Application Note: AN10080

How to use transactions over channels

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use transactions over channels.

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.

Copyright 2016 XMOS Ltd.

1 How to use transactions over channels

By default, channel I/O is synchronous. This means that for every byte/word sent over the channel there is some handshaking taking place and also that the task doing the output is blocked until the input task at the other end of the channel has received the data. The time taken performing the synchronization along with any time spent blocked can result in reduced performance. Transactions provide a possible solution to this issue. They allow 2 cores to engage in a *transaction*, in which a sequence of matching outputs and inputs are communicated over a channel ansynchronously, with the entire transaction being synchronised.

To use a transaction to send data between cores you first need to declare a channel e.g.

```
chan c;
```

You can then pass each end of the channel to each logical core.

```
par {
  f1(c);
  f2(c);
}
```

This function outputs the values of 1, 2 and 3 on the channel 'c' using a master transaction.

```
void f1(chanend c) {
  master {
    c <: 1;
    c <: 2;
    c <: 3;
  }
}
```

This function inputs the values of 1, 2 and 3 from the channel 'c' using a slave transaction.

```
void f2(chanend c) {
  int x;
  slave {
    c :> x;
    printintln(x);
    c :> x;
    printintln(x);
    c :> x;
    printintln(x);
  }
}
```

Copyright © 2016, All Rights Reserved.

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.