Application Note: AN10077

How to use an ordered select statement

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use an ordered select statement.

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 an ordered select statement

A select statement waits for one of a set of inputs to become ready, performs the selected input and then executes a corresponding body of code. Each input is proceeded by the keyword case and its body must be terminated with a break or return statement. Case statements are not permitted to contain output operations. Selects can also be ordered so that priority is given based on the order of the case statements. This allows code to take a decision over which to execute first when events occur at the same time.

In this example the select statement is used to wait for either an input on chnlend_a or an input on chnlend_b. When an input is received from either channel the value of the input is printed. Using `#pragma ordered` the order in which events are processed is based on the order of case statements in the select.

```
#pragma ordered
    select
    {
      case chnlend_a :>  chnl_input_a :
        printstr("Channel Input A Received ");
        printintln(chnl_input_a);
        break;
      case chnlend_b :> chnl_input_b :
        printstr("Channel Input B Received ");
        printintln(chnl_input_b);
        break;
    }
```