ARC® GCC-GDB
GNU Compiler Collection and Debugger

Release Notes 2.3

ARC® GCC-GDB Release Notes 2.3

Virage Logic Corporation

47100 Bayside Parkway
Fremont, California 94538, USA
Tel: +1-510-360-8000
Toll-free: 877-360-6690

www.viragelogic.com

6008-003 March 2010

Contents

Chapter 1 — Introduction

This document provides the latest information on the ARC GNU tools, which consist of the ARC GCC toolchain and the ARC GDB debugger—a complete compiler-debugger toolchain for building and debugging native and Linux-user-mode applications.

ARC GNU includes the ARC GDB Debugger, but applications built with the ARC GCC toolchain can also be debugged using the MetaWare debugger.

The following information about this release is available:

Supported Platforms

ARC GNU supports the following products and platforms:

• Red Hat	Linux	RHEL 3.0 (kernel 2.4), RHEL4.0/5.0 (kernel 2.6)
• Microsoft	Windows XP	2002 Service pack 2

Windows users should use Cygwin™ to build GCC.

Features

The ARC GNU package consists of the following components

- GCC 4.2.1
- binutils 2.17
- GDB/Insight 6.6 (including gdbserver)
- newlib 1.15.0 C runtime library (for bare-metal applications without an operating system)
- uClibc 0.9.29 C runtime library (for user-mode Linux applications)

The package is shipped in source form, allowing it to be built and installed on any suitable platform.

Key Features of the ARC GCC Toolchain v2.3 Package

- Maintenance release with various bug fixes
- DSP and SIMD extension-instruction support is now up-to-date.

- Improved build procedures

Key Features of the ARC GCC Toolchain v2.0 Package

- Supports ARC 600 and ARC 700 CPUs.
  - — Targets 16 and 32 bit instructions.
  - — Supports zero-overhead loops.
- Two toolchains may be built using the provided sources (including GDB).
  - — `ARC-ELF32` for bare-metal applications (no operating system)
  - — `ARC-LINUX-UCLIBC` for user-mode Linux applications
- Provides compiler intrinsics for accessing low-level CPU features including the following:
  - — System instructions (e.g., **BRK**, **SYNC**)
  - — Auxiliary-space access (**LR**, **SR**)
  - — SIMD instructions
- Both the newlib and uClibc libraries use a consistent trap0 mechanism to issue system calls for low level I/O.
  - — Allows immediate use of low-level I/O using the ARC xISS fast instruction set simulator
  - — Simplifies user interface for all low level I/O, as you only need to write a trap handler for the hardware
- Optional hostlink library for low-level I/O via the MetaWare debugger.
- The GDB installation includes Insight and gdbserver
  - — Insight is a graphical user interface for GDB. It can be used either for native debugging (`arc-elf32-insight`) or for Linux ARC7 00 application debugging (`arc-linux-uclibc-insight`).
  - — gdbserver is a Linux user-mode application for debugging user-mode applications running on the ARC Linux port.
  - — GDB supports native debugging on ARC 600 and ARC 700 processors using the CGEN ARC Simulator. This simulator is included in the ARC GDB release (`arc-elf32-gdb`).
  - — GDB supports the xISS (Fast Instruction Set Simulator for ARC processors) for native debugging on ARC 600 and ARC 700 processors. xISS is not included in the ARC GDB release (arc-elf32-gdb).
  - — GDB supports AA4 Hardware Emulator (ARCangel 4) for native debugging on ARC 600 and ARC 700 processors (`arc-elf32-gdb`).
  - — GDB supports ARC 700 Linux application debugging using gdbserver running on the Linux embedded target. GDB communicates with gdbserver via a TCP/IP socket connection (`arc-linux-uclibc-gdb`).

Deliverables List

Product	Components	Description
ARC GNU toolchain	GCC compiler, binutils, Newlib, uClibc, GDB, insight	ARC GNU toolchain – cross-compilation and debugging toolchain

Installation Structure

The ARC GNU toolchain is supplied as a source package (tarball for Linux/Cygwin™) and/or as an installer for pre-built Cygwin™ binaries, which generally includes. ARC GDB.

Table 1 Directory Structure for ARC GNU Sources

Directory/Files	Description
`README.ARC`	Readme file for building ARC GNU.
`README.arcgdb`	Readme file for building GDB/insight toolkit. This includes elf32, Linux versions of GDB/Insight as well as gdbserver ARC 700 Linux.
`build_elf32.sh`	Build script for ELF32 version of toolkit.
`build_uclibc.sh`	Build script for UCLIBC version of toolkit.
`binutils`	Sources for binutils.
`gcc`	Sources for gcc (including newlib).
`insight`	Sources for insight/gdb/gdbserver.
`uClibc-0.9.29`	Sources for uclibc.

Table 2 Installation Structure for Pre-Built ARC GNU binaries

Directory	Files	Description
`$INSTALLDIR/arc/gcc/docs`	`*.pdf`	Additional documentation for the toolkit
`$INSTALLDIR/arc/gcc/elf32`		Bare-metal version of toolkit.
`/bin`	`arc-elf32-*`	Binaries for compiler, binutils, GDB and insight.
`/lib`	`*.lib/*.so.`	- C/C++ libraries (newlib)
`/include`	`*.h`	- C/C++ include files
`/info`	`*.info`	- info manual pages
`/man`		- man manual pages
`$INSTALLDIR/arc/gcc/uclibc`		For user-mode Linux Applications
`/bin`	`arc-linux-uclibc-*`	Binaries for compiler, binutils, GDB and insight.
`/lib`	`*.lib/*.so.`	- C/C++ libraries (uClibc)
`/include`	`*.h`	- C/C++ include files
`/info`	`*.info`	- info manual pages
`/man`		- man manual pages

Table 3 Installation Structure for Pre-Built ARC GDB Binaries (elf32)

Directory	Files	Description
`$INSTALLDIR/elf32/`		gdb/insight install directory after build from sources completed. (bare metal or elf32)
`/lib`	`*.a, *.tcl`	Tk/Tcl libraries, Cgen library, Insight plug in.
`/bin`	`arc-elf32-*`	Binaries for GDB and insight.
`/include`	`*.h`	C/C++ include files to interface with Tcl
`/info`	`*.info`	Info manual pages
`/man`		Man manual pages
`Directory`	`Files`	Description

Table 4 Installation Structure for Pre-Built ARC GDB Binaries (Linux)

Directory	Files	Description
`$INSTALLDIR/linux/`		GDB/insight install directory after build from sources completed. (ARC 700 Linux application debugging)
`/lib`	`*.a, *.tcl`	Tk/Tcl libraries, Cgen library, Insight plug in.
`/bin`	`arc-linux-uclibc-*`	Binaries for GDB and insight.
`/include`	`*.h`	C/C++ include files to interface with Tcl
`/info`	`*.info`	Info manual pages
`/man`		Man manual pages
`Directory`	`Files`	Description

Documents

ARC GNU

ARC GNU is supplied with documentation consistent with other GCC ports (i.e. info pages, README and Changelog files). In addition specific ARC files have been included (with the '.ARC' suffix) to provide information specific to the ARC port within the overall source package.

Separate documentation, in Adobe® PDF format, is located in the *InstallationDir*`/gcc/docs` folder.

Table 5 PDF Documentation.

Directory/Files	Description
`gcc.pdf`	GCC Compiler user manual.
`binutils.pdf`	Binutils user manual.
`as.pdf`	Assembler user manual.
`ld.pdf`	Linker user manual.
`libc.pdf/libm.pdf`	Libraries user manual.
`ARCGCC_GettingStarted.pdf`	ARC specific modifications to compiler toolkit.

Directory/Files	Description
gdb.pdf	GDB debugger's User manual.
ARC_GDB_GettingStarted.pdf	ARC-specific GDB information on features and how to use GDB with ARC processors.

Chapter 2 — Release Change Information

The major changes from previous releases of ARC GNU are described here.

ARC GCC

Improvements in Release 2.3

- Maintenance release with various bug fixes
- DSP and SIMD extension-instruction support is now up-to-date.
- Improved build procedures

ARC-Specific Bugs Fixed in Release 2.3

- 95619 GCC 2.2: cannot compile with option **–fPIC** and **–mlong** calls
- 98561 `QtMapIterator` not working as expected: Relocation entry bad
- 98569 **-ffixed-lp_count** causes a crash
- 93842 mic library issues
- 94994 Shift problem with **double** data type
- 96081 `limits.h` not including the same files as pre-patch version
- 96236 `arc-elf32-as` floating point exception while configuring libgcc
- 92532 Add X/Y memory, **xmac_d16**, and other additional extensions to Gnu Assembler
- 92533 Upgrade GCC assembler support for SIMD to newer version and a large set of additional instructions
- 92995 C7 shared libs.
- 93181 Assembler in candidate 8 is broken for audio codec software.
- 94059 Wrong instruction generated with **-mmul64** on ARC 600
- 94142 Assembler encodes **ld.a** instruction incorrectly
- 94723 `mp3_lp_decode` asm conversions are not correct
- 92164 Assembler generates warning about 8-byte instruction in delay slot; program crashes
- 92194 Assembler error while compiling a benchmark program.

- 92195 Compiler internal error while compiling a benchmark program.
- 92196 **memcpy()** seems slow compared to GCC 3.4 **memcpy()**
- 92442 C++ link with pthread
- 92457 printing floating-point values prints garbage on ARC 600, works on ARC 700
- 92552 **memcmp()**result is not right after the first address is not aligned to DWORD
- 92609 Compiling any program with g++ with **-pg** option gives section-overlap errors
- 92628 Internal compiler error while compiling attached file
- 92664 Internal compiler error with **-mmixed-code** option
- 92665 Bad instructions generated with **-mmixed-code** and **-mARC600** options
- 92732 Compiler error while compiling one of the files from the H264 decoder
- 92996 **-static** should be the default.
- 93836 Compiler generates illegal instructions when building `uClibc`
- 93837 Compiler generates illegal instructions when building Linux UDF filesystem
- 94379 ARC 600 Regression: **printf()** prints garbage for an **int**
- 94420 EEMBC - `pktflow` is failing - `lp_count` is read immediately after write
- 94508 `mpeg2decode` fails with dynamic linker (illegal instruction) - `gtdf2` is resolved incorrectly
- 94684 Compiler produces illegal instruction building `pppd`
- 94782 Zero-overhead loop with only one instruction is generated on ARC 600, which is illegal.
- 94838 **find()** returns `1` rather than `-1` when test value not present in searched string
- 95163 Conversion from a **double** value to an **int** value is incorrect
- 95197 ARC `uClibc` doesn't provide **sched_setaffinity()** and **sched_getaffinity()** calls
- 94329 **pthread_create** crashes application
- 95110 Locale support not working in `uClibc` 0.9.29 port
- 95236 New option **-mmul32x16** causes link failure - unresolved symbols
- 91935 Compile failed: `__ASSEMBLY__` not defined
- 92208 Need hostlink or equivalent solution to run programs with `stdlibc` calls on hardware
- 92917 Linker does not put the correct start address and offset in the ELF program segment header
- 92999 Test suite fails to build with Candidate 7.
- 93238 **gcc** hostlink fails to create a new file using **fopen()**
- 95291 Code generations errors with **-mmul32x16 -mnorm**
- 95438 Warning: 8 byte instructions in delay slots. May be incorrect code generation
- 95510 **openssh** fails to work with default shared-library mode.
- 95556 Application `page_fault` before getting into **main()** if using shared-library link

- 95728 Internal compiler error with **fPIC –mlong**-calls and **-fomit-frame-pointer**
- 95773 Function prologue does not save `r14`, `r15` and `r16` and overwrites the values
- 95949 Code-generation error: 8-byte instruction in the delay slot
- 93610 Unknown symbol `_GLOBAL_OFFSET_TABLE_` while using **insmod** to load driver module
- 98450 GNUtools optimization problem on QT
- 93962 **mmap()** access wrong address with **gcc** option **-mdynamic -D_FILE_OFFSET_BITS=***64*
- 95994 **mul64** assembled with extranous limm
- 96496 Assembler rejects store loop for **-mA6**
- 92626 ARC objdump and **--no-show-raw-insn** switch
- 93609 Wrong expression value with **-O2**
- 95109 Bad instruction generated when building Linux CIFS support with **-Os**
- 95241 Bad instruction seen when building compiler?
- 95349 `ucLibc`: webkit webbrowser cross-compiling error report
- 98148 Create the latest version of ARC GCC toolchain from different patches
- 94078 `MemSiz` reported in program headers is too big compared to sizes in section headers
- 95055 Function **pthread_attr_getstackaddr()** returns `0` stack address
- 95716 Attempting to view system registers in insight causes internal error
- 98419 HTTP test application crash inside **fwrite()**
- 92481 Place new GNU tools release on `projects.arc.com` and `support.arc.com`
- 94039 sda symbol + offset not handled correctly
- 94259 Assembler refuses to assemble loop end after branch
- 95931 ARC GCC Linux tutorial shows incorrect BusyBox **cpio** version
- 92064 Problem with **gas** version of VNOP
- 92810 64-bit instructions need extended `arcextmap` table.
- 92872 Need to supply `arcextmap` definitions for SIMD instructions
- 94353 Accumulator register names pollute symbol namespace
- 94386 Loop with long-immediate store rejected
- 95046 **vmulw.0x1** is not assembled correctly by **gas**
- 95340 Negation of `i` register lane mask not correct.
- 92602 **while** loop exits after first iteration
- 92757 Internal error building `openssl c_skey.c`
- 94690 ARC GNU installer puts forward slashes in Windows `PATH` variable

- 95134 Invalid DWARF for parameter locations
- 95466 C++ link fails with overflow detected in relocation value
- 98225 ARC GCC support question
- 98367 Request to add option to disable zero-overhead loop in **gcc**
- 98368 Reference to `__umoddi3` and `__udivdi3`
- 95254 DSP multiply is assembled incorrectly
- 98310 Document update regarding building Release 2.2 on a 64-bit machine
- 93844 Compiling `uClibc` needs to be done in foreground
- 93845 Procedure required for overall building of Linux
- 94768 **objdump** doesn't disassemble **mpyhu** properly
- 95307 **mul** operator giving internal compiler error for ARC 600 **mul64** and **mul32x16**
- 98395 The **.extinstruction** assembly directive doesn't extend to ARC 600
- 98402 ARC GCC assembly support for DSP instructions
- 98407 **arc-elf32-ld** dumps core with segmentation fault
- 98601 How to set heap and stack size?
- 94076 `crti`and `crtn` need alignment
- 95132 **%f** format specifier gives garbage output
- 95190 **main()** *argv* parameter not set up correctly
- 95315 **info** *frame N* command crashes **gdb**
- 95316 Problems with restarting program execution with **run** command
- 96005 Problem with **pthread** and **fork**
- 97653 `uClibc` build script changes to build with 2.6.30 kernel headers
- 93514 Seg fault while linking u-boot

Improvements in Release 2.0

- Build scripts for automated building of either, or both, targets are included. This simplifies the building of `ARC-ELF32` and `ARC-LINUX-UCLIBC` targets.
- The ARC GCC toolchain has been updated to make greater use of the ARC 600 and ARC 700 instruction sets. This enables it to produce faster and smaller code than previous releases.
- The toolchain includes new command-line switches for taking advantage of additional (optional) instructions (see the *Getting Started* manual for details).

ARC GDB

New Functionality in Release 2.3

Maintenance release with various bug fixes

ARC-Specific Bugs Fixed in Release 2.3

- 95375 Load .ELF binary to `arcjtag` target causes segmentation fault
- 95538 ARC-aux-read/write commands do not work when target is simulator
- 95633 Programs run on AA4 do not terminate
- 95715 Bad formatting in insight assembly window
- 95746 `Can not set up arguments to program` warning from insight
- 95787 data cache mis-handled
- 96046 Seg fault when trying to debug `libstdc++` test-suite binary
- 96160 **arc-reset-board** not effective
- 96585 Architectural version check fails for ARCtangent-A5 executable files
- 96849 No endianness check for ARC JTAG or xISS targets
- 93852 Can a **gcc**-compiled program work on ISS
- 94655 No hardware breakpoints

New Functionality in Release 2.2

- A remote I/O mechanism has been added to `arc-elf32-gdb` for use when connected to targets via the JTAG interface. It supports the basic I/O system calls **open()**, **close()**, **read()**, **write()** and **lseek()**. This means that a program running on an ARCangel 4 target under the control of this debugger can now read and write files on the host, including the `stdin`, `stdout` and `stderr` streams; in particular, **printf()** now works.
- The remote I/O mechanism also intercepts the **exit()** system call performed by the executable program's start-up code after return from the executable's **main()** function, so that when the program terminates a `Program exited normally` message is displayed by the debugger instead of the program going into a silent, endless loop at address 0x100.
- The system calls **gettimeofday()** and **fstat()** are also supported.

- Facilities for tracing the execution of instructions by the xISS (the Fast Instruction Set Simulator) have been added.
- The `arc-elf32-gdb` debugger now sets fields of auxiliary registers to the values they are architecturally required to have when written (as specified by the `onwrite` attributes in the XML file that describes the target's auxiliary-register set).

New Commands Added in Release 2.2

- **target arcxiss** – Use xISS as the target.
- **arc-blast-board** – Configure the ARCangel target-board FPGA using an XBF file
- **arc-set-clock-frequency** – Set the ARCangel target-board clock frequency
- **arc-set-clock-source** – Set the sources of the ARCangel target-board clocks
- **info arc-fpga** – Check whether the ARCangel target-board FPGA has been configured
- **info arc-clock-settings** – Show the ARCangel target-board clock settings
- **set arc-xiss-trace on | off** – Trace or stop tracing the instructions executed by the xISS
- **set arc-xiss-trace-file** – Output the trace of the instructions executed by the xISS to a file
- **set arc-xiss-trace-buffer-size** – Set the size of the instruction address trace buffer
- **arc-xiss-empty-trace-buffer** – Empty the instruction address tracing buffer
- **arc-xiss-save-trace** – Save the contents of the instruction address tracing buffer to a file
- **arc-xiss-list-trace** – List the instructions executed from an instruction address trace

Existing Commands Altered in Release 2.2

- **target arcjtag** – A *noreset* option has been added which allows connection to be made to a target board without resetting it
- **target arcjtag** – An optional *XBF_file* parameter has been added which allows the target FPGA to be blasted upon connection to the target
- **arc-reset-board** – The target board's clock source and frequency settings are restored to their default (power-up) values
- The help or usage information for all ARC-specific commands has been made consistent.

GDB Commands Implemented in Release 2.2

attach – Reattach to an ARCangel target (without a board reset) within the same debugging session

ARC-Specific Bugs Fixed in Release 2.2

This information is taken from the ARC Bugzilla database.

- 95725 bad error message when printing register R61
- 96046 SEGFAULT trying to debug libstdc++ testsuite binary
- 95633 programs run on AA4 do not terminate
- 95722 intermittent failure when selecting executable file in insight

- 95538 arc-aux-read/write commands do not work when target is simulator
- 95530 'info all-registers' command fails when target is simulator
- 95726 bad error messages when trying to set register PCL
- 95883 gdb default architecture is A4, but built-in simulator default architecture is ARC 700
- 95714 segmentation fault in insight when viewing registers
- 95724 extension core registers not handled properly
- 95746 "Can not set up arguments to program" warning from insight
- 95712 insight 'Disconnect' menu item has no effect
- 95732 insight fails to read aux reg AUX_IRQ_LV12 twice
- 95635 insight 'Stop' button does not work when target is arcjtag
- 95787 data cache mishandled
- 95715 bad formatting in insight assembly window
- 95674 main argc/argv parameters not set up for JTAG target
- 95682 Ctrl-C has no effect if program is in I/O loop on arcjtag target
- 95770 unhelpful insight errors message on target connection failure
- 95579 program can not be re-run on arcjtag target
- 95716 attempting to view "system" registers in insight causes internal error
- 96196 no consistency check on simulator / executable endianness

Chapter 3 — Known Limitations and Issues

Up to date information regarding the ARC port of GCC and GDB is located on the customer support site: https://support.arc.com

Generic GCC issues can be found at http://gcc.gnu.org/bugzilla

Generic GCC issues can be found at the following site: http://sourceware.org/gdb/bugs

Known ARC GCC Issues

- You must have **make** version 3.80 or greater installed to build uClibc.
- ARC GCC Release v2.2 cannot be built on a 64-bit machine.
- The small-data section has a limited size, currently the GNU linker cannot select/optimize the use of that section. This means that if an application contains too many modules requiring allocation of the small-data section, then an overflow might occur. In this situation, the linker reports an overflow message. We advise carefully selecting which modules have the most benefit from using the small-data section and compiling the other modules with **-mno-sdata** to disable use of the small-data section.
- Do not use the small-data section for Linux user-mode applications, as the GOT is referenced via the global pointer for dynamic linking.
- The ARCtangent-A4 architecture is neither supported, nor maintained by ARC. This means that the ARC GCC toolchain should not be used to target this architecture.
- ARC officially supports RedHat Enterprise Linux, however users using Fedora Core 8 (or other distributions) may see build issues related to the version of makeinfo installed. Fedora Core 8 includes texinfo 4.11, this package requires texinfo 4.8. Downgrade your copy of texinfo to build on an unsupported platform.

Known ARC GDB Issues

GDB Bugs Not Fixed

- 95217 **gdb** does not send signal to simulator
- There is no support for signals in the ARC CGEN built-in simulator.
- 96209 remote **fstat()** call returns incorrect **st_blksize** and **st_blocks** data
  This is a GDB build-configuration issue.

GCC Bugs Related to GDB

- 95191 DWARF-2 location lists for parameter locations would be useful (enhancement request)
- 95281 incorrect DWARF for K&R-style function parameters
- 95358 extra level of reference in C++ function parameter in DWARF
- 95203 compiling .S file crashes compiler with segmentation violation

ARCangel 4-Related Bug

95691 instruction single-stepping does not work on ARC 600