

GPIO Module Specification

Introduction

The `gpio` module is designed to interface with general-purpose input/output (GPIO) pins, providing a mechanism to read from and write to these pins through a simple bus interface. The module adheres to a basic read/write protocol, allowing for the configuration and monitoring of GPIO states. It is suitable for applications requiring control and status monitoring of digital I/O pins.

Architecture

The `gpio` module is structured as a single top-level module with the following key components:

- **Clock and Reset:** The module operates synchronously with the `sb_clk` clock signal and is reset asynchronously with the `sb_rst_n` signal.
- **Read/Write Channels:** The module supports separate channels for reading and writing data, each with its own set of control signals.
- **GPIO Data Registers:** The module maintains an output data register (`gpio_odr`) to store the state of the GPIO output pins.
- **Data Flow:** The module processes write requests to update the GPIO output register and read requests to provide the current state of the GPIO input or output pins.

Interface

Signal	Width	In/Out	Description
<code>sb_clk</code>	1	In	System clock signal.
<code>sb_rst_n</code>	1	In	Active-low asynchronous reset signal.
<code>sb_arvalid</code>	1	In	Read address valid signal.
<code>sb_arready</code>	1	Out	Read address ready signal.
<code>sb_araddr</code>	32	In	Read address bus.
<code>sb_rvalid</code>	1	Out	Read data valid signal.
<code>sb_rready</code>	1	In	Read data ready signal.
<code>sb_rdata</code>	32	Out	Read data bus.
<code>sb_wvalid</code>	1	In	Write data valid signal.
<code>sb_wready</code>	1	Out	Write data ready signal.
<code>sb_waddr</code>	32	In	Write address bus.
<code>sb_wdata</code>	32	In	Write data bus.
<code>sb_wstrb</code>	4	In	Write strobe signal.
<code>sb_bvalid</code>	1	Out	Write response valid signal.
<code>sb_bready</code>	1	In	Write response ready signal.
<code>sb_bresp</code>	1	Out	Write response signal (always 0).
<code>gpio_i</code>	16	In	GPIO input pins.
<code>gpio_o</code>	16	Out	GPIO output pins.

Timing

- **Latency:** The module processes read and write operations with a latency of one clock cycle. The `sb_rvalid` and `sb_bvalid` signals indicate the validity of read and write responses, respectively.
- **Signal Behavior:** All operations are synchronous to the rising edge of the `sb_clk` signal. The module resets asynchronously with the `sb_rst_n` signal.

Usage

To use the `gpio` module: 1. **Initialization:** Ensure the `sb_rst_n` signal is asserted low to reset the module, then deassert it to begin normal operation. 2. **Write Operation:** - Assert `sb_wvalid` and provide the address on `sb_waddr` and data on `sb_wdata`. - The module will update the `gpio_odr` register if the address is valid. - Monitor `sb_bvalid` to confirm the write operation completion. 3. **Read Operation:** - Assert `sb_arvalid` and provide the address on `sb_araddr`. - The module will output the requested data on `sb_rdata`. - Monitor `sb_rvalid` to confirm the read operation completion. 4. **GPIO Control:** Use `gpio_o` to drive the output pins and monitor `gpio_i` for input pin states.

This specification provides a comprehensive overview of the `gpio` module, enabling hardware engineers to implement and integrate it into larger systems effectively.

Functional Description (Generated by funcgen)

Verilog Module Descriptions

Module: gpio (File: gpio.v)

Purpose

The `gpio` module serves as a General-Purpose Input/Output (GPIO) controller. It interfaces with a system bus for reading from and writing to the GPIO ports. The design supports both input and output operations using a 16-bit data interface.

Parameters

- The module does not include any parameters.

Ports

- `sb_clk`: Input, 1-bit

- The system bus clock signal, used to synchronize all operations within the module.
- **sb_rst_n**: Input, 1-bit
 - Active-low reset signal to initialize or reset the state of the module.
- **sb_arvalid**: Input, 1-bit
 - Asserted by the bus master when it has a valid read address.
- **sb_arready**: Output, 1-bit
 - Asserted by the module when it is ready to accept a read address.
- **sb_araddr**: Input, 32-bit
 - The address from which data is to be read.
- **sb_rvalid**: Output, 1-bit
 - Asserted by the module to indicate that the read data is valid.
- **sb_rready**: Input, 1-bit
 - Asserted by the master indicating it is ready to accept read data.
- **sb_rdata**: Output, 32-bit
 - The data read from the specified address.
- **sb_wvalid**: Input, 1-bit
 - Asserted by the bus master when it has valid write data.
- **sb_wready**: Output, 1-bit
 - Asserted by the module when it is ready to accept write data.
- **sb_waddr**: Input, 32-bit
 - Address at which write data is to be stored.
- **sb_wdata**: Input, 32-bit
 - Data to be written to the specified address.
- **sb_wstrb**: Input, 4-bit
 - Write strobe signals; unused in provided code.
- **sb_bvalid**: Output, 1-bit
 - Asserted by the module when a write transaction is completed.
- **sb_bready**: Input, 1-bit
 - Asserted by the master indicating it is ready to accept the write response.
- **sb_bresp**: Output, 1-bit
 - Fixed to ‘0’, indicating no error in the write response.
- **gpio_i**: Input, 16-bit
 - Input data from GPIO pins.
- **gpio_o**: Output, 16-bit
 - Output data to GPIO pins.

Internal Signals

- **gpio_odr**: reg, 16-bit
 - Register that holds the output data for the GPIO pins.
- **sb_bvalid_r**: reg, 1-bit
 - Register that holds the write response valid signal state.
- **sb_rvalid_r**: reg, 1-bit
 - Register that holds the read response valid signal state.

- **sb_rdata_r**: reg, 32-bit
 - Register that holds the data to be read.

Functionality

Sequential Logic

- **Write Data Handling**:
 - On every positive edge of **sb_clk**:
 - * If **sb_rst_n** is low, **gpio_odr** and **sb_bvalid_r** are reset.
 - * If **sb_wvalid** and **sb_wready** are asserted:
 - If the written address (**sb_waddr[2]**) is 0, the lower 16 bits of **sb_wdata** are loaded into **gpio_odr**.
 - **sb_bvalid_r** is set high to indicate the write transaction is complete.
 - * If **sb_bready** is asserted, **sb_bvalid_r** is reset.
- **Read Data Handling**:
 - On every positive edge of **sb_clk**:
 - * If **sb_rst_n** is low, **sb_rdata_r** and **sb_rvalid_r** are reset.
 - * When **sb_arvalid** and **sb_arready** are high:
 - Depending on **sb_araddr[2]**, **sb_rdata_r** is set to either **gpio_odr** with leading zero padding or **gpio_i** with leading zero padding.
 - **sb_rvalid_r** is set high to signal that read data is ready.
 - * If **sb_rready** is asserted, **sb_rvalid_r** is reset.

Combinational Logic

- Assignments that are purely combinational:
 - **sb_wready** is always high.
 - **sb_arready** is always high.
 - **sb_bresp** is constant at '0'.
 - Outputs (**sb_bvalid**, **sb_rvalid**, **sb_rdata**, **gpio_o**) are assigned from their respective registers or internal signals.

Instantiations

- This module does not instantiate any sub-modules.

Inter-Module Connections

- This standalone **gpio** module does not instantiate or directly interface with other modules in the provided context. Integration of this module into a larger system involves interacting with the system bus through defined signals and handling GPIO pin data via **gpio_i** and **gpio_o**.