

Application Note: AN00198

Getting Started with Simulator Tracing in xTIMEcomposer Studio

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

To get started, simply double click on *Getting Started with Simulator 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 simulator tracing using the xTIMEcomposer studio. It shows you how to run an application on the simulator with tracing enabled. It then explains how to analyze the resulting trace using the built-in trace view.

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_simulator_tracing* project. Open *src/main.xc* by double clicking.

This application calls a single function: *function* which contains a single loop.

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 simulator instruction 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 select the *simulator* option.
- On the *Simulator* tab, In the *Tracing* group, select *Open in trace view*.

Now select the *Run* button to launch the application. On completion, a of *trace.xstr* file will be written to the top level of the project. The perspective will automatically switch to the *xSIM Tracing* perspective, and the trace file will be loaded in the *xSIM Trace* view.

2.3 Browsing the simulator trace

The xSIM trace provides a textual representation of the instructions executed during the run of a given application. This includes the program counter, function, instruction, operands, etc. See the tools user guide for a complete description of each field in a trace file line.

The trace can be filtered for a choosen *Tile*, *Core* or *Function* via the combo boxes in the *xSIM Trace* toolbar. Choose the *function* entry in the *Function* dropdown. This will remove all the trace lines except those pertaining to the choosen function. Double click on a trace line. This will, if possible, open an editor highlighting the corresponding source level line.

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

int function() {
    int result = 0;
    for (unsigned int i = 0; i < 10; ++i) {
        ++result;
    }
    return result;
}

int main() {
    return function();
}
```



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.