AMD EPYC™ 7xx2-series Processors





AOCC compiler (with Flang - Fortran Front-End)

Latest release: 2.1, Nov 2019

https://developer.amd.com/amd-aocc/

Architecture			
Generate instructions that run on 2nd Gen EPYC/ RYZEN	-march=znver2		
Generate instructions for the local machine	-march=native		
Optimization Levels			
Disable all optimizations	-00		
Minimal level speed and code optimization	-01		
Moderate level optimization (default)	-02/-0		
Aggressive optimizations	-03		
Maximize performance	-Ofast		
Enable Link Time Optimization	-fito		
Enable unrolling	-funroll-loops		
Enable aggressive loop optimizations	-enable-loop-versioning-licm -enable-partial-unswitch -unroll-aggressive		
Enable aggressive inline optimizations	-function-specialize -finline-aggressive		
Enable aggressive vectorization	-enable-strided-vectorization -enable-epilog-vectorization		
Enable memory layout optimizations	-fremap-arrays (use with -flto)		
Profile Guided optimizations	-fprofile-instr-generate (1st invocation) -fprofile-instr-use (2nd invocation)		
OpenMP®	-fopenmp		
For enabling streaming stores, memory bandwidth workloads	-fnt-store		
Enable removal of un-used array computation	-reduce-array-computation=3		

Other options			
Enable faster, less precise math operations	-ffast-math -freciprocal-math		
OpenMP® threads and affinity (N number of cores)	export OMP_NUM_THREADS=N export GOMP_CPU_AFFINITY="0- {N-1}"		
Enabling Vector library	-vector-library=LIBMVEC		
Link to Vector library	-L/libm-install-dir/lib -lmvec		
Link to AMD library	-L/libm-install-dir/lib -lamdlibm		

For Fortran workloads		
Compile free form FORTRAN	-ffree-form	
Enable precise math operations in FORTRAN	-kieee	

AMD Optimized Libraries

Latest release: 2.1, Jan 2020

https://developer.amd.com/amd-aocl/

AMD µProf (Performance & Power Profiler)

Latest release: 3.2, Nov 2019

https://developer.amd.com/amd-uprof/

AMD EPYC™ 7xx2-series Processors

Compiler Options Quick Reference Guide



GNU compiler collection (gcc, g++, gfortran)

Latest release: 9.3, Mar 2020 Recommended version: 9.3

http://gcc.gnu.org

Microsoft Visual Studio 2019

Latest stable release: 16.5, Mar 2020

https://www.visualstudio.com/

User Guide

Architecture		Architecture	
Generate instructions that run on 2nd Gen EPYC/ RYZEN	-march=znver2	Generate instructions that run on 2nd Gen EPYC/ RYZEN	/arch:[AVX AVX2]
Generate instructions for the local machine	-march=native	Optimize for 64-bit AMD processors	/favor:AMD64 /d2vzeroupper
Optimization Levels		Optimization Levels	
Disable all optimizations (default)	-00	Disable optimizations	/Od
Minimal level speed and code optimizations	-01/-0	Maximum optimizations (favor space)	/01
Moderate level optimizations	-02	Maximum optimizations (favor speed)	/02
Aggressive optimizations	-03	[link.exe] Eliminate unrefer-	/OPT:REF
Maximize performance	-Ofast	enced function and/ or data	
Additional Optimizations		[link.exe] Perform identical	/OPT:ICF
Link time optimization	-flto	COMDAT folding	
Enable unrolling	-funroll-all-loops	Output an informational mes- /Qv sage for loops that are auto-vectorized	/Qvec-report:[1 2]
Generate memory preload instructions	-fprefetch-loop-arraysparam prefetch-latency=300		
Profile-guided optimization	-fprofile-generate (1st invocation) -fprofile-use (2nd invocation)	Enable automatic parallelization of loops, used in conjunction with #pragma loop() directive	/Qpar
OpenMP®	-fopenmp	Output an informational mes-	/Qpar-report:[1 2]
Other options		sage for loops that are auto- parallelized	
Enable generation of code that follows IEEE arithmetic	-mieee-fp	Additional Optimizations	
Enable faster, less precise math operations	-ffast-math	Maintain the precision for floating-point operations	/fp:precise
Compile free form FORTRAN	-ffree-form	through proper rounding	
OpenMP® threads and affinity (N number of cores)	export OMP_NUM_THREADS=N export GOMP_CPU_AFFINITY="0-{N-1}"	Optimize floating-point code for speed at the expense of floating -point accuracy and correctness	/fp:fast
Link to AMD library	-L/libm-install-dir/lib -lamdlibm		/GL
GlibC Latest release: 2.31, Jan 2020 Recommendation: 3.26 or later		Whole Program Optimization /GL (link-time code generation)	/GL
		Profile-guided optimization	LTCG:PGI and /LTCG:PGO

Recommendation: 2.26 or later https://www.gnu.org/software/libc/

Binutils

Recommendation: 2.26.1 or later https://www.gnu.org/software/binutils/

AMD EPYC™ 7xx2-series Processors





Intel compilers (icc, icpc, ifort)

Latest release: 19.1

http://software.intel.com

Architecture Generate instructions that run on 2nd Gen EPYC/RYZEN -axCORE-AVX2 Optimization Levels Disable all optimizations Speed optimization without code growth Enable optimization for speed including vectorization Aggressive optimization Aggressive optimizations Additional Optimizations Aggressive unrolling Disable improved precision floating divides Enable vectorization OpenMP® -qopenmp Prefetch optimization -voc Use optimized header definitions Other options Floating point accuracy tuning -march=core-avx2 (preferred) OR -axCORE-AVX2 -march=core-avx2 (preferred) OR -axCORE-AVX2 -march=core-avx2 (preferred) OR -axCORE-AVX2 -march=core-avx2 (preferred) OR -axCORE-AVX2 -volume -volume -volume -volume -value -vec -no-prec-div -prof-gen and -prof-use -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model -free			
2nd Gen EPYC/RYZEN -axCORE-AVX2 Optimization Levels Disable all optimizations -O0 Speed optimization without code growth Enable optimization for speed including vectorization Aggressive optimization -O3 Maximize performance -Ofast Additional Optimizations Aggressive unrolling -unroll-aggressive Disable improved precision floating divides Enable vectorization -vec Inter procedural Optimization OpenMP® -qopenmp Prefetch optimization -prof-gen and -prof-use Use optimized header definitions Floating point accuracy tuning -fp-model	Architecture		
Optimization Levels Disable all optimizations -O0 Speed optimization without code growth Enable optimization for speed including vectorization Aggressive optimization -O3 Maximize performance -Ofast Additional Optimizations Aggressive unrolling -unroll-aggressive Disable improved precision floating divides Enable vectorization -vec Inter procedural Optimization -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions Floating point accuracy tuning -fp-model		-march=core-avx2 (preferred) OR	
Disable all optimizations Speed optimization without code growth Enable optimization for speed including vectorization Aggressive optimization Aggressive optimization Additional Optimizations Aggressive unrolling Disable improved precision floating divides Enable vectorization Inter procedural Optimization OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions Floating point accuracy tuning -fp-model	2nd Gen EPYC/RYZEN	-axCORE-AVX2	
Speed optimization without code growth Enable optimization for speed including vectorization Aggressive optimization Aggressive optimization Additional Optimizations Aggressive unrolling Disable improved precision floating divides Enable vectorization OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions Floating point accuracy tuning -fp-model	Optimization Levels		
growth Enable optimization for speed including vectorization Aggressive optimization Additional Optimizations Aggressive unrolling Disable improved precision floating divides Enable vectorization OpenMP® Prefetch optimization Profile generated optimization Use optimized header definitions Ploating point accuracy tuning -O2 -O2 -O3 -O3 -Ofast -unroll-aggressive -no-prec-div -no-prec-div -no-prec-div -rop-prec-div -rop-prefetch -prof-gen and -prof-use -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Disable all optimizations	-00	
including vectorization Aggressive optimization Additional Optimizations Aggressive unrolling Disable improved precision floating divides Enable vectorization OpenMP® Prefetch optimization Profile generated optimization Use optimized header definitions Floating point accuracy tuning -O3 -O3 -O5 -O6st -U10 -U10 -U10 -U20 -U30 -		-01	
Maximize performance -Ofast Additional Optimizations Aggressive unrolling -unroll-aggressive Disable improved precision -no-prec-div floating divides Enable vectorization -vec Inter procedural Optimization -ipo OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model		-02	
Additional Optimizations Aggressive unrolling -unroll-aggressive Disable improved precision -no-prec-div floating divides Enable vectorization -vec Inter procedural Optimization -ipo OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Aggressive optimization	-03	
Aggressive unrolling -unroll-aggressive Disable improved precision -no-prec-div floating divides Enable vectorization -vec Inter procedural Optimization -ipo OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Maximize performance	-Ofast	
Disable improved precision floating divides Enable vectorization -vec Inter procedural Optimization OpenMP® -qopenmp Prefetch optimization -qopt-prefetch -profile generated optimization Use optimized header definitions Other options Floating point accuracy tuning -no-prec-div -vec -ipo -qop -qop -qopenmp -qopt-prefetch -prof-gen and -prof-use -use-intel-optimized-headers Other options -fp-model	Additional Optimizations		
floating divides Enable vectorization -vec Inter procedural Optimization -ipo OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Aggressive unrolling	-unroll-aggressive	
Inter procedural Optimization -ipo OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	· ·	-no-prec-div	
OpenMP® -qopenmp Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Enable vectorization	-vec	
Prefetch optimization -qopt-prefetch Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Inter procedural Optimization	-ipo	
Profile generated optimization -prof-gen and -prof-use Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	OpenMP®	-qopenmp	
Use optimized header definitions -use-intel-optimized-headers Other options Floating point accuracy tuning -fp-model	Prefetch optimization	-qopt-prefetch	
Other options Floating point accuracy tuning -fp-model	Profile generated optimization	-prof-gen and -prof-use	
Floating point accuracy tuning -fp-model	Use optimized header definitions	-use-intel-optimized-headers	
	Other options		
Compile free form FORTRAN -free	Floating point accuracy tuning	-fp-model	
	Compile free form FORTRAN	-free	