Application Note: AN10029

How to use interfaces to connect to multiple tasks

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use interfaces to connect to multiple tasks.

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 use interfaces to connect to multiple tasks

The following example shows three tasks running in parallel and communicating. The task3 function performs transactions with either task1 or task2.

```
void task1(interface my_interface client c) {
  c.fA(5, 10);
}

void task2(interface my_interface client c) {
  c.fA(20, 25);
}

void task3(interface my_interface server c,
           interface my_interface server d) {
  for (int i=0; i < 2; i++) {
    // wait for either fA or fB over connection c.
    select {
    case c.fA(int x, int y):
      printf("Received fA from interface end c: %d, %d\n", x, y);
      break;
    case d.fA(int x, int y):
      printf("Received fA from interface end d: %d, %d\n", x, y);
      break;
    }
  }
}

int main(void) {
  interface my_interface c;
  interface my_interface d;
  par {
    task1(c);
    task2(d);
    task3(c, d);
  }
  return 0;
}
```

You can also connect to multiple tasks over the same interfaces using interface arrays.