

# piso.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/04/15

---

## INFORMATION

---

### Brief

---

PISO (parallel in serial out) The idea is to keep this core simple, and let the control logic be handled outside of this core.

### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## piso

---

```
module piso #(
    parameter
    BUS_WIDTH
    =
    1
) ( input clk, input rstn, input ena, input rev, input load, input [BUS_WID
```

parallel in serial out

### Parameters

**BUS\_WIDTH** width of the parallel data input in bytes.

## Ports

<b>clk</b>	global clock for the core.
<b>rstn</b>	negative synchronus reset to clk.
<b>ena</b>	enable for core, use to change output rate. Enable serial shift output.
<b>rev</b>	reverse, 0 is MSb first out, 1 is LSB first out.
<b>load</b>	load parallel data into core. Reset for next data message to send. This can be done at any time.
<b>pdata</b>	parallel data input, registered at load only.
<b>sdata</b>	serialized data output.
<b>dcount</b>	Number of bits to shift out. When the count hits zero, the parallel data register is empty and last bit is output on sdata.