

VectorCAST RSP ARM FVP

Version 1.0

2022-04-25

Application Note AN-ACT-1-102

Author Tim Schneider

Restrictions Public Document

Abstract Information on the VectorCAST RSP for ARM FVP Simulator

Table of Contents

1	Overview	2
2	Use Cases	3
	2.1 Selecting the Proper Configuration	3
	2.2 Setup.....	4
	2.3 Debugging.....	5
3	Additional Resources	7
4	Trademarks	7
5	Contacts.....	7

1 Overview

VectorCAST Runtime Support Package (RSP) is an extension of the VectorCAST product. It provides an interface layer that allows you to use the VectorCAST testing techniques and methods on an embedded target processor. The VectorCAST RSP will always run on your host platform (the same platform as the compiler). Only the VectorCAST generated test harness will be downloaded to, and executed on, the embedded target.

A VectorCAST RSP is always customized to a particular Target CPU, Cross Compiler, and Run-Time environment (or kernel). As such, the information presented here contains sections for this specific RSP.

This application note provides details on configuring VectorCAST to run tests on the ARM Fixed Virtual Platform (FVP) Simulators.

VectorCAST uses the ARM Clang v6 compiler and linker to build the VectorCAST test harness into an ARM AXF application. The application is loaded into the FVP for direct execution.

Once the application is in the simulator, the FVP starts running and VectorCAST uses the standard out from the FVP simulator to collect the test results.

VectorCAST provides support for two major architectures:

- > ARM
- > ARM64

Unique processor configurations support:

- | | |
|---------------|----------------|
| > Cortex-M0 | > Cortex-A35 |
| > Cortex-M0+ | > Cortex-A35MP |
| > Cortex-M1 | > Cortex-A53 |
| > Cortex-M3 | > Cortex-A53MP |
| > Cortex-M4 | > Cortex-A57 |
| > Cortex-M4F | > Cortex-A57MP |
| > Cortex-M7 | > Cortex-A72 |
| > Cortex-M7F | > Cortex-A72MP |
| > Cortex-A5 | > Cortex-A73 |
| > Cortex-A5MP | > Cortex-A73MP |
| > Cortex-A7 | > Cortex-A76 |
| > Cortex-A7MP | > Cortex-A76MP |
| > Cortex-A9 | > Cortex-A77 |
| > Cortex-A9MP | > Cortex-A77MP |
| > Cortex-A32 | |

The RSPs released for the ARM FVP support C, C++, C++11, C++14 and C++17.

If you need support for a different architecture or run into any issues configuring VectorCAST, please contact support@vector.com.

2 Use Cases

2.1 Selecting the Proper Configuration

The first thing you need to do when configuring your ARM Clang FVP RSP is to select the architecture (ARM or ARM64), processor and language seen in Figure 1.

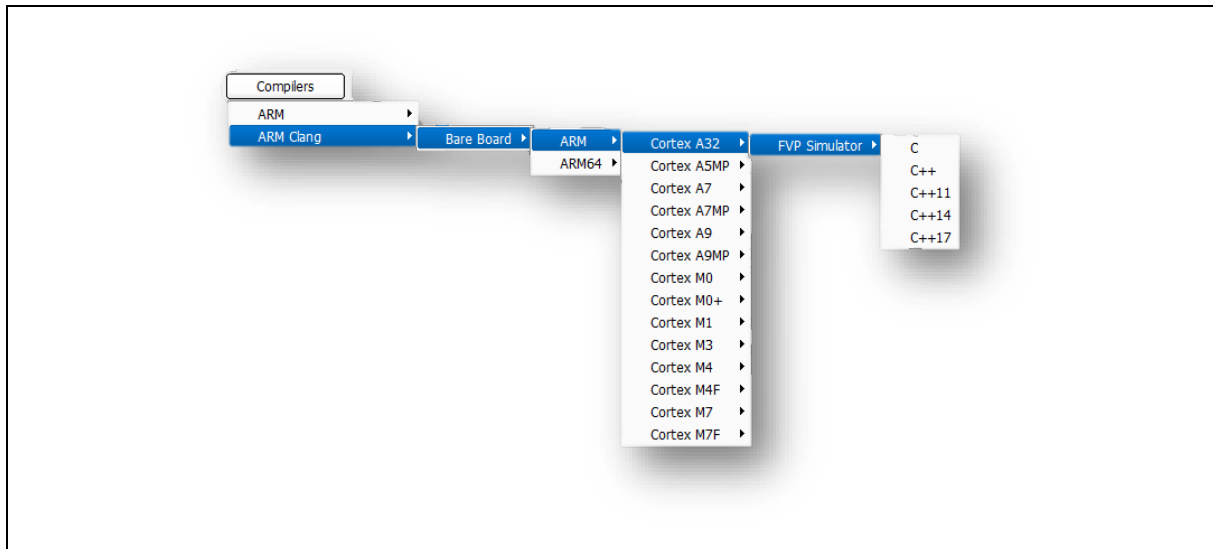


Figure 1 – Selecting the Configuration

Once you have selected the RSP, the configuration will show up in your project and the compiler node will be populated with the proper data as shown in Figure 2.

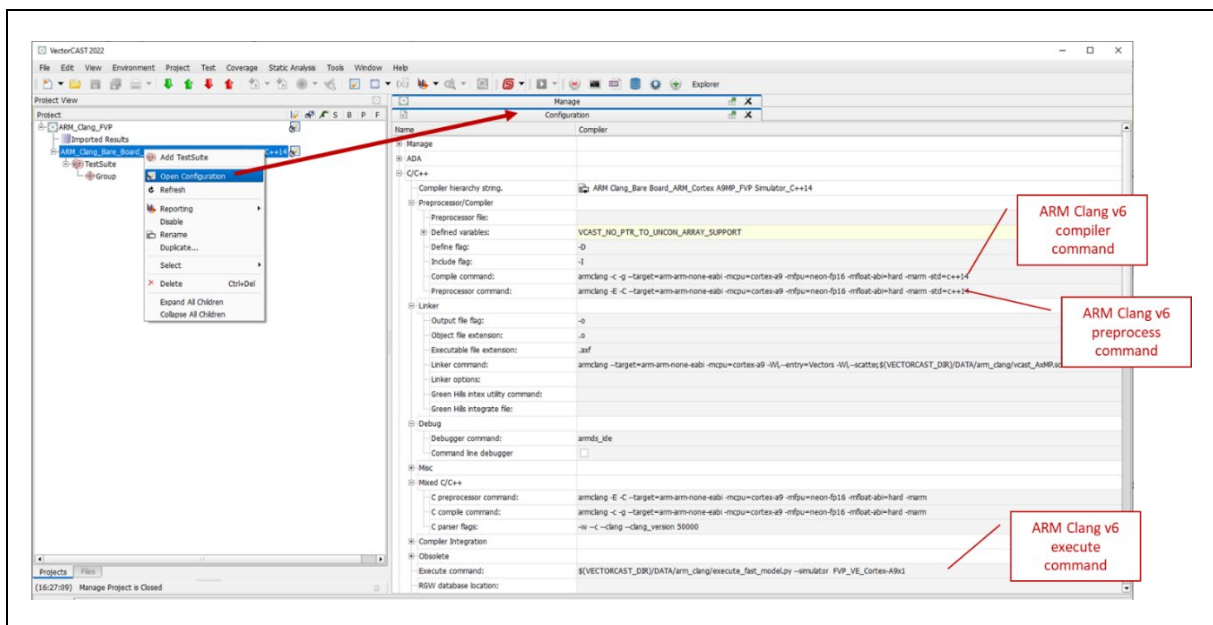


Figure 2 – Viewing Connection Information

2.2 Setup

In order to make the ARM Clang FVP RSPs as generic as possible, the RSP uses environment variables to allow it to be easily adapted to different tool setups. The following environment variables are used in this RSP:

- VCAST_ARM_INSTALL_DIR – Set to the base directory of the ARM DS-5 or ARM Development Studio tools installation
 - set VCAST_ARM_INSTALL_DIR=C:\tools\ARM-DS
- VCAST_ARM_COMPILER_DIR – Set to the compiler base directory. This will include the ARM compiler version and is based on VCAST_ARM_INSTALL_DIR.
 - set VCAST_ARM_COMPILER_DIR=%VCAST_ARM_INSTALL_DIR%/sw/ARMCompiler6.15/
- VCAST_ARM_STARTUP_DIR – Set to the startup directory in your Eclipse project.
 - set VCAST_ARM_STARTUP_DIR=C:\arm-DS\workspace\startup_Armv8-Ax1_AC6
- PATH – The path will need to be updated to include the compiler and IDE directories:
 - set PATH=%PATH%;%VCAST_ARM_COMPILER_DIR%\bin;%VCAST_ARM_INSTALL_DIR%\bin

These environment variables are usually setup in a batch file (Setup_VectorCAST.bat):

```
set VCAST_ARM_INSTALL_DIR=C:\tools\ARM-DS
set VCAST_ARM_COMPILER_DIR=%VCAST_ARM_INSTALL_DIR%/sw/ARMCompiler6.15/
set VCAST_ARM_STARTUP_DIR=C:\arm-DS\workspace\startup_Armv8-Ax1_AC6
set PATH=%PATH%;%VCAST_ARM_COMPILER_DIR%\bin;%VCAST_ARM_INSTALL_DIR%\bin
```

The Setup_VectorCAST.bat script will need to be called before execution of any VectorCAST CLI commands (such as %VECTORCAST_DIR%\manage or %VECTORCAST_DIR%\clicast).

The Setup_VectorCAST.bat script will also need to be called before launching the VectorCAST GUI. We recommend that you create another script called Start_VectorCAST.bat that will call Setup_VectorCAST.bat and then launch the VectorCAST GUI.

Your Start_VectorCAST.bat will look like this:

```
call %~dp0setup_VectorCAST.bat
start %VECTORCAST_DIR%\vcastqt
```

The %~dp0 is a windows batch syntax that will expand the directory path so that Setup_VectorCAST.bat will be called from the same directory that Start_VectorCAST.bat exists in. This allows you to create a shortcut to Start_VectorCAST.bat and if Setup_VectorCAST.bat is in the same directory as Start_VectorCAST.bat then the shortcut will work.

You will need to use Start_VectorCAST.bat to launch VectorCAST or you will not be able to build or execute unit test environments.

2.3 Debugging

Debugging VectorCAST testcases is done using the ARM-DS IDE. When executing a testcase using debug mode, the user will be prompted with a Debug Help screen. Once you select “Start Debugger”, the ARM-DS IDE will open and you can follow the instructions from the Debug Help dialog.

When the ARM-DS IDE launches, select the “Run > Debug Configurations...” menu option:

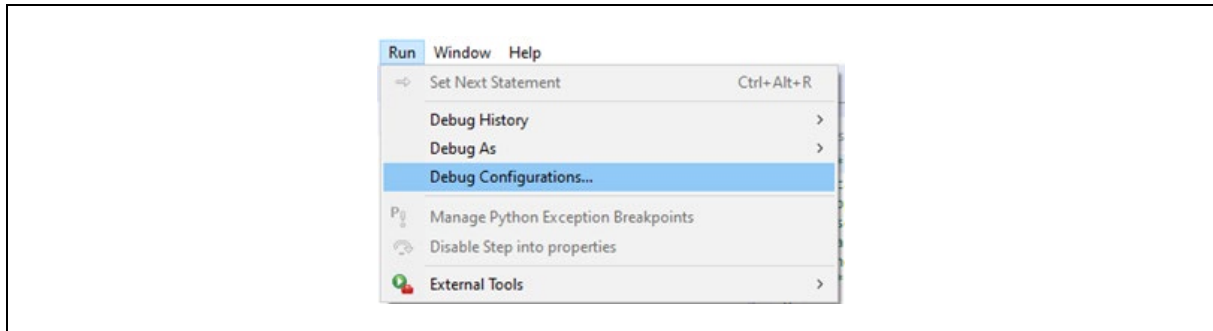


Figure 3 – Selecting the Debug Configurations

Duplicate an existing, working debug launch and change its name to something recognizable later (e.g. VCAST_Debug, etc).

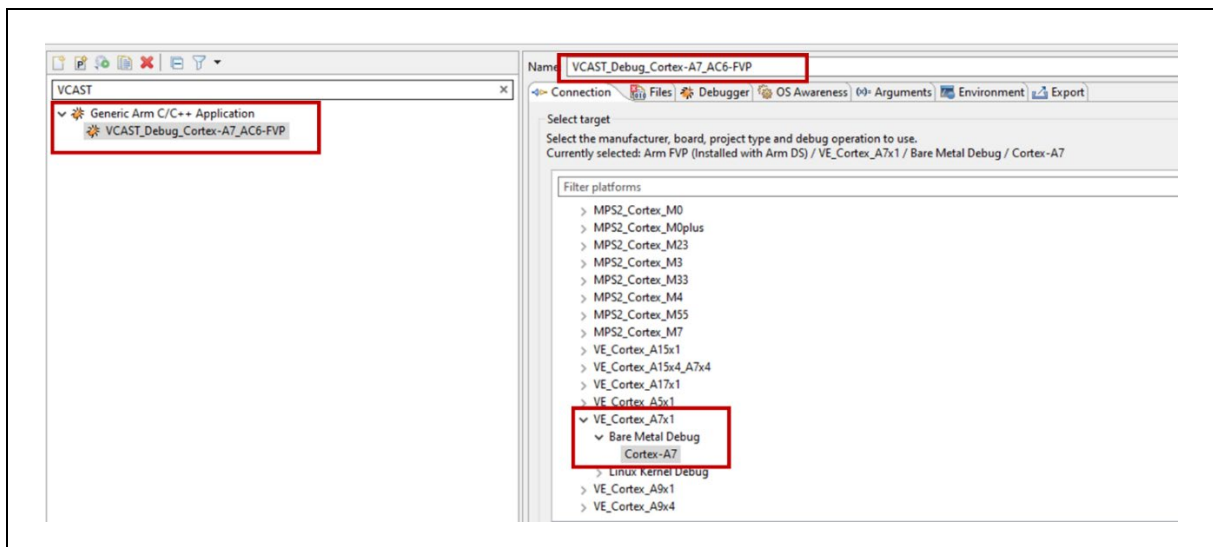


Figure 4 – Creating a New Debug Configuration

Once duplicated, select the Files tab and change the “Application on host to download:” text field to be the harness you want to debug. Select “Apply” to save the changes and then “Debug” to start the FVP simulator and start debugging.

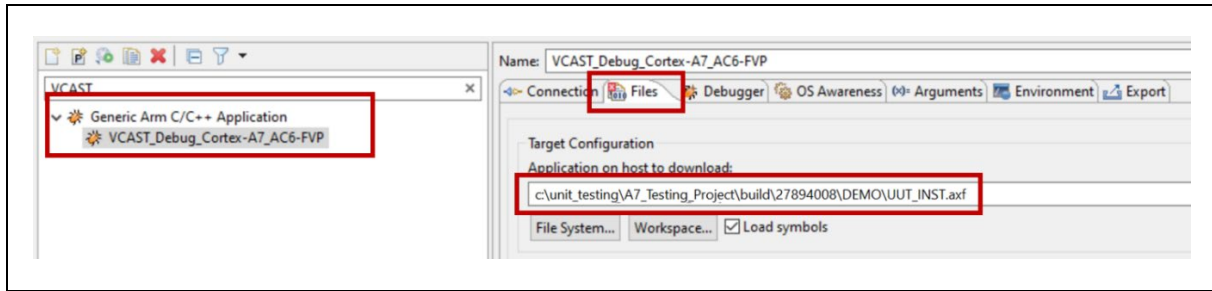


Figure 5 – Selecting the Harness

When you start debugging, execution will stop on main in the VectorCAST test harness. You can set a breakpoint on the function under test. Once you do, you can click go. Execution should stop on the function under test and be able to debug your code.

3 Additional Resources

<https://developer.arm.com/tools-and-software/simulation-models/fixed-virtual-platforms>

4 Trademarks

All mentioned names are either registered or unregistered trademarks of their respective owners.

5 Contacts

For a full list with all Vector locations and addresses worldwide, please visit <http://vector.com/contact/>.