

Application Note: AN10047

How to use buffering for port output

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

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 output

You can set a port into *buffered output* port. This tells the Xmos architecture to set the port into a mode that serializes the data. Any outputs will be placed into a small fifo (the *transfer register*) and writing to the port will fill this fifo. The fifo will then be output independently at the clock rate of the port. 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 outputs to be driven to the pins independently to the instructions of the program.

The declaration

```
out buffered port:32 out_port = XS1_PORT_1A;
```

declares a buffered output port named `out_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 buffered port declaration using the colon. The allowable sizes for this FIFO on XS1 devices are: 1, 4, 8 and 32. Output to the port is performed as normal.

```
out_port <: counter;
```

The port buffers the data so that the processor can continue executing subsequent instructions. On each falling edge of the clock, the port takes the next bit of data from its buffer and drives it on its pins.