

# 基于RISC-V指令集的混合架构 处理器内核设计

Hybrid Architecture Processor Design Based on RISC-V  
Instruction Set

报告人: 王旭

厦门半导体投资集团有限公司

# 目录

1 项目基本情况



2 Egret处理器内核设计

3 Egret处理器内核验证

4 开发成果

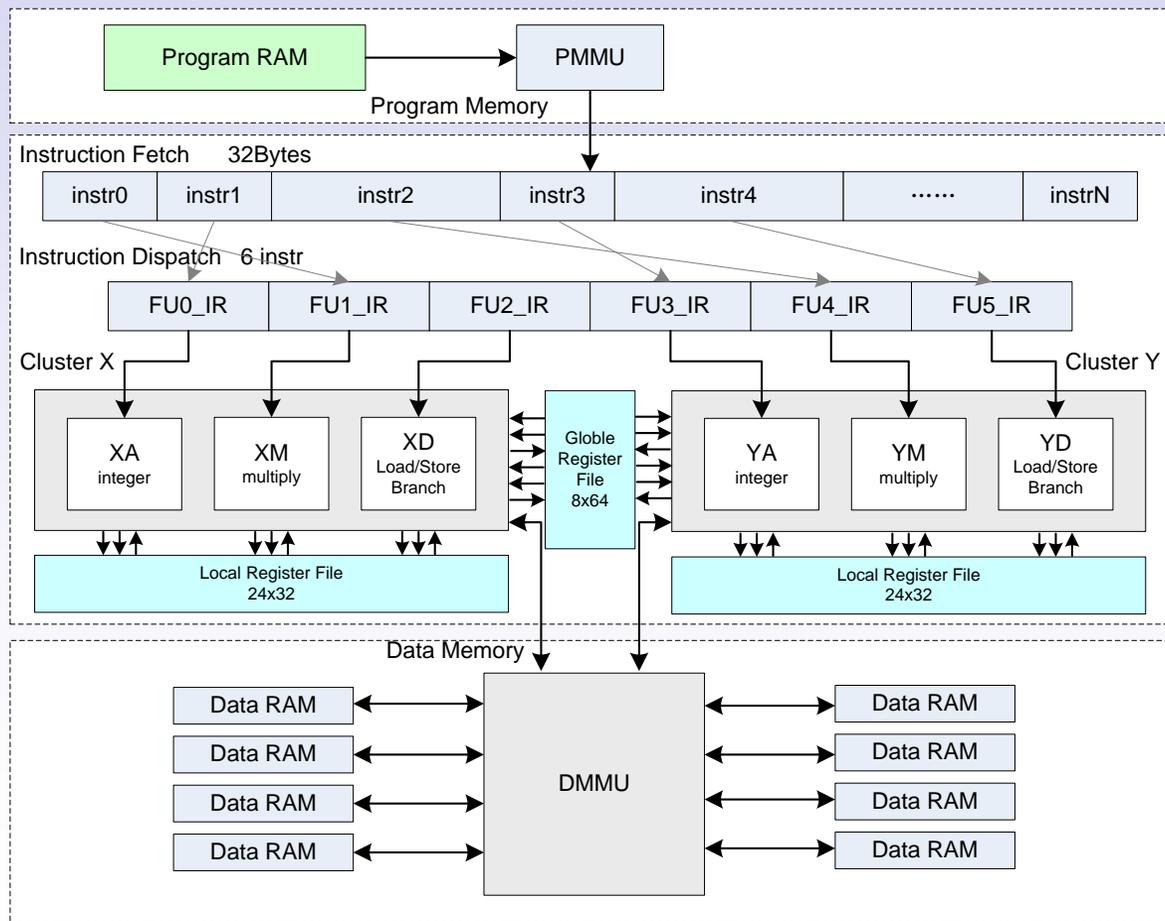
5 前景展望

# 项目基本情况

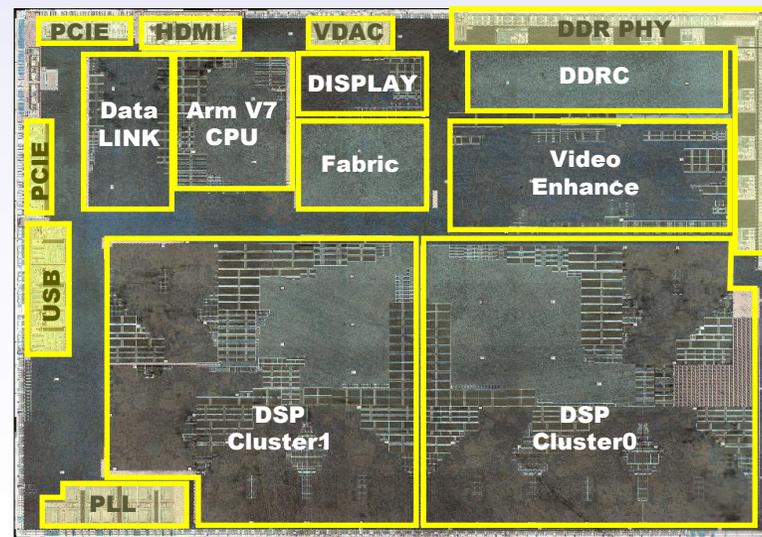
- ▶ 基于RISC-V指令集，实现与ARM A系列性能相当的处理器IP
- ▶ 指令双发射，九级流水线架构
- ▶ Dhrystone 1.76 (Best Effort) , Coremark 2.88
- ▶ 完全兼容ARM CPU接口，可实现SoC中处理器的原位替代
- ▶ 研发团队具有多年处理器设计积累



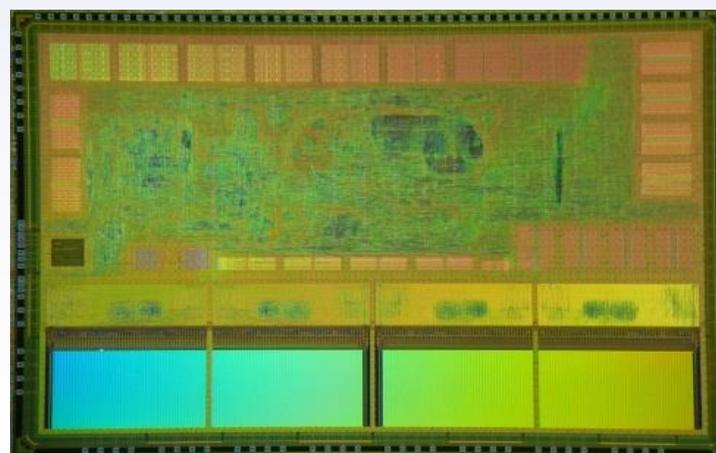
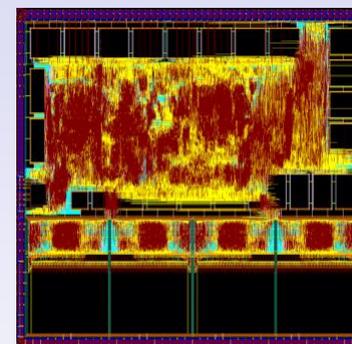
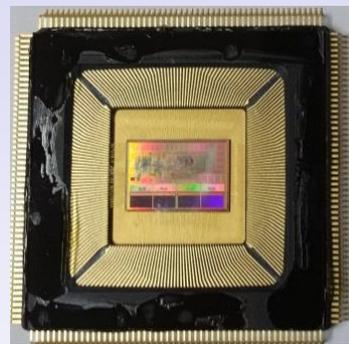
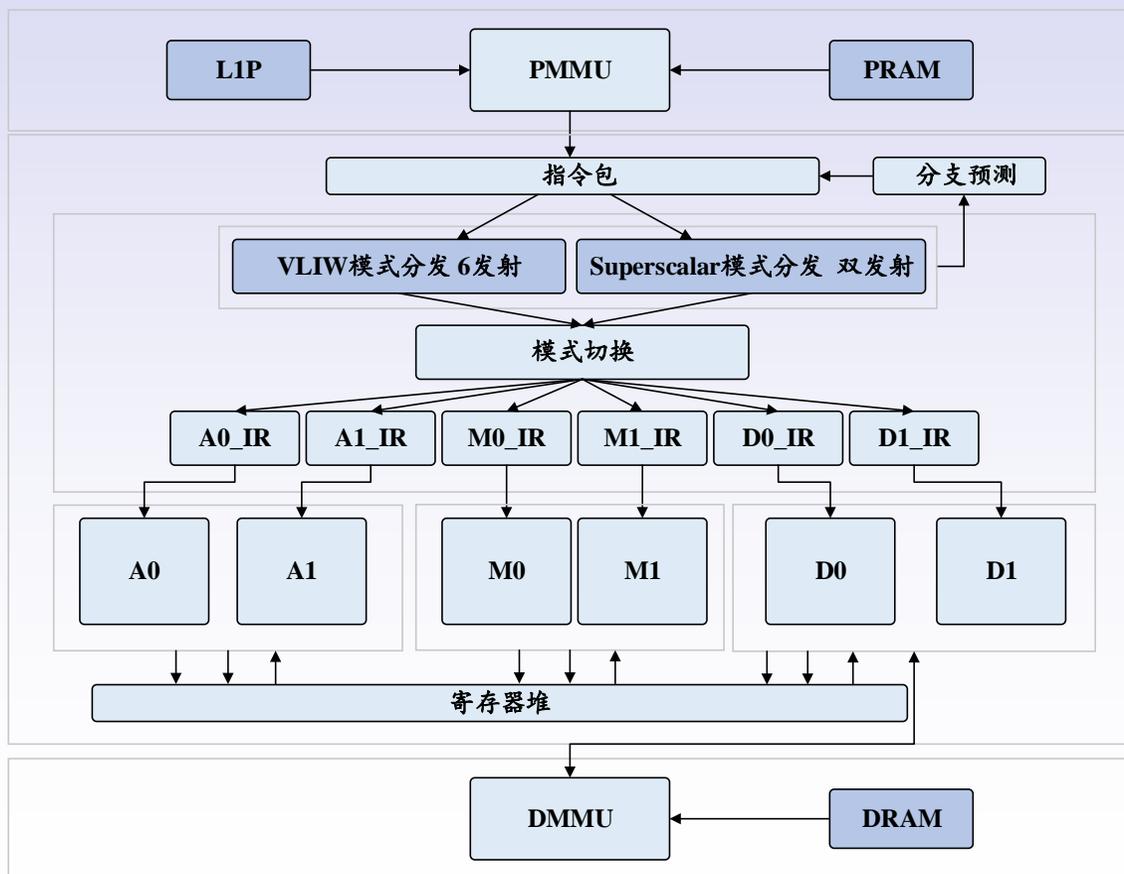
- 自研指令集、VLIW高性能32位密码处理器IP
- 6发射，11级流水线，双数据并行



## 异构多核DSP



- 自研混合架构，兼容ARM v7指令集，双发射Superscaler + 六发射VLIW
- 把CPU和DSP统一到同一个指令集下，带来了巨大的成本优势，能够以一套指令集、软件工具链，获得较好的通用性能和数据处理能力，1.95 DMIPS/MHz



# 目录

1 项目基本情况

2 Egret处理器内核设计 ←

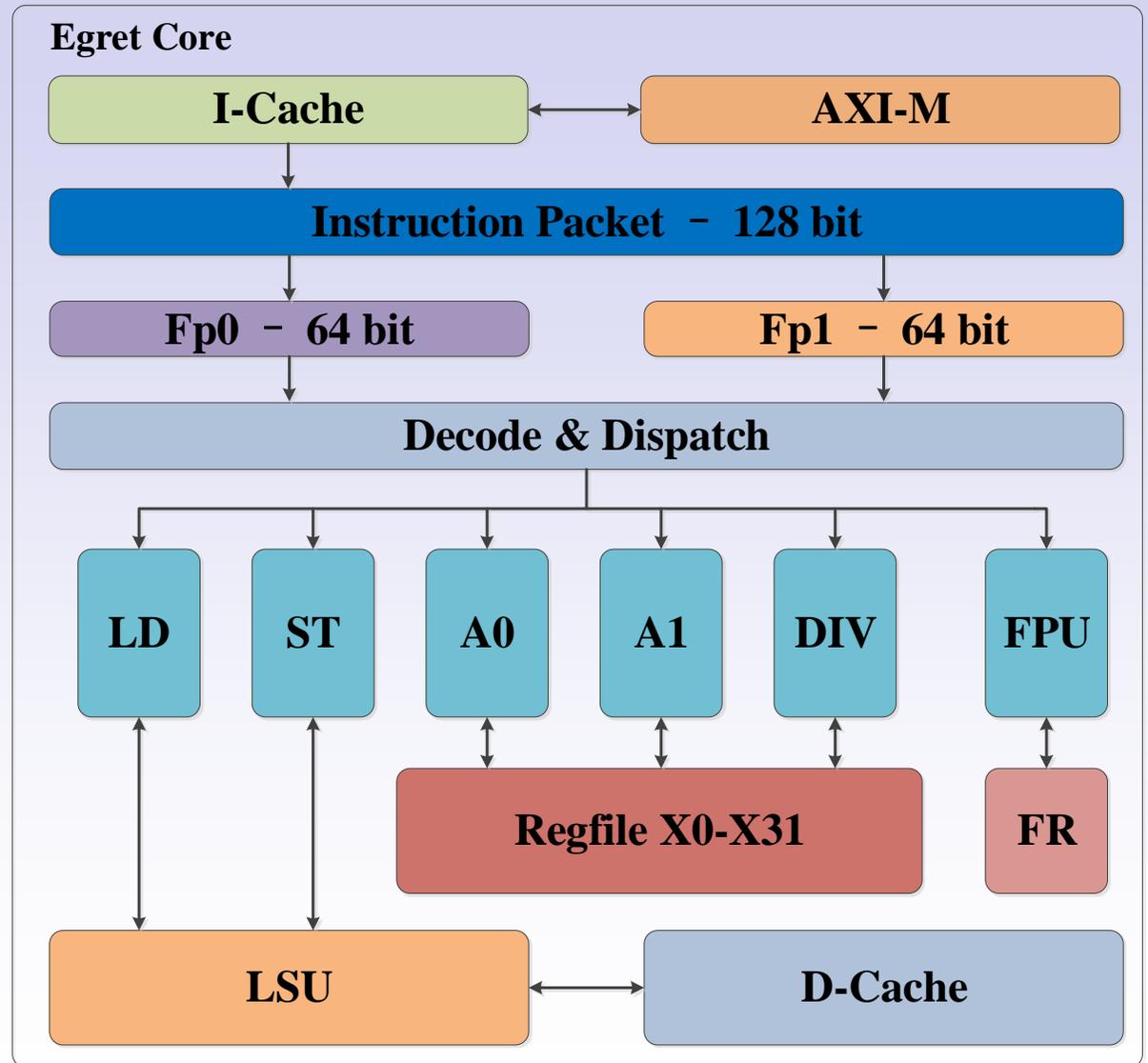
3 Egret处理器内核验证

4 开发成果

5 前景展望

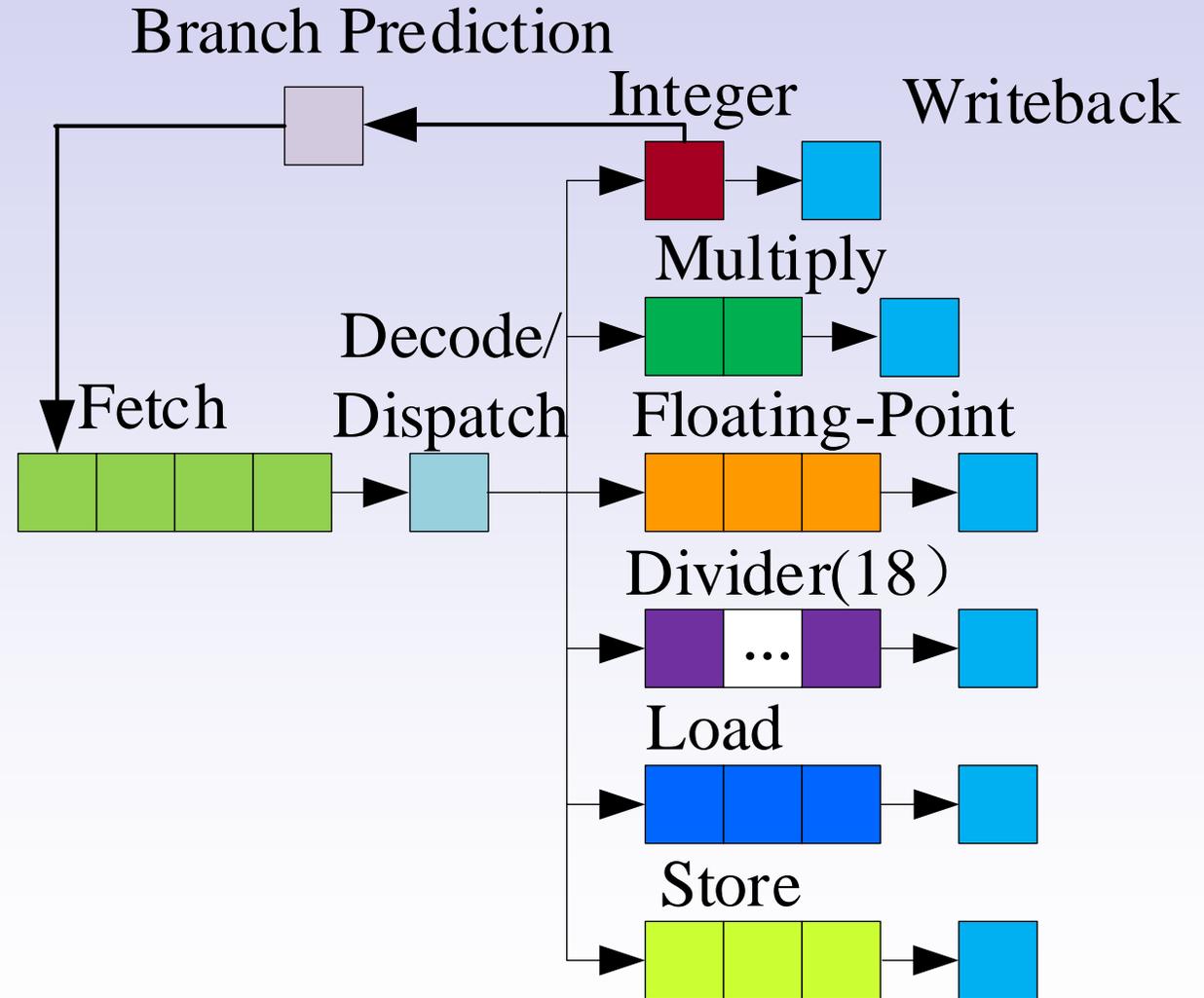
# Egret处理器内核设计

- 自研架构，双发射
- RV32IMAFC
- 6个独立的执行单元
- 兼容AXI接口
- 内核集成FPU，中断控制器
- Dhrystone 1.76
- Coremark 2.88
- 可扩展自定义指令
- 预留VLIW实现方式



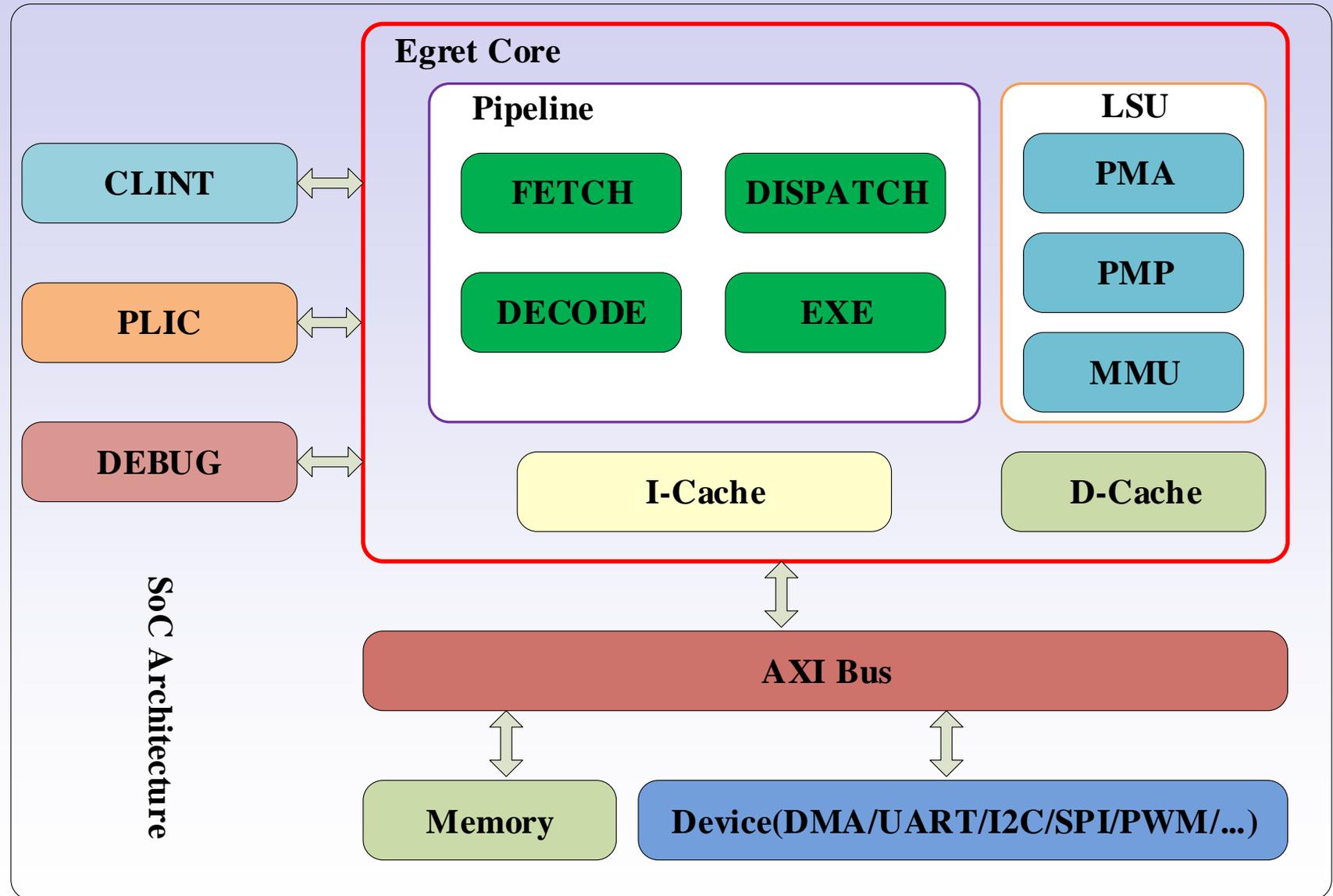
# Egret处理器流水线架构

- 九级流水线
- 指令获取：4级
- 指令分发：1级
- 指令执行：1/2/3级
- 指令写回：1级
- 动态分支预测
- 低功耗设计



# SoC结构图

- 支持外部中断，定时器中断，软中断
- 支持Debug模式
- I-Cache可配置
- D-Cache可配置
- MMU, PMP
- AXI接口
- 原位替换ARM Cortex-A7



## ➤ 自定义扩展指令

### 软件工具链

#### ● 软件实现自定义扩展指令

- ✓ 根据指令定义，软件实现指令功能，并在spike中仿真添加功能
- ✓ 针对扩展功能，把操作流程封装成函数，后续使用可直接调用

### 硬件结构

#### ● 硬件支持扩展指令功能

- ✓ 根据指令定义，硬件解码出操作数及操作码，分发到执行模块
- ✓ 执行模块根据指令的操作码，实现相应功能的硬件模块



- ▶ 支持RISC-V指令集标准
  - (1) 支持基本指令集 RV32I：32位整数指令与地址空间
  - (2) 支持扩展指令集 M：整数乘法与除法指令
  - (3) 支持扩展指令集 A：存储器原子操作指令
  - (4) 支持扩展指令集 F：单精度（32bit）浮点指令
  - (5) 支持扩展指令集 C：压缩指令
  - (6) 支持自定义扩展指令
- ▶ 自研，顺序双发射，九级流水线架构
- ▶ PLIC支持64个外部中断，CLINT中包含1个定时器中断，1个软中断
- ▶ I-Cache，D-Cache均可配置，32KB/16KB/8KB/4KB
- ▶ 兼容AXI接口，可将Egret做成硬核，在SoC中实现原位替换

# 目录

1 项目基本情况

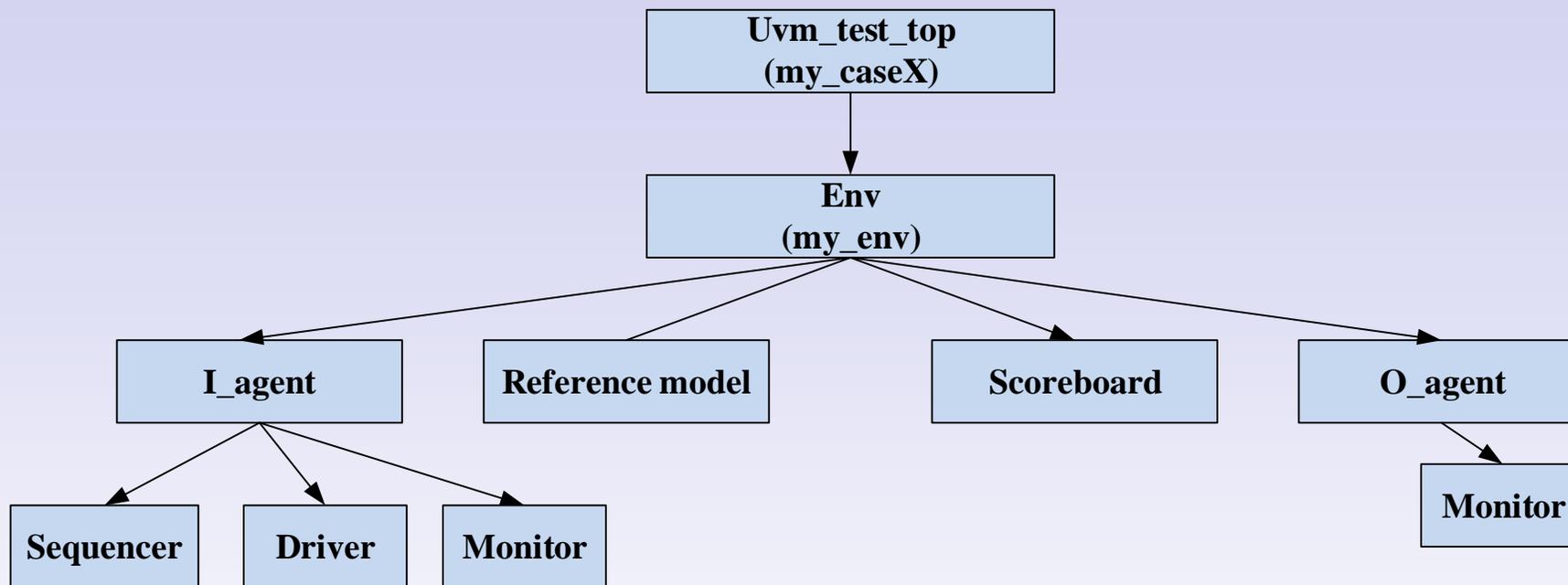
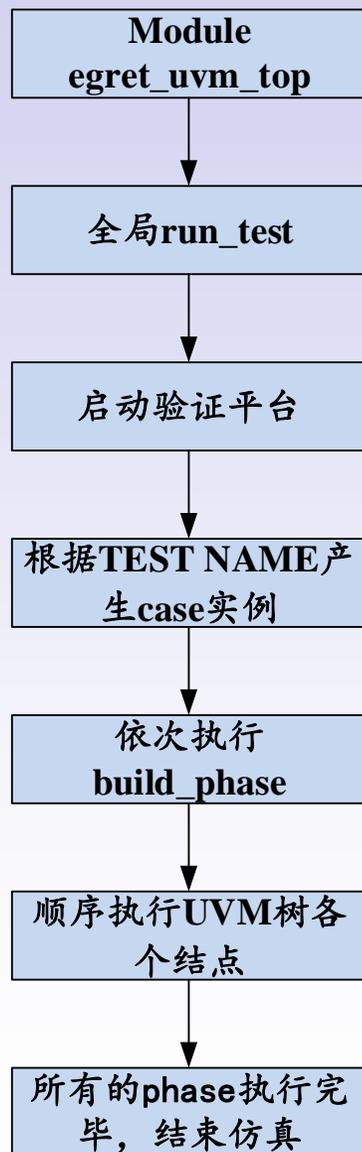
2 Egret处理器内核设计

3 Egret处理器内核验证 ←

4 开发成果

5 前景展望

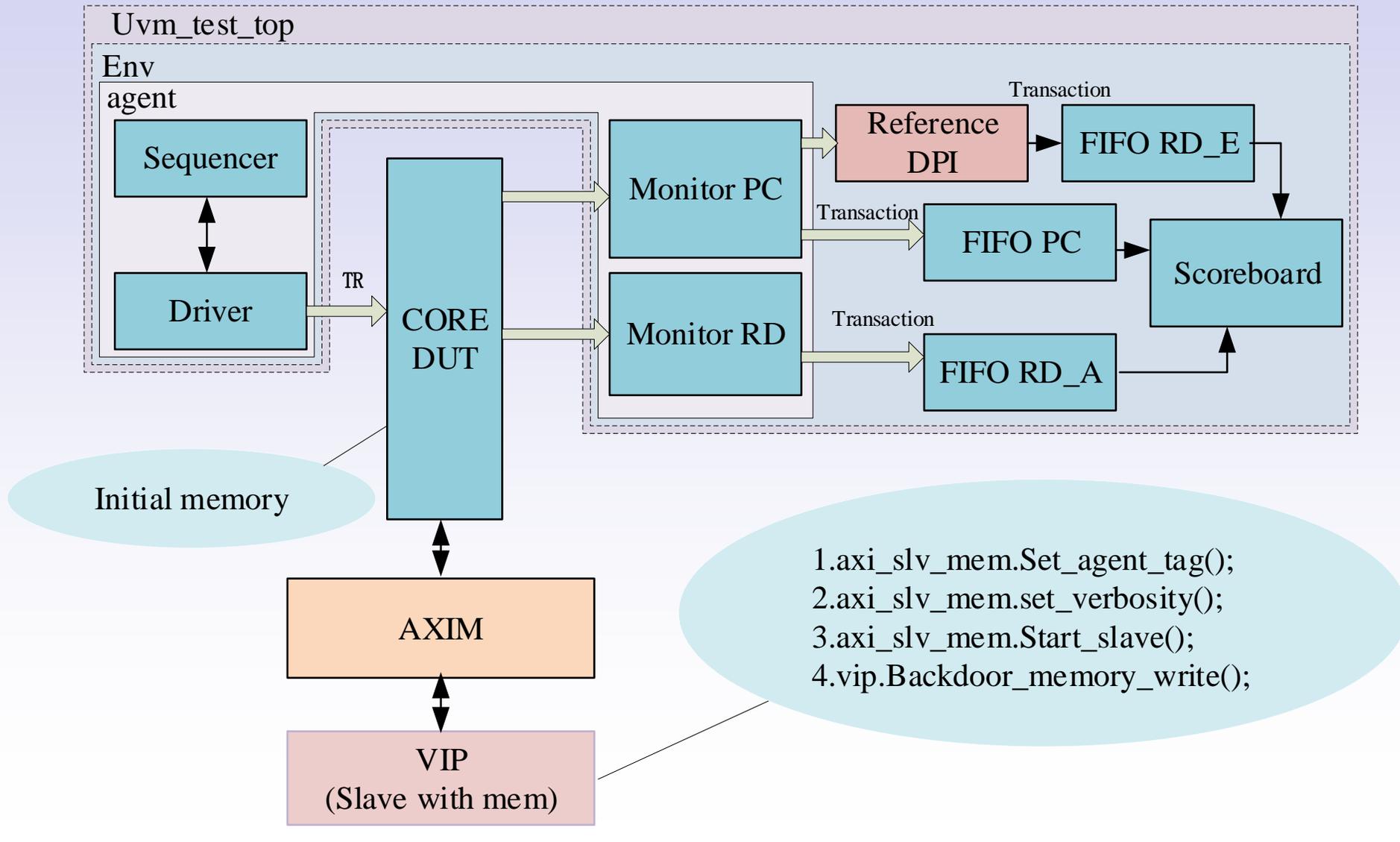
# Egret处理器UVM验证流程



1. Driver用于产生测试用例激励
2. Monitor用于监控设计中信号
3. Refernce model用于产生期望数据
4. Scoreborad用于完成期望值与实际值的对比

# Egret处理器UVM验证平台

Egret\_uvm\_top



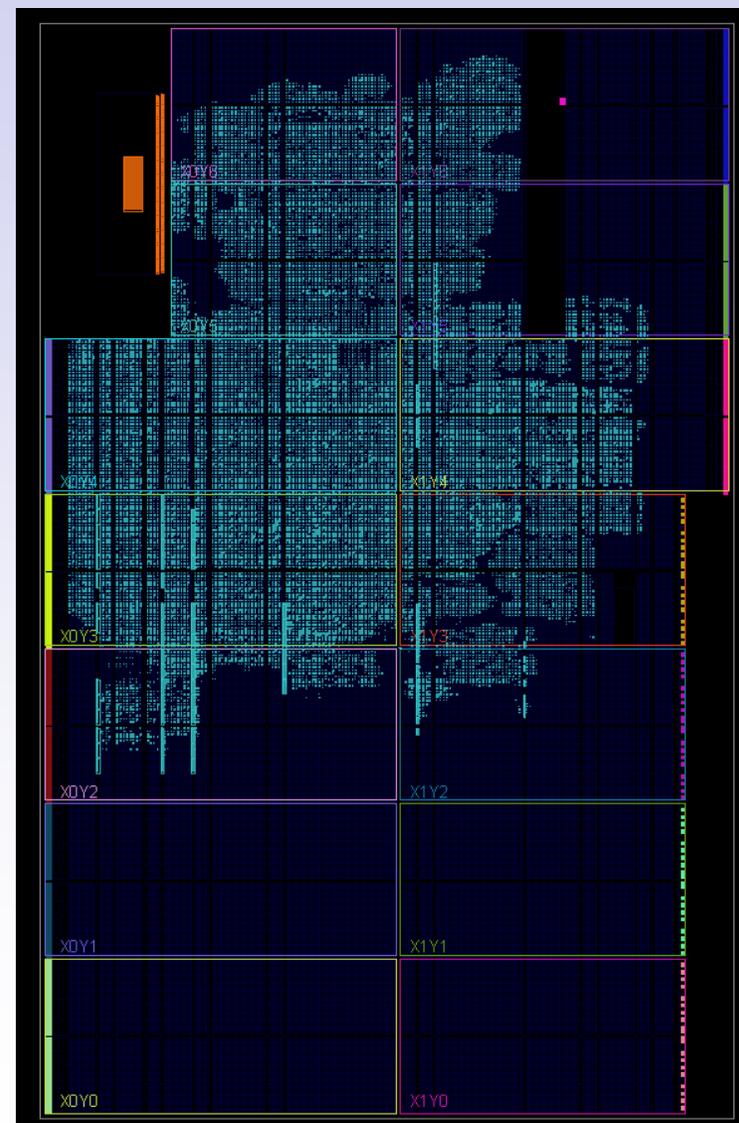
- 1.axi\_slv\_mem.Set\_agent\_tag();
- 2.axi\_slv\_mem.set\_verbosity();
- 3.axi\_slv\_mem.Start\_slave();
- 4.vip.Backdoor\_memory\_write();

## ■ FPGA综合结果

- FPGA工程中，使用ZC706开发板。替换仿真SRAM为FPGA对应的BRAM资源，修改门控时钟，行为级仿真
- 顶层文件中添加时钟和复位等模块，综合并布局布线，门级仿真
- 生成Bitstream并下载到FPGA中，观察ILA波形，调试得到正确的

结果

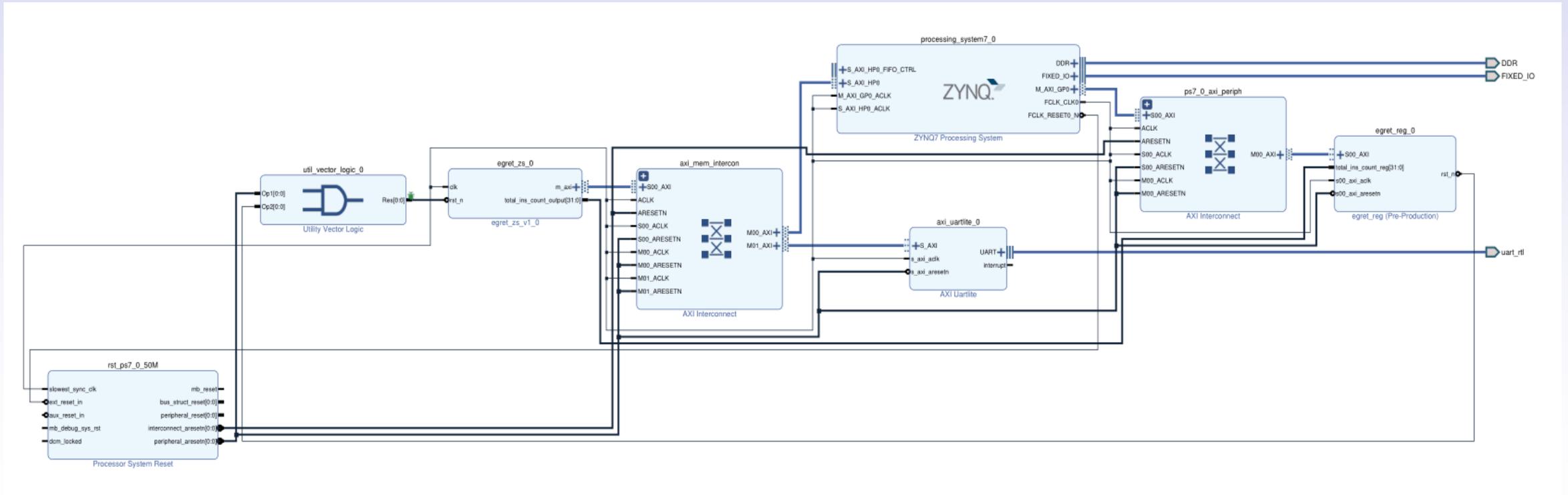
Site Type	Used	Fixed	Available	Util%
Slice LUTs	77111	0	218600	35.27
LUT as Logic	76974	0	218600	35.21
LUT as Memory	137	0	70400	0.19
LUT as Distributed RAM	62	0		
LUT as Shift Register	75	0		
Slice Registers	43460	0	437200	9.94
Register as Flip Flop	43242	0	437200	9.89
Register as Latch	218	0	437200	0.05
F7 Muxes	3557	0	109300	3.25
F8 Muxes	965	0	54650	1.77



# Egret处理器FPGA验证平台

## ■ FPGA验证平台

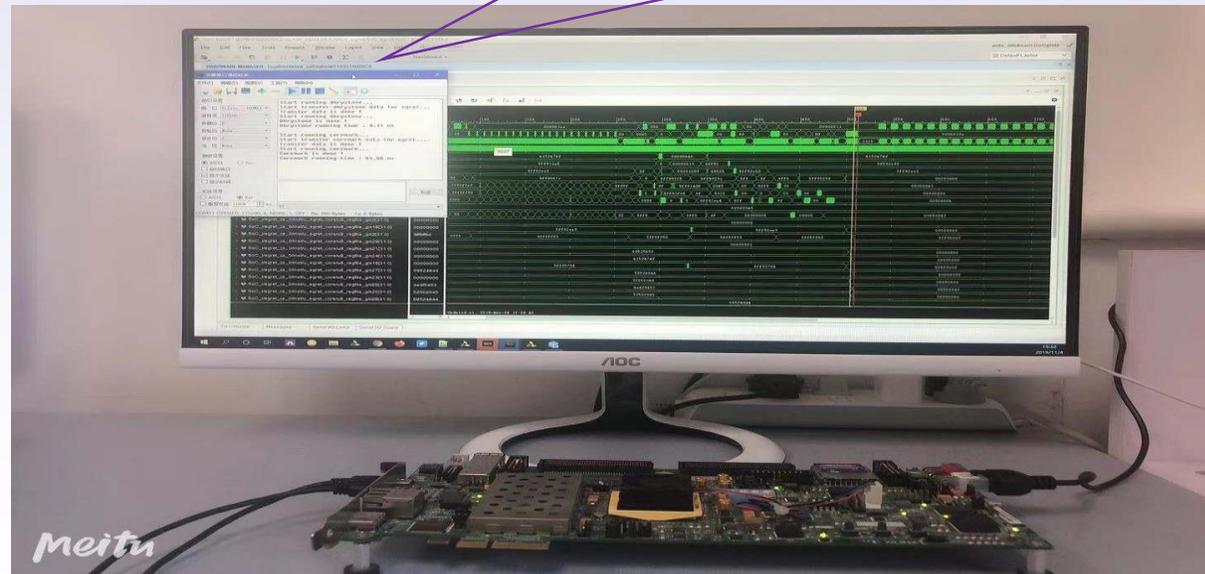
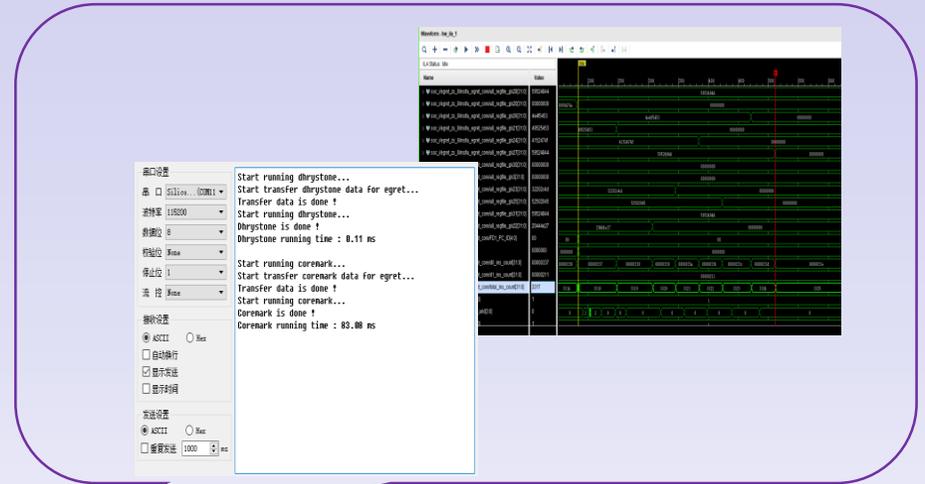
- 封装Egret处理器为AXI4总线接口的IP，构建SoC系统，并添加Zynq、AXI互联，配置寄存器和UART等IP
- 综合，添加Debug调试信号，布局布线，生成Bitstream



# Egret处理器FPGA验证平台

## ■ FPGA验证平台

- 实现Egret处理器的FPGA设计，完成FPGA上板调试，频率100MHz
- 在仿真验证后的RTL代码基础上，完成FPGA版本处理器的开发与调试
- 构建处理器的SoC系统，在FPGA上运行得到正确的结果
- Dhrystone, Coremark, 其他SoC的case均能正确执行，与预期一致
- 可执行其他测试程序，打印调试信息，辅助软件开发



# 目录

1 项目基本情况

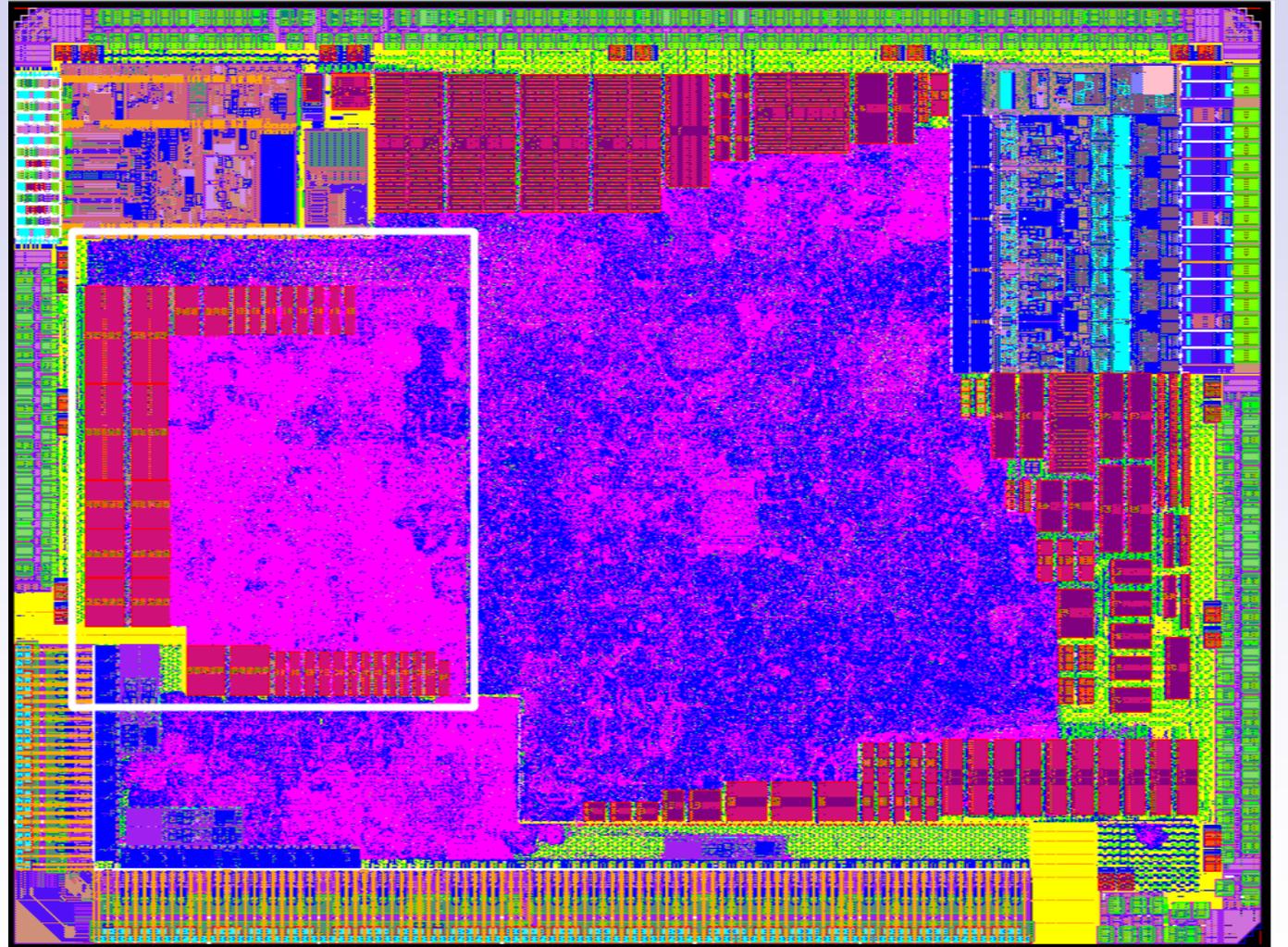
2 Egret处理器内核设计

3 Egret处理器内核验证

4 开发成果 

5 前景展望

- SMIC 40nm LL
- 面积  $1\text{mm}^2$
- Egret后端做成硬核
- Egret替换ARM Cortex-A7
- SoC中其他部分不需要任何  
改动
- C代码重新用RISC-V软件  
工具链编译即可



# Egret vs ARM Cortex-A7

	Egret	ARM Cotex-A7
指令集	RV32IMAFC;可扩展专用指令	ARMv7-A, AArch32
压缩指令	16位压缩指令	32位无压缩
内存管理	PMP和MMU	只有MMU
中断控制器	集成在内核中	需要额外授权, 并联调
浮点单元	集成在内核流水线中	FPU作为协处理器
指令并行度	2	Partial Dual Issue
Dhrystone	1.76 DMIPS/MHz	1.90 DMIPS/MHz
Coremark	2.88 Coremark/MHz	2.57 Coremark/MHz

# 目录

1 项目基本情况

2 Egret处理器内核设计

3 Egret处理器内核验证

4 开发成果

5 前景展望 





- RTL代码12万行，完全自研，从源头上确保处理器芯片的自主可控
- 由单核架构向多核架构递进，支持AI人工智能硬件加速器
- 完全支持RISC-V标准

## 设计完全自研

## 简化开发流程

- 可根据具体的应用需求进行特定的优化，扩展指令功能，满足定制化的需求
- 软硬件开发流程规范，易于实现
- 简化SoC设计，处理器核原位替换



- Egret与ARM A系列性能相当，可广泛应用于移动终端类设备
- 采取时钟门控的方法降低整个芯片时钟树的功耗，支持睡眠，中断唤醒等多种低功耗设计方法

## 高性能低功耗

## 助力RISC-V发展

- Egret已经进行了比较充分的验证，原位替换ARM Cortex-A7
- 向产业界、学术界授权基于RISC-V指令集的系列处理器内核IP，推动RISC-V的发展

# 谢谢！