
Application Note: AN10046

How to use buffering for port input

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use buffering for port input.

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 buffering for port input

You can set a port into *buffered input* port. This tells the XMOS architecture to set the port into a mode that de-serializes the data. Any inputs will be places into a small fifo (the *transfer register*) and reading the port will read out the fifo when it is full. The size of the transfer register is user specified and does not need to be the same as the port width.

This mode can improve performance since it allows inputs to be sampled off the pins independently to the instructions of the program.

The declaration

```
in buffered port:32 in_port = XS1_PORT_1A;
```

declares a buffered input port named `in_port`, which refers to the 1-bit port identifier `XS1_PORT_1A`. Note that the size of the port FIFO is specified as part of the port declaration using the colon. The allowable sizes for this FIFO on XS1 devices are: 1, 4, 8 and 32. Input from the port is performed as normal.

```
in_port :> data;
```

At most one value is sampled by the port and inserted to the FIFO per period of its clock. If the FIFO is full, the oldest value is dropped to make room for the most recently sampled value.