

Application Note: AN00194

Getting Started with VCD Tracing in xTIMEcomposer Studio

This application note shows how to get started with VCD tracing using the xTIMEcomposer studio. It shows you how to run an application in simulation with VCD tracing enabled. It then explains how to analyze the resulting trace using the built-in VCD waveform viewer.

To get started, simply double click on *Getting Started with VCD tracing in xTIMEcomposer Studio* in the Examples view, and click finish in the resulting import dialog. The sample project will then be imported and you will be switched to the XMOS edit perspective. The getting started pdf is then accessible from the *doc/pdf* folder at the top level of the imported project.

Required tools and libraries

- xTIMEcomposer Tools - Version 14.0

Required hardware

This application note is designed to run on the XMOS simulator.

Prerequisites

None

1 Overview

1.1 Introduction

Our comprehensive development tools suite provides everything you need to write, debug and test applications based on xCORE multicore microcontrollers. The full xTIMEcomposer tool set includes unique capabilities such as the xSCOPE logic analyzer and XMOS Timing Analyzer, that let you get the best performance from the deterministic xCORE architecture. With our collection of libraries and examples, it's easy to create and deliver xCORE applications.

xTIMEcomposer features:

- Eclipse graphical environment + plus command line tools
- LLVM C, C++ and xC compilers
- xDEBUG: GDB multicore debugger
- xSIM: Cycle accurate simulator
- xSCOPE: In-circuit instrumentation + real-time logic analyzer
- XTA: Static timing analysis
- Multiple platform support: Windows, OS X, Linux
- Enterprise/Community editions: Tools support for everyone

This application note shows how to get started with binary analysis using the xTIMEcomposer studio. It shows you how to open a binary in the binary analysis tool, browse the resource usage, modify the source code and see the change reflected in the analysis output.

2 Getting Started

Ensure you are in the edit perspective by clicking on the *Edit* perspective button on the left hand side toolbar.

In the *Project Explorer* view you will see the *app_getting_started_with_vcd_tracing* project. Open *src/main.xc* by double clicking.

This application outputs 25 ticks of a clock on port 1A.

2.1 Build the application

To build the application, select 'Project -> Build Project' in the menu, or click the 'Build' button on the toolbar. The output from the compilation process will be visible on the console.

2.2 Create and launch a run configuration

To create/view a VCD trace you first need to create a run configuration. The xTIMEcomposer allows multiple configurations to exist, thus allowing you to store configurations for running on different targets/with different runtime options and arguments.

Right-click on the generated binary in the *Project Explorer* view, and select *Run As -> Run Configurations*. In the resulting dialog, double click on *xCORE Application*, then perform the following operations:

- On the *Main* tab check the *simulator* option.
- On the *Simulator* tab, check *Enable signal tracing*.
- In the *Tile Trace Options* group, click the *Add* button and check the *Ports* option.

Now select the *Run* button to launch the application. On completion, the VCD file *app_getting_started_with_vcd_tracing.vcd* will be written to the top level of the project. The perspective will automatically switch to *VCD Tracing*, and the VCD file will be loaded into the *Signals* view.

In the *Signals View*, expand the *Ports -> XS1_PORT_1A* nodes in the tree, then double click on the *tile[0]_XS1_PORT_1A* node to view the waveform for this signal.

2.3 Waveform Navigation

Click the *Zoom Fit* button in the *Waves* view toolbar to show the signal content for the entire duration of the run. You should be able to see the 25 clocks output by the application, located toward the end of the trace. You can zoom into a desired portion of the trace by right clicking in the *Waves* view, holding and dragging to define a region. Alternatively, zooming can be achieved via the toolbar buttons or the mouse wheel.

Left clicking in the *Waves* view will relocate the red marker bar. The middle of the 3 boxes above shows the time at the marker bar location. The left hand box shows the absolute time of the mouse position and the right hand box displays the time relative to the marker bar.

Hover over the port 1A output clock signal. The mouse pointer will show a hand indicating that click-through to the source is available. Double clicking will bring up the editor highlighting the line that caused the current transition.

3 References

XMOS Tools User Guide

<http://www.xmos.com/published/xtimecomposer-user-guide>

XMOS xCORE Programming Guide

<http://www.xmos.com/published/xmos-programming-guide>

4 Full source code listing

4.1 Source code for main.xc

```
// Copyright (c) 2016, XMOS Ltd, All rights reserved

#include <xs1.h>

port p = XS1_PORT_1A;

int main() {
    int value = 1;
    for (unsigned int i = 0; i < 25; ++i) {
        p <: value;
        value = value == 0 ? 1 : 0;
    }
    return 0;
}
```



Copyright © 2016, All Rights Reserved.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the “Information”) and is providing it to you “AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.