APB GPIO Specification

Introduction

The `apb_gpio` module is a General Purpose Input/Output (GPIO) controller designed to interface with an Advanced Peripheral Bus (APB). It provides a configurable interface for controlling and monitoring GPIO pins, supporting interrupt generation and pad configuration. The module adheres to the APB protocol, ensuring compatibility with standard APB-based systems.

Architecture

The `apb_gpio` module is structured to handle GPIO operations through a series of registers accessible via the APB interface. The architecture includes:

- **Top-Level Structure**: The module is parameterized to support a configurable address width (`APB_ADDR_WIDTH`), defaulting to 12 bits, allowing for a 4KB address space.
- **Key Operations**:
  - **Data Synchronization**: Input signals are synchronized to the system clock to mitigate metastability issues.
  - **Interrupt Handling**: The module supports edge-triggered interrupts, configurable for rising, falling, or both edges.
  - **Clock Gating**: Utilizes clock gating for power efficiency, enabling or disabling clocks based on GPIO enable signals.
- **Implementation Details**:
  - **Clock Domains**: Operates within a single clock domain (`HCLK`), with reset (`HRESETn`) active low.
  - **Optimization**: Employs clock gating to reduce power consumption when GPIOs are not in use.

Interface

Parameters

Parameter	Type	Default	Description
`APB_ADDR_WIDTH`	`logic`	12	Width of the APB address bus

Ports

Clock and Reset

Signal	Width	In/Out	Description
`HCLK`	1	In	System clock signal
`HRESETn`	1	In	Active-low reset signal

APB Interface

Signal	Width	In/Out	Description
PADDR	APB_ADDR_WIDTH	In	APB address bus
PWDATA	32	In	APB write data bus
PWRITE	1	In	APB write enable
PSEL	1	In	APB select signal
PENABLE	1	In	APB enable signal
PRDATA	32	Out	APB read data bus
PREADY	1	Out	APB ready signal
PSLVERR	1	Out	APB slave error signal

GPIO Interface

Signal	Width	In/Out	Description
gpio_in	32	In	Input data from GPIO pins
gpio_in_sync	32	Out	Synchronized input data
gpio_out	32	Out	Output data to GPIO pins
gpio_dir	32	Out	GPIO direction control
gpio_padcfg	32x6	Out	GPIO pad configuration
interrupt	1	Out	Interrupt signal

Internal Signals

Signal	Width	Description
r_gpio_inten	32	GPIO interrupt enable register
r_gpio_inttype0	32	GPIO interrupt type register 0
s_gpio_inttype0	32	Synchronized interrupt type register 0
r_gpio_inttype1	32	GPIO interrupt type register 1
s_gpio_inttype1	32	Synchronized interrupt type register 1
r_gpio_out	32	GPIO output register
r_gpio_dir	32	GPIO direction register
r_gpio_sync0	32	First stage synchronization register
r_gpio_sync1	32	Second stage synchronization register
r_gpio_in	32	Registered GPIO input
r_gpio_en	32	GPIO enable register
r_gpio_lock	32	GPIO lock register
s_gpio_rise	32	Detected rising edges
s_gpio_fall	32	Detected falling edges
s_is_int_rise	32	Rising edge interrupt status
s_is_int_rifa	32	Rising or falling edge interrupt status
s_is_int_fall	32	Falling edge interrupt status

Signal	Width	Description
s_is_int_all	32	Combined interrupt status
s_rise_int	1	Combined interrupt signal
s_apb_addr	5	APB address decoding
r_status	32	Interrupt status register
s_clk_en	8	Clock enable signals for each GPIO group
s_clkg	8	Gated clock signals for each GPIO group

Timing

- **Latency**: The module operates with a single clock cycle latency for read and write operations through the APB interface.
- **Signal Sampling**: Inputs are sampled on the rising edge of HCLK, and outputs are updated synchronously with the clock.
- **Interrupts**: Interrupts are generated based on edge detection and are synchronized to the system clock.

Usage

To use the apb_gpio module: 1. **Initialization**: Ensure the module is reset by asserting HRESETn low. 2. **Configuration**: Configure GPIO direction, output values, and interrupt settings via the APB interface. 3. **Operation**: Monitor and control GPIO pins by reading from and writing to the appropriate registers. 4. **Interrupt Handling**: Configure interrupt types and enable interrupts as needed. Monitor the interrupt output for interrupt events. 5. **Clock Gating**: Utilize the clock gating feature to reduce power consumption when GPIOs are not in use.

This specification provides a comprehensive overview of the apb_gpio module, enabling hardware engineers to design and integrate the module into their systems effectively.

Functional Description (Generated by funcgen)

Module Documentation

Module: apb_gpio

File: apb_gpio.sv

Purpose

The apb_gpio module functions as a GPIO (General Purpose Input/Output) interface, capable of handling inputs and outputs for up to 32 GPIOs. It is

designed to interface with an APB (Advanced Peripheral Bus), allowing for configuration and control of the GPIOs through memory-mapped registers. The module supports interrupt generation based on edge-detection of input signals.

Parameters

- **APB_ADDR_WIDTH**: Default value is 12. This parameter specifies the address width of the memory-mapped interface, supporting a default 4KB space for the APB peripheral.

Ports

- **HCLK (input, 1-bit)**: The main clock signal driving sequential logic within the module.
- **HRESETn (input, 1-bit)**: An active-low reset signal, used to reset the module's state.
- **dft_cg_enable_i (input, 1-bit)**: Test enable signal for clock gating, typically used in DFT (Design For Test).
- **PADDR (input, `APB_ADDR_WIDTH` bits)**: Address bus for APB interface, used for addressing internal registers.
- **PWDATA (input, 32-bit)**: Data bus for writing data into registers.
- **PWRITE (input, 1-bit)**: Write signal for APB transactions; high indicates a write operation.
- **PSEL (input, 1-bit)**: Select signal indicating that the module is the target of a current APB transfer.
- **PENABLE (input, 1-bit)**: Enable signal timing the second cycle of an APB access.
- **PRDATA (output, 32-bit)**: Data bus for reading data from registers.
- **PREADY (output, 1-bit)**: Indicates the slave is ready to transfer data.
- **PSLVERR (output, 1-bit)**: Indicates a slave error response.
- **gpio_in (input, 32-bit)**: Input port for external GPIO signals.
- **gpio_in_sync (output, 32-bit)**: Synchronized version of `gpio_in` used for internal processing.
- **gpio_out (output, 32-bit)**: Output signals to control external GPIOs.
- **gpio_dir (output, 32-bit)**: Data direction for GPIO pins, output or input.
- **gpio_padcfg (output, [31:0][5:0] bits)**: Configuration bits for each pin defining drive strength and pull settings.
- **interrupt (output, 1-bit)**: An interrupt line indicating an event on GPIO inputs.

Internal Signals

- **r_gpio_inten (32-bit)**: Register keeping track of interrupts enabled for each GPIO.
- **r_gpio_inttype0/s_gpio_inttype0 (32-bit)**: Define the interrupt trigger type (falling edge, rising edge).

- **r_gpio_out (32-bit)**: Register that holds the current output state of the GPIOs.
- **r_gpio_dir (32-bit)**: Register that keeps track of the direction (input/output) of each GPIO.
- **r_gpio_sync0/r_gpio_sync1 (32-bit)**: Intermediate registers used for input synchronization to prevent metastability.
- **r_gpio_in (32-bit)**: Register of current synchronized GPIO inputs, used for edge detection.
- **r_gpio_en (32-bit)**: Enable signals for GPIOs.
- **r_gpio_lock (32-bit)**: Controls access restriction to other registers for locking configuration.
- **s_gpio_rise/s_gpio_fall (32-bit)**: Detect rising/falling edges on inputs.
- **s_is_int_rise/s_is_int_fall/s_is_int_rifa/s_is_int_all (32-bit)**: Intermediate calculations for interrupt conditions.
- **s_rise_int (1-bit)**: Indicates whether any interrupt condition has been met.
- **s_apb_addr (5-bit)**: Holds a truncated version of `PADDR` for register selection.
- **r_status (32-bit)**: Holds the current interrupt status for GPIOs.
- **s_clk_en (8-bit)**: Enable signals for clock gating on blocks of GPIOs.
- **s_clkg (8-bit)**: Resultant gated clock outputs used to drive different portions of synchronization.

Functionality

Sequential Logic

- The module uses a series of flip-flops to synchronize GPIO input signals (`gpio_in`) to the internal clock domain (`HCLK`) to resolve metastability, distributing these across 8 clock-gated domains.
- Registers like `r_gpio_out`, `r_gpio_dir`, `r_gpio_inten`, among others, are updated based on APB writes when the proper conditions (`PSEL`, `PENABLE`, `PWRITE`) along with address decoding are met.
- The module generates an interrupt when an enabled edge-triggered condition on a GPIO line occurs, where conditions are processed through a series of logical checks and stored in `r_status`.

Combinational Logic

- Address decoding logic is used to determine which register an APB transaction targets.
- Conditions for rising/falling interrupts are checked through bitwise operations to derive intermediate signals (`s_is_int_rise`, etc.) that inform edge detection.
- Outputs are computed based on the state of the internal registers and various combinational checks, enabling output GPIOs or indicating their

directions.

Instantiations

- **pulp_clock_gating**: A clock gating cell is instantiated 8 times for grouping GPIOs into blocks to efficiently manage the clock domains. Each instance connects `HCLK`, `s_clk_en`, and `dft_cg_enable_i` to produce `s_clkg` outputs, which feed into the synchronization registers.

Inter-Module Connections

The `apb_gpio` module represents a standalone GPIO interface block, anticipating expansion around an APB-based peripheral system. It directly consumes the `HCLK` and `HRESETn` for synchronous operations and APB signals for configuration management. The `pulp_clock_gating` instantiations manage local clock domains, simplifying the design's scalability and power efficiency when handling GPIO synchronizations. The module can seamlessly integrate within a larger SoC framework, offering programmable IO control coupled with interrupt support.