

Application Note: AN10055

# How to de-serialize input data from a port

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

A port can deserialize data, reducing the number of instructions required to input data. The example below performs a 4-to-8 bit conversion on an input port, controlled by a 25MHz clock.

The program declares `inP` to be a 4-bit wide port with an 8-bit transfer width, meaning that two 4-bit values can be sampled by the port before they must be input by the processor.

```
in buffered port:8 inP = XS1_PORT_4A;
```

Data is sampled on the rising edges of the clock and, when de-serializing, the least significant nibble is read first. Subsequent data is added to the least significant part of the input word with earlier inputs shifted left. The sampled data is available in the port's buffer for input for two clock periods.

```
configure_clock_rate(clk25, 100, 4);
configure_in_port(inP, clk25);
start_clock(clk25);
while (1) {
    int x;
    inP := x;
    f(x);
}
```