Calling between C/C++ and XC

IN THIS DOCUMENT

In certain cases, it is possible to pass arguments of one type in XC to function parameters that have different types in C/C++, and vice versa.

To help simplify the task of declaring common functions between C/C++ and XC, the system header file `xccompat.h` contains various type definitions and macro defines. See the header file for documentation.

1 Passing arguments from XC to C/C++

A function defined in C/C++ with a parameter of type `unsigned int` can be declared in XC as taking a parameter of type `port`, `chanend` or `timer`.

A function defined in C/C++ with a parameter of type “pointer to `T`” can be declared in XC as taking a parameter of type “reference to `T`” or “nullable reference to `T`.”

A function defined in C/C++ with a parameter of type “pointer to `T`” can be declared in XC as taking a parameter of type “array of `T`.”

2 Passing arguments from C/C++ to XC

A function defined in XC with a parameter of type `port`, `chanend` or `timer` can be declared in C/C++ as taking a parameter of type `unsigned int`.

A function defined in XC with a parameter of type “reference to `T`” or “nullable reference to `T`” can be declared in C/C++ as taking a parameter of type “pointer to `T`.”

A function defined in XC with a parameter of type “array of `T`” can be declared in C/C++ as taking a parameter of type “pointer to type `T`.” In this case, the xCORE linker inserts code to add an implicit array bound parameter equal to the maximum value of the `unsigned int` type.

Publication Date: 2013/11/11 REV C

Xmos Ltd. is the owner or licensee of the information in this document 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.