Application Note: AN10068

How to use the select statement with a guard

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

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 the select statement with a guard

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. A guard can be used within a select case to enable or disable the input operation. In this example a guard is used to control input between chnlend_a and chnlend_b. The initial state is set so that input from chnlend_a is enabled and input from chnlend_b is disabled.

```
int is_a = 1;
int is_b = 0;
```

The select body receives 10 inputs on chnlend_a at which point chnlend_a is disabled and chnlend_b is enabled to perform the same input sequence. The program continues to alternate receiving from the two inputs.

```
select
{
  case is_a => chnlend_a :> int chnl_input_a :
    printstr("Channel Input A Received ");
    printintln(chnl_input_a);
    if (++counter == 10)
    {
      is_a = 0;
      is_b = 1;
      counter = 0;
    }
    break;
  case is_b => chnlend_b :> int chnl_input_b :
    printstr("Channel Input B Received ");
    printintln(chnl_input_b);
    if (++counter == 10)
    {
      is_a = 1;
      is_b = 0;
      counter = 0;
    }
    break;
}
```