



基于RISC-V向量拓展的 高性能开源GPGPU设计

(股票代码: 688262)

苏州国芯科技股份有限公司

www.china-core.com





研发背景

➤ 发展背景:

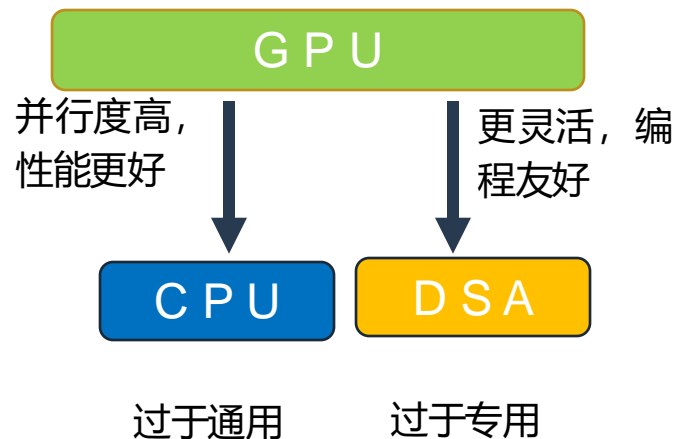
- 通用图形处理单元 (GPGPU) 是高性能并行计算处理器芯片的一种。GPGPU因为其可编程性、易移植性, 面对新兴应用具有更低的跟随成本, 所以始终占据着高性能计算市场最大的份额。

➤ 痛点:

- GPU研发资源极度分散, 内卷及其严重。
- GPU研发软件生态不够统一, 各自封闭。

➤ 解决以上痛点

- 采用开源的RSIC-V架构实现GPU设计解决生态问题。
- 基于LLVM开发编译器, 结合linker、runtime library、math library实现对OpenGL、OpenGL ES、Vulkan和OpenCL C的完整支持。



GPU的核心就是生态, 以上痛点也成为了国内GPU长远发展的瓶颈!



GPGPU主要技术规格

➤ 指令集架构

- RISC-V : RV32IMA_zicsr_zfinx + Vector 扩展
- 自定义线程束的分支和同步、多线程束执行指令

➤ 特性

- 统一分配SM线程调度器CTA
- 每个SM支持最大32线程
- 支持Private、Share、Global存储模型
- 支持256向量和64标量寄存器堆
- 支持barrier、fence同步指令
- 支持endprg、mask线程控制指令
- 可配置Tensor Core

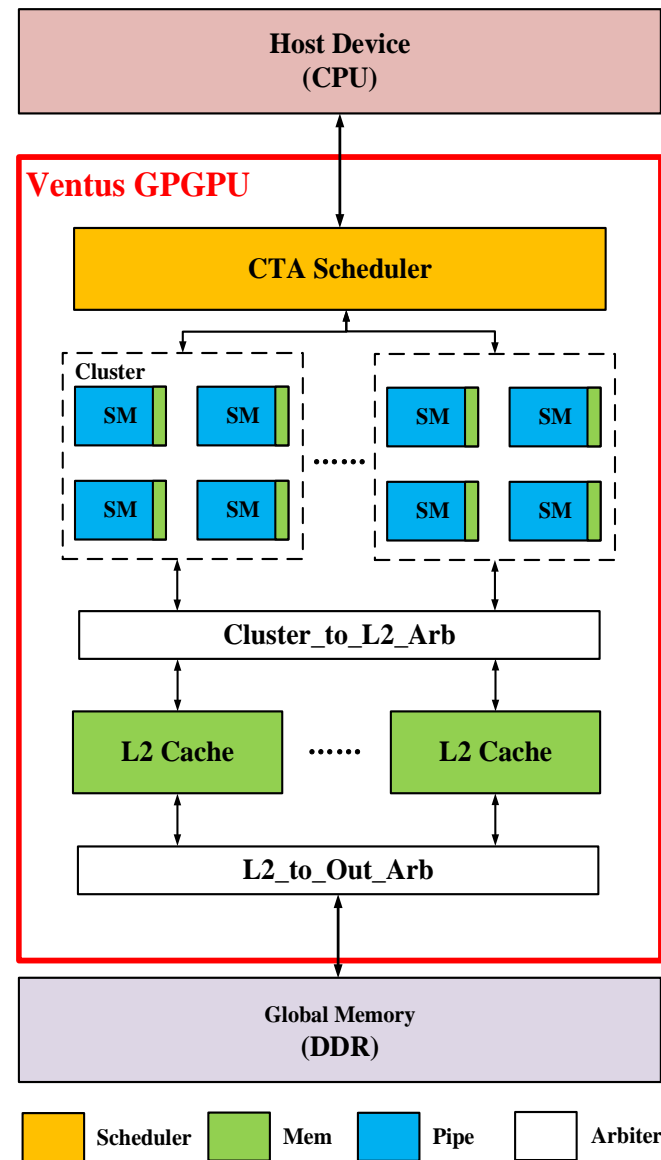
➤ 缓存架构

- 在RVWMO的基础上，“乘影”进一步部署了释放连贯性指导的缓存一致性（RCC）。让SM私有缓存间具备一致性功能的同时，避免了硬件一致性协议带来的高昂硬件成本和运行时带宽开销

➤ 工程化RTL设计

- Verilog语言设计

扩展	指令类型	GPGPU
V(zve32f)	配置指令	部分支持
	加载和存储	支持
	整型指令	支持
	浮点指令	支持fp32
I	基本指令	支持
M	乘法指令	支持
A	原子指令	支持
zicsr	CSR操作	支持
F(zfinx)	单精度浮点指令	支持





GPGPU主要技术规格

➤ 主要功能模块

■ 1. 总调度器设计 (CTA Scheduler) :

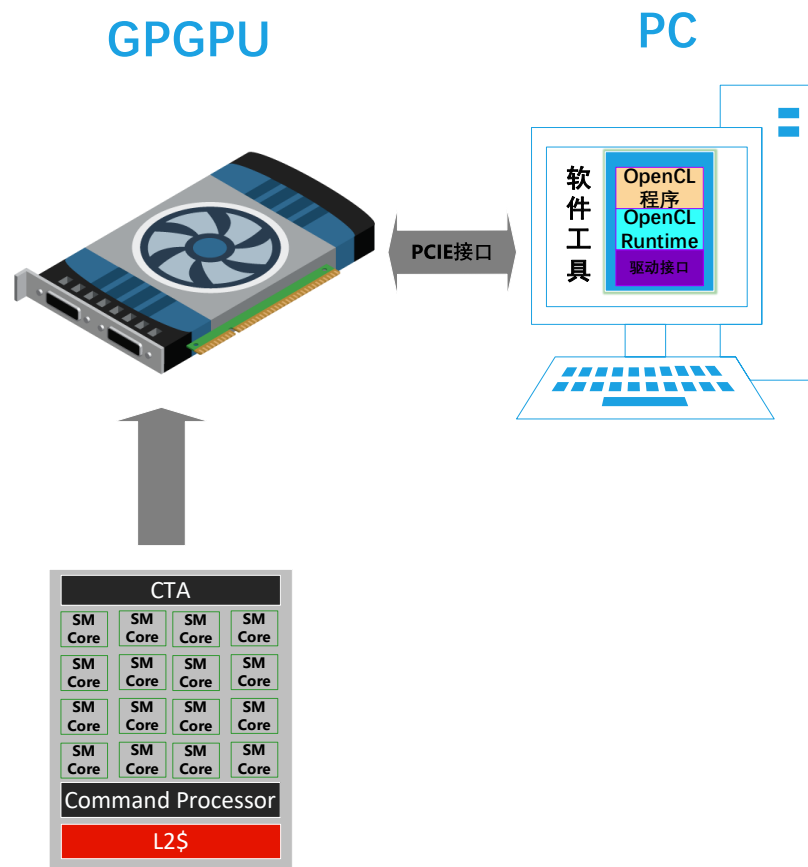
- 负责以workgroup为单位接收CPU分配的总任务，并统计资源
- 将任务以warp为单位逐个分配到空闲的SM上

■ 2. 流多处理器设计 (SM) :

- 前端包括：取指、译码、指令缓冲、操作数收集、发射、记分牌、warp调度（单指令操作）
- 后端包括：ALU、vALU、vFPU、MUL、LSU、CSR、SFU、SIMT-stack、TC、warp控制、写回（多线程执行）

■ 3. 内存系统设计 (Cache) :

- 单个SM内部：Regfile、L1 DCache、L1 ICache、Share Mem
- 多个SM共享L2 Cache

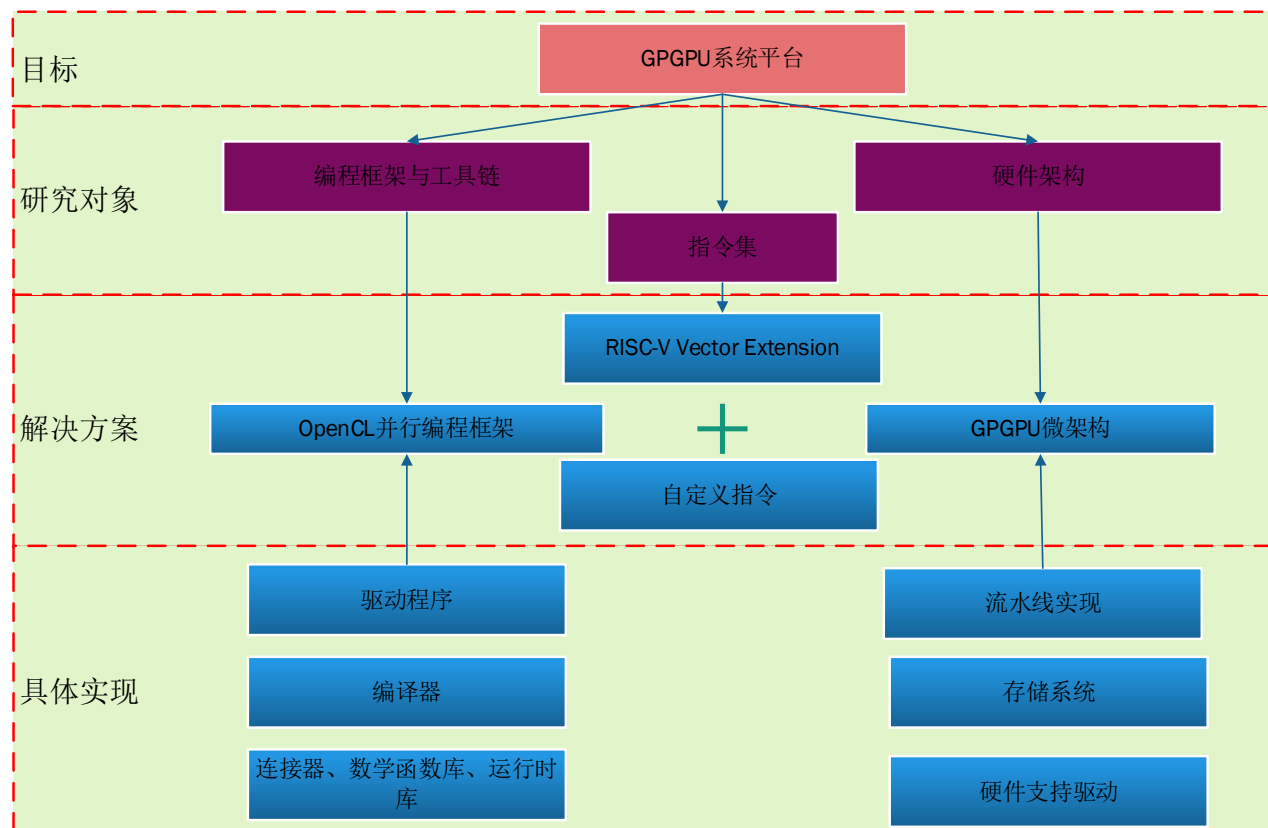




GPGPU 研发平台简介

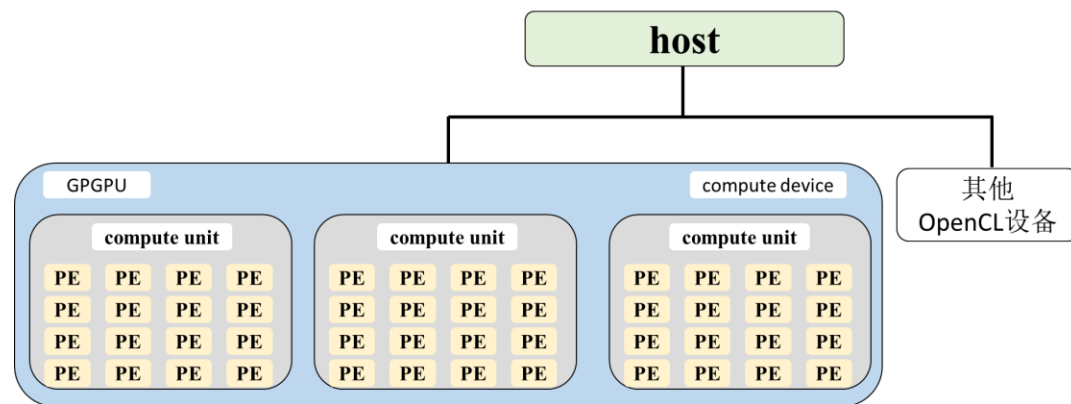
➤ 研发框架

- 围绕开源RISC-V向量扩展搭建GPGPU平台，完成包括开源指令集、微架构设计、OpenCL软件工具链、专用张量计算单元在内的全套实现。
 - 参照开源软硬件项目“乘影” GPGPU平台而设计
 - 具有高度可定制性和可扩展性，具有完整的开源编译器，驱动程序和运行时软件堆栈
 - 是支持基于RISC-V ISA扩展的GPGPU

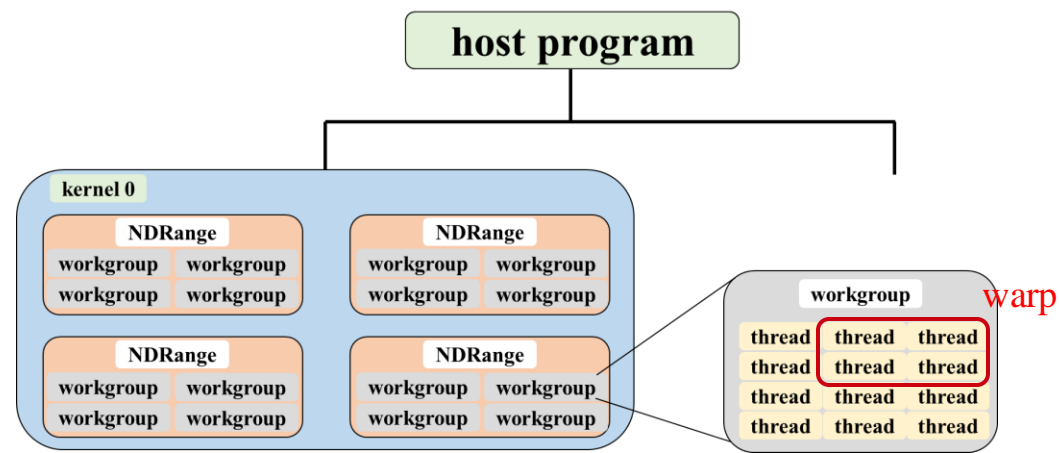
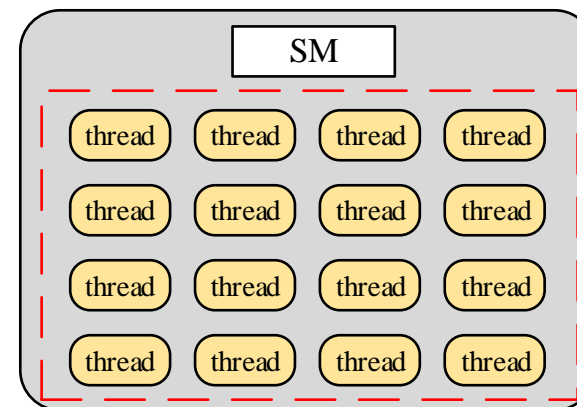




OpenCL编程模型



OpenCL平台模型



OpenCL执行模型

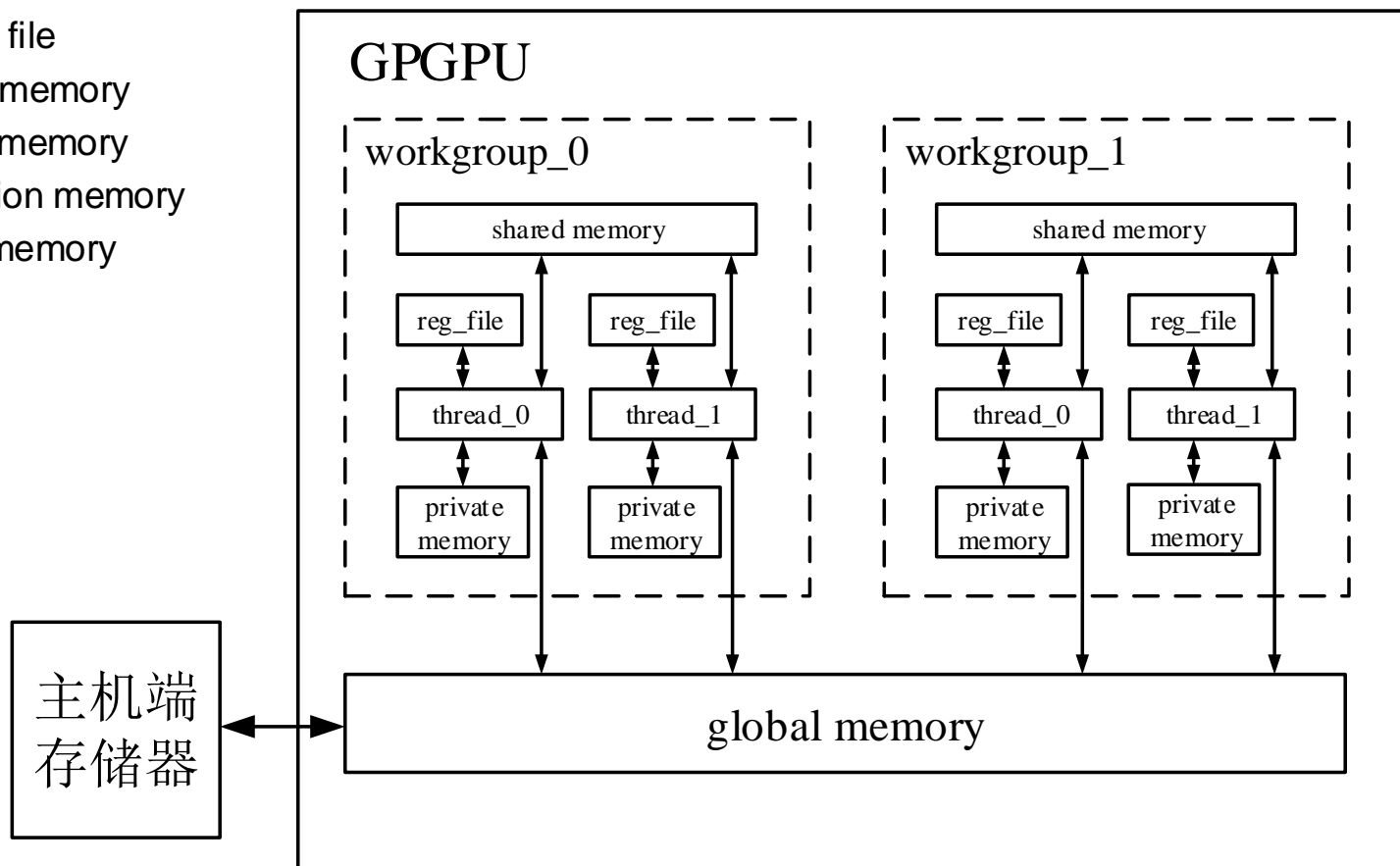
- workgroup被拆分成warp的形式在CU/SM内执行（1个SM执行1个workgroup）
- warp：单个SM内在硬件上有多少个线程
- 单个SM最多同时执行8个warp
- 同一warp中，所有线程运行相同的指令
- SM可以通过warp切换来掩藏延时



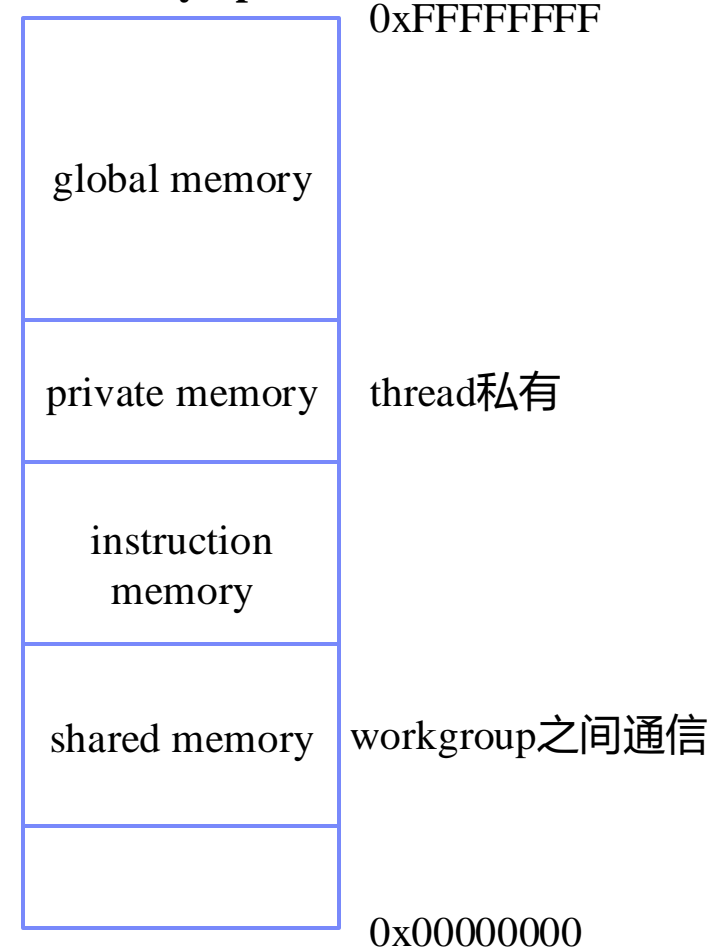
OpenCL存储模型

➤ 从软件视角看，GPGPU内部的存储资源包括：

- register file
- shared memory
- private memory
- instruction memory
- global memory



Memory Space

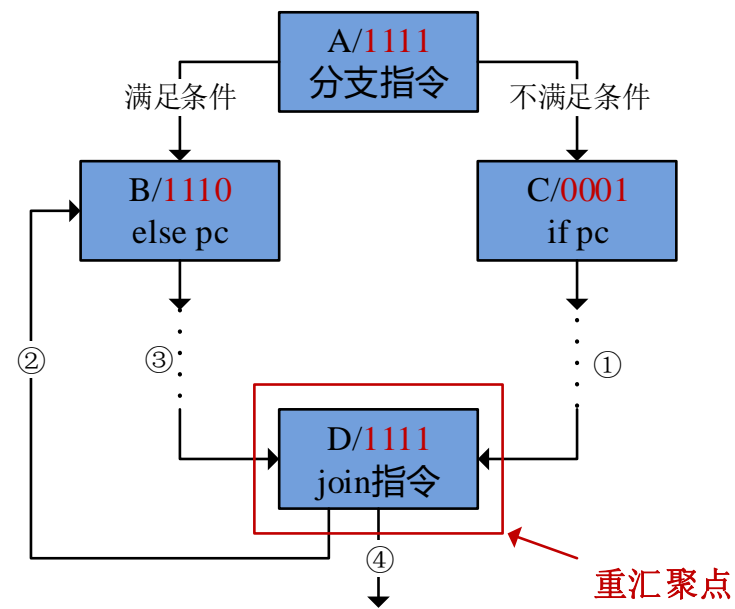
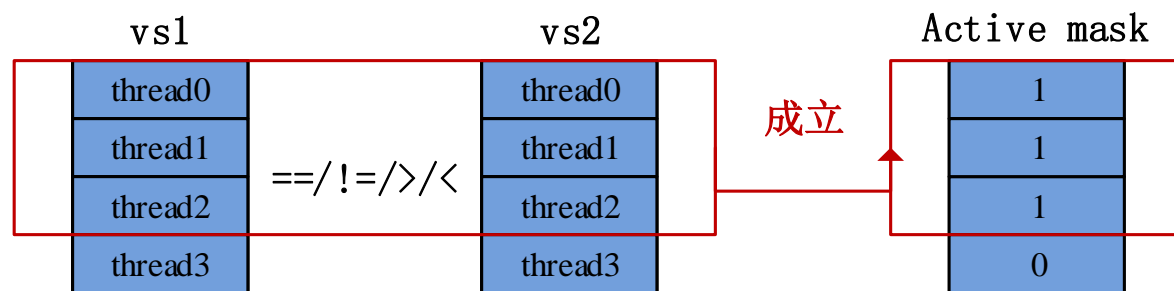




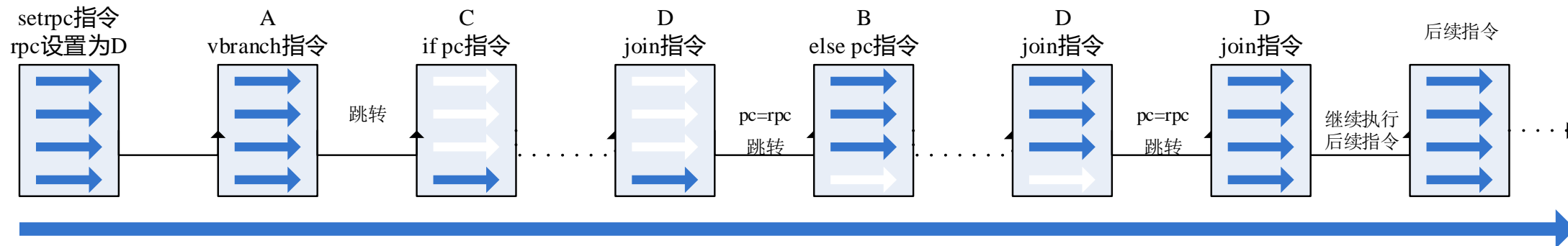
自定义指令介绍

➤ GPGPU支持SIMT执行模型，由硬件来管理分支

- 分支控制指令：vbeq/vben、vblt、vbge
- 线程分支汇聚指令：join
- 重汇聚pc设置指令：setrpc



➤ 发生分支时，GPGPU的执行过程





自定义指令介绍

➤ 同步和任务控制指令

- 栅栏指令: barrier
- 线程退出指令: endprg

示例:

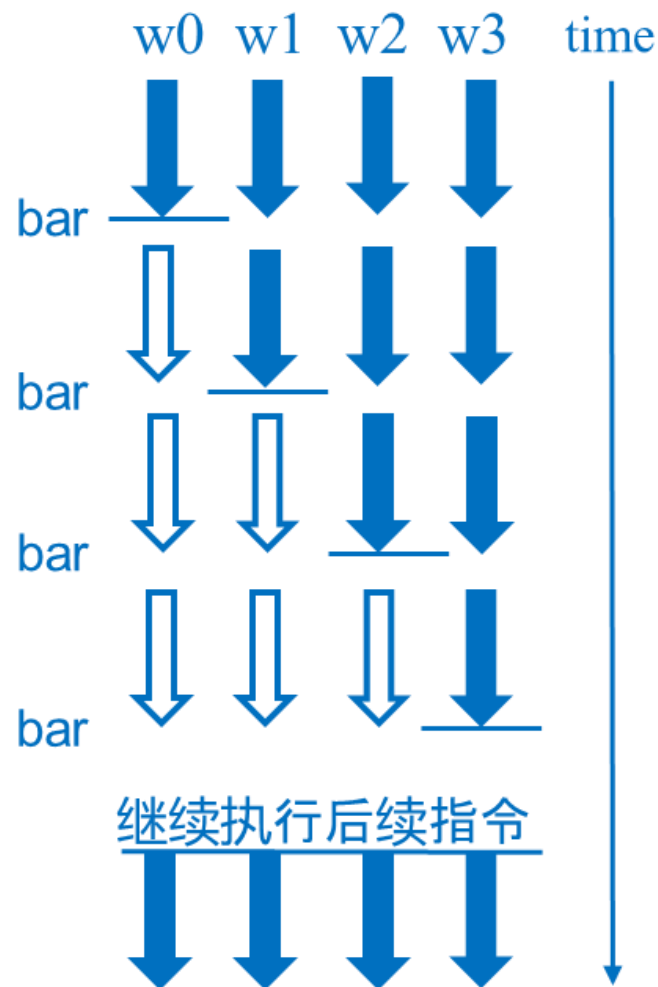
- barrier x0, x0, imm
- endprg x0, x0, x0: 用于指示当前warp已经执行结束

➤ 寄存器扩展指令: GPGPU需要索引比CPU更多的寄存器

- 立即数编码扩展: regexti
- 寄存器编码扩展: regext

示例:

- regext x0, 0b000_001_000_010
- vadd.vx v16, v20, x8
- 两条指令的执行结果等价于vadd.vx v80, v20, x40





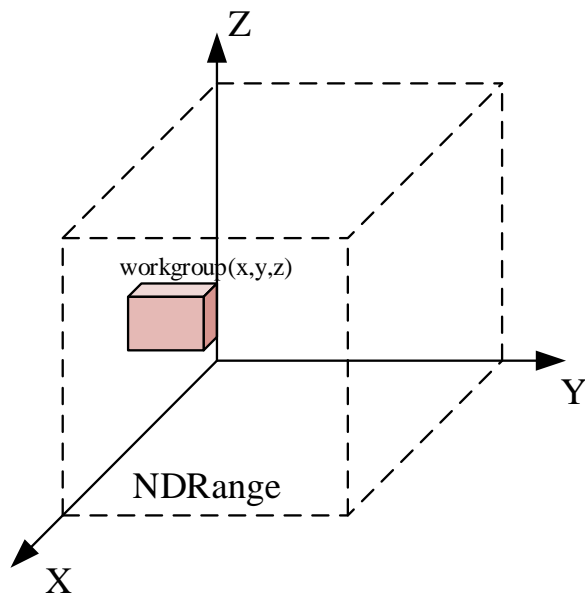
自定义指令介绍

➤ 访存指令

- `vlw12.v/vsw12.v`: 支持 $\text{addr} = \text{rs1} + \text{imm}$, 而非RVV的 $\text{rs1} + \text{vs1}$
- `vlw.v/vsw.v`: 用于访问线程私有内存, 与地址偏移`thread_idx`有关

示例:

- `vlw12.v vd, offset(vs1)` 功能: $\text{vd} = \text{mem}[\text{addr}]$, $\text{addr} = \text{vs1} + \text{offset}$
- `vlw.v vd, offset(vs1)` 功能: $\text{vd} = \text{mem}[\text{addr}]$, $\text{addr} = (\text{vs1} + \text{imm}) * \text{num_thread_in_wg} + \text{thread_idx} + \text{csr_pds}$



`thread_idx`:

warp0	warp1	warp2	warp3
0	31 32	63 64	95 96 127



自定义指令介绍

➤ 计算指令

- vadd12.v: 带12位立即数的向量帧数加减指令
- vftta.vv: 张量计算指令

示例:

- vadd12.vi vd, vs1 功能: $vd = vs1 + imm$
- vftta.vv vd, v2, v1, v0.mask 功能: $vd = vs1 \text{ conv } vs2 + vd$

➤ 5*5的输入特征图与一个3*3的卷积核进行卷积: 转换为矩阵乘积进行计算

0	0	0	0	4	3	0	1	3
0	0	0	4	3	1	1	3	0
0	0	0	3	1	0	3	0	0
0	4	3	0	1	3	0	4	3
4	3	1	1	3	0	4	3	2
3	1	0	3	0	0	3	2	0
0	1	3	0	4	3	0	0	0
1	3	0	4	3	2	0	0	0
3	0	0	3	2	0	0	0	0

 \times

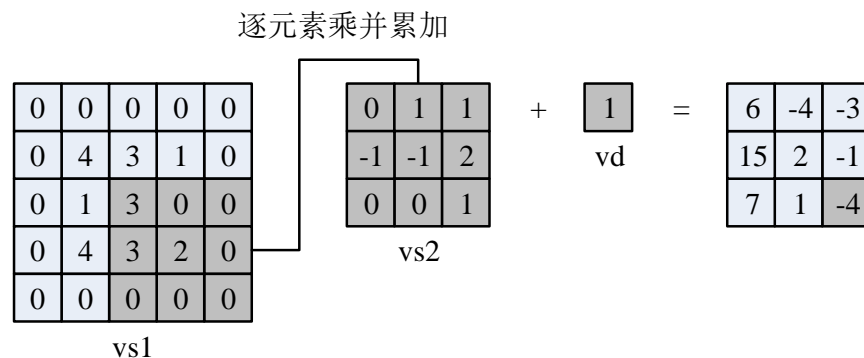
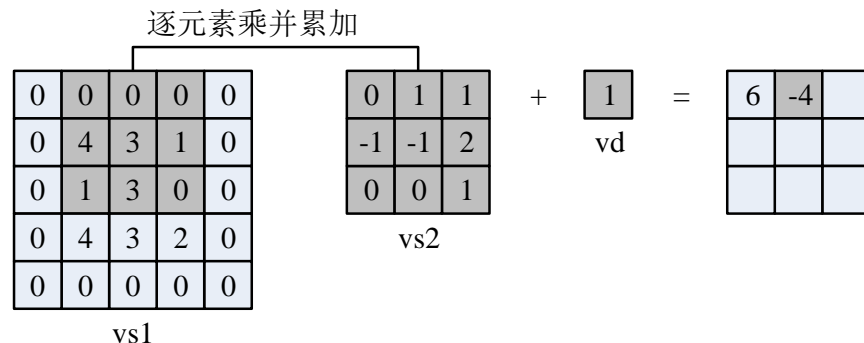
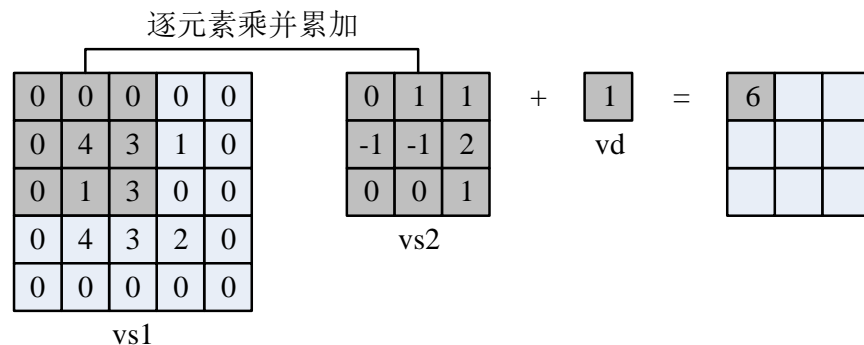
0
1
1
1
-1
-1
2
0
0
1

 $+$

1
1
1
1
1
1
1
1
1
1

 $=$

6
-4
-3
15
2
-1
7
1
-4

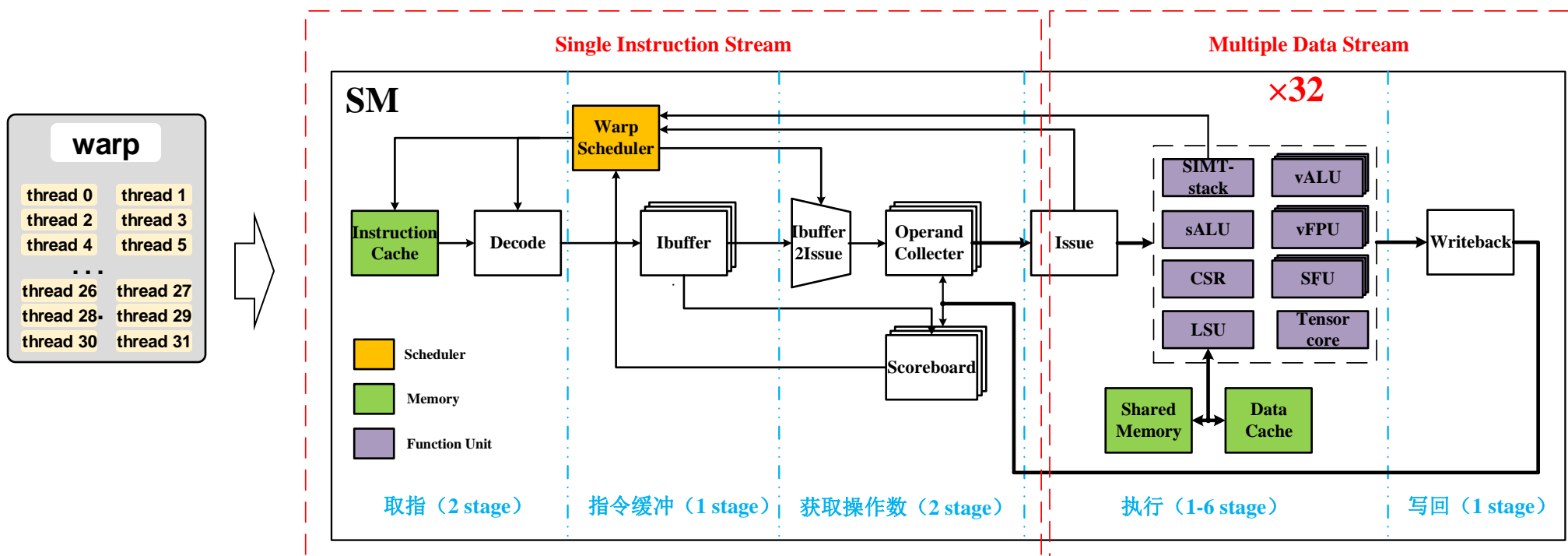




GPGPU 硬件架构简介

➤ SM流多处理器设计

- 支持多warp调度的RISC-V向量处理器
- 单个SM支持最多8个warp同时调度
- 单个warp支持最多32个向量同时执行

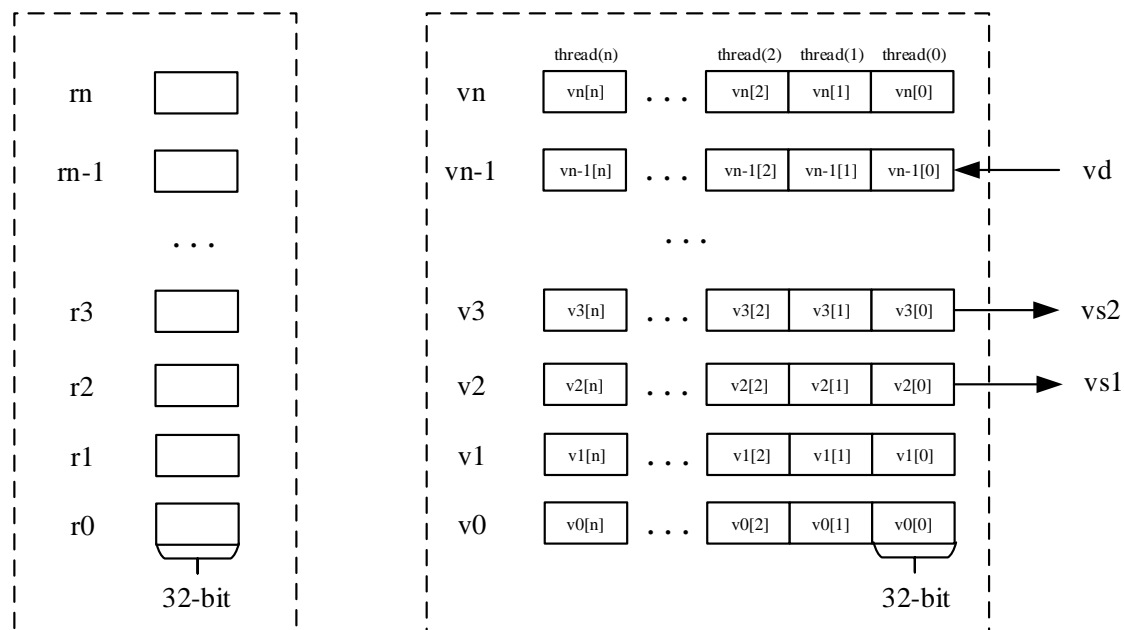




GPGPU 硬件架构简介

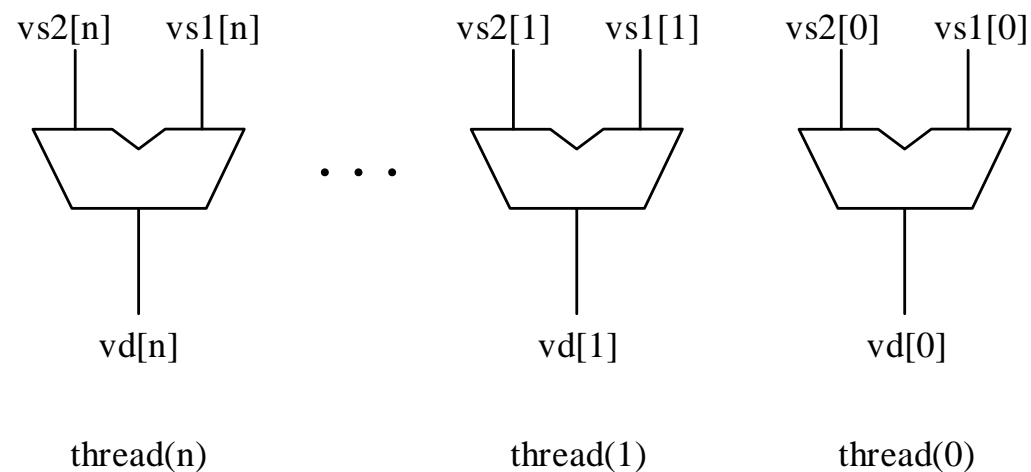
➤ 寄存器堆

- SM内部有1024个sgpr和1024个vgpr
- 每个warp最多拥有256个sgpr和256个vgpr
- sgpr为warp内所有线程共有
- vgpr对应warp内各线程私有gpr



➤ 向量指令执行原理

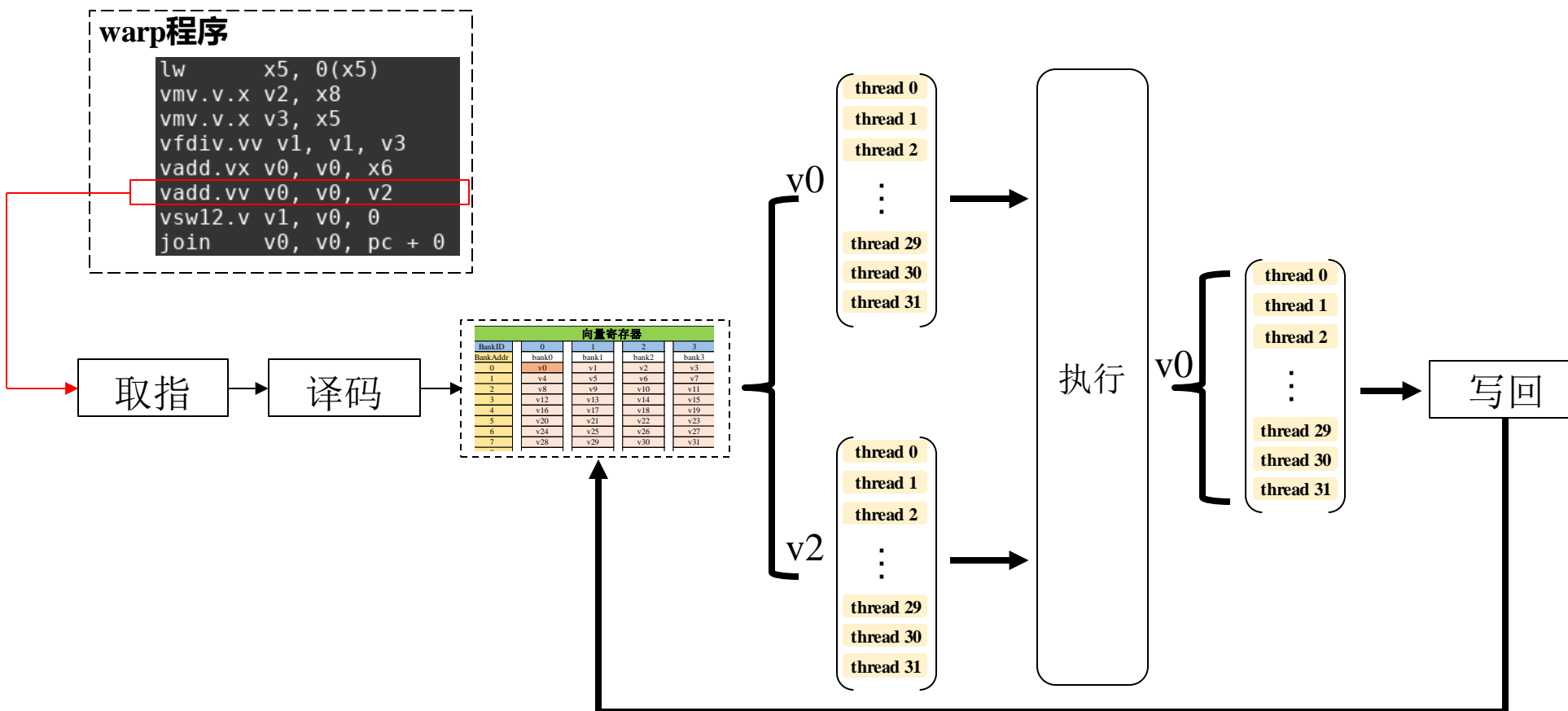
- 指令发送到执行单元后，按照SIMD的方式执行
- 从单个线程的角度看，执行的是标量指令
- 执行结束后，所有线程的结果同时写回某一个向量寄存器中





GPGPU 硬件架构简介

- 一段包含多个thread的RVV程序

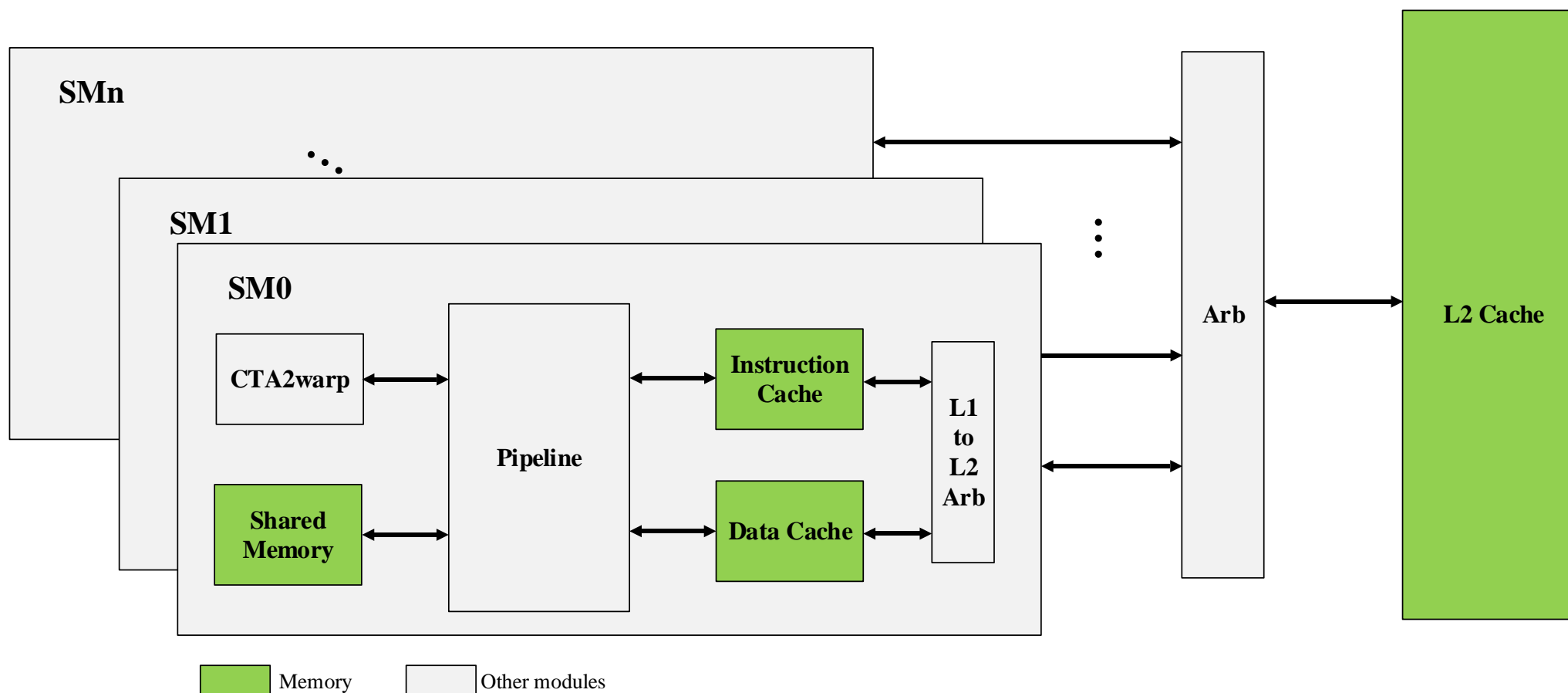




GPGPU 硬件架构简介

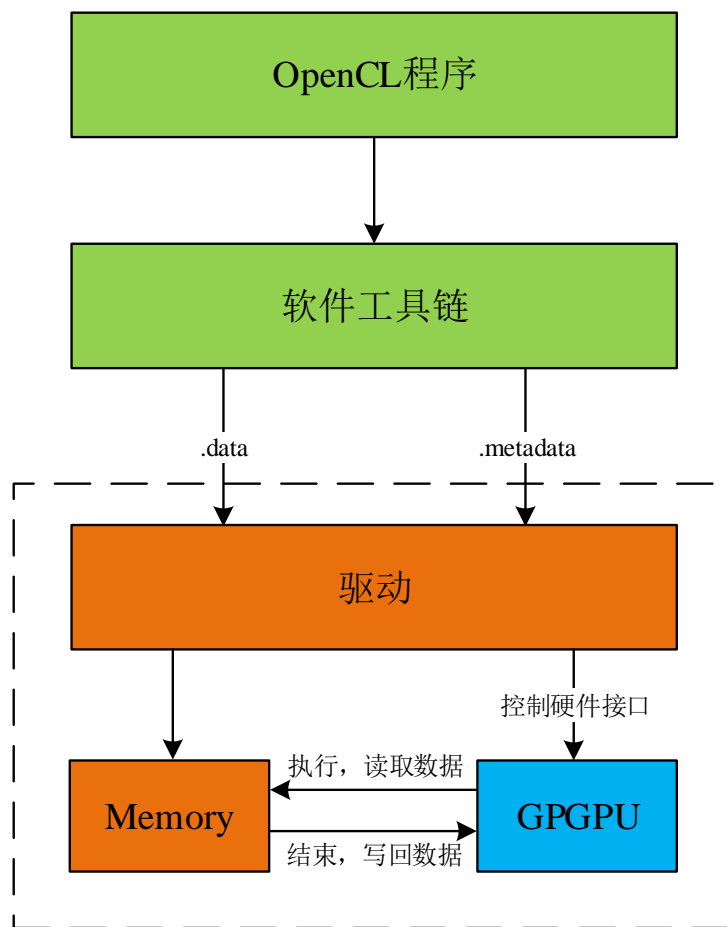
➤ 硬件存储结构

- shared memory用于仅用于SM内部通信
- private memory实际被保存在global memory中





GPGPU 测试仿真



测试平台:

- ① Testbench (Verilog) 模仿驱动行为, 向GPGPU发起任务请求。
- ② GPGPU和Memory之间采用AXI4接口
- ③ **.data**: 硬件运行的指令和数据
- ④ **.metadata**: 指令和数据存放的地址, 以及控制硬件接口所要包含的信息 (wg size, pds addr等)

Pattern分类:

- ① 针对不同的测试用例要编写不同的Testbench
- ② 不断添加边界测试用例



GPGPU 测试举例

➤ 以Gaussian消元举例

$$\begin{cases} -0.6x_1 - 0.5x_2 + 0.7x_3 + 0.3x_4 = -0.85 \\ -0.3x_1 - 0.9x_2 + 0.3x_3 + 0.7x_4 = -0.68 \\ -0.4x_1 - 0.5x_2 - 0.3x_3 + 0.80 = 0.24 \\ -0.1x_2 + 0.2x_3 + 0.9x_4 = -0.53 \end{cases}$$

开始执行前, 将这个矩阵存入global mem

增广矩阵

$$\left[\begin{array}{cccc|c} -0.6 & -0.5 & 0.7 & 0.3 & -0.85 \\ -0.3 & -0.9 & 0.3 & 0.7 & -0.68 \\ -0.4 & -0.5 & -0.3 & -0.8 & 0.24 \\ 0.0 & -0.1 & 0.2 & 0.9 & -0.53 \end{array} \right]$$

① 提取系数: 0.5, 0.67, 0

kernel0: GPGPU调用4个thread来完成计算,
结果存入global mem

② 消元

$$\left[\begin{array}{cccc|c} -0.6 & -0.5 & 0.7 & 0.3 & -0.85 \\ 0.0 & -0.65 & -0.05 & 0.55 & -0.255 \\ 0.0 & -0.167 & -0.76 & -1.0 & 0.8 \\ 0.0 & -0.1 & 0.2 & 0.9 & -0.53 \end{array} \right]$$

重复上述步骤

化成上三角矩阵

$$\left[\begin{array}{cccc|c} -0.6 & -0.5 & 0.7 & 0.3 & -0.85 \\ 0.0 & -0.65 & -0.05 & 0.55 & -0.255 \\ 0.0 & 0.0 & -0.754 & -1.14 & 0.872 \\ 0.0 & 0.0 & 0.0 & 0.501 & -0.251 \end{array} \right]$$

kernel1: GPGPU调用16个thread来完成计算, 结果存入global mem

最终结果存入global mem



GPGPU 评估



➤ 性能评估

硬件配置	SM	1
	Wrap	8
	Thread	32
	执行单元	32
	L1 Icache	512B
	L1 Dcache	512B
	L2 Cache	64B
工艺	T28, 9T cell, HVT & SVT	
综合频率	620MHz	
综合面积	1.93mm ²	

➤ 常用测试Pattern示例

测试集	warp/thread	是否通过	执行周期	说明
vecadd: 向量加	4w16t	Pass	1800	64个元素向量加
	4w8t	Pass	2696	32个元素向量加
	4w32t	Pass	2164	128个元素向量加
	8w4t	Pass	2899	32个元素向量加
matadd: 矩阵加	1w32t	Pass	2801	4*4矩阵相加
	1w16t	Pass	2500	4*4矩阵相加
	2w8t	Pass	2640	4*4矩阵相加
	4w4t	Pass	4054	4*4矩阵相加
nn: 最近邻内插法	2w16t	Pass	2031	19个点中找最近的5个点
	4w8t	Pass	4033	28个点中找最近的5个点
	4w16t	Pass	2269	53个点中找最近的5个点
	8w4t	Pass	3382	19个点中找最近的5个点
	8w8t	Pass	2038	53个点中找最近的5个点
gaussian: 高斯消元	1w16t	Pass	10151	四元一次方程组消元
	2w8t	Pass	11670	四元一次方程组消元
	4w4t	Pass	11537	四元一次方程组消元
	4w8t	Pass	15940	五元一次方程组消元
bfs: 宽度优先算法	2w16t	Pass	20938	
	4w8t	Pass	22730	
	4w32t	Pass	36114	
	8w4t	Pass	40888	



最后，愿国芯与您携手合作，共创双赢！

重要声明：本公司致力于为客户持续提供自主可控高可靠芯片产品，产品规格如有变化，恕不另行通知，谨以最新技术资料及线下咨询为准。