

AMD Embedded R-Series SOC

Compiler Options Quick Reference Guide

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

Latest release: 5.2.0, July 2015

<http://gcc.gnu.org>

Intel compiler (icc, icpc, ifort)

Latest release: 15.0.3, April 2015

<https://software.intel.com>

Architecture	
Generate instructions that run on Embedded R-Series SOC	-march=bdver4
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	
Unroll 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 aggressive loop optimizations to preserve code correctness	-fno-aggressive-loop-optimizations
Instruction reordering	-fschedule-insns2 -fsched-pressure
Enable Link-time optimization	-flto
Don't split AVX unaligned store	-mno-avx256-split-unaligned-store
Reduce register live range	-flive-range-shrinkage
Enable vectorization	-ftree-vectorize
Local arrays are put on stack	-fstack-arrays
Perform branch target register load optimization	-fbranch-target-load-optimize
AMD Math library	-L/opt/ACML/acml-6-1-0-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"

Architecture	
Generate instructions that run on Embedded R-Series SOC	-march=core-avx2
Optimization Levels	
Disable optimizations	-O0
Code size and locality optimization	-O1
Code speed optimization (default)	-O2
More aggressive loop and memory optimizations, prefetching	-O3
Maximize code speed	-fast
Additional Optimizations	
General compiler-guided loop unrolling	-unroll
Complete loop unrolling for loops with small counts	-unroll-aggressive
Improve precision of floating division	-prec-div
Inter-procedural optimizations	-ipo
Loop vectorization enabled	-vec
Compiler analysis based loop vectorization	-vec-threshold100
Enable code vectorization	-simd
Use and replace transcendental functions with faster but less precise implementations	-fast-transcendentals
More aggressive optimization for float	-fp-model fast=1
Use alternate malloc algorithm and configure it for speed	-qopt-malloc-options=2
Profile-guided optimization	-prof-gen, -prof-use
Enable prefetching	-qopt-prefetch
AMD Math library	-L/opt/ACML/acml-6-1-0-ifort-64bit/ifort64/lib -lacml
Other options	
Enable OpenMP directives	-qopenmp
Generate multi-threaded code for loops	-parallel

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

AMD Embedded R-Series SOC

Compiler Options Quick Reference Guide

PGI compilers (pgcc, pgcpp, pgfortran)

Latest release: 15.7, July 2015

<http://www.pgroup.com>

Architecture	
Specify target processor	-tp piledriver
Optimization Levels	
Local optimizations (default)	-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
Non-temporal moves	-Mmovnt
Profile guided optimizations	-Mpfi, -Mpfo
AMD Math library	-L/opt/ACML/acml-5-3-1-pgi-64bit/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 (openc, openCC, openf95)

Latest release: 4.5.2.1 , March 2013

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>

AMD Embedded R-Series SOC

Compiler Options Quick Reference Guide

Microsoft Visual Studio 2015

Latest Release: July 2015

<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 Embedded R-Series SOC	/arch:AVX2
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>

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors, is subject to change and may be rendered inaccurate for many reasons, including but not limited to new product releases, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.