### 1. GPIO VIP Interface Signals (Detailed)

A robust GPIO VIP should model all the external and internal signals that interact with the GPIO block. Here’s a breakdown:

#### ****Essential Signals****

* **gpio\_in[N-1:0]**  
  Type: Input  
  Description: Represents the external signals coming into the GPIO block. Each bit corresponds to a physical pin.  
  Usage in VIP: The VIP drives these signals to simulate external events (e.g., button press, sensor input).
* **gpio\_out[N-1:0]**  
  Type: Output  
  Description: Signals driven by the GPIO block to the outside world. Each bit is an output pin.  
  Usage in VIP: The VIP monitors these signals to check if the DUT is driving the correct values.
* **gpio\_dir[N-1:0]**  
  Type: Input/Output (depends on implementation)  
  Description: Direction control for each pin. 1 = Output, 0 = Input.  
  Usage in VIP: The VIP can configure this (if register-mapped) or monitor it to ensure correct directionality.
* **gpio\_oe[N-1:0]** (Optional)  
  Type: Input/Output  
  Description: Output enable for each pin, used for tri-state or open-drain configurations.  
  Usage in VIP: The VIP can drive or monitor this to test tri-state behavior.
* **gpio\_intr[N-1:0]** (Optional)  
  Type: Output  
  Description: Interrupt lines, one per pin or a shared line, indicating an event (e.g., edge or level detected).  
  Usage in VIP: The VIP monitors these to verify correct interrupt generation.
* **clk**  
  Type: Input  
  Description: System clock for synchronous operation.  
  Usage in VIP: Drives timing for register accesses and pin sampling.
* **rst\_n**  
  Type: Input  
  Description: Active-low reset.  
  Usage in VIP: Used to reset the DUT and VIP state.

#### ****Example SystemVerilog Interface****

systemverilog

Copy Code

interface gpio\_if #(parameter N = 32) (input logic clk, input logic rst\_n);

logic [N-1:0] gpio\_in; // External input to DUT

logic [N-1:0] gpio\_out; // Output from DUT

logic [N-1:0] gpio\_dir; // Direction control

logic [N-1:0] gpio\_oe; // Output enable (optional)

logic [N-1:0] gpio\_intr; // Interrupts (optional)

endinterface

### 2. GPIO VIP Register Map (Detailed)

The register map defines how software (or the VIP) configures and queries the GPIO block. Here’s a typical, feature-rich map:

| **Address** | **Register Name** | **Width** | **Description** | **Access** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 0x00 | DATA\_IN | N | Current value of GPIO input pins | RO | Reflects external state on gpio\_in |
| 0x04 | DATA\_OUT | N | Value to drive on GPIO output pins | RW | Written by SW/VIP, drives gpio\_out when direction is output |
| 0x08 | DIR | N | Direction control: 1 = output, 0 = input | RW | Each bit controls one pin |
| 0x0C | OE | N | Output enable: 1 = enabled, 0 = tri-state | RW | Optional, for tri-state/open-drain |
| 0x10 | INTR\_MASK | N | Interrupt mask: 1 = enable interrupt for pin, 0 = disable | RW | Optional, per-pin interrupt enable |
| 0x14 | INTR\_STATUS | N | Interrupt status: 1 = interrupt pending | RO/WC | Write 1 to clear |
| 0x18 | INTR\_TYPE | N | Interrupt type: 0 = level, 1 = edge | RW | Optional, per-pin |
| 0x1C | INTR\_POLARITY | N | Interrupt polarity: 0 = low/falling, 1 = high/rising | RW | Optional, per-pin |

**Legend:**

* **RW**: Read/Write
* **RO**: Read Only
* **WC**: Write 1 to Clear

#### ****Register Field Details****

* **DATA\_IN:**  
  Read-only. Reflects the current state of the physical pins (gpio\_in).
* **DATA\_OUT:**  
  Read/Write. Value to be driven on output pins (gpio\_out) when direction is set to output.
* **DIR:**  
  Read/Write. Each bit sets the direction for a pin (1 = output, 0 = input).
* **OE:**  
  Read/Write. Enables output driver for each pin (for tri-state or open-drain support).
* **INTR\_MASK:**  
  Read/Write. Enables or disables interrupt generation for each pin.
* **INTR\_STATUS:**  
  Read/Write-Clear. Indicates which pins have pending interrupts. Writing 1 clears the interrupt.
* **INTR\_TYPE:**  
  Read/Write. Selects between level-sensitive or edge-sensitive interrupts for each pin.
* **INTR\_POLARITY:**  
  Read/Write. Selects between high/rising or low/falling for interrupt detection.