cadence

# General Purpose I/O Device (GPIO)

## Technical Data Sheet

Part Number: T-CS-PE-0002-100

Document Number: I-IPA01-0025-USR Rev 04

October 2002

# General Purpose I/O Device (GPIO)



## Proprietary Notice

# General Purpose I/O Device (GPIO)

## Features

- Up to 32 independently programmable I/O ports
- Each port provides input, output and output enable for bi-directional I/O pins
- Each port can be programmed as input or output
- Each port can be bypassed to or from a separate peripheral device
- Each port can separately trigger the GPIO interrupt on several event types
- Interfaces to AMBA version 2.0 APB
- Supports 100 MHz APB, when synthesized using 0.25 µm technology

## Description

The GPIO module provides up to 32 programmable I/O ports. Each port can be independently programmed.

### Modes

The following modes control the flow of data through the module. Bypass mode and GPIO mode are mutually exclusive per pin. Input mode and output mode are mutually exclusive per pin.

#### Bypass Mode

A direct through path between the I/O port and the bypass port is made. The signals are not latched or registered. Both directions are bypassed.

#### GPIO Mode

The I/O port is available to programmable registers.

#### Input mode

Data is routed through from the I/O port to either the `input_value` register or to both the `input_value` register and the bypass port.

#### Output mode

Data is routed through to the I/O port from either the `output_value` register or the bypass port.

### Output Enable

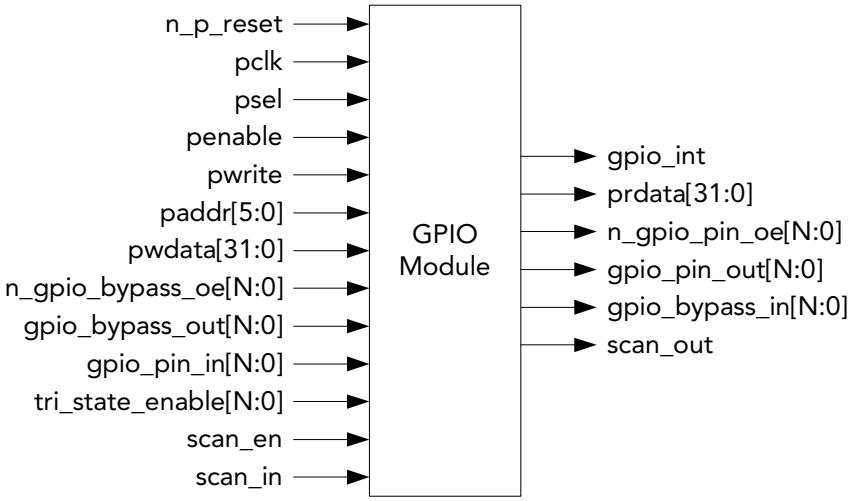
An output enable can be set for each pin. This can be used by external circuitry to control bi-directional pads.

### Interrupts

An interrupt trigger is generated for each pin if a pre-defined event type is seen on the pin input. The event can be specified as high or low level, rising or falling edge, or any edge. The event type that triggers an interrupt is defined in registers `int_type`, `int_value` and `int_on_any`. The event is recorded in a read-only register `int_status`. This register is cleared whenever it is read.

An interrupt is output if an interrupt trigger is seen for any pin, whose `int_mask` register bit is clear.

## Signal Interfaces




### AMBA APB Interface

| Signal Name | I/O | Function |
|---|---|---|
| `n_p_reset` | I | AMBA reset (`npreset`). This signal is negated asynchronously with `pclk` to reset the module, and asserted synchronously to allow the module to clock. |
| `pclk` | I | Peripheral bus clock (`pclk`). |
| `psel` | I | Peripheral Select for the GPIO (`psel`). This indicates that a valid access is being made to a peripheral register. |
| `penable` | I | Peripheral enable (`penable`). This indicates the second clock cycle of an access and indicates that the write data may be strobed into a register on the next rising edge of `pclk`, or that the read data is expected to be valid at the next rising edge of `pclk`. |
| `pwrite` | I | Peripheral write strobe (`pwrite`). This indicates that a write access is taking place (if `psel` is active). |
| `paddr[5:0]` | I | Address bus of selected master (`paddr`). This indicates which register is being accessed. |
| `pwdata[31:0]` | I | Write data. Data to be written into the addressed register. |
| `prdata[31:0]` | O | Read data. Data read from the addressed register. |

### Bypass Interface

| Signal Name | I/O | Function |
|---|---|---|
| `n_gpio_bypass_oe[N:0]` | I | Bypass mode output enable control. Indicates that the relevant bit of `gpio_bypass_out` is valid |
| `gpio_bypass_out[N:0]` | I | Bypass mode output value. |
| `gpio_bypass_in[N:0]` | O | Bypass mode input data value |

### GPIO Interface

| Signal Name | I/O | Function |
|---|---|---|
| `n_gpio_pin_oe[N:0]` | O | Output enable signal to the pin |
| `gpio_pin_out[N:0]` | O | Output value to the pin |
| `gpio_pin_in [N:0]` | I | Input data value from the pin |
| `gpio_int` | O | Interrupt indicating input pin event |

### Test Interface

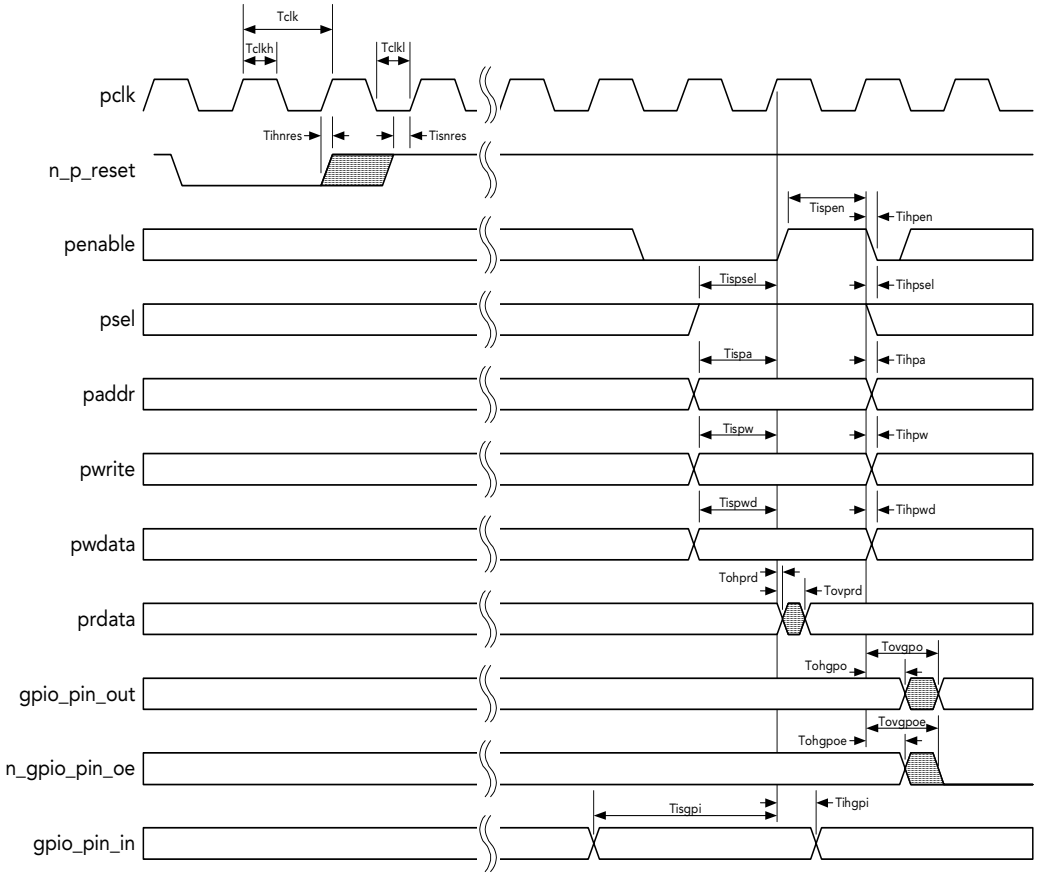
| Signal Name | I/O | Function |
|---|---|---|
| `scan_out` | O | Scan chain output data |
| `scan_in` | I | Scan chain input data |
| `scan_en` | I | Enable scan shift |
| `tri_state_enable[N:0]` | I | Disables each output enable |

## Timing Requirements

### APB/GPIO Timings

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| Tclk | Clock period, T01 (Note 1) | 25 | DC | ns |
| Tclkh | Clock high | 40% | - | × T0 |
| Tclkl | Clock low | 40% | - | × T0 |
| Tihnres | `n_p_reset` hold after `pclk` | 5% | - | × T0 |
| Tisnres | `n_p_reset` setup before `pclk` | 35% | - | × T0 |
| Tihpen | `penable` hold after `pclk` | 5% | - | × T0 |
| Tispen | `penable` setup before `pclk` | 35% | - | × T0 |
| Tihpsel | `psel` hold after `pclk` | 5% | - | × T0 |
| Tispsel | `psel` setup before `pclk` | 35% | - | × T0 |
| Tihpa | `paddr` hold after `pclk` | 5% | - | × T0 |
| Tispa | `paddr` setup before `pclk` | 35% | - | × T0 |
| Tihpw | `pwrite` hold after `pclk` | 5% | - | × T0 |
| Tispw | `pwrite` setup before `pclk` | 35% | - | × T0 |
| Tihpwd | `pwdata` hold after `pclk` | 5% | - | × T0 |

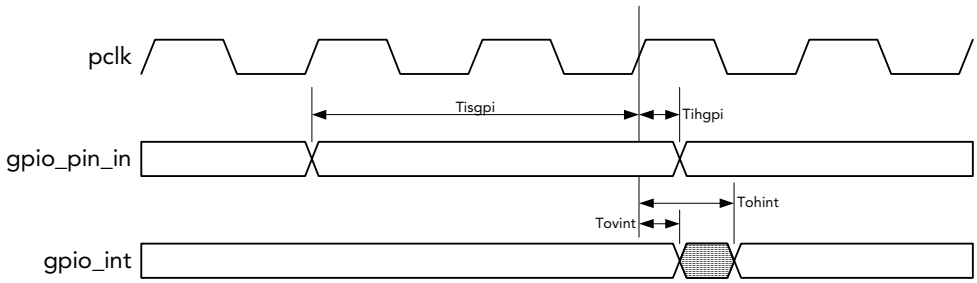
| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| Tispwd | `pwdata` setup before `pclk` | 35% | - | × T0 |
| Tihgpi | `gpio_pin_in` hold after read access | 5% | - | × T0 |
| Tisgpi | `gpio_pin_in` setup before read access (Note 2) | 35% | 3 | cycles |
| Tohprd | `prdata` hold after `pclk` | 5% | - | × T0 |
| Tovprd | `prdata` valid after `pclk` | - | 35% | × T0 |
| Tohgpvo | `gpio_pin_out` hold after `pclk` | 5% | - | × T0 |
| Tovgpvo | `gpio_pin_out` valid after `pclk` | - | 40% | × T0 |
| Tohgpoe | `n_gpio_pin_oe` hold after `pclk` | 5% | - | × T0 |
| Tovgpoe | `n_gpio_pin_oe` hold after `pclk` | - | 40% | × T0 |




*Note 1: All timings are specified relative to the target clock speed T0. These are coded into the provided synthesis script. These have been achieved with a typical technology, but are for guidance only.*

*Note 2: The inputs `gpio_pin_in` are asynchronous with `pclk`. There is a latency of 2 or 3 cycles before a change will be available for an APB read cycle.*

### Interrupt Timing




| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| Tihgpi | `gpio_pin_in` hold after read access | 0 | - | cycles |
| Tisgpi | `gpio_pin_in` setup before read access (see Note) | 2 | 3 | cycles |
| Tohint | `gpio_int` hold after `pclk` | 5% | - | × T0 |
| Tovint | `gpio_int` valid after `pclk` | - | 40% | × T0 |

*Note: The inputs `gpio_pin_in` are asynchronous with `pclk`. There is a latency of 2 or 3 cycles before a change will be available for an APB read cycle.*

## Programming Interface

The following registers can be configured to determine the functionality of the pin. Each pin has a register within each address and is of the format shown below. The maximum value of N is 31, where (N+1) is the number of I/O pins configured for the device.

| Bit | 31:~~ | N:1 | 0 |
|---|---|---|---|
| Field | Reserved (N<31) | Pin N:Pin 1 | Pin 0 |

### Register Map

| Offset | Name | Access | Reset Value | Description |
|---|---|---|---|---|
| 0x00 | `bypass_mode` | RW | [N:0] | If bit I is 1: set pin I to bypass mode. If bit I is 0: set pin I to GPIO mode. |
| 0x04 | `direction_mode` | RW | [N:0] | If bit I is 1: set pin I to input mode. If bit I is 0: set pin I to output mode. |
| 0x08 | `output_enable` | RW | [N:0] | If bit I is 1: set pin I to output enabled. If bit I is 0: set pin I to output disabled. Ignored if the pin is set to bypass mode. |
| 0x0C | `output_value` | RW | [N:0] | This register contains the value to be driven out of the pins. The output will only appear at the port if the pin is set to GPIO and output modes. |
| 0x10 | `input_value` | RO | [N:0] | The input value is read from this register, regardless of the pin mode. |

| Offset | Name | Access | Reset Value | Description |
|---|---|---|---|---|
| 0x14 | int_mask | RO | [N:0] | This register is used to mask interrupt events being signalled by gpio_int. If Bit I is 1, bit I of int_status is ignored. If Bit I is 0 an event on Bit I will set an interrupt. Bits are set and cleared using the registers int_enable and int_disable. |
| 0x18 | int_enable | WO | [N:0] | If bit I is 1, bit I of int_mask is cleared. |
| 0x1C | int_disable | WO | [N:0] | If bit I is 1, bit I of int_mask is set. |
| 0x20 | int_status | RO | [N:0] | If bit I is 1 an interrupt-generating event has occurred on bit I of input_value (int_status is set regardless of int_mask). |
| 0x24 | int_type | RW | [N:0] | If bit I is 1, interrupt is level triggered. If bit I is 0, interrupt is edge triggered. |
| 0x28 | int_value | RW | [N:0] | If bit I is 1, interrupt is triggered on high level or rising edge, depending on int_type value. If bit I is 0, interrupt is triggered on low level or falling edge, depending on int_type value. |
| 0x2C | int_on_any | RW | [N:0] | If bit I is 1 edge triggering occurs on any edge, otherwise edge specified in int_value triggers an interrupt (int_on_any is ignored if int_type=1). |

## Physical Estimates

| | |
|---|---|
| Gate count | ~230 2i/p NAND equivalents per GPIO pin |
| FF count | 13 FFs per GPIO pin |
| SOC-Internal pins (in) | 13 + 4 per GPIO pin |
| SOC-Internal pins (out) | 2 + 2 per GPIO pin |
| SOC-External pins (in) | 1 per GPIO pin |
| SOC-External pins (out) | 2 per GPIO pin |

## Verification

All our IP modules are verified to one of the following levels.

**Gold** IP has been to target silicon.
**Silver** IP has been to silicon in FPGA.
**Bronze** IP has been verified in simulation with logical timing closure.
**In development** IP has not yet been verified.

Please contact the IPGallery™ (ipgallery@cadence.com) for the latest verification information.

## Deliverables

The full IP package comes complete with:

- Verilog HDL
- Envisia (BuildGates) and Synopsys Design Compiler synthesis scripts
- Verilog testbench
- GPIO User Guide with full programming interface, parameterization instructions and synthesis instructions

## Cadence Design Foundry

Cadence Design Foundry, a division of Cadence Design Systems, Inc. (NYSE:CDN), is a leading provider of electronic design and supply chain management services. Leading and emerging technology companies around the globe leverage Cadence Design Foundry's engineering services and intellectual property for the design of complex integrated circuits and systems-on-chip (SoC). Cadence is head quartered in San Jose, California. For more information, please contact us: North America 1 877 473 2924, Europe +44 (0) 1506 595955, Asia +81 (0) 454 757756, by e-mail IPGallery@cadence.com, or visit us on the World Wide Web at: http://www.cadence.com.