



lib_xassert: Assertions library

Publication Date: 2024/10/15

Document Number: XM-006382-UG v4.3.1

IN THIS DOCUMENT

1	Introduction	2
2	API	2
	2.1 Assertions	2
	2.2 Unreachable	2
	2.3 Fail	2
3	Controlling assertions	3

1 Introduction

This library provides a lightweight and flexible replacement for the standard C header `assert.h`.

The assertions in this library can be enabled/disabled and configured as to how much information they show. This configuration can be per *xassert unit* (i.e. for sets of files).

`lib_xassert` is intended to be used with the [XCommon CMake](#), the XMOS application build and dependency management system.

2 API

To use this module, include `lib_xassert` in the application's `APP_DEPENDENT_MODULES` list and include the `xassert.h` header file.

2.1 Assertions

An assertion can be inserted into code with the `assert` macro e.g.:

```
assert(i < n);
```

Optionally a debug message can be added with the `msg` macro:

```
assert(i < n && msg("i must be less than the array bound"));
```

If assertions are enabled and the expression in the assertion is false then a trap will occur.

2.2 Unreachable

If the logic of a program dictates that certain code cannot be reached, the `unreachable` macro can be used e.g.:

```
switch (message) {
case 0:
    . . .
case 1:
    . . .
default:
    unreachable("message must be 0 or 1");
    break;
}
```

If assertions are enabled then this macro will cause a trap if executed.

2.3 Fail

A failure can be indicated with the `fail` macro e.g.:

```
if (reg_value != 0xA5)
    fail("device not connected properly")
```

A fail will always cause a trap if executed. A failure differs from unreachable in that an unreachable macro should never execute in a correct program whereas a fail could happen in catastrophic circumstances even if the program is correct.

3 Controlling assertions

Assertions can be enabled/disabled via command line options to your application build. The following defines can be set by using the `-D` option to the compiler. For example, the following in your application `CMakeLists.txt` will enable line numbers in assertion messages:

```
set(APP_COMPILER_FLAGS -DXASSERT_ENABLE_LINE_NUMBERS=1)
```

The following defines can be set:

XASSERT_ENABLE_ASSERTIONS

This define can be used to turn assertions on or off (defaults to 1).

XASSERT_ENABLE_DEBUG

This define will cause assertions to print out the failing expression before trapping (defaults to 0). Note that this option could significantly increase the code size of your application.

XASSERT_ENABLE_LINE_NUMBERS

This define will cause assertions to print the file and line number of the assertion before trapping. Note that this option could significantly increase the code size of your application.

If `XASSERT_UNIT` is defined when `xassert.h` is included then all the assertions in that file belong to that unit. Assertions can then be controlled per debug unit. The mechanism is similar to that used in `module_logging`.

XASSERT_ENABLE_ASSERTIONS_[xassert unit]

Enable asserts for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_ASSERTIONS` for that debug unit.

XASSERT_ENABLE_DEBUG_[xassert unit]

Enable debug messages for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_DEBUG` for that debug unit.

XASSERT_DISABLE_ASSERTIONS_[xassert unit]

Disable asserts for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_ASSERTIONS` for that debug unit.

XASSERT_DISABLE_DEBUG_[xassert unit]

Disable debug messages for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_DEBUG` for that debug unit.



Copyright © 2024, 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.

XMOS, xCore, xcore.ai, and the XMOS logo are registered trademarks of XMOS Ltd in the United Kingdom and other countries and may not be used without written permission. Company and product names mentioned in this document are the trademarks or registered trademarks of their respective owners.

