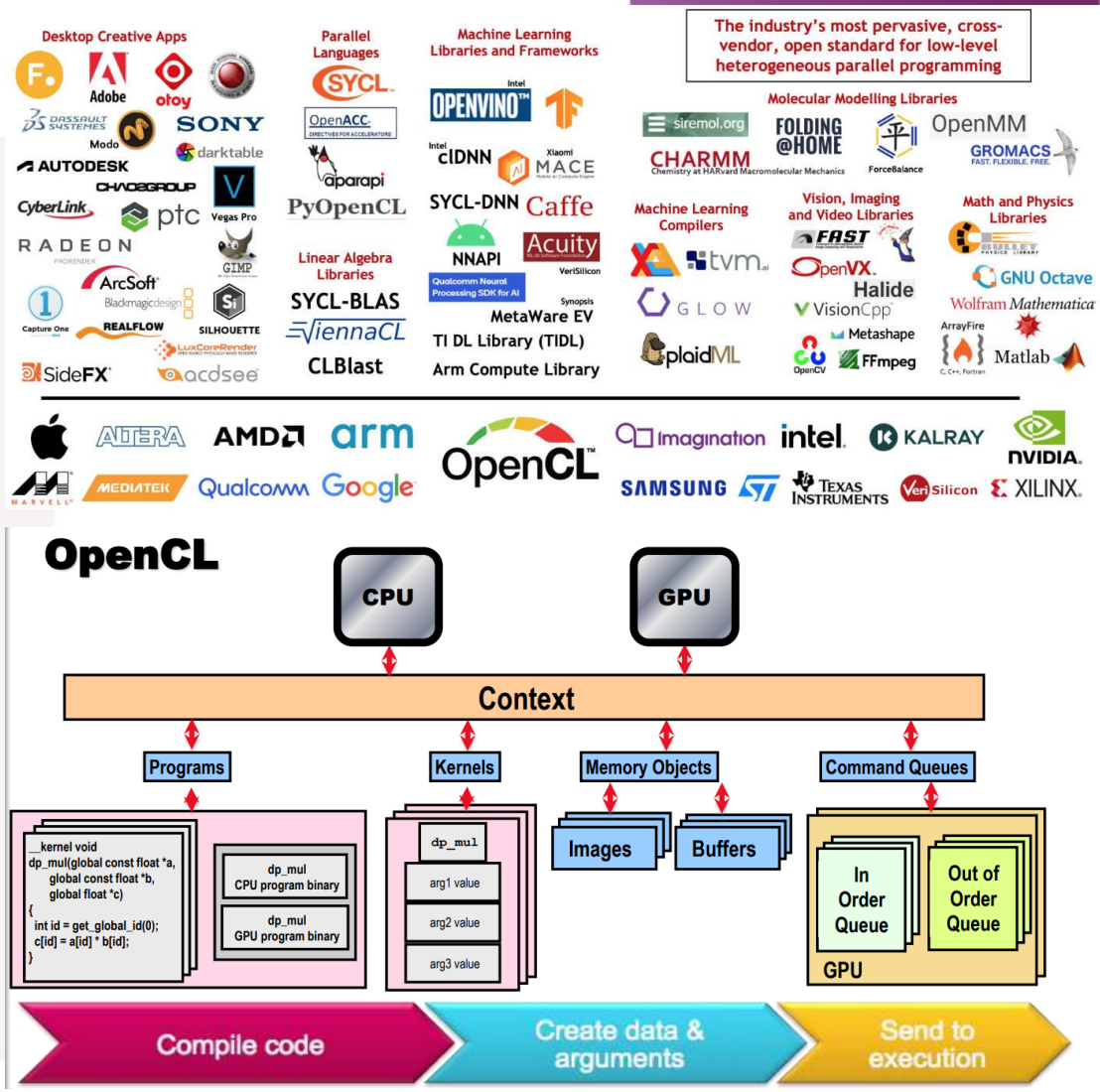
“乘影” 软件工具链

杨泽夏

清华大学集成电路学院

“乘影” GPGPU软件工具链

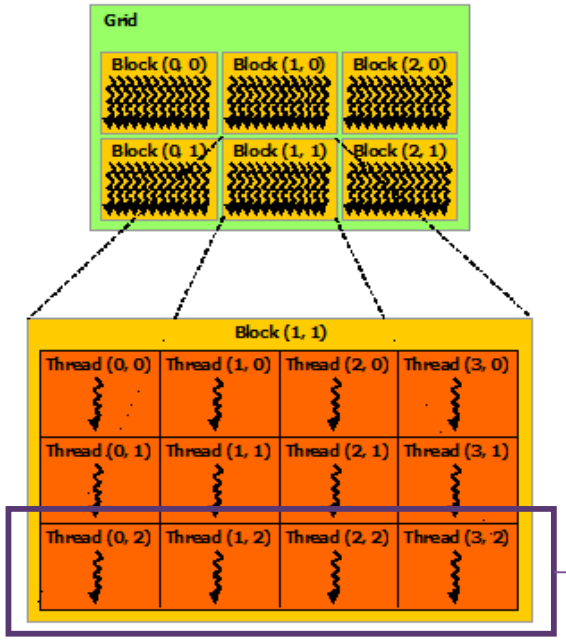
- OpenCL编程模型(kernel)
- 驱动程序框架(platform)
- 仿真器

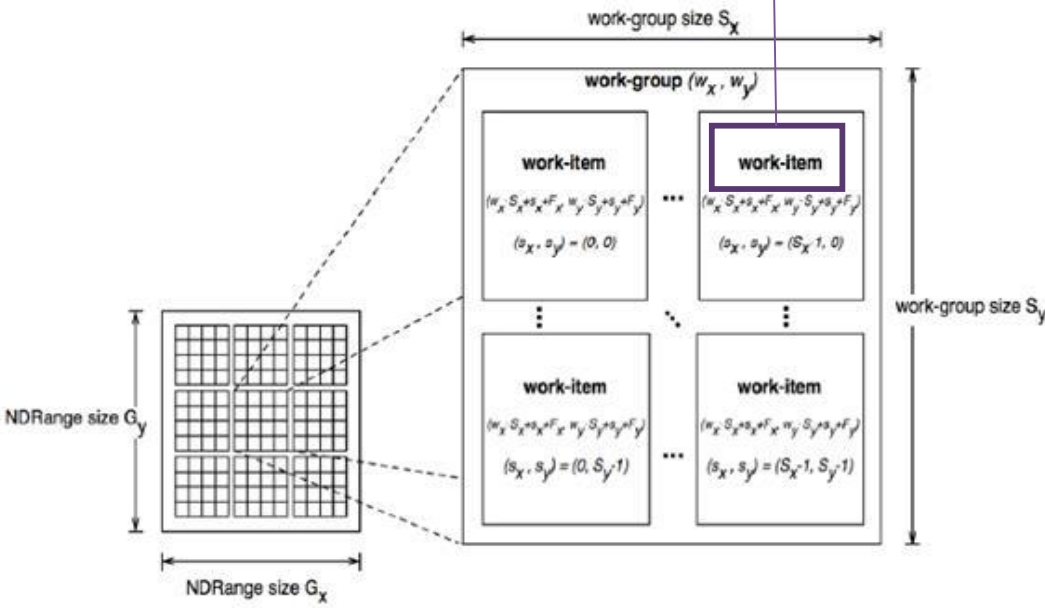


*图片来自OpenCL Overview - The Khronos Group Inc

“乘影” GPGPU软件工具链

- OpenCL编程模型(kernel)：
  - NDRange(Grid) – WorkGroup(CTA/Block) – workitem(thread)
  - 程序员声明需要的thread数目，然后对单个thread的行为进行描述
  - warp由一定数目的thread组成，硬件将WorkGroup中的thread组织起来，以warp为单位映射到硬件上执行

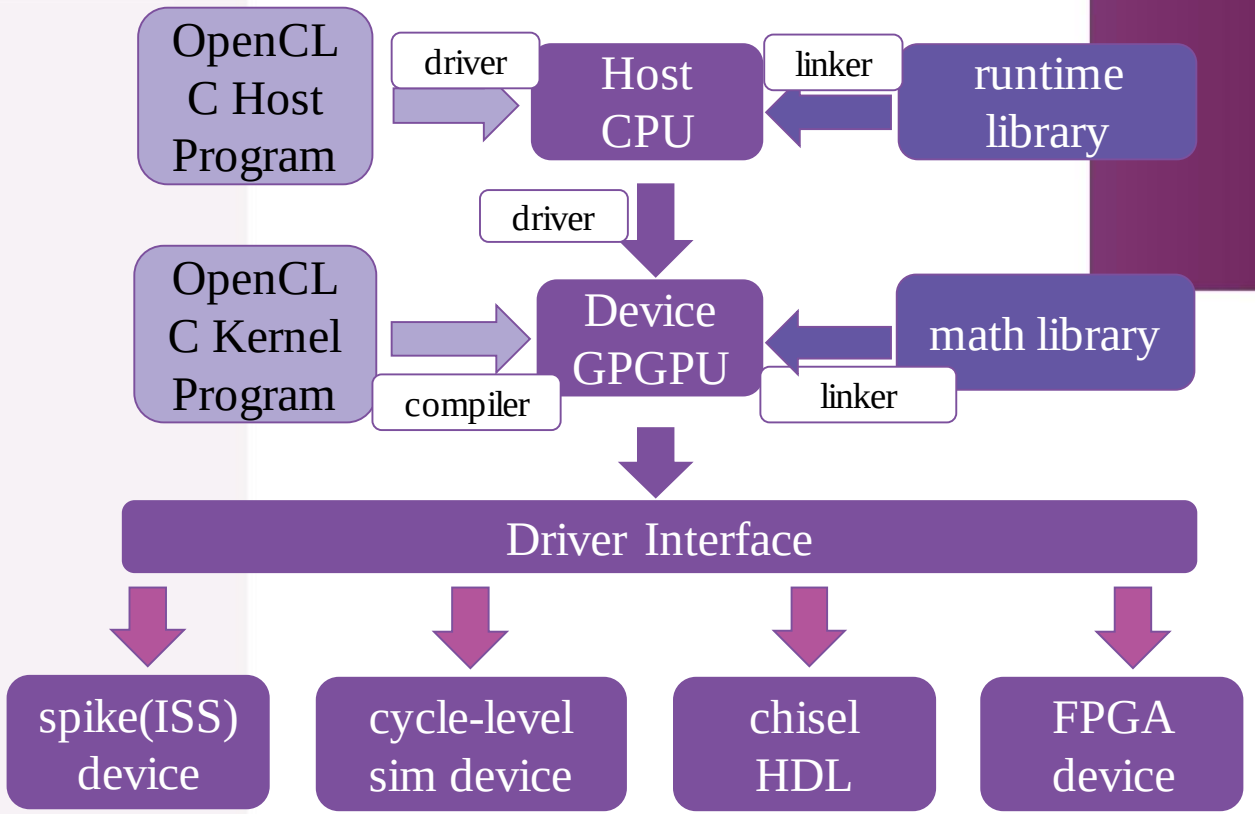




*图片来自Programming Guide :: CUDA Toolkit Documentation (nvidia.com)和The OpenCL Specification (khronos.org)

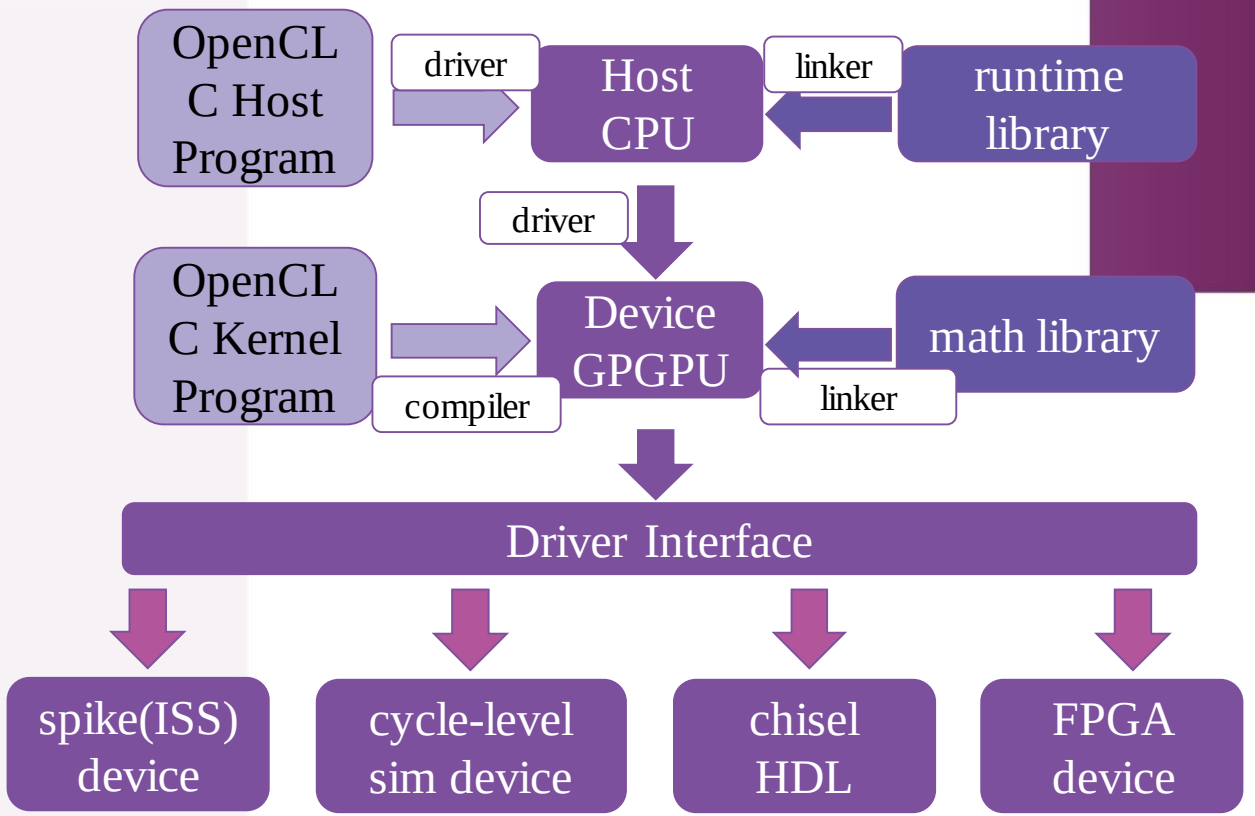
“乘影” GPGPU软件框架

- 驱动程序框架(platform)
  - 基于开源Pocl框架开发driver
  - 基于LLVM实现compiler，结合linker、runtime library、math library实现对OpenCL C 2.0的完整支持
  - 具有多个硬件设备或仿真器，实现了不同的设备驱动程序



软件工具链功能

- 设计和实现平台发现机制：开发平台驱动程序发现系统中可用的OpenCL平台。检测和识别支持OpenCL的硬件设备、驱动程序和运行时环境。
- 对设备进行初始化和配置：计算单元、内存分配和配置缓存等。
- 编译和优化支持：设备驱动程序接入编译器，将OpenCL代码转化为设备可执行的指令。



软件工具链功能 – 实现OpenCL API

- 支持OpenCL函数调用：实现OpenCL API中定义的函数调用，保证程序员能正常使用API 编写OpenCL程序
- 内存管理：实现OpenCL API中的内存管理函数，以便应用程序能够分配、释放和传输数据到设备的内存。
  - 涉及到设备和主机之间的内存传输和数据同步操作
- 同步和事件管理：实现OpenCL API中的同步和事件管理函数，支持进行同步和事件通信。
  - 等待事件完成、事件间的依赖关系、事件状态查询等

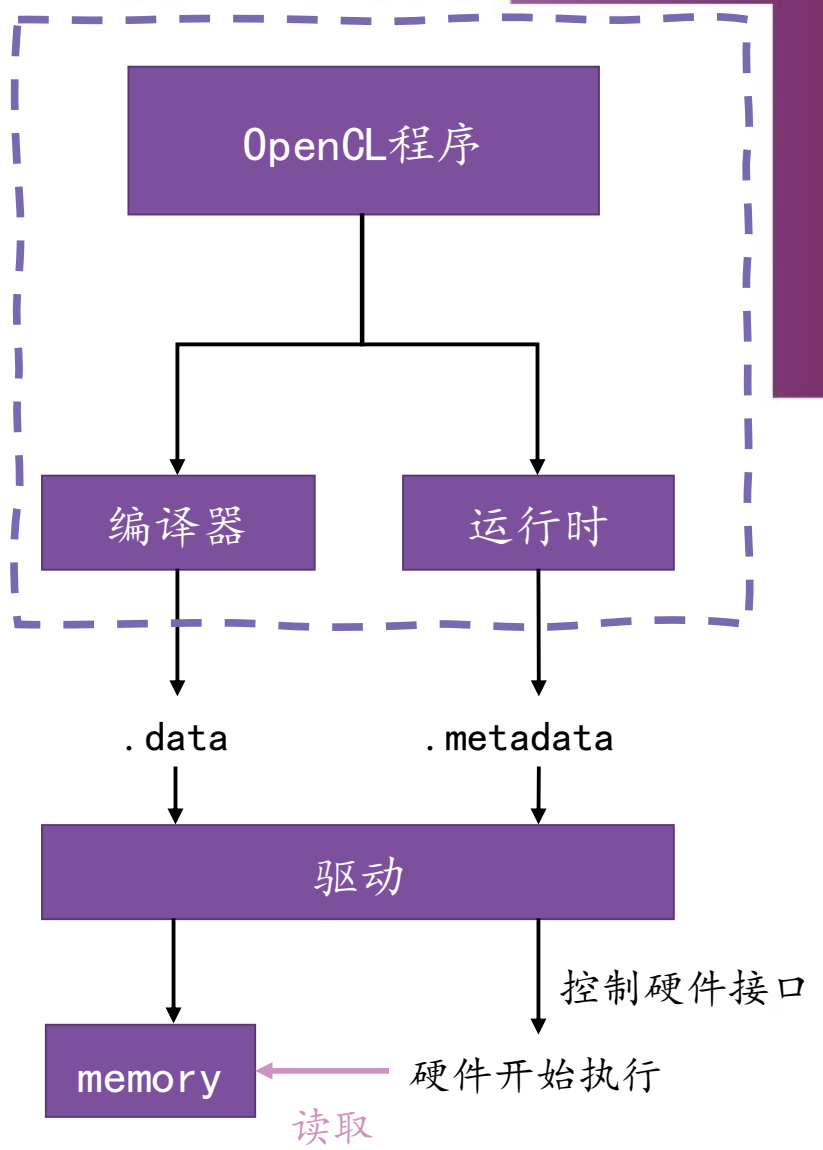
类型	API
The OpenCL Runtime	clCreateCommandQueueWithProperties clRetainCommandQueue clReleaseCommandQueue clGetCommandQueueInfo
Buffer Objects	clCreateBuffer clCreateSubBuffer clEnqueueReadBuffer clEnqueueReadBufferRect clEnqueueWriteBuffer clEnqueueWriteBufferRect clEnqueueCopyBuffer clEnqueueCopyBufferRect clEnqueueMapBuffer
Kernel Objects	clCreateKernel clRetainKernel clReleaseKernel clSetKernelArg clGetKernelInfo clGetKernelArgInfo clEnqueueNDRangeKernel
Program Objects	clCreateProgramWithSource clReleaseProgram clGetProgramBuildInfo
Event Objects	clCreateUserEvent clSetUserEventStatus clWaitForEvents clGetEventInfo clReleaseEvent

软件工具链功能 – 内存管理

- 每个warp有独立的地址空间：
  - warp地址空间：LocalMemory（堆栈）
  - WorkGroup地址空间： SharedMemory
  - kernel地址空间：GlobalMemory（数据）、ConstantMemory（数据、指令）
- 驱动完成地址分配
- 针对不同空间数据采用不同指令/地址范围进行访问

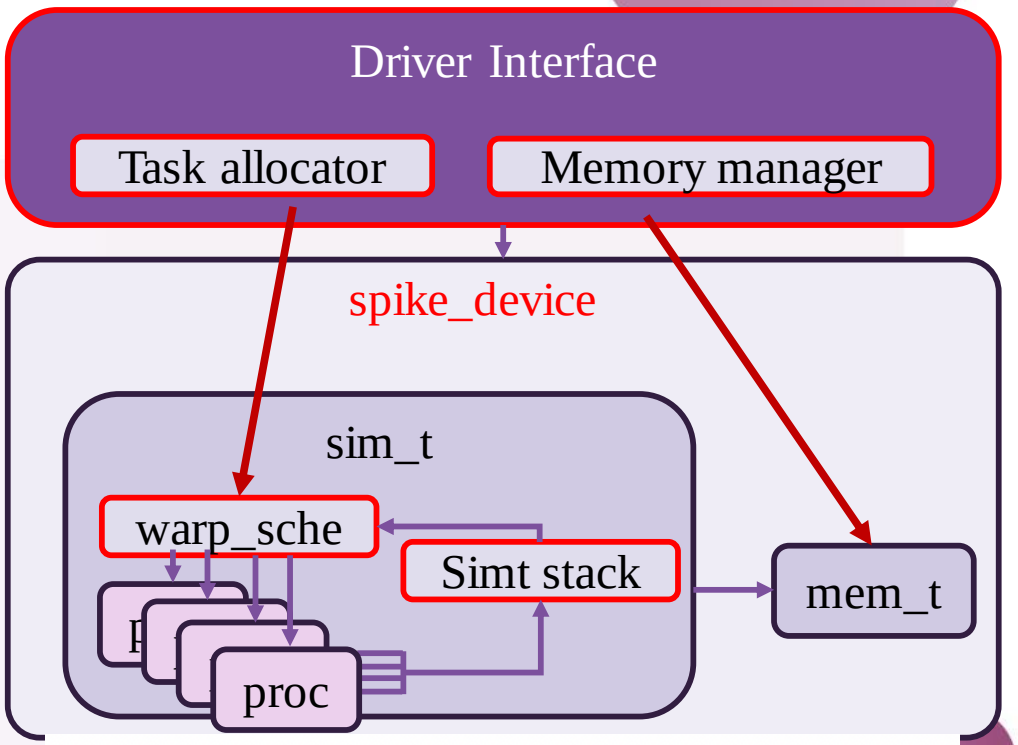
软件工具链功能 – 程序生成和加载

- 平台驱动程序调用编译器，生成可以被GPGPU执行的指令和数据
- .meta文件为设备接口有关的数据结构，直接与硬件进行对接，由设备驱动程序进行管理
- 驱动程序将.data文件加载到内存，将.meta文件加载到硬件接口



基于spike的指令精度仿真器

- warp级抽象，每个core代表一个warp
- 添加自定义指令
- 与硬件接口关键信号保持一致，可通过命令行选项配置lds_size，lds_baseaddr
- 实现GPU组件：
  - SIMT-stack
  - register extend
  - warp_scheduler(仅用于barrier)



type	instruction name	usage
kernel response	endprog	endprg x0,x0,x0
synchronization	barrier, barriersub	barrier x0,x0,imm barriersub x0,x0,imm
branch control	vbeq, vbne, vblt vbge, vbltu, vbgeu	vbeq vs2, vs1, offset vbne vs2, vs1, offset
branch control	join, setrpc	join v0,v0,0 setrpc rd,rs1,offset
register index extension	regext,regexti	regext x0,x0,imm regexti x0,x0,imm
memory access	vlw.v, vlh(u).v, vlb(u).v vsw.v, vsh.v, vsb.v	vlw.v vd,offset(vs1) vsw.v vs2,offset(vs1)
memory access	vlw12.v, vlh(u)12.v, vlb(u)12.v vsw12.v, vsh12.v, vsb12.v	vlw12.v vd, offset(vs1) vsw12.v vs2,offset(vs1)
memory access	vadd12.vi	vadd12.vi vd, vs1, imm
tensor related	vfexp.v, vftta.v	vfexp vd,v2,v0.mask vftta.vv vd,v2,v1,v0.mask

软件工具链运行效果展示

OpenCL语言

```
__kernel void Fan1(__global float *m_dev,
                   __global float *a_dev,
                   __global float *b_dev,
                   const int size,
                   const int t) {
    int globalId = get_global_id(0);

    if (globalId < size-1-t) {
        *(m_dev + size * (globalId + t + 1)+t) =  \
        *(a_dev + size * (globalId + t + 1) + t) / *(a_dev +
size * t + t);
    }
}
```

支持自定指令的RISC-V汇编

```
800000ac <Fan1>:
addi        sp, sp, 16
sw          ra, -16(sp)
lw          t0, 12(a0)
sw          t0, -4(sp)
sw          a0, -12(sp)
lw          t0, 16(a0)
sw          t0, -8(sp)
vmv.v.x     v0, zero
jal         <get_global_id>
lw          s0, -8(sp)
lw          t2, -4(sp)
not         t0, s0
add         t0, t0, t2
vmv.v.x     v1, t0
auipc       t1, 0
setrpc      zero, t1, 108
vbge        v0, v1, <.LBB0_2>
lw          t1, -12(sp)
lw          t0, 4(t1)
lw          t1, 0(t1)
vmv.v.x     v1, s0
vmv.v.x     v2, t2
vmv.v.x     v3, t0
vadd.vv     v0, v0, v1
vadd.vi     v0, v0, 1
vmul.vv     v0, v0, v2
vsll.vi     v0, v0,
vadd.vv     v4, v3, v0
……
```

```
yangzx@sw-001-VirtualHost:~/ventus/gpu-rodinia/opencl/gaussian$
```

```
Fan1_0.data       Fan1_2.data       Fan2_1.data       gaussianElim.cpp          Makefile
Fan1_0.log        Fan1_2.log        Fan2_1.log        gaussianElim.h            object.cl
Fan1_0.metadata   Fan1_2.metadata   Fan2_1.metadata   gaussianElim_kernels.cl   object.dump
Fan1_1.data       Fan2_0.data       Fan2_2.data       gaussian.out              object.riscv
Fan1_1.log        Fan2_0.log        Fan2_2.log        gettimeofday.cpp          object.vmem
Fan1_1.metadata   Fan2_0.metadata   Fan2_2.metadata   gettimeofday.h            OriginalParallel.c
```

版本发布

release_v2.0.2-alpha.tar.gz

Date	Download	Release Notes
2024-01-24 17:04	download	release notes

总结

- 目前支持将kernel函数编译为RVV程序并按照OpenCL框架在多个仿真设备上运行的完整流程

- 已验证的benchmark

Benchmark	vecadd	gaussian	nn	bfs	hybridsort	kmeans	nw
指令级仿真	✓	✓	✓	✓	✓	✓	✓
RTL仿真	✓	✓	✓	✓			

- OpenCL CTS: Basic 103/115、Api 96/105、Computeinfo 5/5、 Compiler 26/66

THANK YOU

OpenGPGPU
乘影

上海清华国际创新中心
集成电路研究平台
International Innovation Center of Tsinghua University Shanghai
Integrated Circuit Research Platform