How to serialize output data to a port

version	1.1.0
scope	Example. This code is provided as example code for a user to base their code on.
description	How to serialize output data to a port
boards	Unless otherwise specified, this example runs on the SliceKIT Core Board, but can easily be run on any XMOS device by using a different XN file.

A clocked port can serialise 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 initialisation 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 serialisation 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.

```
int x = 0xAA00FFFF;
configure_clock_src(clk, inClock);
configure_out_port(outP, clk, 0);
start_clock(clk);

while (1) {
  outP <: x;
  x = f(x);
}
```

Publication Date: 2013/7/18 REV A