AMD A10, A8, A6 7000 Series APUs

Compiler Options Quick Reference Guide

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

Latest release: 4.9.0, April 2014

http://gcc.gnu.org

Architecture	
Generate instructions that run on X2200 Series APU	-march=bdver3
Generate instructions for the local machine	-march=native
Optimization Levels	
Disable all optimizations (default)	-O0
Speed and code optimizations	-O1
More optimizations	-O2
Aggressive optimizations	-O3
Maximize performance	-Ofast
Additional Optimizations	
Unwind all loops	-funroll-all-loops
Generate memory preload instructions	-fprefetch-loop-arrays --param prefetch-latency=300 (300-700)
Profile-guided optimization	-fprofile-generate, -fprofile-use
Disable loop optimization to preserve code correctness	-fno-aggressive-loop-optimizations
Inline string operations	-minline-all-stringops
Enable Link-time optimization	-flto
Software pipelining during selective scheduling	-fsel-sched-pipelining -fselective-scheduling
Partial redundancy elimination	-ftree-pre
Disable vectorization	-fno-tree-vectorize
Local arrays are put on stack	-fstack-arrays
Instructions reordering	-fschedule-insns2
AMD Math library	-L/opt/ACML/acml-5-3-1-gfortran-64bit/gfortran64/lib –lacml
Other options	
Generate IEEE arithmetic code	-mieee-fp
Enable faster, less precise math operations	-ffast-math
Enable OpenMP #pragma	-fopenmp
OpenMP threads and affinity	export OMP_NUM_THREADS=4 export GOMP_CPU_AFFINITY="0-3"

Intel compiler

Latest release: 14.0.2, February 2014

https://software.intel.com

Architecture	
Generate instructions that run on X2200 Series APU	-msse4.2
Optimization Levels	
Disable optimizations	-O0
Code size and locality optimization	-O1
Code speed optimization	-O2
More aggressive loop and memory optimizations, prefetching	-O3
Maximize code speed across the entire program	-fast
Additional Optimizations	
General compiler-guided loop unrolling	-unroll
Complete loop unrolling for loops with small counts	-unroll-aggressive
Selection of an optimal method for registers allocation	-opt-ra-region-strategy
Improve precision of floating point divides	-prec-div
Loop vectorization enabled	-vec
Compiler analisys based loop vectorization	-vec-threshold100
Enable code vectorization	-simd
Replace calls to transcendental function with faster but less precise implementations	-fast-transcendentals
More aggressive optimization on floating point data	-fp-model fast=1
Use optimized performance headers definitions	-use-intel-optimization-headers
Enable prefetching	-opt-prefetch
AMD Math library	-L/opt/ACML/acml-5-3-1-ifort-64bit/ifort64/lib –lacml
Other options	
Enable OpenMP directives	-openmp

For more information, visit http://developer.amd.com

Advanced Micro Devices
One AMD Place
P.O. Box 3453
Sunnyvale, CA 94088-3453

AMD A10, A8, A6 7000 Series APUs

Compiler Options Quick Reference Guide

PGI compilers (pgcc, pgcpp, pgfortran)

Latest release: 14.4, April 2014

http://www.pgroup.com

Architecture	
Specify target processor	-tp bulldozer
Optimization Levels	
Local optimizations	-O1
Global optimizations	-O2
Local, global optimizations, code hoisting, scalar replacement	-O3
Guarded invariant floating point expressions code hoisting	-O4
Choose generally optimal optimization flags	-fast
Additional Optimizations	
Enable auto-concurrentization of loops on all available CPU cores	-Mconcur=allcores
Alternative code generation for parallelized loops	-Mconcur=altcode:32
Enable loop parallelization with reductions	-Mconcur=assoc
Automatically bind threads to system's cores	-Mconcur=bind
Parallelize code with block|cyclic distribution.	-Mconcur=dist:block|cyclic
Use thread affinity for NUMA architectures	-Mconcur=numa
Interprocedural analysis	-Mipa=const|fast|inline|libc
Remove functions which are never used	-Mipa=vestigial
Nontemporal moves, optimal flags selection for SSE processors	-Mmovnt -fastsse
Profile guided optimizations	-Mpfi, -Mpfo
AMD Math library	-L/opt/ACML/pgi64/lib -lacml
Other options	
Enable OpenMP directives	-mp
Enable OpenACC directives	-ta=radeon –acc
Output OpenACC debug information	export PGI_ACC_DEBUG=1 export PGI_ACC_TIME=1

Open64 compilers (opencc, openCC, openf95)

Latest release: 4.5.2.1 , March 2013

http://developer.amd.com/open64

Architecture	
Specify target processor	-march=bdver2
Create instruction for the local machine	-march=auto
Optimization Levels	
Local optimizations	-O1
Global optimizations	-O2
Local and global optimizations, loops optimization, prefetching	-O3
A selection of optimizations	-Ofast
Additional Optimizations	
Optimizations on multi-core systems for scalability	-mso
Interprocedural analysis	-ipa
Function inline processing	-finline
Link against special versions of standard library routines	-ffast-stdlib
Calculate square root using reciprocal square root operation	-OPT:fast_sqrt=ON
Enable loop fission	-LNO:fission=ON
Enable loop fusion	-LNO:fusion=ON
Profile based optimization	-fb_create, -fb_opt
Prefetch a number of cache lines	-LNO:prefetch_ahead=3
Relaxed IEEE rules/specifications	-ffast-math
AMD Math library	-L/opt/ACML/open64_64/lib -lacml
Other options	
Enable OpenMP directives	-mp
OpenMP threads and affinity	export OMP_NUM_THREADS=4 O64_OMP_AFFINITY_MAP=0-3
Permits storing floating point variables in registers	-ffloat-store
Use "huge" 2 MB pages for heap, text and data segments	-HP
Set FP operations accuracy level	-fp-accuracy

For more information, visit http://developer.amd.com

Advanced Micro Devices
One AMD Place
P.O. Box 3453
Sunnyvale, CA 94088-3453

AMD A10, A8, A6 7000 Series APUs

Compiler Options Quick Reference Guide

Microsoft Visual Studio 2013

Latest Release: November 2013

http://msdn.microsoft.com/en-us/library/fwkeyyhe.aspx

http://msdn.microsoft.com/en-us/library/vstudio/60k1461a.aspx

http://www.microsoft.com/visualstudio

Architecture	
Generate instructions that run on X2200 Series APU	/arch:AVX
Favor AMD architecture (x64 only)	/favor:AMD64
Optimization Levels	
Maximize performance	/O2
Eliminate unreferenced function and/or data	/OPT:REF
Perform identical COMDAT folding	/OPT:ICF
Output an informational message for loops that are auto-vectorized	/Qvec-report:1
Enable automatic parallelization of loops marked with the #pragma loop() directive	/Qpar
Additional Optimizations	
Maintain the precision for floating-point operations through proper rounding	/fp:precise
Optimize floating-point code for speed at the expense of floating-point accuracy and correctness	/fp:fast
Whole Program Optimization	/GL
Profile guided optimization	/LTCG:PGI and /LTCG:PGO
AMD Math library	-L\opt\ACML\ifort64 \lib\libacml_dll.lib

For more information, visit http://developer.amd.com

Advanced Micro Devices
One AMD Place
P.O. Box 3453
Sunnyvale, CA 94088-3453