Target	Task	Time	Type	GPUs	Vk Frequency	Process
RX7 - Navassa_engine	Frame #1	1.93ms	Vulkan	Navi 31	1800 MHz	[3419] [Local]

Summary | Details | Source | Control | Comments | Raw | Session

Compare | Notes | Export

GPU Speed Of Light Throughput

Section provides a high-level overview of the utilization for compute and memory resources of the GPU. For each unit, the throughput reports the achieved percentage of utilization with respect to the theoretical maximum. Breakdowns show the throughput for each individual sub-metric of Compute and Memory. Depending on the workload characteristics, High-level overview of the utilization for compute and memory resources of the GPU presented in the Speed Of Light section.

Compute (SM) Throughput [%]	0.06	Duration [ms]	1.25
Memory Throughput [%]	0.06	Elapsed Cycles [cycle]	1,869,312
L1/TEX Cache Throughput [%]	0.06	SM Active Cycles [cycle]	113,984.00
L2 Cache Throughput [%]	0.06	Compute (SM) [%]	0.06
VRAM Throughput [%]	0.07	Memory Frequency [GHz]	1.20

Latency bound this kernel exhibits low compute throughput and memory bandwidth utilization relative to the peak performance of this device. Achieved compute throughput and/or memory bandwidth below 60.0% of peak typically indicate latency issues. Look at Scheduler Statistics and Warp State in order to gather potential reasons.

Roofline Analysis The sample of peak float [GFLOP] is double [GFLOP] performance on this device is 0.0. This kernel achieved 0% of this peak float performance and 0% of its fp64 peak performance. See the Roofline chart for more details on roofline analysis.

GPU Throughput

Speed Of Light [%]

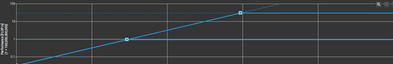
Compute Throughput Breakdown

Int Instr Executed Per cycle [%]	3.09
SM Issue Active [%]	2.15
Int Instr Executed [%]	1.98
SM Alu Pipe Active Cycles Active [%]	1.98
SM Fma Pipe Active Cycles Active [%]	1.98
SM Issue [%]	1.98
SM Pipe Alu Cycles Active [%]	1.98
SM Pipe Fma Cycles Active [%]	1.98
Int Instr Executed Pipe Fma Pred On Arg [%]	0.06
Int Instr Executed Pipe Alu [%]	0.06
Int Instr Executed Pipe Fma [%]	0.06
Int Instr Executed Pipe Lsb Mem Op [%]	0.06
Int Instr Executed Pipe Tex [%]	0.06
SM Warps Active Cycles Active [%]	0.06
SM Warps Active Cycles Elapsed [%]	0.06
Pipe Shared Cycles Active [%]	0.06
Pipe Tex Cycles Active [%]	0.06
FE: Request Cycles Active [%]	0.06
Sm Sysldesp Internal Stall Warp Issue Active [%]	0.06
Int Instr Executed Per Sub Partition [%]	0.06
Int Instr Executed Per Cycle [%]	0.06
Int Instr Executed Per Sub Partition [%]	0.06

Memory Throughput Breakdown

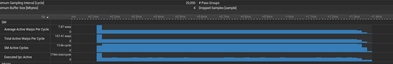
L1 Lsmem Requests [%]	3.09
L1 Data Pipe Lsu Wavefronts [%]	2.15
DRAM: Cycles Active [%]	1.98
DRAM: Read Cycles Active [%]	1.98
DRAM: Write Cycles Active [%]	1.98
L1: All Global Read Sectors [%]	1.98
L1: Tag Requests [%]	0.06
L1: Tag Cycles Active [%]	0.06
GPU Compute Memory Access Throughput Internal Active [%]	0.06
L1: Tag Requests [%]	0.06
L1: Tag Cycles Active [%]	0.06
L1: All Global Read Sectors [%]	0.06
L1: Wavefronts Cycles Active [%]	0.07
L1: Data Cycles Active [%]	0.07
L1: Lsmem Wavefronts Cycles Active [%]	0.07
L1: Lsmem Input Cycles Active [%]	0.07
L1: Data Bank Reads [%]	0.06
L1: Tag Cycles Stalled [%]	0.06
L1: Data Cycles Stalled [%]	0.06
L1: Tag Stalled [%]	0.06
L1: Tag Writeback Active [%]	0.06
L1: Tag Cycles Active [%]	0.06
L1: Tag Cycles Active [%]	0.06
L1: Tag Cycles Active [%]	0.06
L1: Tag Cycles Active [%]	0.06
L1: Tag Writeback Active [%]	0.06
L1: Data Cycles Active [%]	0.06
GPU: Compute Memory Request Throughput Internal Active [%]	0.06

Floating Point Operations Roofline



PM Sampling

Provides a view of the warps's cycle-by-cycle activity. Sampling is performed at a fixed interval, independent of the kernel's execution. Depending on the workload characteristics, Maximum Sampling Interval [cycle] 20,000 P-Para Groups Sampling Interval [cycle] 20,000 Dropped Samples [sample] 0

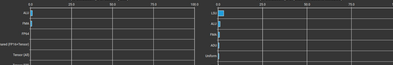


Compute Workload Analysis

Detailed analysis of the compute resources of the streaming multiprocessor (SM), including the achieved instructions per clock (IPC) and the utilization of each available pipeline. Pipelines with very high utilization often limit the overall performance.

Executed Ipc Elapsed [inst/cycle]	0.08	SM Busy [%]	2.17
Executed Ipc Active [inst/cycle]	3.09	Issue Slots Busy [%]	1.98
Issued Ipc Active [inst/cycle]	3.09		

Low Utilization All compute pipelines are under-utilized. Either this kernel is very small or it doesn't issue enough warps per scheduler. Check the Launch Statistics and Scheduler Statistics sections for further details.



Memory Workload Analysis

Detailed analysis of the memory resources of the GPU. Memory can become a limiting factor for the overall kernel performance when fully utilizing the control flow and compute pipelines (see above), or when the workload exhibits many memory accesses. Depending on the workload characteristics, Memory Throughput [Gbyte/s] 0.07 Mem Busy [%] 0.06 Max Bandwidth [%] 0.06 L1/TEX Hit Rate [%] 0.00 L2 Hit Rate [%] 0.00 Mem Pipes Busy [%] 0.06

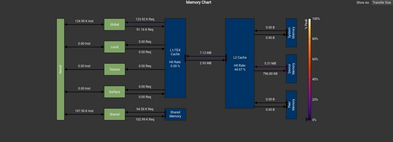
Memory L2 Compression The optional metric lts__sectors_compressed_ratio_sector_read.sum could not be found. Collecting it as an additional metric could enable the rule to provide more guidance.

Memory Pattern The memory access pattern for global loads might not be optimal. On average, only 31.4 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the Memory Chart section for additional details.

L1TEX Global Store Access Pattern The memory access pattern for global stores to L1TEX might not be optimal. On average, only 4.0 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the Memory Chart section for additional details.

Shared Load Bank Conflicts The memory access pattern for shared loads might not be optimal and causes on average a 2.0 - way bank conflict across all 5453824 shared load requests.This results in 14219 bank conflicts, which represent 12.70% of the overall 111955 cycles spent for shared loads. Check the Source Counters section for uncoalesced shared loads.

Memory Chart



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	84,304	84,304	111,955	0.38	14,219
Shared Load Matrix	0	0	0		
Shared Store	102,992	102,992	155,209	0.52	0
Shared Atomic	0	0	0		
Shared Reduce	0	0	0		
Total	187,296	187,296	404,931	1.38	14,219

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit Rate	Bytes	Sector Misses to L2	% Peak to L2	Returns to SM
Local Load	0	0	0		0						
Local Store	63,798	63,798	63,798	0.21	1,993,688	31.25	0.00	63,817,216	1,993,688	6.49	124,674,...
Surface Load	0	0	0		0						
Texture Load	0	0	0		0						
Global Store	1,024	1,024	1,024	0.00	1,024	1.00	0.00	32,768	1,024	0.00	
Global Atomic	0	0	0		0						
Surface Store	0	0	0		0						
Global Reduction	0	0	0		0						
Surface Atomic	0	0	0		0						
Texture Reduction	0	0	0		0						
Global Atomic CAS	0	0	0		0						
Global Atomic Add	90,131	90,131	90,336	0.31	90,997	1.00	0	2,912,704	90,997	0.02	see above
Global Reduction Add	0	0	0		0						
Surface Atomic CAS	0	0	0		0						
Surface Atomic Add	0	0	0		0						
Total	63,798	63,798	63,798	0.21	1,993,688	31.25	0.00	63,817,216	1,993,688	6.49	124,674,...
Pipe Lsmem Op Ld	84,304	84,304	111,955	0.38	14,219	0.13	0.00	3,582,464	14,219	0.05	
Pipe Lsmem Op St	90,131	90,131	90,336	0.31	90,997	1.00	0	2,912,704	90,997	0.02	

L2 Cache

	Requests	Sectors	Sectors/Req	% Peak	Hit Rate	Throughput	Sector Misses to Device	Sector Misses to System	Sector Writes to Peer
L1/TEX Load	76,751	1,993,688	25.98	0.06	98.07	6,227,640	5,944,414,977	127,073	0
L1/TEX Atomic ALU	1,007	32,896	32.67	0.00	98.07	102,938	26,295,168	0	0
L1/TEX Atomic CAS	0	0				0	1,600,010,136	0	0
L1/TEX Atomic Add	90,335	90,374	1.00	0.00	100.00	2,950,688	2,106,576,417,952	0	0
L1/TEX Reduction	0	0				0	0	0	0
L1/TEX Store	90,127	2,912,704	32.31	0.09	97.18	2,912,704	3,154,444,800,000	127,073	0
EDC Total		2,986,966	2.00	0.09	98.07	9,293,282	0	254,294,146	0
VPN Total	166,769	2,035,584	12.16	0.06	44.18	6,330,578	7,950,010,977,120	127,073	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	165,655	1.10	5,298,240	4,234,240,000.00
Store	32,900	0.22	756,800	605,044,106.50
Total	198,555	1.32	6,055,040	4,839,284,106.50

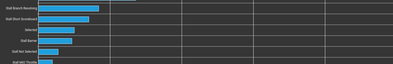
Scheduler Statistics

Summary of the activity of the schedulers issuing instructions. Each schedulers maintains a pool of warps that it can issue instructions for. The upper bound of warps in the pool (Theoretical Warps) is limited by the launch configuration. On every cycle each scheduler checks the state of all warps in this pool (Eligible Warps) and issues an instruction, if at least one warp is ready and can issue an instruction. Front End Stalls indicate that the scheduler is not able to issue an instruction, Issued Warps, on cycles with no eligible warps. Ideally, the scheduler should be able to issue an instruction every cycle. The scheduler's ability to issue an instruction depends on the number of Eligible Warps. Low Eligible Warps indicates that the kernel's performance is latency bound. Every cycle with no eligible warp, a single warp from which to issue one is stalled for a number of reasons. Look at the Warp State section for more information on the reasons for warp stalls.

Active Warps Per Scheduler	19.57	Avg. Active Threads Per Warp	14.13
Warp Cycles Per Executed Instruction [cycle]	22.98	Avg. Not Predicated Off Threads Per Warp	10.00
Eligible Warps Per Scheduler [warp]	0.08		

Issue Slot Utilization Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 1.5 cycles. This might lower bandwidth requirements and improve overall performance. Out of the maximum of 4 issue slots per scheduler, this kernel allocates an average of 1.96 active warps per scheduler, but only an average of 0.12 warps were eligible per cycle. Eligible warps are the subset of active warps that are ready to issue their next instruction. Every cycle with no eligible warp results in a dedicated stall issue slot. Most instructions are issued only a single time, but some require multiple issues. To increase the number of eligible warps, reduce the time the active warps are stalled for a kernel to be issued.

Warps Per Scheduler



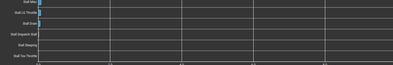
Warp State Statistics

Analysis of the stalls in which all warps spent cycles during the kernel execution. The warp states describe a warp's readiness or inability to issue its next instruction. The warp cycles per instruction define the latency between two consecutive instructions. The higher the value, the more warp parallelism is required to hide this latency. For each warp state, the chart shows the average number of cycles spent in that state per issued instruction. Stalls cycles are a subset of these cycles. Warps can be stalled for many different reasons, either because the warp is waiting on a high-latency instruction, or because a hardware resource is saturated.

Warp Cycles Per Issued Instruction [cycle]	19.57	Avg. Active Threads Per Warp	14.13
Warp Cycles Per Executed Instruction [cycle]	22.98	Avg. Not Predicated Off Threads Per Warp	10.00

Long Scoreboard Stalls On average, each warp of this kernel spends 8.3 cycles being stalled waiting for a scoreboard dependency on a L1TEX (local, global, surface, texture) operation. Find the instruction producing the data being waited upon to identify the culprit. To enhance latency hiding, this may be caused by non-optimized memory accesses, or by too many in-flight L1TEX operations. For Global memory, coalesce the data access. For Texture or Surface memory, ensure that the data layout is optimal for the access pattern. Consider increasing the kernel's occupancy or reorder non-dependent instructions to increase the distance between dependent instructions.

Warp Stall Check the Warp State (All Cycles) table for the top stall reasons on warps based on sampling data. The Source Counters section provides more detailed in-kernel stall reasons.

Thread Divergence Instructions are executed in warps, which are groups of 32 threads. Optimal instruction throughput is achieved if all 32 threads of a warp execute the same instruction. The chosen launch configuration, warp thread assignment, and divergent flow do not allow the kernel to achieve optimal thread divergence efficiency. This kernel's average thread divergence efficiency is 100.0% (22.0 threads active per warp). This is further reduced to 12.0 threads per warp due to predication. The compiler may use predication to avoid an actual branch. Instead, all instructions are scheduled, but a per-thread condition code or predicate controls which threads execute the instruction. Try to avoid different execution paths within a warp where possible.

Warp State (All Cycles)