

Using GNU RISC-V GDB

Abstract

This document describes the configuration of TESSY's adaption for the GNU RISC-V GDB debugger. At the time of writing this document version 9.1 of the GNU RISC-V GDB has been used for the adaption. Interactive debugging is only supported by utilizing TESSY's special built-in data binary feature. See chapter 4 for further details. The default configuration was tested using the GNU RISC-V 64-bit compiler (version 10.1.0) with the integrated simulator of its GNU RISC-V 64-bit GDB. After adapting the `Compiler Install Path` to your local GNU RISC-V compiler installation path, TESSY's example test should run out-of-the box.

Important Note: Check your local project configuration settings for the proper CPU and ABI values of the compiler.

Please note: The GNU RISC-V GDB adaption does not support interactive debugging features when executing tests with TESSY. (See 4 to learn how to debug interactively having your test data statically built into the target binary.)

Table of Contents

Abstract	1
1 Introduction.....	2
2 TESSY Environment Settings.....	2
2.1 GDB Simulator	2
2.2 Optional GDB server	2
3 Parallel Built and Parallel Execution.....	3
4 Interactive Debugging.....	3
5 Troubleshooting.....	5

1 Introduction

The communication between TESSY and GNU RISC-V GDB is based partially on GDB scripting language and on direct commands sent to the GDB client. By default, TESSY supports GDB's integrated simulator. If a GDB server is needed, TEE attribute **GDB Server Debugger** has to be set accordingly. Refer to chapter 2.2 for further details.

In order to debug the test application interactively with the test case data provided from TDE you need to rebuild the test application in a special mode, i.e. the input values will be compiled into the target binary. Please refer to chapter 4 for further details on how to generate this special target binary.

2 TESSY Environment Settings

At the time of writing this document, only the combination of GNU RISC-V compiler with GNU RISC-V GDB was available. So, at first, set the **Compiler Install Path**. If errors remain, check the attribute's value of **Compiler Base Name** and compare the name with GCC binary of your GNU RISC-V compiler installation. Next, check the attributes **CPU** and **ABI**. If you are using a CPU with 32-bit base integer instruction set, you will have to set attribute **Base Integer Instruction Set** to `32-bit`. The default settings do not need any additional startup code or linker file. But if you adapted attributes CPU and ABI, it may be necessary to utilize attribute **InitObjDir** and **Linker File**. See application note *063 Startup Code of the Test Program*.

2.1 GDB Simulator

By default, TESSY supports GDB's integrated simulator. That means that attribute **GDI Parameter** is set to `sim`.

2.2 Optional GDB server

Using a GDB server instead of the GDB's built-in simulator, you may set attribute **Debugger Path** to the absolute path of the GDB server executable and use it on the command line given by attribute **GDB Server Debugger**. Please enable TEE's expert mode to display the attributes in the TEE. TESSY provides a feature to determine the port number from the debugger's output messages. TESSY uses the contents of `String` type attribute **Port Trigger**, which has to be created in TEE, to find the unique text fragment right before the port number. So, the port number of the debugger's output message itself is expected right after the given text of attribute **Port Trigger**. Make sure that the GDB server choose a port number on its own. So, it is strongly recommended **not** to use option `-p [NUMBER]`, so that the debugger may let the operating system choose a convenient port. Leave attribute **GDB Port Number** at 0. Finally, attribute **GDI Parameter** has to be set to `extended-remote localhost` in that case or to `extended-remote localhost:$(GDB Port Number)` if you want to use a predefined GDB server port number.

3 Parallel Built and Parallel Execution

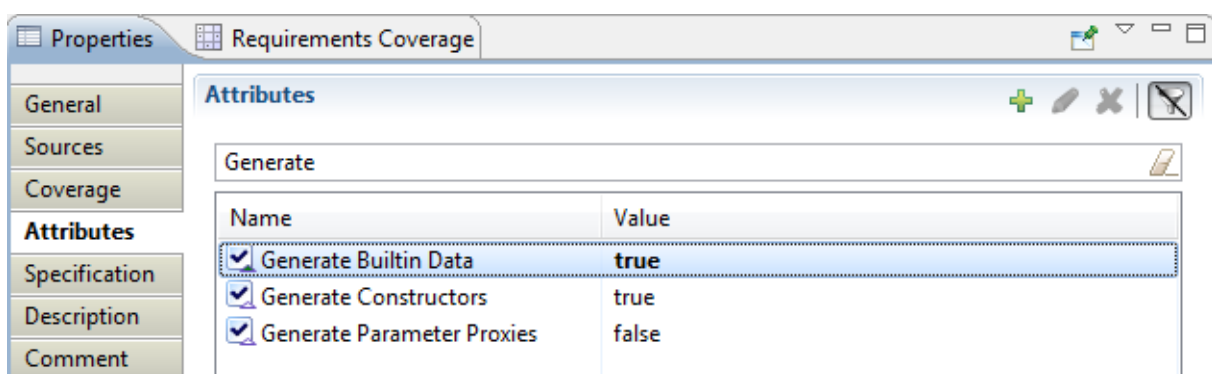
The amount of test objects built in parallel is given by **Compiler Concurrency**, which is by default set to 4, while 20 source files per test object are compiled in parallel. So, no more than 80 processes run by default in parallel for the built. You can disable the parallel built by setting TEE attribute **Compiler Concurrency** to 1 and by clearing the TEE attribute **Make Options**.

The parallel execution represented by attribute **Target Concurrency** is set to 10. That means ten module tests run at a time by default. If you are using the same target devices, it might be possible to run multiple target tests each on its own target device in parallel. The corresponding master of TESSY supports it.

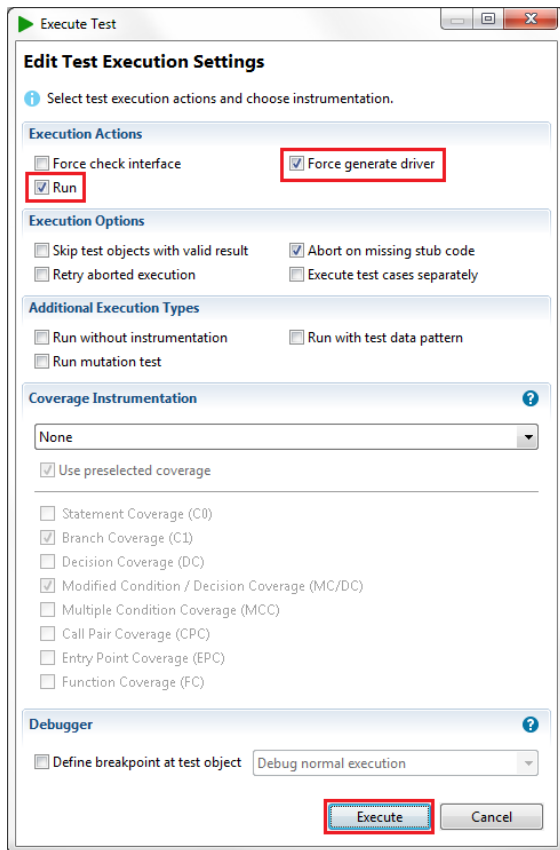
Too many compiler or debugger processes in parallel may slow down the test execution. It depends on your computer system. So, feel free to test different values for **Compiler Concurrency** and **Target Concurrency**. You have to **Enable Expert Mode** from the **Attributes** view's toolbar to alter these attributes.

4 Interactive Debugging

In order to debug the test application interactively with the test case data provided from TDE you need to rebuild the test application in a special mode, i.e. the input data will be compiled into the target binary. TESSY provides the TEE attribute **Generate Builtin Data** to facilitate this task. The attribute is of type `Boolean` and, if set to `true`, TESSY will rebuild your target binary during the next test run having the selected test data built-in, i.e. TESSY will not actually perform the test run but instead create the target binary with test data built-in to it. After the build process TESSY will launch a command shell with a running GDB client and a proper script to run the built binary and stop at the test object. To disable this feature, you have to set the attribute to `false`.



Open the **Execute Test** dialog and make sure **Force Generate Driver** is selected.



Now execute the test by pressing the **Execute** button. After the built, TESSY displays the path to the generated built-in target binary within its **Console** view and displays some helpful commands for the started GDB client. The default GDB script TESSY uses to launch the client is configured for GDB's built-in debugger. If you need to start a GDB server beforehand, please add the command line to the Makefile template, copy TESSY GDB client script into your TESSY project, and adjust the file accordingly.

```

+-----+
| Compile/Link Driver [IsValueInRange/is_value_in_range] |
+-----+

***** Compiling Builtin Data *****
***** Compiling Communication Modules *****
***** Compiling Slave *****
***** Linking Slave *****
*
* Generated target binary 'E:\Projects\TESSY_5.1\GNU_RISC-
V\tessy\work\4\2\1\ts_00000403_d.elf' containing builtin data.
*
* Launching the command line debugger in a command shell.
* Please switch to the opened command shell.
*
* These simple commands may be helpful:
* c          continue the execution
* i args     display the function parameter values
* l          list source code lines around current line
* n          step over the current statement
* s          step into the subfunction or step over
* p variable print the value of the given variable
* q          quit the debugging session
* tcs       list current test case
* CTRL-d    quit the debugging session
*

```

TESSY's GDB script will automatically setup the GDB client, to start the simulator, load the binary, set a breakpoint at your test object function, and run to the function. Please notice the useful GDB commands listed in TESSY's **Console** view.

```
***** Test Case 1.1 *****
35 // erroneous: v1 == 7 results "yes" instead of "no"!
36 //
37
38 result is_value_in_range (struct range r1, value v1)
39 {
40     if (v1 < r1.range_start)
41         return no;
42
43     if (v1 > (r1.range_start + r1.range_len))
44         return no;
45
46     return yes;
47
48 }
49
50
51 /*****
52 /* T E S S Y C O D E */
53 /*****
54
55 #define __TS_THIS_IS_THE_SLAVE__
#0 is_value_in_range (r1=..., v1=15) at
E:\Projects\TESSY_5.1\GNU_RI~1\tessy\work\4\2\1\ts_src01.c:40
40     if (v1 < r1.range_start)
(gdb)
```

5 Troubleshooting

Please contact support@razorcat.com if you encounter any unsolvable problems.