[Codito Technologies]

ARC GDB User Doc.

< www.codito.com/arc >

arc-support@codito.com

CONTENTS

INTRODUCTION

This manual provides an overview of the GDB provided for the ARC700 and ARC600 processors.

This document is intended to be used in conjunction with the online documentation provided with GDB . Please refer to Appendix A for more information on using the GDB online documents.

Versions of tools provided in the release:

arc-linux-uclibc-gdb : 6.3_codito_20060518
arc-linux-uclibc-gdbserver : 6.3_codito_20060518
arc-elf32-gdb : 6.3_codito_20060518
arc-a4-elf32-gdb : 6.3_codito_20060518

Building GDB

This chapter covers the details of building GDB and GDBserver for the ARC platform.

2.1 Configurations

- **arc-elf32:** This is for a ARC600 / ARC700 build for debugging using JTAG. With the JTAG target the gpio driver provided with Seecode can be used for debugging. This is a requirement for gdb to work on Linux. Currently gdb has been tested with JTAG support on GNU / Linux platforms only.
- **arc-a4-elf32:** This is for a A4 build for debugging using JTAG. With the JTAG target the gpio driver provided with Seecode can be used for debugging. This is a requirement for gdb to work on Linux. Currently gdb has been tested with JTAG support on GNU / Linux platforms only.
- **arc-linux-uclibc:** This is for a build that supports Linux Application debugging using gdbserver on the ARC700 platform. This configuration allows for remote debugging of GNU / Linux applications on the ARC700 Linux platform.

2.2 Compiling and Installing

2.2.1 Building GDB

2.2.1.1 Software Required

- Host GNU C Compiler.
- Autotools.
- C Development Environment on the Host where GDB is being built.

2.2.1.2 Instructions

On the shell prompt, please follow the following instructions to build gdb.

```
$> mkdir build-dir
$> cd build-dir
$>$SRC_DIR/configure --target=$TARGET_OPT --prefix=$INSTALL_DIR
$> make
$> make install
```

(In case TCL libraries are not present on the host system, please disable gdbtui by giving the configure option –disable-tui.)

2.2.2 Building GDBServer (only for ARC700)

2.2.2.1 Software Required

- ARC GNU Tools (GCC 3.4.x / binutils 2.15) : arc-linux-uclibc 20060222 release
- uClibc 0.9.27 – 20050515 release.

2.2.2.2 Build Instructions

To build gdbserver, first build the uclibc libraries with **pthreads debug** support **ON**. Then use the following script:

(On the host).

```
#!/bin/sh

if [ $# -lt 1 ]
then
        echo "Usage: $0 SOURCE_DIR"
        return
fi

SOURCE_DIR=$1
export CFLAGS="-mA7"
export CC=arc-linux-uclibc-gcc

$1/gdb/gdbserver/configure –target=arc-linux-uclibc

make
```

Using GDB

3.1 Software required for running GDB

GDB can be run on any standard linux distribution. The only requirement for software is the presence of the gpio driver. GDB reuses the driver provided as gpio for its communications with the JTAG module.

3.2 Remote Debugging

Typically on embedded processors, it is not possible to run a debugger on the embedded target due to footprint reasons. Hence in such circumstances, remote debugging is the only way forward. In case of the GDB port for the ARC, standard GDB remote debugging is allowed with the arc-linux-uclibc target. In case of the JTAG target, a separate command to connect to the JTAG target has been added to GDB. (Note: A target in GDB refers to the mechanism used by the debugger to communicate to the debuggee.)

3.3 Commands Added

target arcjtag

Usage: target arcjtag

Connect to the arcjtag target.

This target expects the ARC board to be connected to the parallel port on the host. Currently we support debugging only on GNU/Linux hosts. This target uses the gpio device to access the parallel port. You must have the gpio driver installed and you must have read/write privileges to /dev/gpio.

arc-reset-board

Usage: arc-reset-board

Reset the board.

For this command to work, you must be connected to the arcjtag target, by using the command *target arcjtag*.

arc-aux-read

Usage: arc-aux-read <REG-FROM> [<REG-TO>]

Read and show a range of auxillary registers. REG-FROM and REG-TO can be any expressions that evaluate to integers. REG-TO is optional; if it is not specified, only one register is displayed.

For example:

```
(gdb) arc-aux-read 0x400 0x406
00000400: 00000100 80001abc 00001620 00200000
00000404: 00000100 00000000 00000000
```

For this command to work, you must be connected to the arcjtag target, by using the command *target arcjtag*.

arc-aux-write

Usage: arc-aux-write <REG> = <VALUE>

Write to an auxillary register. REG and VALUE can be any expressions that evaluate to integers.

For example:

```
(gdb) arc-aux-write 6 = 0x123
```

For this command to work, you must be connected to the arcjtag target, by using the command *target arcjtag*.

info arc-bcr-registers

Usage: info arc-bcr-registers

Show all the build configuration registers.

For example:

```
(gdb) info arc-bcr-registers
[61] DCCM_BASE_BUILD : 0x1010121
[62] CRC_BASE_BUILD : 0x00
[63] BTA_LINK_BUILD : 0x1010121
[64] DVBF_BUILD : 0x00
[65] TEL_INSTR_BUILD : 0x00
[67] MEMSUBSYS : 0x01
[68] VECBASE_AC_BUILD : 0x01
[69] P_BASE_ADDRESS : 0xfc0001
[6f] MMU_BUILD : 0x1010121
[70] ARCANGEL_BUILD : 0x1010121
[72] D_CACHE_BUILD : 0x12001
[73] MADI_BUILD : 0x00
```

```
[74] DCCM_BUILD : 0x00
[75] TIMER_BUILD : 0x303
[76] AP_BUILD : 0x00
[77] ICACHE_BUILD : 0x22001
[78] ICCM_BUILD : 0x1010121
[79] DSPRAM_BUILD : 0x1203
[7a] MAC_BUILD : 0x00
[7b] MULTIPLY_BUILD : 0x01
[7c] SWAP_BUILD : 0x01
[7d] NORM_BUILD : 0x02
[7e] MINMAX_BUILD : 0x00
[7f] BARREL_BUILD : 0x02
```

For this command to work, you must be connected to the arcjtag target, by using the command *target arcjtag*.

set arcjtag-debug-statemachine

Usage: set arcjtag-debug-statemachine ARG

Switch on JTAG state machine debugging messages if ARG is non-zero. Switch them off if it is zero.

set arcjtag-debug-target

Usage: set arcjtag-debug-target ARG

Switch on JTAG target debugging messages if ARG is non-zero. Switch them off if it is zero.

3.4 Tutorial

Target remote

The gdbserver binary must be accessible from a shell on the board. (You could do this by putting it in an NFS share.)

Telnet to the board. Suppose the file you want to debug is called foo. Then run the command

```
[arclinux]$ ./gdbserver :4444 foo
Process ./foo created; pid = 102
Listening on port 4444
```

On the host, start arc-linux-uclibc-gdb:

```
$ arc-linux-uclibc-gdb
(gdb) file /path/to/foo
Reading symbols from /path/to/foo...done.
```

```
(gdb) target remote arcboard:4444
Remote debugging using arcboard:4444
0x00000098 in start ()
(gdb) continue
```

Here, arcboard is the hostname of the board. The above example uses port 4444, you can use some other port if you want to.

Note the use of continue. Do not use run.

Target arcjtag

This must be run on a machine that is connected to the ARC board. You'll need the gpio driver and read/write permissions on /dev/gpio.

```
$ arc-elf32-gdb
(gdb) file /path/to/foo
Reading symbols from /path/to/foo...done.
(gdb) target arcjtag
Connected to the arcjtag target.
(gdb) load
Loading section .text, size 0x3d0 lma 0x74
Loading section .data, size 0x34 lma 0x2444
Start address 0x74, load size 1028
Transfer rate: 8224 bits in <1 sec, 342 bytes/write.
(gdb) run
```

For help on other commands, see the GDB user manual.

3.5 Frontends

GDB can be used with a number of industry standard frontends. GDB is used with the Eclipse CDT plugin (C Development Toolkit) front end and DDD (the Data Display Debugger.)

arc-linux-uclibc-gdb has been used with the Eclipse CDT plugin. Please refer to the UsingEclipseCDT doc for more details. The DDD debugger frontend can be used with arc-linux-uclibc-gdb as well as arc-elf32-gdb.

In addition arc-linux-uclibc-gdbtui is a standard TCL based fullscreen command line frontend available for GDB. This is available on building gdb with –enable-tui option (by default.)

References

- Codito ARC Linux user guide : ARC Linux User guide, build and install ARC Linux / uClibc and ARC GNU Tools
- Codito ARC GNU Tools User Guide : ARC GCC/Binutils user guide.
- GDB Online Documentation : Complete GDB reference manual. Contains documentation for all GDB commands. ARC specific commands can be found in this document.
- DDD Frontend : DDD frontend to the GDB debugger.
- DDD Manual : DDD Online manual.

Appendix A

1. Using GNU Info

GNU Info is a program, which can be used for reading info documents. The GNU Project distributes most of its on-line manuals in the Info format.

To access GNU Info or Info help system, use the following command

info -f filename

The info files for GNU C Compiler Toolkit are installed under /usr/local/arc/info directory. For example, GCC documentation can be accessed via info as follows:

info –f <install-path>/gdb.info