Application Note: AN10012

How to examine the register state in a core

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to examine the register state in a core.

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 examine the register state in a core

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 main() {
  return 0;
}
```

2 From within xTIMEcomposer Studio

Create a new debug configuration via *Run->debug Configurations->xCORE Applications*. Set a breakpoint at the start of *main* then start debugging. Execution will now break when *main* is reached. The contents of the registers, and some core specific internal registers, are visible via the *Registers* view.

3 From the command line

On the command line, register state can examined using the *print* command. For example, start XGDB, connect to the simulator and set a breakpoint on *main*. When run, execution will break at the start of *main*. You can now display the register contents using the *print* command as follows:

```
> xgdb a.xe
...etc...
(gdb) connect -s
0xffffc04e in ?? ()
(gdb) b main
Breakpoint 1 at 0x100b0: file examining_core_state.xc, line 9.
(gdb) r
...etc...
Breakpoint 1, main () at examining_core_state.xc:9
9     return 0;
(gdb) print /x $r0
$1 = 0x0
(gdb) print /x $r1
$2 = 0x10240
(gdb) print /x $r2
$3 = 0x1f
(gdb) print /x $lr
$4 = 0x10062
(gdb) print /x $pc
$5 = 0x100b0
...etc...
```

Note: If required, the *info registers* command can be used to display the complete state of all of the registers.