Application Note: AN10064

How to run an executable using XGDB

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to run an executable using XGDB.

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 run an executable using XGDB

For example, compile the following *Hello world* code ensuring that debug is enabled (-g):

```
#include <print.h>

int main() {
  printstr("Hello World!\n");
  return 0;
}
```

2 To run using XGDB from xTIMEcomposer Studio

Create a new debug configuration via *Run->Debug Configurations->xCORE Applications*. You must then choose the target, for example, either one of the currently connected development boards or the simulator. Clicking on *Debug* will start XGDB, connect to the selected target and start executing the program.

3 To run using XGDB from the command line

Start XGDB passing the resulting executable as an argument:

```
xgdb a.xe
```

Next you will need to choose the target. The XGDB *connect* command with no arguments will connect a development board, if there is only a single one connected. If there is more than one connected, then the chosen target must be specified. Alternatively, use *connect -s* to target the simulator. The *run* command will then start execution on the chosen target. For example, executing the above code on a connected development board will result in the following session:

```
> xgdb a.xe
GNU gdb (XGDB) 12.1.0 (build 7669)

...etc...

(gdb) connect
0x00010000 in _start ()
(gdb) run
Loading image to XCore 0
Loading section .text, size 0x1ec lma 0x10000

...etc...

Start address 0x10000, load size 764
Transfer rate: 149 KB/sec, 63 bytes/write.
Hello World!

Program exited normally.
(gdb)
```