

OpenCPI

GPO Component Data Sheet

OpenCPI Release: v2.4.8

Revision History

Revision	Description of Change	Date
v1.6	Initial Release	11/2019

Summary - GPO

Name	gpo
Worker Type	
OpenCPI Release	v2.4.8
Last Update	11/2019
Component Library	ocpi.assets.devices
Workers	gpo.hdl
Tested Platforms	matchstiq_z1, xsim

Functionality

The GPO device worker controls the GPIO pins present on an embedded platform. It is configured for general purpose output.

Worker Implementation Details

The GPO device worker can control the GPIO pins via the property `mask_data`, and its (optional) input port.

The (optional) input port uses a protocol that has one opcode and contains data and a mask. See `gpio_protocol.xml` in `ocpi.core/specs/gpio_protocol.xml` for further details on the protocol.

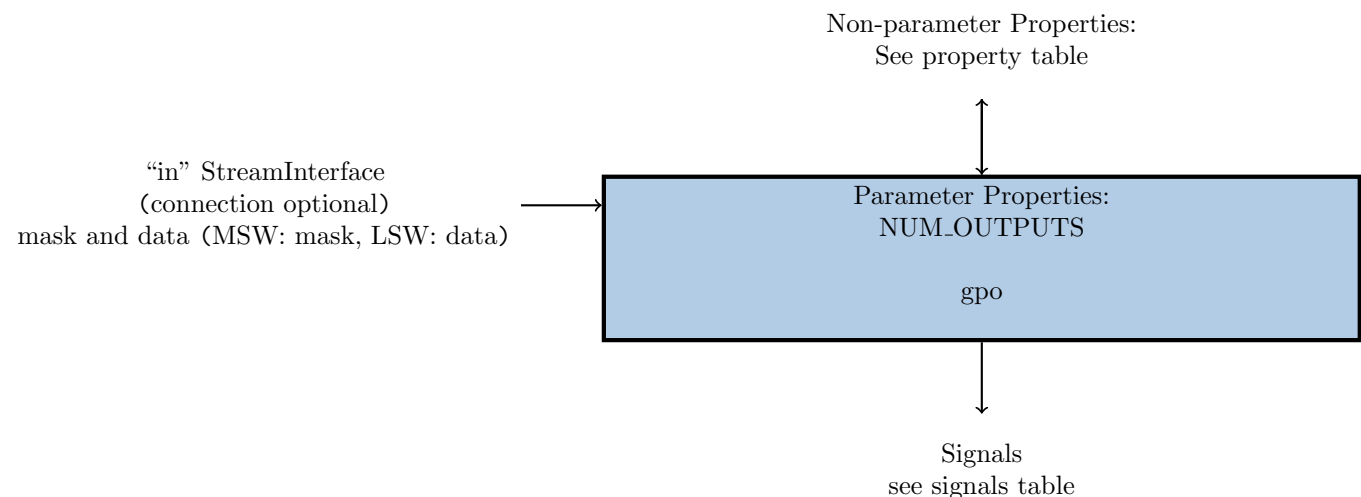
The port and the `mask_data` property set the values of pins using a data and mask. In order for a GPIO pin to take on the value of a data bit, the corresponding mask bit has to be set to 1. Pins that don't have a mask bit set to 1 are left untouched. The MSW of `mask_data` and the port data, must be the mask and the LSW must be the data.

The `mask_enable` property provides a way for enabling or disabling the use of the property "`mask_data`"; or the input port data and mask; if it desired to enable or disable any one of the ways of controlling the GPIO pins. By default both are enabled.

There also is a `pin_state` property that stores the current GPIO pin state.

Block Diagrams

Top level



Source Dependencies

gpo.hdl

- assets/hdl/devices/gpo.hdl/gpo.vhd
- assets/primitives/util_prims/set_clr/src/set_clr.vhd

gpo_em.hdl

- assets/hdl/devices/gpo_em.hdl/gpo_em.vhd

Worker Properties

Name	Type	Default	SequenceLength	ArrayLength	ArrayDimensions	Accessibility	Usage
NUM_OUTPUTS	uChar	1	-	-	-	Parameter	Number of GP Outputs. The max number of GP Outputs supported is 16.
mask_enable	uChar	0x03	-	-	-	Writable	Bitfield that allows enabling or disabling the use of the property 'mask_data'; or input port 'data' and 'mask'. Bit 0 is the property and bit 1 is the input port. If a bit is a 1 then the corresponding way of controlling the GPIO pin is enabled.
mask_data	uLong	0	-	-	-	Writable	Bitfield containing the data to write the GPIO pins and the mask. The mask allows setting GPIO pins on or off in a single operation and only affects the corresponding data pin; mask bit 0 will allow for data bit 0 to be set. The MSW must be the mask and LSW must be the data. For example if mask_data= 0x00010003, the mask = 0x0001 and data = 0x0003. This would set gpo-pin[0] to 1 and leave all other pins untouched.
pin_state	uShort	-	-	-	-	Volatile	Current state of the GPIO pins.

Component Ports

Name	Protocol	Producer	Optional	Usage
in	gpio_protocol	false	true	Data to write GPIO pins

Worker Interfaces

gpo.hdl

Type	Name	DataWidth (b)	Advanced	Usage
StreamInterface	in	32	-	Data to write GPIO pins

Signals

Name	Type	Width (b)	Description
gpo_pin	out	NUM_OUTPUTS	Output to GPIO pins

Control Timing and Signals

The GPO device worker uses the clock from the Control Plane and standard Control Plane signals.

Worker Configuration Parameters

gpo.hdl

Performance and Resource Utilization

gpo.hdl

Test and Verification

The gpo device worker unit test has two test cases and utilizes the gpo_em device emulator. Case 1 tests controlling the GPIO pins via the `mask_data` property and case 2 tests controlling them via the input port.

For the `mask_data` property and the data port, the tests that are done are: setting data bits high but not setting masks, setting data bits high and setting the appropriate mask bits high, and clearing data bits.

The gpo_em device emulator detects if the state of the GPIO pins have changed and sends the current state of the GPIO pins as message if there has been change.

The `generate.py` script generates the input data and generates a `golden.dat` file. For case 1 the input file is a 0 byte file since the input port data is ignored in this case.

The `verify.py` script checks that the output data matches the expected output data contained in the `golden.dat` files.