

OpenCPI

GPI Component Data Sheet

OpenCPI Release: v2.4.8

Revision History

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

Summary - GPI

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

Functionality

The GPI device worker reads the GPIO pins present on an embedded platform. It is configured for general purpose input.

Worker Implementation Details

The GPI device worker reads GPIO pins and stores the values in the property `mask_data` and send the values to its (optional) output port.

It's (optional) output port uses a protocol that has one opcode and contains data and a mask. It sends data regardless of whether or not the worker downstream is ready, so it is something that cannot be piped directly to software. See `gpio_protocol.xml` in `ocpi.core/specs/gpio_protocol.xml` for further details on the protocol.

For the `mask_data` property and the output port data, the MSW is a mask and the LSW are the values read from the GPIO pins.

The mask allows knowledge of which GPIO pins changed since the previous read cycle. The mask is the current data XOR the previous data.

There are parameter properties that enable/disable build-time inclusion of 3 different circuits.

The `USE_DEBOUNCE` property enables/disables build-time inclusion of a debounce circuit which debounces input from mechanical switches. There are the `CLK_RATE_HZ` and `DEBOUNCE_TIME_SEC` properties used by the GPI device worker to calculate the width of the debounce circuit's counter. The counter width is calculated by using the formula: $\text{ceil}(\log_2(\text{CLK_RATE_HZ} * \text{DEBOUNCE_TIME_SEC}))$.

It takes $2^{\text{counterwidth}} + 3$ clock cycles until the input is ready at the debounce circuit's output. If the debounce circuit is not enabled, metastability flip flops are used instead. These metastability flip flops have a latency of two clock cycles.

The `EDGE_MODE` property enables/disables build-time inclusion of an edge detector circuit. The edge detector detects rising and falling edge of an input on the rising edge of the clock. There is a `RISING` property that is used to select between using the rising edge output of the edge detector or the falling edge. If `RISING` is true the rising edge is used otherwise the falling edge is used. The outputs of the edge detector have a latency of one clock cycle. The output of either the debounce circuit or metastability flip flops is fed to the edge detector. If the edge detector is not used, the output of debounce circuit or metastability flip flops is used instead.

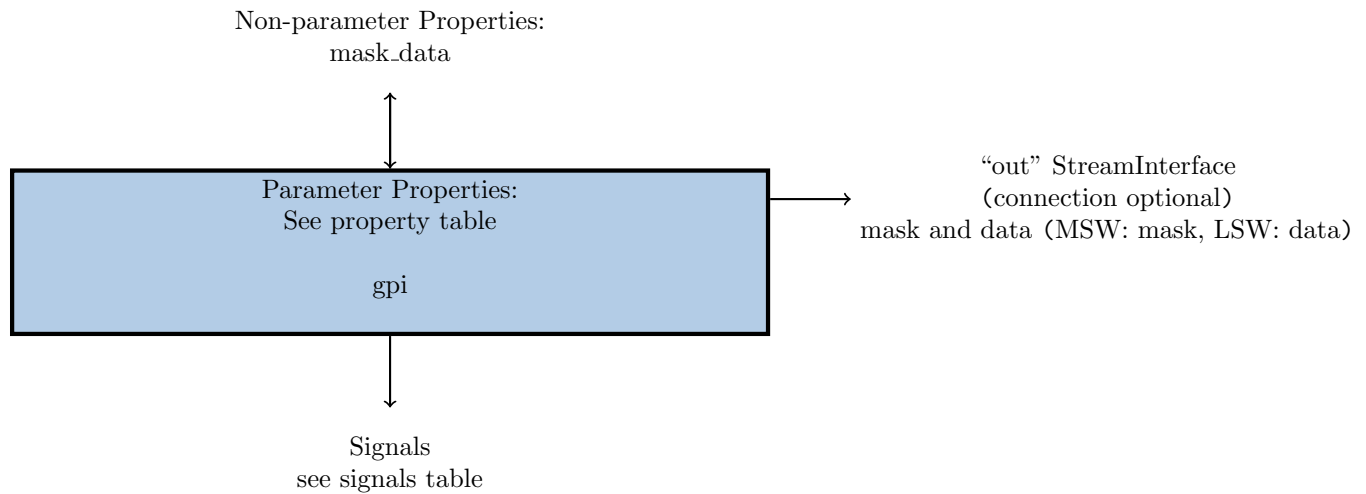
The `USE_TOGGLE` property enables/disables build-time inclusion of a toggle circuit. The toggle circuit has a register that stores the result of XOR'ing the output of the edge detector, debounce circuit, or metastability flip flops with the current value of the register. If toggle circuit is not used the output of the edge detector, debounce circuit, or metastability flops is used instead.

The output of these chain of circuits is fed to device worker's output port.

There is a parameter property called `EVENT_MODE`. If `EVENT_MODE` is set to true the output port's valid signal is driven high only when there is a change in the data to be sent to the output port. If it is set to false, the output port valid signal is always high and the data is valid every clock cycle.

Block Diagrams

Top level



Source Dependencies

`gpi.hdl`

- `assets/hdl/devices/gpi.hdl/gpi.vhd`
- `assets/primitives/misc_prims/edge_detector/src/edge_detector.vhd`
- `assets/primitives/misc_prims/debounce/src/debounce.vhd`

`gpi_em.hdl`

- `assets/hdl/devices/gpi_em.hdl/gpi_em.vhd`
- `assets/primitives/util_prims/set_clr/src/set_clr.vhd`

Worker Properties

Name	Type	Default	SequenceLength	ArrayLength	ArrayDimensions	Accessibility	Usage
NUM_INPUTS	uChar	1	-	-	-	Parameter	Number of GP Inputs. The max number of GP Inputs supported is 16.
USE_DEBOUNCE	bool	false	-	-	-	Parameter	Enable/Disable build-time inclusion of debounce circuit.
CLK_RATE_HZ	double	100e6	-	-	-	Parameter	The clock rate of the clock feeding the debounce circuit in Hz.
DEBOUNCE_TIME_PSEC	double	1e10	-	-	-	Parameter	The desired debounce time for the debounce circuit in picoseconds.
EDGE_MODE	bool	false	-	-	-	Parameter	Enable/Disable build-time inclusion of edge detector circuit.
RISING	bool	true	-	-	-	Parameter	True - Selects the edge detector's rising edge output. False - Selects the edge detector's falling edge output.
USE_TOGGLE	bool	false	-	-	-	Parameter	Enable/Disable build-time inclusion of toggle circuit.
EVENT_MODE	bool	false	-	-	-	Parameter	True - Output port data valid only when there is change in the data to be sent the output port. False - Output port data valid every clock cycle.
mask_data	uLong	0	-	-	-	Volatile	Bitfield containing the data read from the GPIO pins and the mask. The mask allows knowledge of which GPIO pins changed since the previous read cycle. The MSW must be the mask and LSW must be the data. The mask is the current data XOR the previous data.

Component Ports

Name	Protocol	Producer	Optional	Usage
out	gpio_protocol	true	true	Mask and the data read from GPIO pins

Worker Interfaces

gpi.hdl

Type	Name	DataWidth (b)	Advanced	Usage
StreamInterface	out	32	-	Mask and the data read from GPIO pins

Signals

Name	Type	Width (b)	Description
gpi_pin	out	NUM_INPUTS	Input from GPIO pins

Control Timing and Signals

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

Worker Configuration Parameters

gpi.hdl

Performance and Resource Utilization

gpi.hdl

Test and Verification

The `gpi` device worker unit test has five test cases and utilizes the `gpi_em` device emulator. The first case tests the `gpi` device worker by toggling GPIO pins on and then off with the debounce circuit, edge detector circuit, and toggle circuit disabled. Test case two toggles the pins on and off with only the debounce circuit enabled. The third test case toggles the pins on and off with only the edge detector circuit enabled. Test case four toggles the pins on, off, on, and then off with only the toggle circuit enabled. The fifth test case tests toggling the GPIO pins on, off, and then on with the debounce circuit, edge detector circuit, and toggle circuit enabled.

The `gpi_em` device emulator uses the data received from a data source to write the GPIO pins.

The `generate.py` script generates the input data for the `gpi_em` device emulator and the `verify.py` script checks the output file for expected length and data contents.