Application Note: AN10071

How to set a breakpoint

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to set a breakpoint.

Required tools and libraries

This application note is based on the following components:

- xTIMEcomposer Tools - Version 14.0.0

Required hardware

Programming how-tos are generally not specific to any particular hardware and can usually run on all XMOS devices. See the contents of the note for full details.

1 How to set a breakpoint

Instruction breakpoints are used to allow XGDB to halt the execution of a the program at a user defined position in the code. For example, compile the following code ensuring that debug is enabled (-g):

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

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

2 To set a breakpoint from xTIMEcomposer Studio

Create a new debug configuration via *Run->debug Configurations->xCORE Applications*. Double click in the left-hand margin of the *return 0* line in function *f*. This will create a new instruction breakpoint, thus when run, execution will pause when this line is reached.

3 To set a breakpoint from the command line

Breakpoints are set using the *breakpoint* command. This takes the breakpoint location as an argument, which can be either a function name or a file/linenumber combination. For example, the XGDB command: *breakpoint setting_a_breakpoint.xc:2* will install a breakpoint on the first line of the function *f* in the above code.

Using the above code, an example XGDB session using breakpoints could be as follows:

```
> xgdb a.xe
...etc...
(gdb) connect -s
0xffffc04e in ?? ()
(gdb) breakpoint f
Breakpoint 1 at 0x100ac
(gdb) run
...etc...
Breakpoint 1, 0x000100ac in f ()
(gdb) c
```