

How to examine the contents of memory

IN THIS DOCUMENT

- ▶ From within xTIMEcomposer Studio
 - ▶ From the command line
-

version 1.1.0

scope Example. This code is provided as example code for a user to base their code on.

description How to examine the contents of memory

boards Unless otherwise specified, this example runs on the SliceKIT Core Board, but can easily be run on any XMOS device by using a different XN file.

XGDB can be used to examine the contents of memory at a given point in time. For example, compile the following code ensuring that debug is enabled (-g):

```
int global_variable[5] = {0, 1, 2, 3, 4};

int f(int index) {
    return global_variable[index];
}

int main() {
    f(2);
    return 0;
}
```

1 From within xTIMEcomposer Studio

Create a new debug configuration via *Run->debug Configurations->xCORE Applications*. Set a breakpoint at the start of *f* then start debugging. Execution will now break when *f* is reached. The memory contents occupied by the global variable *global_variable* can be seen in the *Memory* view. Click on the *add memory monitor* button on the view toolbar. In the resulting dialog, input '*global_variable*' then press *ok*. This will show the address of *global_variable* and the contents of this memory location.

2 From the command line

On the command line, memory contents can be examined using the `x` (examine) command. For example, start XGDB, connect to the simulator and set a breakpoint on `f`. When run, execution will break at the start of `f`. You can now display the contents of the memory occupied by `global_variable` using the `x` command as follows:

```
> xgdb a.xe
...etc...
(gdb) connect -s
0xfffffc04e in ?? ()
(gdb) break f
Breakpoint 1 at 0x100b2: file examining_memory.xc, line 11.
(gdb) run
...etc...
Breakpoint 1, f (index=2) at examining_memory.xc:11
11      return global_variable[index];
(gdb) x global_variable
0x10274 <global_variable>: 0x00000000
```

Note: The print command accepts an argument specifying the desired format. For example, `x/d` will display the memory contents as decimal instead of the default of hex.

Also, to see the content of an offset other than 0 in the `global_variable` array, (for example, `index = 1`), the following command can be used:

```
x &global_variable[1]
```



Copyright © 2013, 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.