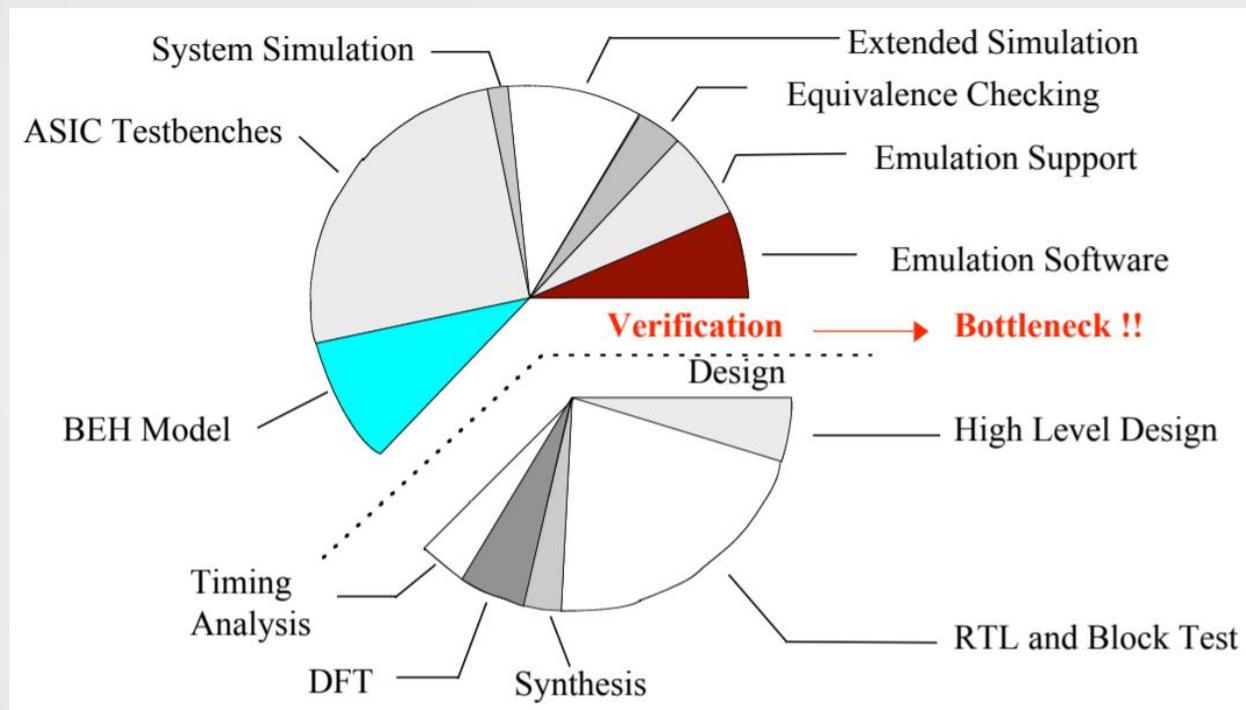
# 芯片验证的压力

## 芯片设计成本日益高涨

- 芯片规模不断变大 (~百亿逻辑门)
- 软件内容持续增多 (多核, 异构核)
- 系统测试复杂费时 (覆盖率&软件)

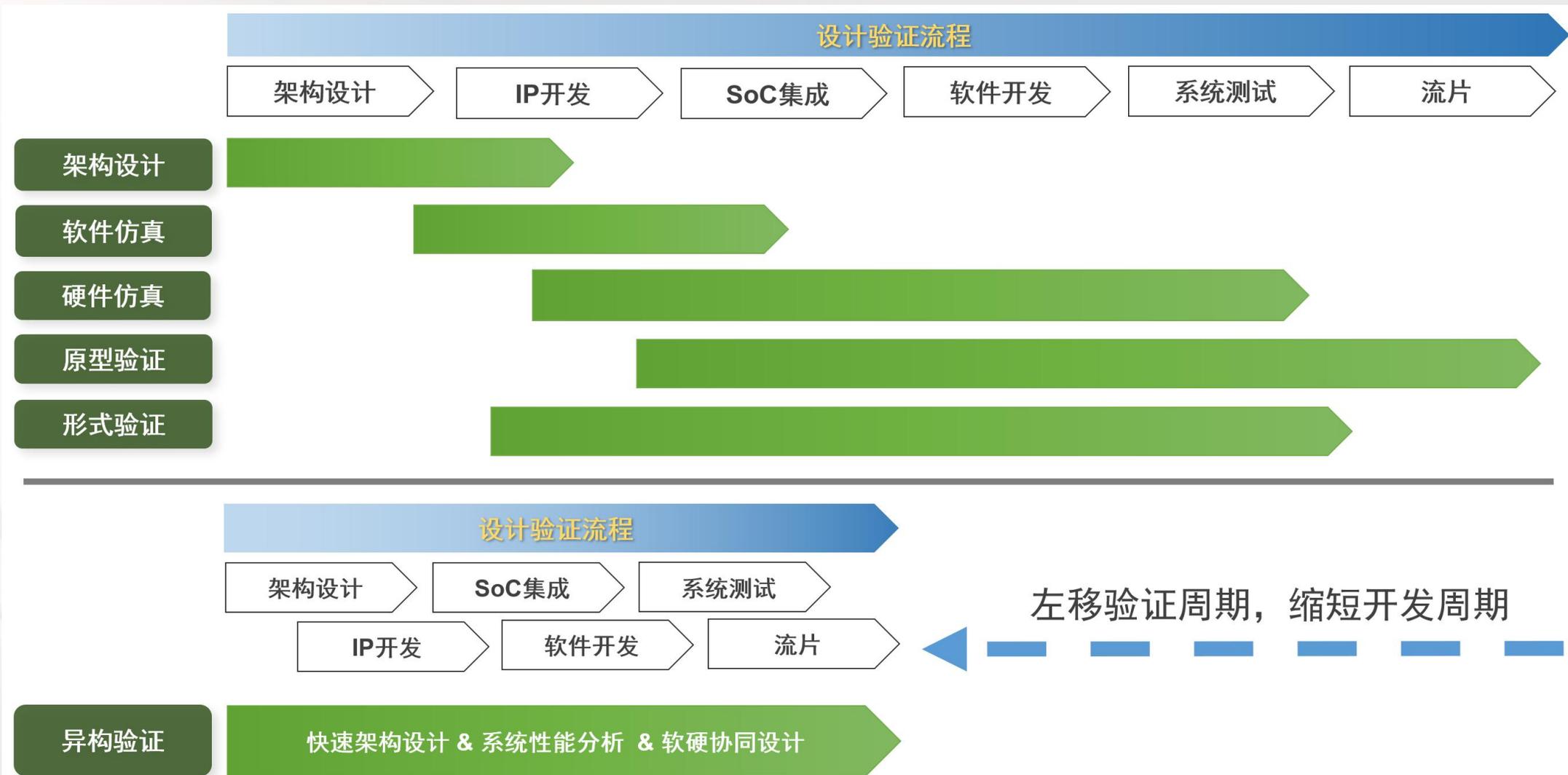
## 系统级芯片验证极其复杂

- 逻辑设计错误是导致流片失败的第一因素
- 验证和仿真约占7成左右的研发时间
- 不仅要确定芯片设计正确, 还要确定设计正确的芯片

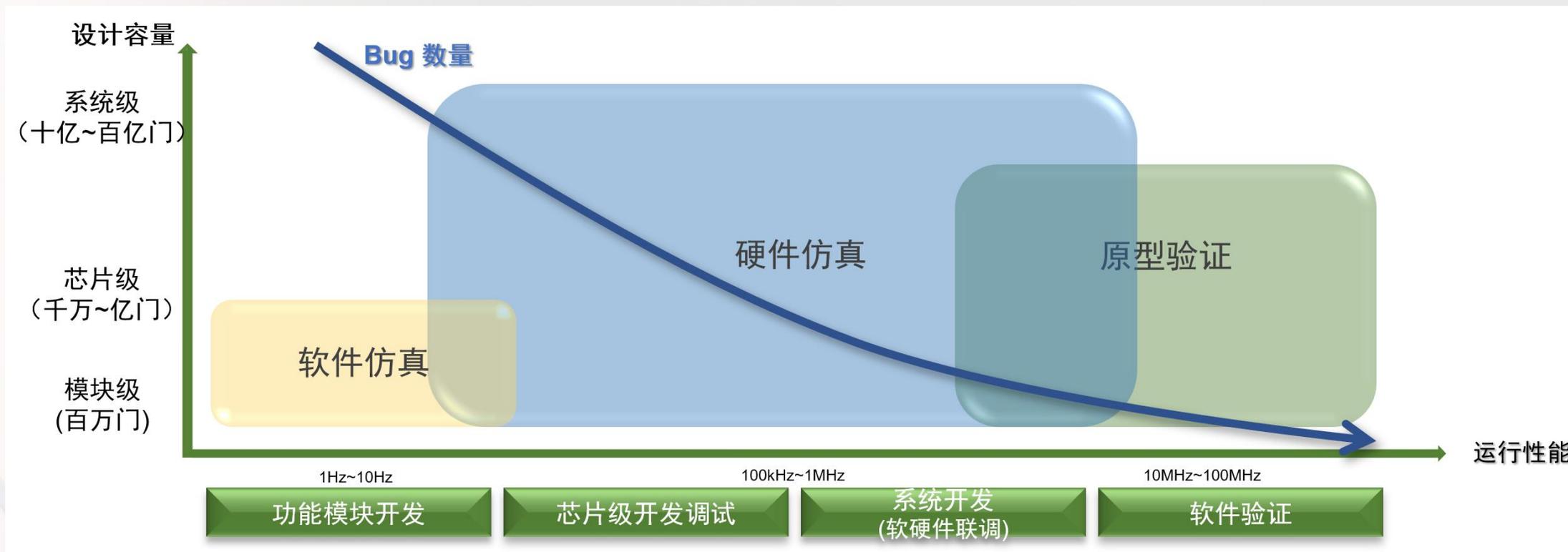


功能验证和提高验证效率非常重要!

# 异构验证方法学



# 验证工具对比



## 软件仿真

- 方便易用
- 仿真速度慢
- 适合 IP / 模块级仿真



## 硬件仿真

- 设计容量大
- 调试能力强, 调试模式多样
- 适合大型设计从模块级、芯片级到系统级的仿真验证



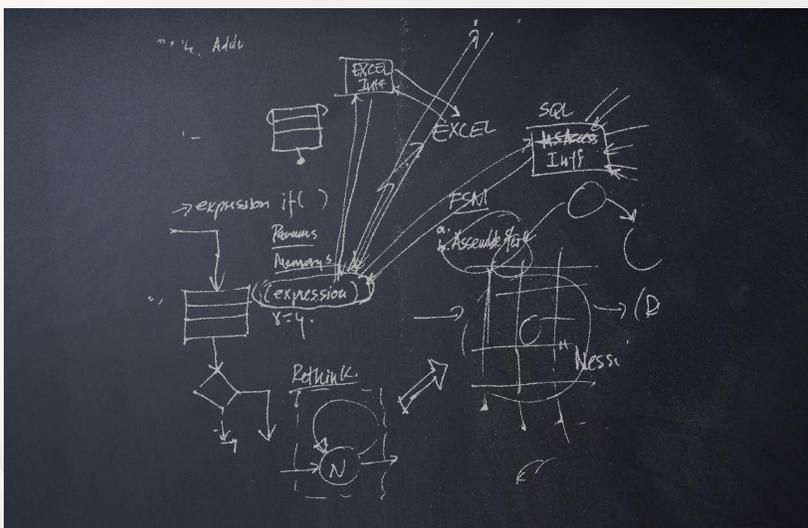
## 原型验证

- 运行速度快
- 调试能力较弱
- 适合系统级验证和软件开发

# 架构分析工具 - Genesis

## 市场需求：

随着设计的规模不断增大，容易由于各子系统之间相互协调不好导致系统失效；半导体与系统公司之间也需要更有效地合作。



过去的系统设计形式

## 解决方案：

- 进行架构探索、优化、验证和交流决策
- 实现图形可视化项目的规格设计
- 提供可运行和仿真参数的规格参考



# 架构设计在SoC验证中的应用

## ■ 架构比对 (Architecture comparison)

- 不同数量处理器内核的配置，总线类型，存储的选择及不同的软件搭配
- 不同场景下吞吐率和功耗的测量

## ■ 功能安全分析 (Functional safety analysis)

- 分析硬件、软件、网络、操作系统或者功耗失效后系统的相关处理

## ■ 性能优化 (Performance optimization)

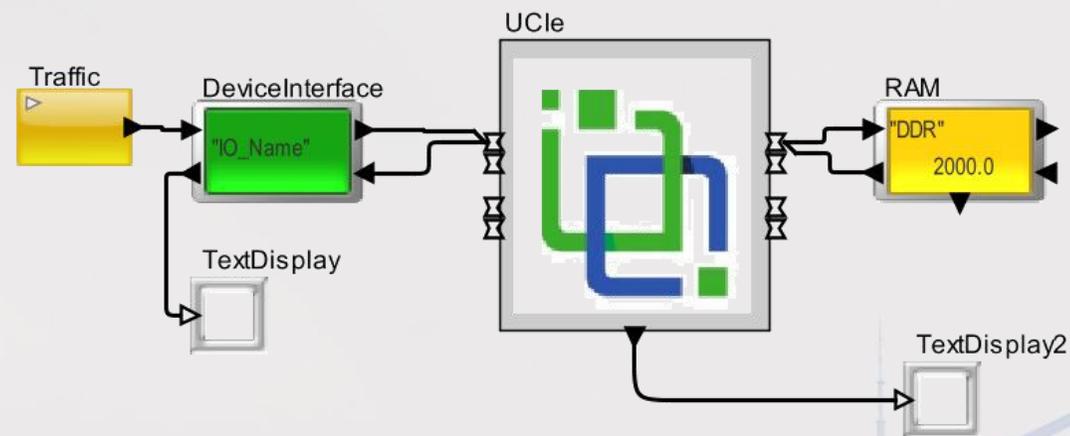
- 使cache hit-ratio最大化并且寻找缩短访问存储的时间
- 分析总线的通信量，端到端延迟和系统的吞吐率

## ■ 虚拟原型性能 (Virtual performance prototyping)

- 基于实际的SW代码来做性能优化
- 功耗，软硬件协同，占用缓存等权衡

# 应用案例： Chiplet

- 模拟Chiplet芯片的多个die通过UCIe互联的情况下，各die传输存在传输性能瓶颈
- 可定制化配置封装类型、端口传输速率、RAM参数等影响数据传输性能的参数
- 可支持配置多协议、多Module等UCIe支持的特性
- 支持监控总体传输状态（实时带宽、延时等）及内部传输过程（各端口收发包情况）



模拟从RAM中读取数据，并在端口间进行传递，统计实时带宽、传输时延、端口通信质量等信息



# 依据配置得到相应的仿真结果

```

DISPLAY AT TIME          ----- 3.6000000000000000 sec -----
{DS_Name                 = "UCIe_Switch_UCIe_Stats",
UCIe_Switch_UCIe_Port_1_Drop_Count      = 0,
UCIe_Switch_UCIe_Port_1_Rx_MBps        = 0.027231111111111,
UCIe_Switch_UCIe_Port_1_Total_MBps     = 0.035551111111111,
UCIe_Switch_UCIe_Port_1_Tx_MBps        = 0.00832,
UCIe_Switch_UCIe_Port_1_to_Port_5_Max_Latency = 1.2207062027869E-9,
UCIe_Switch_UCIe_Port_1_to_Port_5_Mean_Latency = 1.2207060024794E-9,
UCIe_Switch_UCIe_Port_1_to_Port_5_Min_Latency = 1.2207057586977E-9,
UCIe_Switch_UCIe_Port_2_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_2_Rx_MBps        = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_2_Total_MBps     = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_2_Tx_MBps        = 0.0,
UCIe_Switch_UCIe_Port_3_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_3_Rx_MBps        = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_3_Total_MBps     = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_3_Tx_MBps        = 0.0,
UCIe_Switch_UCIe_Port_4_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_4_Rx_MBps        = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_4_Total_MBps     = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_4_Tx_MBps        = 0.0,
UCIe_Switch_UCIe_Port_5_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_5_Rx_MBps        = 0.0083511111111,
UCIe_Switch_UCIe_Port_5_Total_MBps     = 0.0355511111111,
UCIe_Switch_UCIe_Port_5_Tx_MBps        = 0.0272,
UCIe_Switch_UCIe_Port_5_to_Port_1_Max_Latency = 7.5957102829705E-9,
UCIe_Switch_UCIe_Port_5_to_Port_1_Mean_Latency = 4.4394579881518E-9,
UCIe_Switch_UCIe_Port_5_to_Port_1_Min_Latency = 1.2832059859136E-9,
UCIe_Switch_UCIe_Port_6_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_6_Rx_MBps        = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_6_Total_MBps     = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_6_Tx_MBps        = 0.0,
UCIe_Switch_UCIe_Port_7_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_7_Rx_MBps        = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_7_Total_MBps     = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_7_Tx_MBps        = 0.0,
UCIe_Switch_UCIe_Port_8_Drop_Count     = 0,
UCIe_Switch_UCIe_Port_8_Rx_MBps        = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_8_Total_MBps     = 3.1111111111111E-5,
UCIe_Switch_UCIe_Port_8_Tx_MBps        = 0.0}
    
```

## 端口传输监控

- 实时传输带宽
- 丢包统计
- 传输时延

内部传输状态监控：各端口收发包、包类型等信息

```

DISPLAY AT TIME          ----- 2.200000369158210 sec -----
Flit_Disassembler_UCIe_Port_1 <Rx> ::: NOP

DISPLAY AT TIME          ----- 2.200000369236336 sec -----
Replay_Buffer_UCIe_Port_5 ::: ACK received ::: Flit Num = [223]           Num Ack = 1

DISPLAY AT TIME          ----- 2.200000369658210 sec -----
Flit_Disassembler_UCIe_Port_1 <Rx> ::: FLIT [225]           Destination Port = 1   TLPs = 1

DISPLAY AT TIME          ----- 2.200000369775399 sec -----
Flit_Assembler_UCIe_Port_5 <Tx> ::: FLIT [227]           Destination Port = 1   TLPs = 1           TLPs to

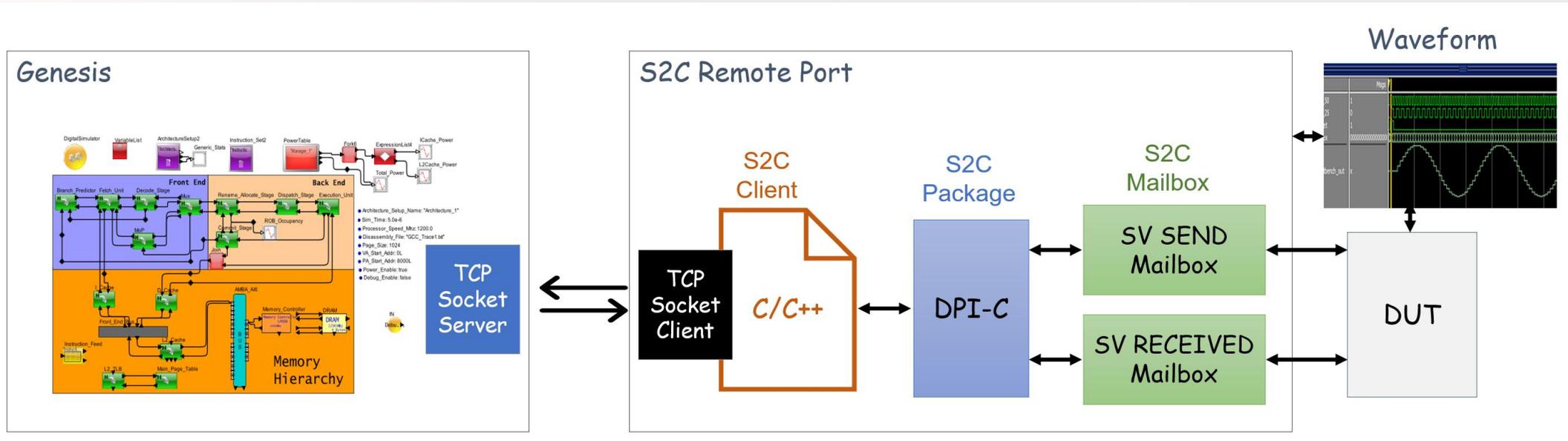
DISPLAY AT TIME          ----- 2.200000370736336 sec -----
Flit_Disassembler_UCIe_Port_1 <Rx> ::: NOP

DISPLAY AT TIME          ----- 2.200000370814462 sec -----
Replay_Buffer_UCIe_Port_5 ::: ACK received ::: Flit Num = [225]           Num Ack = 1

DISPLAY AT TIME          ----- 2.200000371236336 sec -----
Flit_Disassembler_UCIe_Port_1 <Rx> ::: FLIT [227]           Destination Port = 1   TLPs = 1
    
```

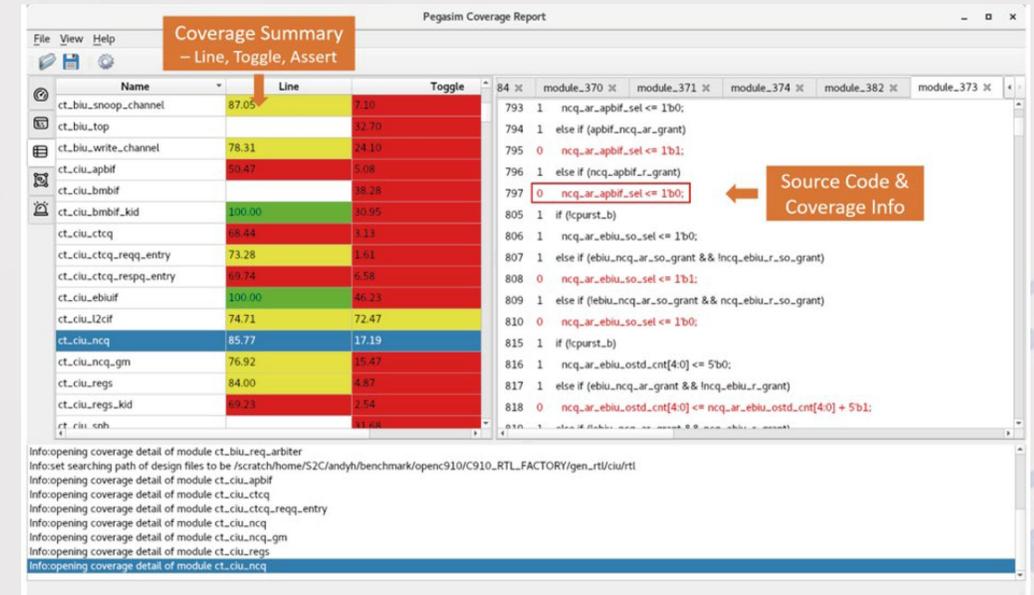
# Genesis & Simulator 联合仿真

- TCP socket 通信 (其他快速方案也在研究中)
- S2C通过远端接口
  - S2C 客户端: 提供API用于连接, 发送和接收数据
  - S2C 发包: DPI-C package
  - S2C SV Mailbox: 允许数据在DUT和客户的交互



# PegaSim 软仿特性

- 支持多种设计与验证语言： IEEE1800-2017 SystemVerilog , IEEE1364 Verilog 和 IEEE1076-1993 VHDL
- SVA 断言、约束求解器、代码与功能覆盖率分析
- 支持UVM, VMM, OVM 方法学 ( IEEE 1800.2)
- 支持 DPI & VPI 混合协同仿真
- 支持时序反标 (SDF 和 Timing Check)
- 支持设计增量编译和并行编译
- 支持多种处理器架构, 如X86, ARM , RISC-V
- VCD和自研波形格式, 灵活支持多种调试手段



The screenshot displays the 'Pegasim Coverage Report' window. It features a 'Coverage Summary' table with columns for Name, Line, and Toggle. Below the table, source code snippets are shown with corresponding coverage information. An orange callout box labeled 'Source Code & Coverage Info' points to a specific line in the code.

Name	Line	Toggle
ct_biu_snoop_channel	87.05	7.10
ct_biu_top		32.70
ct_biu_write_channel	78.31	24.10
ct_ciu_apbif	30.47	5.08
ct_ciu_bmbif		38.28
ct_ciu_bmbif_kid	100.00	30.95
ct_ciu_ctcq	66.44	3.13
ct_ciu_ctcq_reqq_entry	73.28	1.61
ct_ciu_ctcq_respq_entry	69.74	6.58
ct_ciu_ebluif	100.00	46.23
ct_ciu_l2cif	74.71	72.47
ct_ciu_ncq	85.77	17.19
ct_ciu_ncq_gm	76.92	15.47
ct_ciu_regs	84.00	4.87
ct_ciu_regs_kid	69.23	2.54
ct_ciu_snh		33.68

Source Code & Coverage Info:

```

793 1 ncq_ar_apbif_sel <= 1'b0;
794 1 else if (apbif_ncq_ar_grant)
795 0 ncq_ar_apbif_sel <= 1'b1;
796 1 else if (ncq_apbif_r_grant)
797 0 ncq_ar_apbif_sel <= 1'b0;
805 1 if (fcpurst_b)
806 1 ncq_ar_ebiu_so_sel <= 1'b0;
807 1 else if (ebiu_ncq_ar_so_grant && lncq_ebiu_r_so_grant)
808 0 ncq_ar_ebiu_so_sel <= 1'b1;
809 1 else if (lebiu_ncq_ar_so_grant && ncq_ebiu_r_so_grant)
810 0 ncq_ar_ebiu_so_sel <= 1'b0;
815 1 if (fcpurst_b)
816 1 ncq_ar_ebiu_ostd_cnt[4:0] <= 5'b0;
817 1 else if (ebiu_ncq_ar_grant && lncq_ebiu_r_grant)
818 0 ncq_ar_ebiu_ostd_cnt[4:0] <= ncq_ar_ebiu_ostd_cnt[4:0] + 5'b1;

```

# PegaSim & 形式验证工具协同

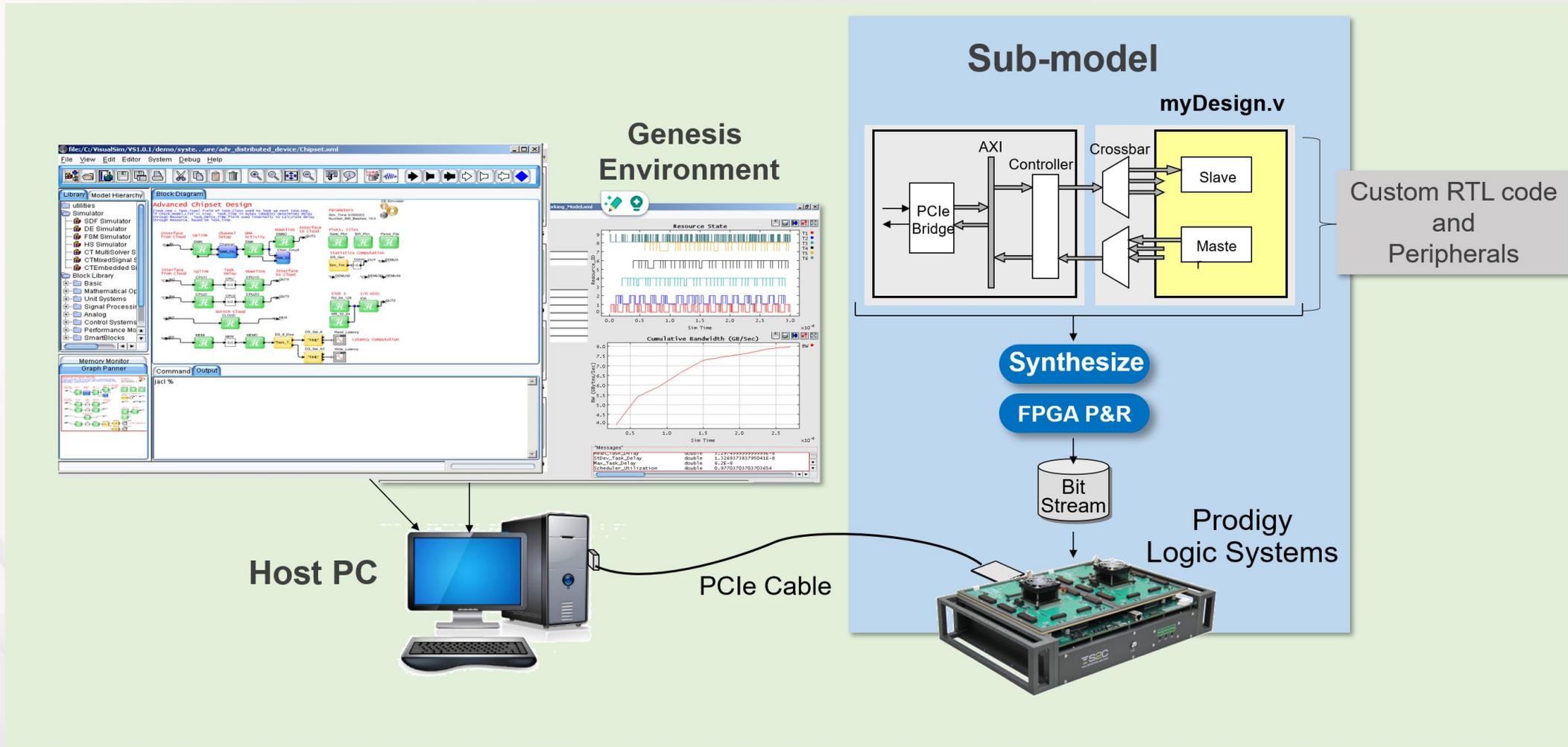
- 生成Coverage 数据库
- 形式验证工具分析
- 自动生成测试激励
- 再次仿真

```

-----
No.      Model      File                                  Line  Signal      Status      TB_VCD
-----
[0]      LINE          ../design/vlog_street_ctrl_fsm.v     42, 43 \            COVERED     work/cov.../covtask/trace0,17,19,22,28,3
[1]      TOGGLE        ../design/vlog_street_ctrl_fsm.v     4      rst         UNREACHABLE \
[2]      EXPRESSION    ../design/vlog_street_ctrl_fsm.v     41     \            COVERED     work/cov.../covtask/trace0,16,17,23,27,28
[3]      LINE          ../design/vlog_street_ctrl_fsm.v     59, 62 \            COVERED     work/cov.../covtask/trace0,23,8
[4]      LINE          ../design/vlog_street_ctrl_fsm.v     64, 65 \            COVERED     work/cov.../covtask/trace16,17,27,28
[5]      LINE          ../design/vlog_street_ctrl_fsm.v     76, 78 \            UNREACHABLE \
[6]      LINE          ../design/vlog_street_ctrl_fsm.v     83, 85 \            COVERED     work/cov.../covtask/trace1,14,18,2,25,7
[7]      LINE          ../design/vlog_street_ctrl_fsm.v     58, 59 \            COVERED     work/cov.../covtask/trace0
[8]      LINE          ../design/vlog_street_ctrl_fsm.v     74, 76 \            UNREACHABLE \
[9]      LINE          ../design/vlog_street_ctrl_fsm.v     76, 78 \            UNREACHABLE \
[10]     LINE          ../design/vlog_street_ctrl_fsm.v     99, 106 \           COVERED     work/cov.../covtask/trace15,26
[11]     LINE          ../design/vlog_street_ctrl_fsm.v     83, 85 \           COVERED     work/cov.../covtask/trace1,14,18,2,25,7
[12]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     6      waiting     COVERED     work/cov.../covtask/trace10,11,12,9
[13]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     7      waiting_cross COVERED     work/cov.../covtask/trace10,11,12,9
[14]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     8      state_cross UNREACHABLE \
[15]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     8      state_cross UNREACHABLE \
[16]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     8      state_cross UNREACHABLE \
[17]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     10     state_out   UNREACHABLE \
[18]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     10     state_out   UNREACHABLE \
[19]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     23     next_state  UNREACHABLE \
[20]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     23     next_state  UNREACHABLE \
[21]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     23     next_state  UNREACHABLE \
[22]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     26     timer       UNREACHABLE \
[23]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     26     timer       UNREACHABLE \
[24]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     4      rst         UNREACHABLE \
[25]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     6      waiting     COVERED     work/cov.../covtask/trace10,11,12,9
[26]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     7      waiting_cross COVERED     work/cov.../covtask/trace10,11,12,9
[27]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     8      state_cross UNREACHABLE \
[28]     TOGGLE        ../design/vlog_street_ctrl_fsm.v     8      state_cross UNREACHABLE \
-----
出  调试控制台  终端  窗口
-----
COMMENT [VERI-1482] Analyzing Verilog file '../design/vlog_street_ctrl_fsm.v'
COMMENT [VERI-1482] Analyzing Verilog file '../TB/tb.v'
read_db -file s2c.db
handle_db_file -verify_db_coverage -write_sva
-----

```

# Genesis & Prototype 混仿

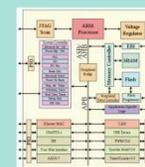


# 原型验证解决方案

协同仿真套件  
ProtoBridge



自动原型编译软件  
PPro-CT



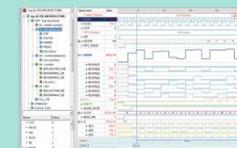
实时控制软件  
PPro-RT



原型验证系统  
FPGA Prototyping Platform



深度调试套件  
PPro-DT



外置应用库  
Prototype Ready IP



云管理软件  
Neuro



# 原型验证软件工具

- 从 RTL 到比特流
- RTL / Netlist 分割
- Constraint 分割
- 时钟, 模块复制
- 时序分析
- 端口时分复用
- 注入调试模块
- 并行布局布线
- 增量编译

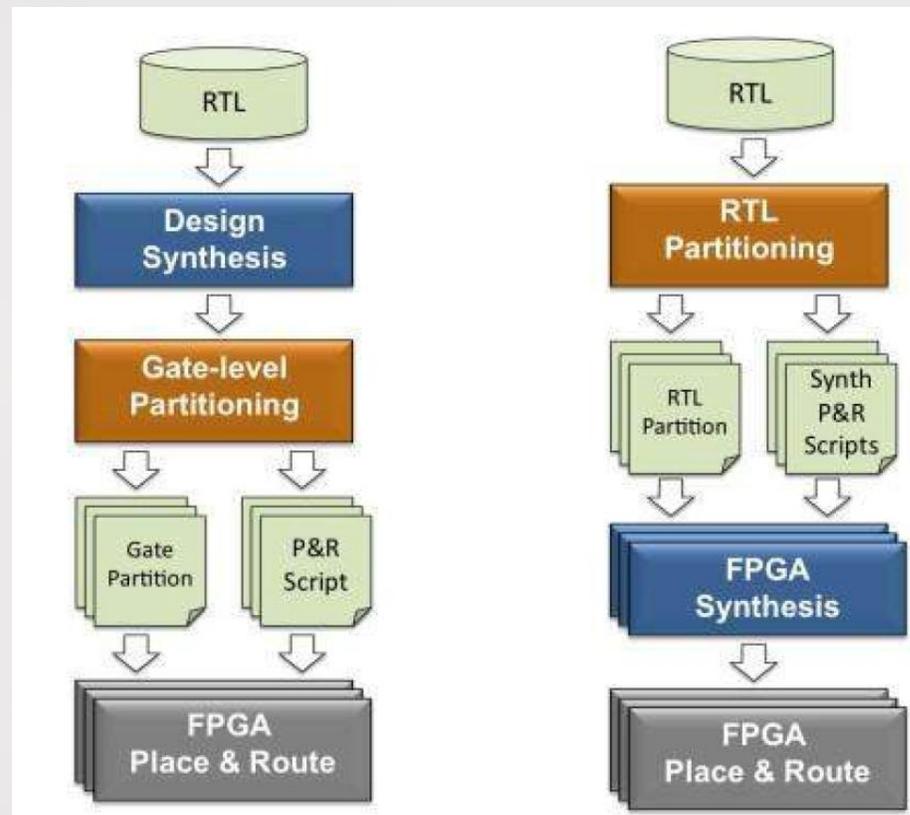
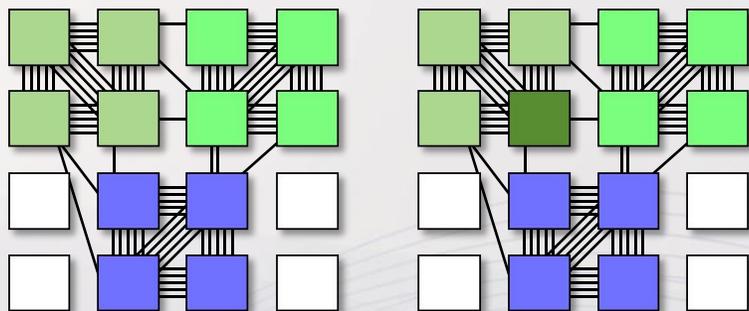


- 系统管理: 授权, 日志, 电源, OTA
- 远端操作: 多元下载, 时钟, 复位
- 运行控制-增强调试:
  - 后门通道: NTBus/vAXI
  - 虚拟IO / 虚拟UART
  - 动态探针 (获取内部信号状态)

- 多FPGA深度调试
- 自动编译模式和IP插入模式
- 独立资源
  - 不占用USER FPGA
  - 64GB 深度
  - 8K+ \* 8 groups
- 采样频率
- FSM脚本模式
- 动态、静态探针

# 设计自动编译

- RTL布局规划
- 并行综合
- IP版本管理
- 增量编译



# 自动时分复用 IP 插入

## ■ 支持 LVDS TDM

- 最高复用比: 256:2

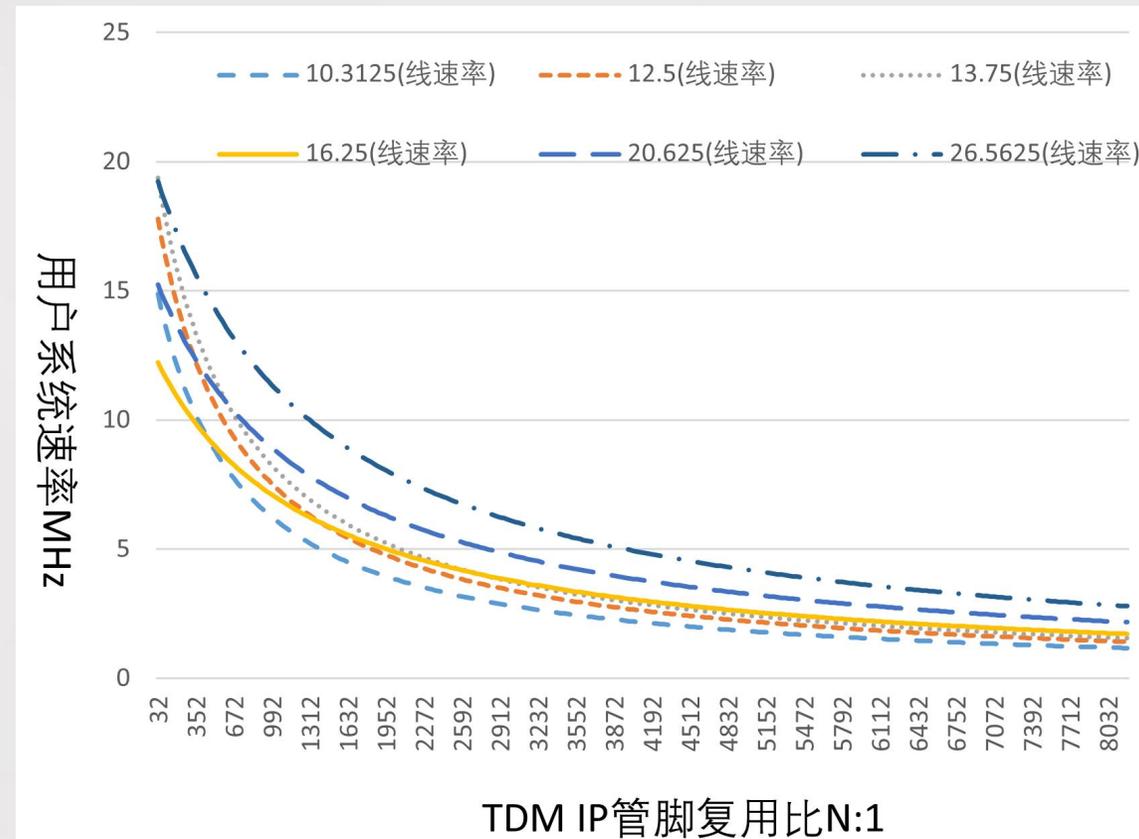
## ■ 支持SerDes TDM

- 最高复用比: 8192:1
- 长距离传输 - 光纤
- 自带误码检测

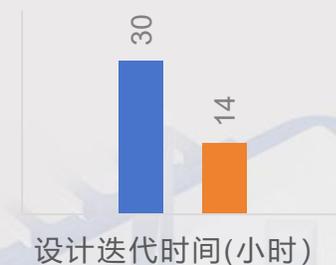
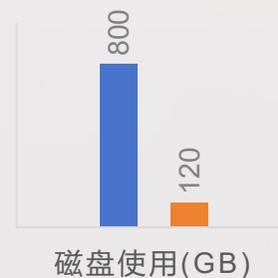
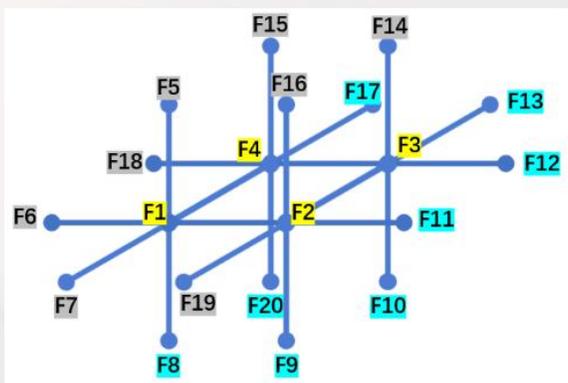
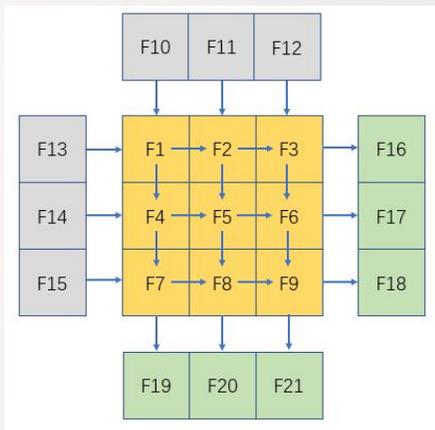
## ■ 支持总线级分割

- AXI 3 / 4

## ■ Timing Driven TDM



# 逻辑重用 & 复制



**编译流程缩短，迭代效率加速，减少人力消耗**

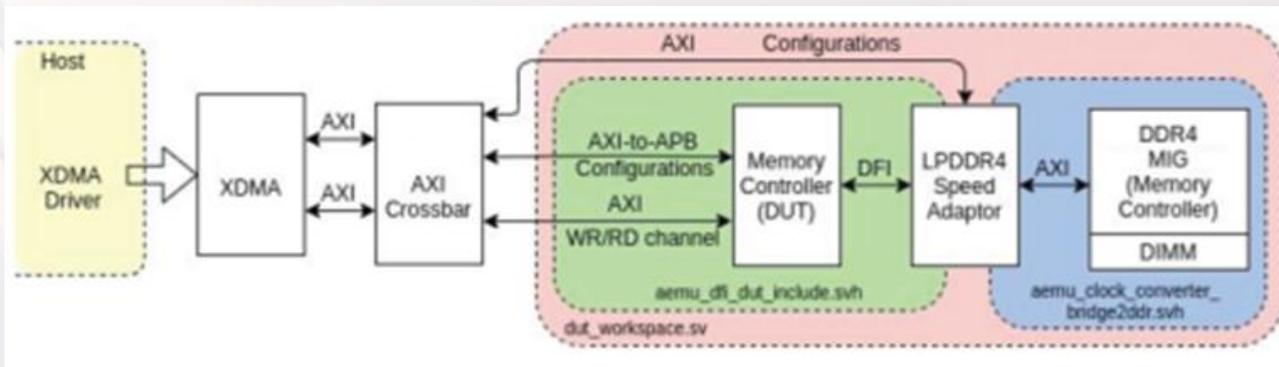
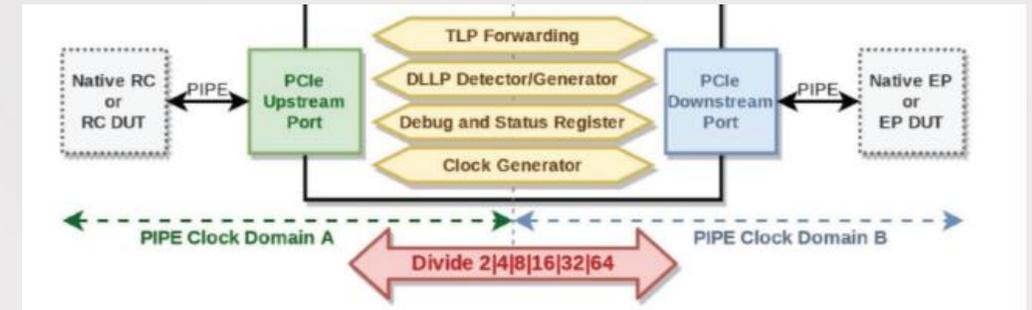
# PCI 降速桥 & 内存模型

## ■ PCIe降速桥

- 支持PCIE Gen3/ Gen4 / Gen5
- 支持RC、EP 模式

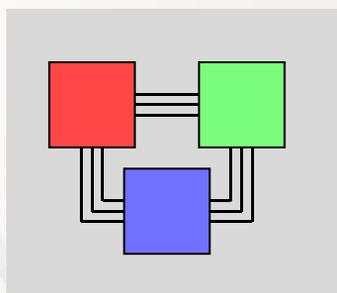
## ■ 内存模型

- 通过DFI验证Memory
- 支持DFI 4/DFI 5
- 支持DDR, LPDDR, GDDR, HBM...

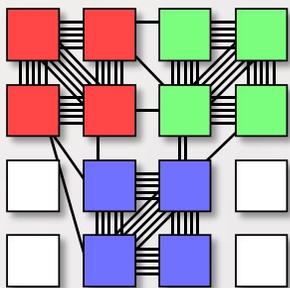


# 系统管理

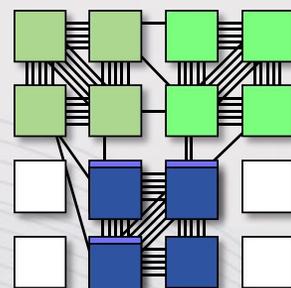
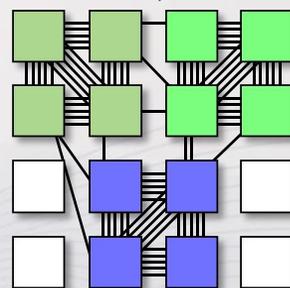
- 组网可靠性
- Bitstream下载
  - 版本管理
- 运行监控
- 资源复用, 多用户复用



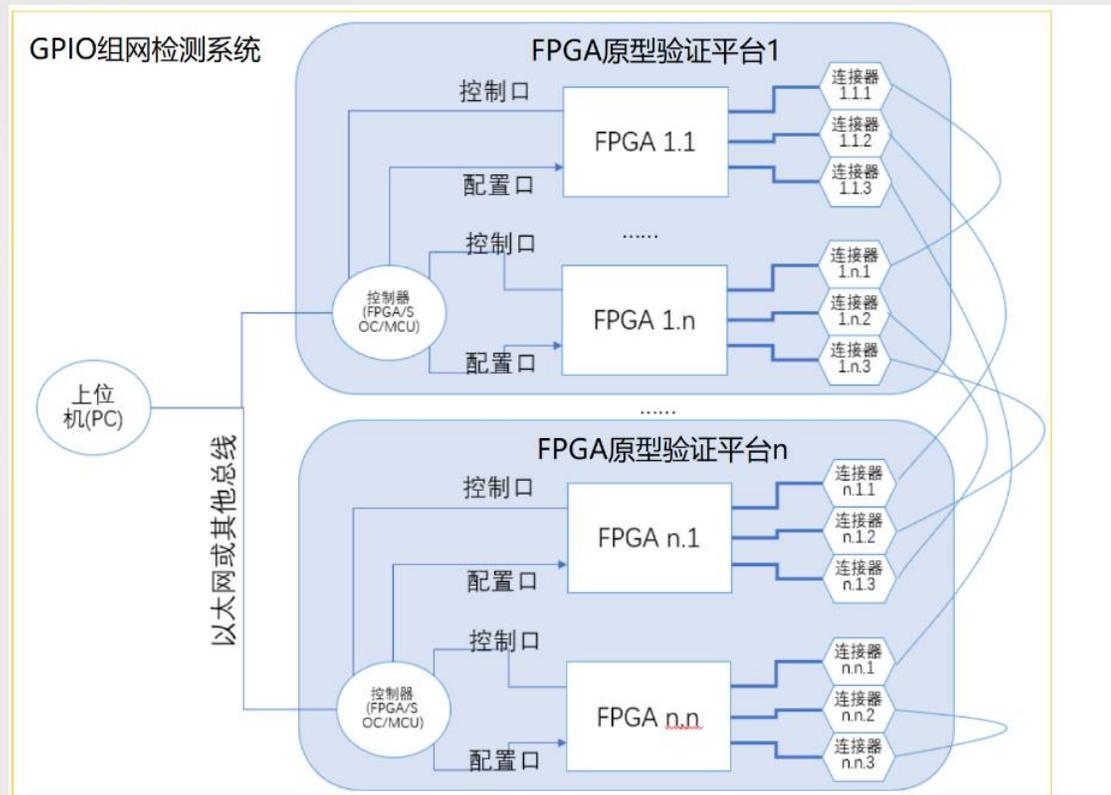
F1,F2,...,F16



F1A,F2A,...,F5A,F6A,...F16



F1A,F2A,...,F5A,F6A,...F15A,  
F16



# 基于Web的管理工具

## ■ 硬件和计算资源集中管理

- 本地部署或全球部署
- 虚拟机管理

## ■ 提高生产率

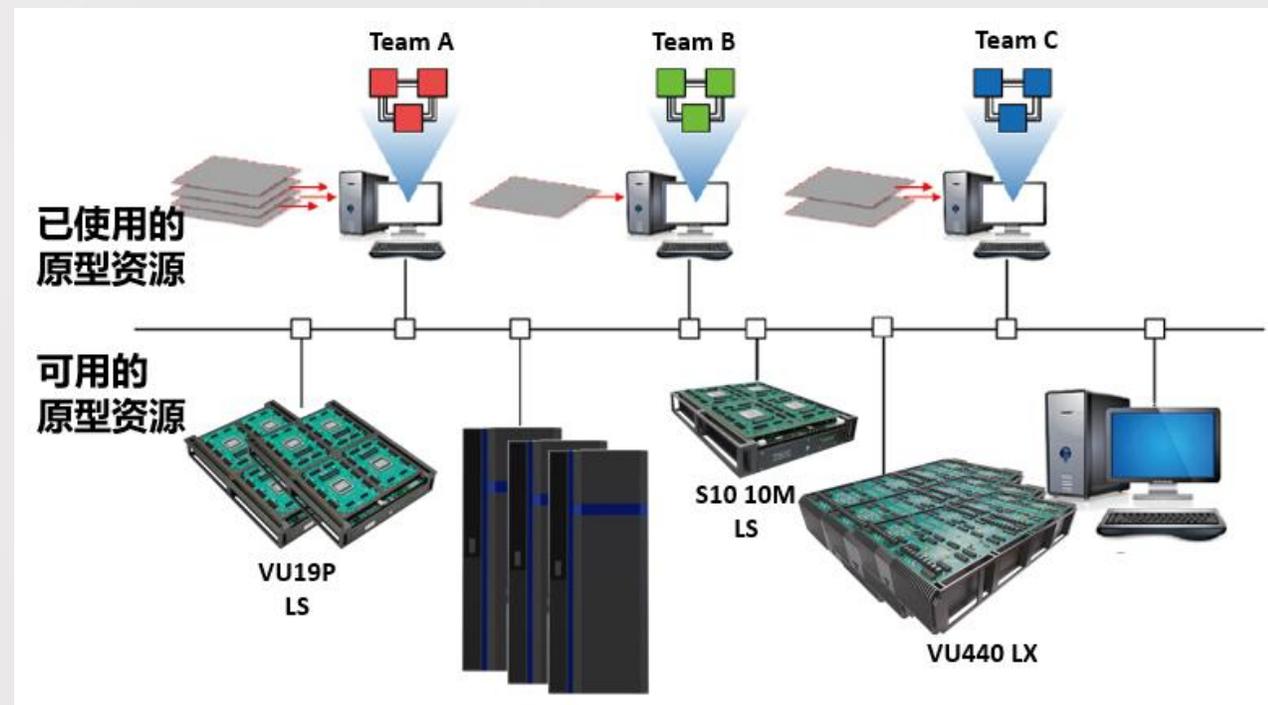
- 资源、项目与用户多维度管理，避免冲突
- 远程实时控制与监测

## ■ 节省成本，保护投资

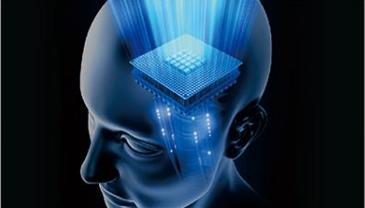
- 通过Neuro Server 实现多项目组资源共享
- 可根据优先级或者使用率报告，灵活分配调度资源

## ■ 易于维护

- 维护方便，最小化系统停机时间
- 资源使用率监测与操作可追溯性



# 总结

架构设计 Modeling	软件仿真 Simulation	硬件仿真 Emulation	原型验证 Prototyping	验证云 Cloud Service
				
<ul style="list-style-type: none"><li>• 设计架构规划</li><li>• 系统性能分析</li><li>• 虚拟化应用场景</li><li>• 快速IP建模</li></ul>	<ul style="list-style-type: none"><li>• 支持主流HDL语言</li><li>• 周期和时间驱动模式</li><li>• 高速并行仿真引擎</li><li>• 高阶建模接口</li></ul>	<ul style="list-style-type: none"><li>• 超大逻辑容量</li><li>• 自动设计编译</li><li>• 信号全可视调试</li><li>• 支持ICE/SBA/TBA</li></ul>	<ul style="list-style-type: none"><li>• 高性能高密多核硬件</li><li>• 自动设计分割</li><li>• 深度调试套件</li><li>• 丰富的外置应用库</li></ul>	<ul style="list-style-type: none"><li>• 弹性算力</li><li>• 资源集中管理</li><li>• 节省成本</li><li>• 保护投资</li></ul>

异构验证满足不同阶段不同角色的验证需求

# 超大规模集成设计 硬件仿真挑战与实现



# CONTENTS

---

**01** 硬件仿真加速系统技术演进

---

**02** 超大规模设计仿真面临的挑战

---

**03** 芯神鼎硬件仿真系统特性解析

---

**04** 芯神鼎在高端芯片系统验证的应用

---



# 硬件仿真加速系统技术演进



# 硬件仿真加速系统技术演进

硬件仿真加速系统（Emulator）从诞生至今，已经有近40年的历史。随着集成电路产业的发展，硬件仿真加速系统在大规模芯片设计前端验证领域展示了巨大优势。另外，硬件仿真加速系统以其容量、性能、可调试性方面的独特优势，已经成为最重要的验证工具之一，同时也推动着芯片前端验证技术的不断革新。



Palladium



ZeBu



Veloce



# 硬件仿真加速系统未来趋势

- 在极短时间内完成用户设计的移植和部署
- 实现对超大规模设计的全自动编译
- 快速发现和定位深层次的隐藏问题
- 快速设计足够验证case，满足不同应用场景的验证需求
- 在短时间完成对大规模设计的系统性充分验证

# 超大规模设计仿真面临的挑战



# 超大规模设计仿真面临的挑战

## ■ 依赖验证工具

芯片验证旨在找到所有芯片设计功能缺陷，高验证覆盖率对验证工具的要求高

- 准确：系统级功能验证环境极其复杂，需要足够多的验证case验证设计正确性
- 调试：能快速定位问题
- 便捷：高级语言编译验证case，环境搭建容易
- 快速：70%的研发时间用来验证，缩短验证周期短
- 丰富：丰富的IP和验证场景，满足用户各种验证场景

## ■ 人力资源投入大

芯片设计工程师和芯片验证工程师配比为1：1到 1：5

## ■ 无法保证

- 60%-70%项目不能按照原计划项目进度完成，无法保证上市时间
- 验证不充分，重新流片，时间和成本损失大

# 芯片设计前端验证主要方法

## 软件仿真

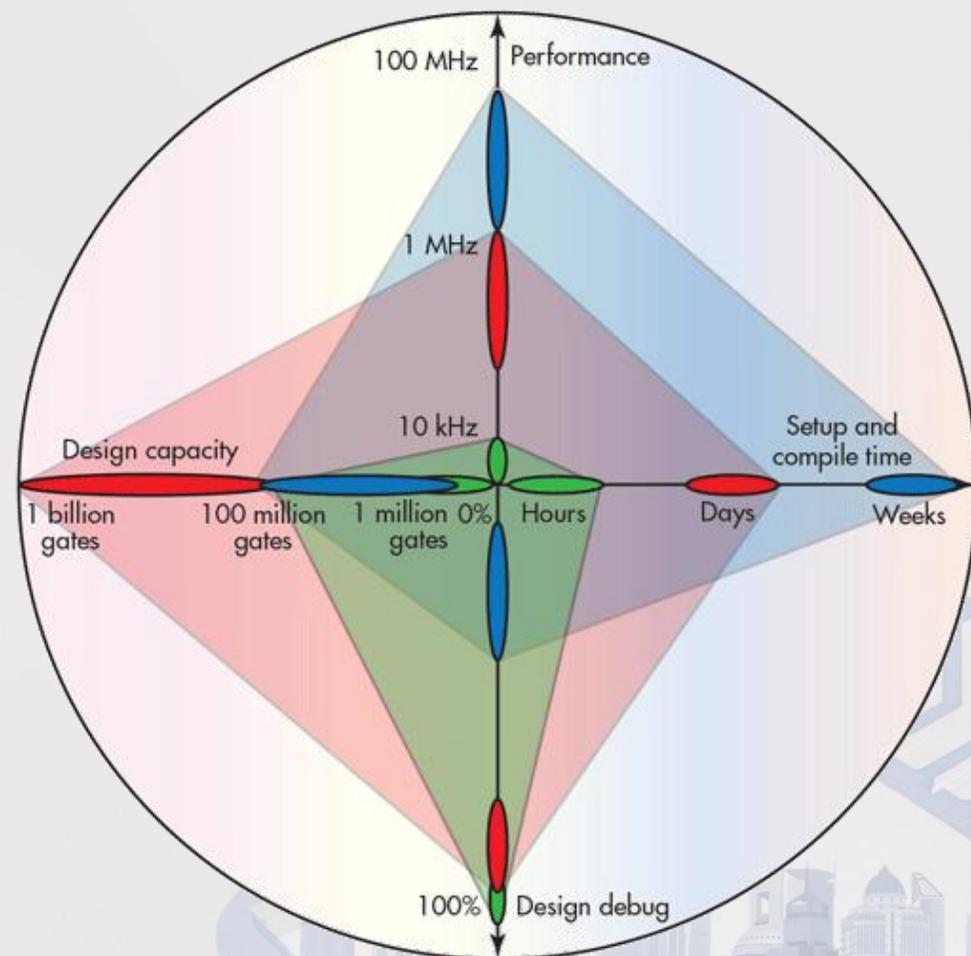
- 方便易用
- 仿真速度慢
- 适合小型设计和模块级仿真

## 硬件仿真

- 设计容量大，可扩展性好
- 仿真速度快
- 调试能力强，调试模式多样
- 适合大型设计从模块级、芯片级到系统级的仿真验证

## 原型验证

- 运行速度快
- 调试能力较弱
- 适合系统级验证



Simulator  
Prototype  
Emulator

# 芯神鼎硬件仿真系统特性解析



# OmniArk芯神鼎硬件仿真系统

## 可扩展大容量

设计容量2.5-20亿门

## 快速自动编译流程

全自动流程，用户“零”干预

## MHz级仿真加速

最大运行频率可达2MHz



## 强大调试能力

灵活探针，信号全可视

## 多种仿真验证模式

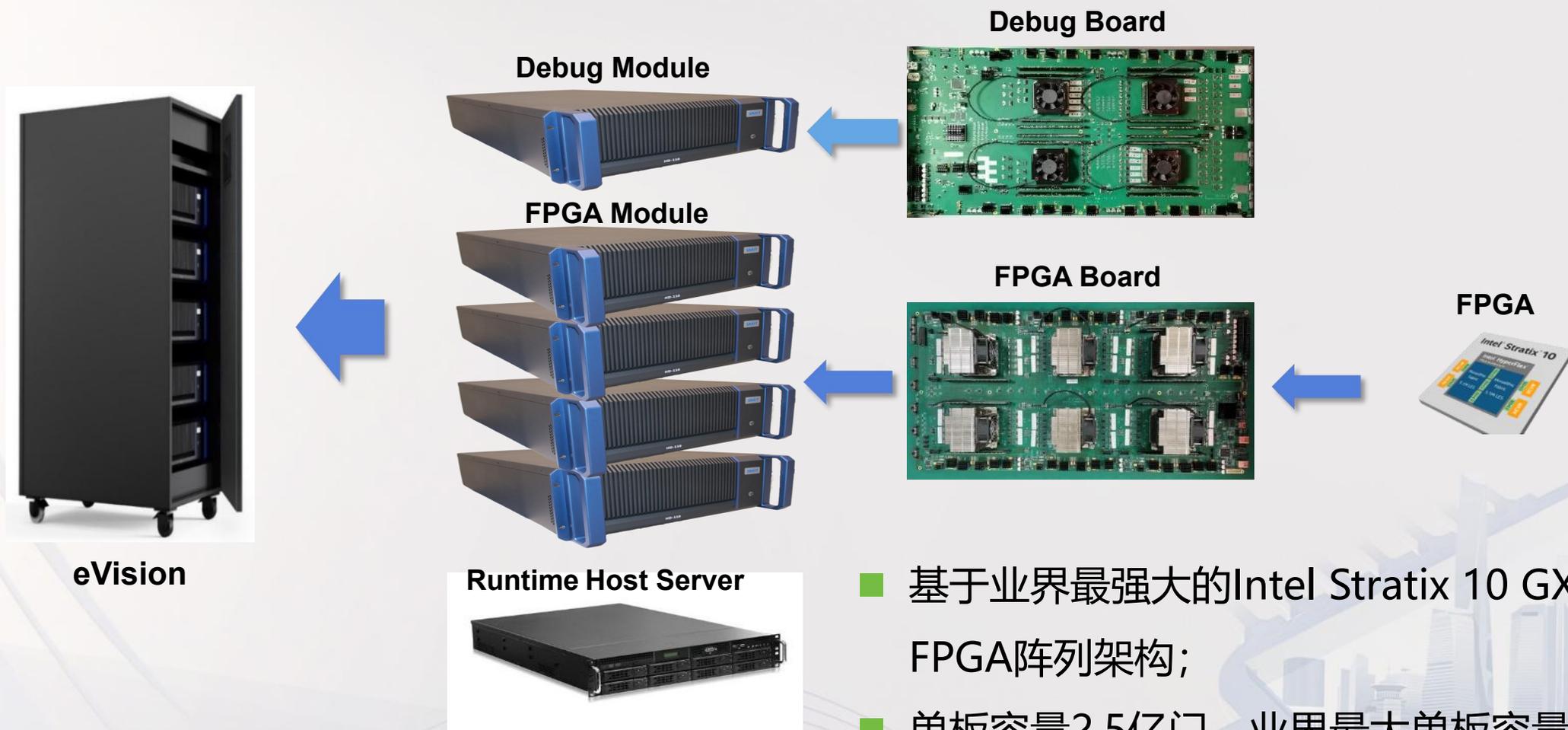
支持TBA、ICE、QEMU等

## 丰富的VIP库

支持AHB、AXI、DDR、ETH等  
20+ 验证IP

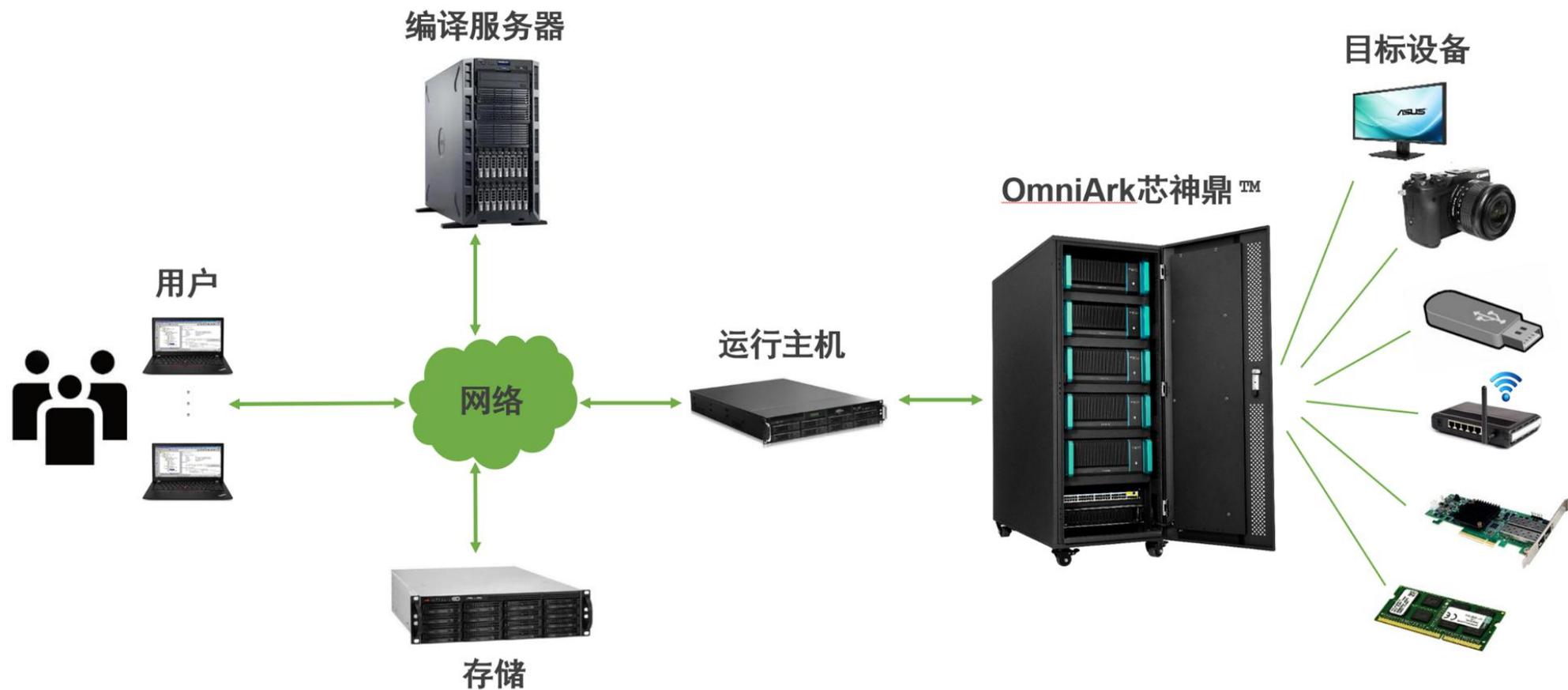
拥有完整知识产权的国产硬件仿真加速系统

# OmniArk芯神鼎硬件仿真系统



- 基于业界最强大的Intel Stratix 10 GX 10M FPGA阵列架构;
- 单板容量2.5亿门, 业界最大单板容量
- 可扩展结构, 最大设计规模20亿门

# OmniArk芯神鼎硬件仿真系统架构



# OmniArk芯神鼎仿真加速系统软件工具



OmniArk 芯神鼎

**编译**

通过综合、分割、布局布线将RTL源码编译成FPGA Bit stream文件。

**运行**

控制DUT运行，实时监测运行状态，支持多用户管理。

**调试**

支持信号采样，Trigger设置，波形下载和分析。

# 芯神鼎在高端芯片验证应用



# 芯神鼎在高端芯片验证应用

- 采用多重并行处理技术，大幅度减少编译时间
- 自动化程度超高得分割算法，资源评估准确率高
- Smart P&R 算法，智能参数优化，提升PR成功率
- 解决超大规模设计的性能挑战，降低服务器内存需求，降低用户成本
- 系统级时序建模及时序分析，精确估算最大运行频率，为时序驱动算法提供反馈

# 芯神鼎在高端芯片验证应用



- 极少的用户干预，实现用户设计的快速移植和部署
  - 用户设计语法自动纠错
  - 门控时钟自动处理
  - 用户内存自动建模
  - 支持Verilog, System Verilog等常用开发语言
- 全自动编译流程
  - 设计导入
  - RTL分析处理，逻辑综合
  - 设计分割，路由优化
  - FPGA布局布线



**Runtime Host Environment**



# 强大的信号分析调试能力

- 灵活的信号采集手段
  - 静态探针/动态探针
- 巨大波形存取深度，触发位置可设置
- 信号100%全可见
- 存储器后门读写
- 支持强制/释放功能，方便故障注入测试
- 内置波形查看工具，支持信号与RTL代码反标

The screenshot displays the eVision software interface with several key features highlighted:

- 标记探针 (Marked Probe):** A red label points to the 'Add to watchpoint' menu option in the RTL code editor, which is used for setting static or dynamic probes on specific signals.
- 波形查看 (Waveform View):** A red label points to the 'Waves' window, which shows a detailed digital waveform of the selected signals over time.
- Trigger设置 (Trigger Settings):** A red label points to the RTL code editor, where specific conditions are set to trigger the waveform capture.

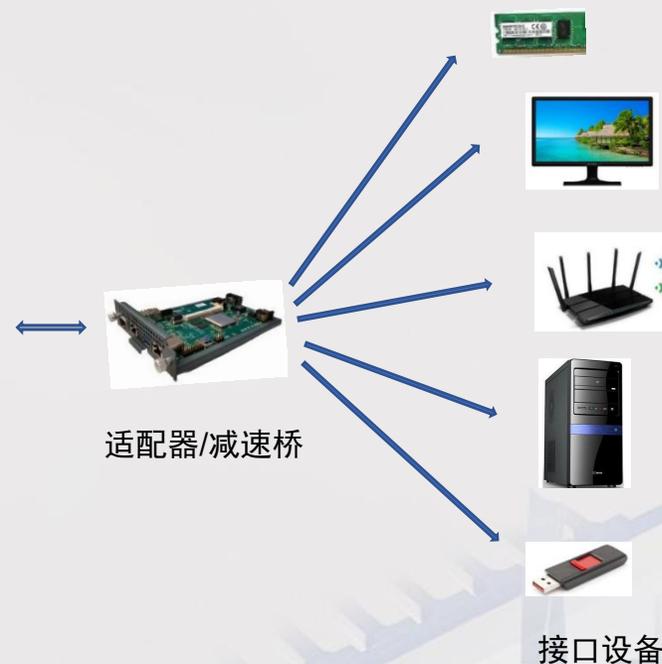
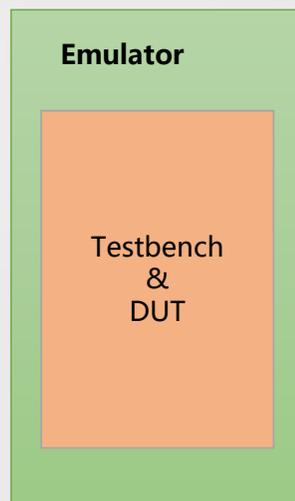
The interface includes a Hierarchy tree on the left, a central RTL code editor, and a Signals list on the right. The waveform window shows multiple signals like HBURSTH, HMASTLOCK, HREADYM0M, etc., with their digital values over time.

# 芯神鼎ICE模式在高端芯片验证应用

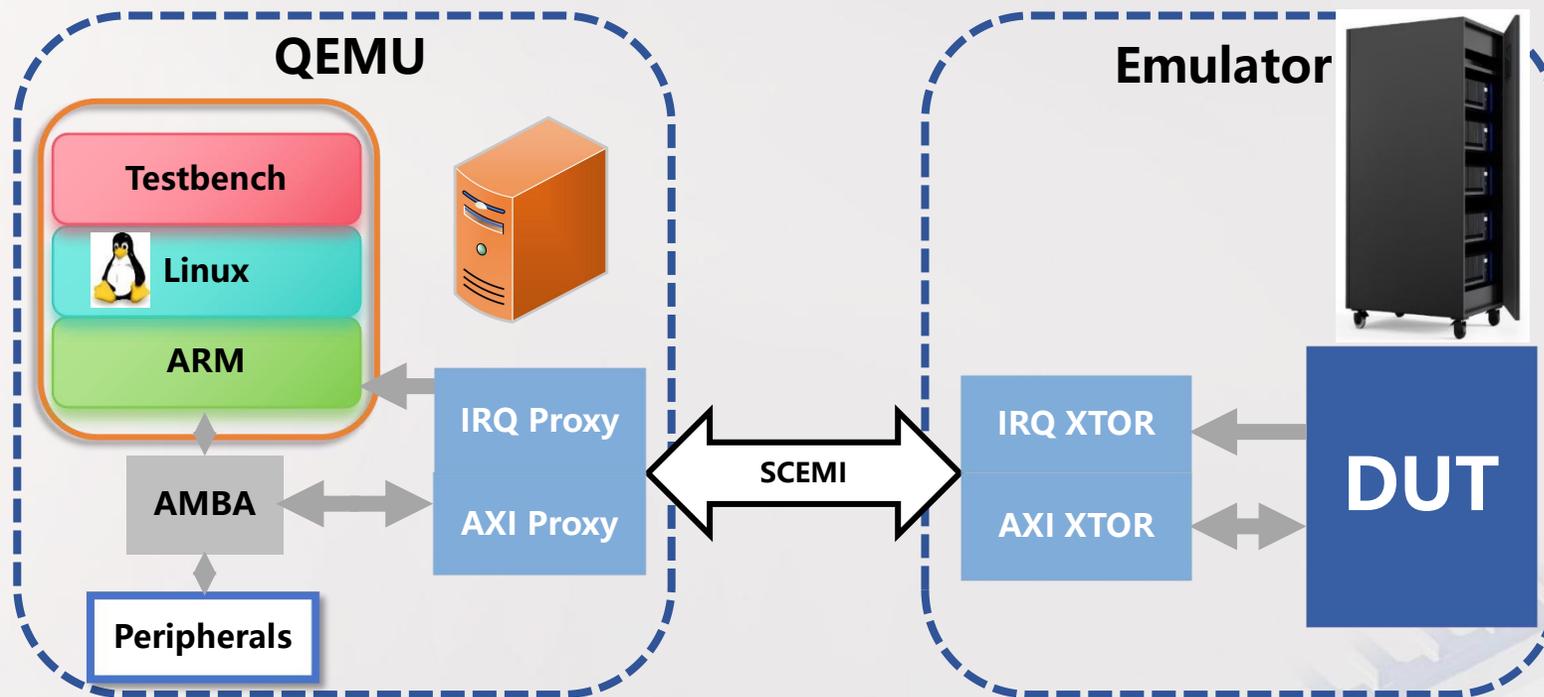
- 支持最多24个适配器子卡
- 子卡地址自动分配及路由
- 高速接口减速桥
  - Ethernet
  - USB
  - PCIe
  - .....



eVision



# 芯神鼎Hybrid混合模式在高端芯片验证应用



# Thank you!

思尔所需 加速芯未来

