How to use interfaces to connect to multiple tasks

`version` 1.1.1

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

`description` How to use interfaces to connect to multiple tasks

`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.

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.

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.