Application Note: AN10056

How to serialize output data to a port

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

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 serialize output data to a port

A clocked port can serialize data, reducing the number of instructions required to perform an output. This example outputs a 32-bit value onto 8 pins, using a clock to determine for how long each 8-bit value is driven.

The following declares the port outP to drive 8 pins from a 32-bit shift register. The type port:32 specifies the number of bits that are transferred in each output operation (the transfer width). The initialization XS1_PORT_8A specifies the number of physical pins connected to the port (the port width).

```
out buffered port:32 outP    = XS1_PORT_8A;
```

By offloading the serialization to the port, the processor has only to output once every 4 clock periods. On each falling edge of the clock, the least significant 8 bits of the shift register are driven on the pins; the shift register is then right-shifted by 8 bits.

```
unsigned int x = 0xAA00FFFF;
configure_clock_src(clk, inClock);
configure_out_port(outP, clk, 0);
start_clock(clk);
while (1) {
  outP <: x;
  x = x + 1;
}
```

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.