INICORE

Datasheet

GPIO MODULE

Version 1.2

INICORE INC.
5600 Mowry School Road
Suite 180
Newark, CA 94560
t: 510 445 1529 *f:* 510 656 0995 *e:* info@inicore.com
www.inicore.com

Table of Contents

Table of Figures

1 Introduction

The GPIOmodule is a general purpose input/output controller, offering some unique features that eases system integration and use.

Each GPIO port can be configured for input, output or bypass mode. Output data can be set in one access or single or multiples bits can be set or cleared. Every GPIO port can serve as an interrupt source and has its own configuration options:

- Level sensitive, single edge triggered or level change
- Active high or low respectively rising edge or falling edge
- Individual interrupt enable register and status flags

1.1 Features

Following special features are available:

- Up to 32 ports
- Each port can be input, output or bypass
- Data set or bitwise set and clear control
- Input data synchronization
- Flexible interrupt generation for each GPIO pin
- Synchronous bus interface
  - Zero wait-states
  - Supports system bus' such as AMBA APB version 2.0
- Full synchronous design
- Synthesis Options:
  - CPU readback enable
  - CPU bus width
  - Number of ports
- Technology independent

1.2 Implementation Features

The core provides several synthesis options to ease the system integration and minimize the gate count:

- Selectable CPU bus width: default options are 8/16/32-bit
- Selectable number of GPIO ports
- Readback enable: Register readback can be disabled to optimize gate count.
- With a separate APB wrapper, the core can be used in ARM subsystems

1.2.1 Deliverables

- VHDL or Verilog RTL source code
- Simulation testbench
- Timing constraints file
- Synthesis script
- User Guide

1.3 Block Diagram

A high-level block diagram of the GPIOmodule is shown in the figure below:

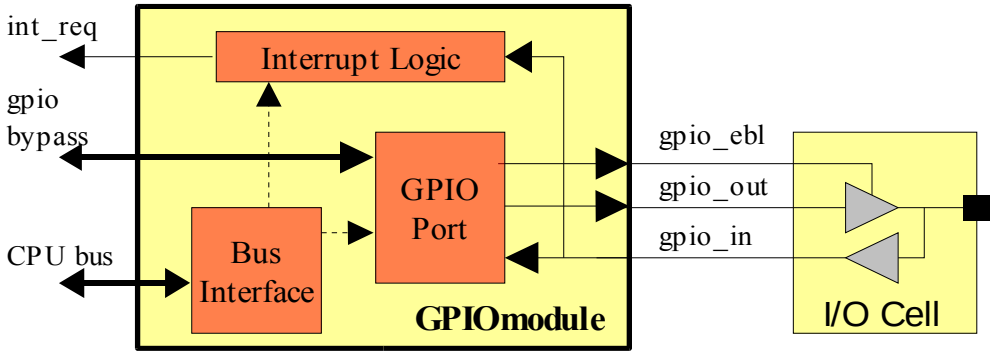

Figure 1: Block Diagram

2 Signal Descriptions

The following paragraph lists the input and output ports of the GPIOmodule. Please refer to the chapter 4 for a definition of the interface timings.

2.1 Symbol

The figure below shows all interfaces of the GPIOmodule core.

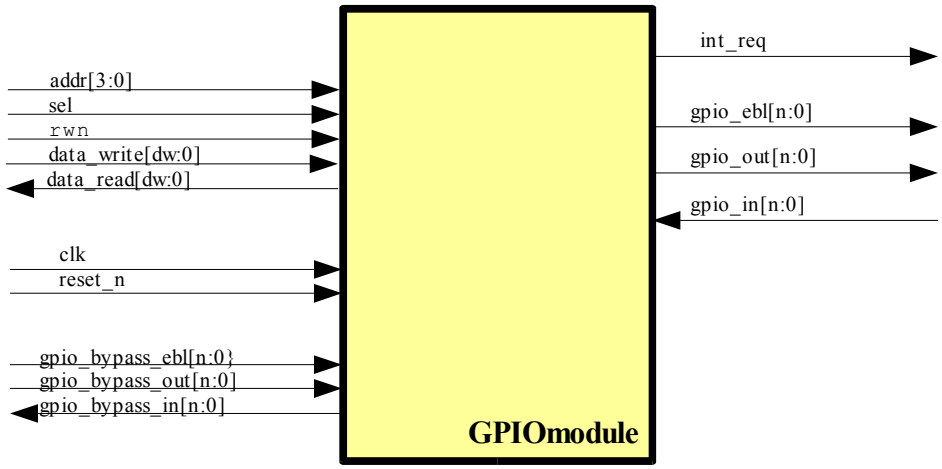

Figure 2: Symbol with I/Os

2.2 General Inputs

The module is initialized with one asynchronous, active low reset input. All registers are clocked with the system clock, there are no internal generated asynchronous clocks and resets.

Pin Name	*Type*	*Description*
clk	in	System clock
reset_n	in	Asynchronous system reset, active low

2.3 Local Bus

A synchronous zero wait-states system bus interface provides access to all status and configuration registers. The timing diagram is shown in chapter 4. For optimal system integration, this core can be configured to supported CPU systems that have a 8, 16, or 32-bit wide data bus.

Pin Name	*Type*	*Description*
addr[3:0]	in	Address bus input
data_write[dw:0]	in	Data bus input
data_read[dw:0]	out	Data bus output
sel	in	Module chip select, active high
rwn	in	Read/write control signal '0': Write '1': Read
int_req	out	Local interrupt request

2.4 GPIO Bypass Ports

Using the bypass port, the GPIO ports can be driven from internal system logic instead from the GPIOmodule.

Pin Name	*Type*	*Description*
gpio_bypass_in[n:0]	out	GPIO input from I/O cell
gpio_bypass_out[n:0]	in	GPIO output to I/O cell
gpio_bypass_ebl[n:0]	in	GPIO output enable, active HIGH

2.5 GPIO Ports

The GPIOmodule's ports have to be connected to the on-chip I/O cell.

Pin Name	*Type*	*Description*
gpio_in[n:0]	in	GPIO driver input
gpio_out[n:0]	out	GPIO driver output
gpio_ebl[n:0]	out	GPIO driver output enable, active HIGH

Figure 3 shows how the I/O cell has to be connected to the GPIOmodule.

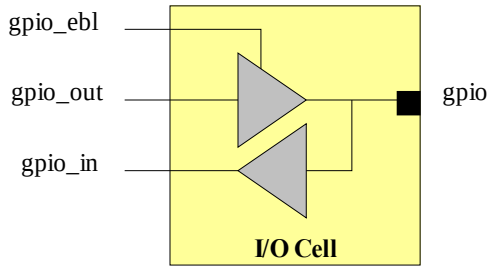

Figure 3: I/O Cell

3 Programming Model

3.1 Memory Mapping

The table below shows the entire memory mapping of the GPIOmodule. A detailed description of the functionalities of these registers can be found in paragraph 3.2.

Address	*Name*	*R/W*	*Description*
+0x00	gpio_data	R/W	GPIO Data Register
+0x01	gpio_ds	W	GPIO Data Set Register
+0x02	gpio_dc	W	GPIO Data Clear Register
+0x03	gpio_dir	r/W	I/O Direction Register
+0x04	gpio_bypass	r/W	GPIO Bypass Register
+0x05	int_status	R/W	Interrupt Status
+0x06	int_ebl	r/W	Interrupt Enabled
+0x07	int_type	r/W	Interrupt Type
+0x08	int_level	r/W	Interrupt Level
+0x09	int_edge	r/W	Interrupt Edge Type

The following acronyms are used in the table above:

r: Data readback

R: Data read

W: Data write

3.2 Register Description

3.2.1 GPIO Control

Each port can be either driven by the GPIOmodule or, if by-passed, driven by the internal functions.

Address	*Name*	*R/W*	*Description*
+0x00	gpio_data	R/W	GPIO Data Register Write: Set value of GPIO pins Read: Read value of GPIO pins
+0x01	gpio_ds	r/W	GPIO Data Set Register '0': no action '1': output is set to '1'
+0x02	gpio_dc	r/W	GPIO Data Clear Register '0': no action '1': output is set to '0'
+0x03	gpio_dir	r/W	I/O Direction Register '0': Input '1': Output
+0x04	gpio_bypass	r/W	GPIO Bypass Register '0': GPIO '1': Bypass

3.2.2 Interrupt Generation

Each GPIO port can serve as an interrupt source. Several configuration registers are provided for each input to configure them properly. Each interrupt source is synchronized to the local system clock domain using a dual stage synchronization.

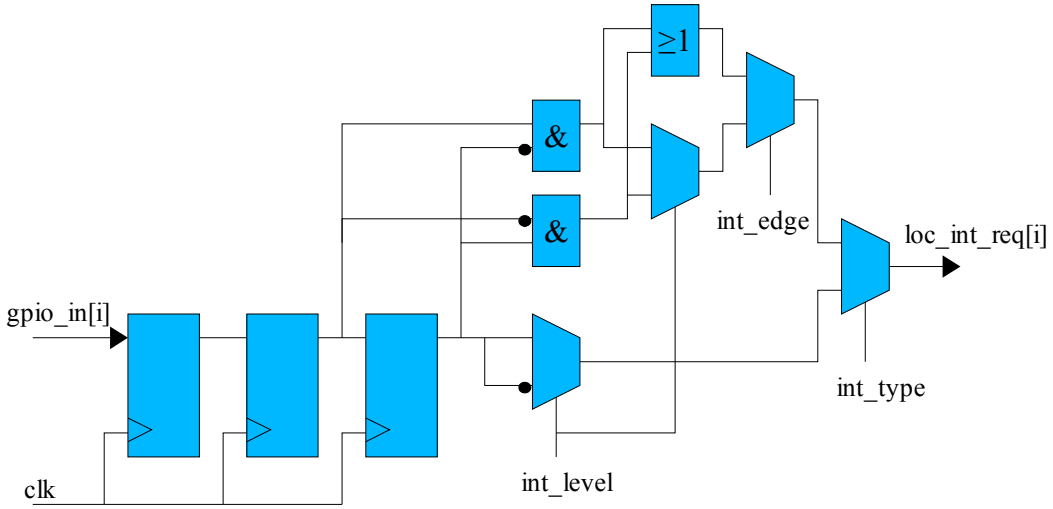

Figure 4: Interrupt Generation

Address	Name	R/W	Description
+0x05	int_status	R	Interrupt Status '0': No interrupt pending '1': Interrupt pending
		W	Interrupt Acknowledge '0': No action '1': Acknowledge pending interrupt
+0x06	int_ebl	r/W	Interrupt Enabled '0': Interrupt source disabled '1': Interrupt source enabled
+0x07	int_type	r/W	Interrupt Type '0': Level sensitive '1': Edge sensitive

Address	Name	R/W	Description
+0x08	int_level	r/W	Interrupt Level '0': Active low, falling edge '1': Active high, rising edge
+0x09	int_edge	r/W	Interrupt Edge Type '0': Single edge '1': Falling and rising edges

4 Timing Diagram

Timing numbers depend on the chosen device, synthesis options, place&route constraints. Typical numbers will be provided.

4.1 Local Bus Interface

The microprocessor interface is designed to operate on a full synchronous bus with zero wait-states. The internal registers are selected using cs_n, wr_n or rd_n respectively and addr. The selected register is written on the rising edge of the system clock. data_out is asynchronously generated.

Following figures shows the interface timing diagram.

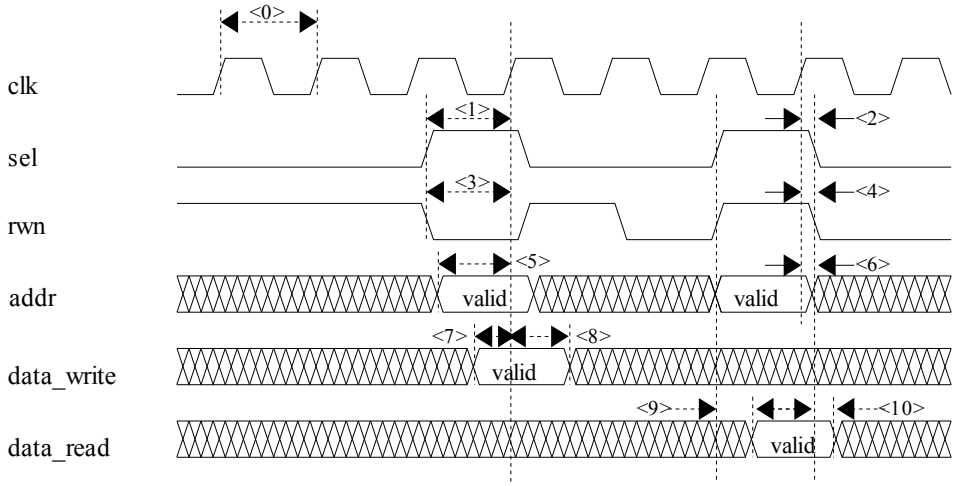

Figure 5: Local Bus Interface Timing

The timing numbers are technology dependent and can only be given once a target technology is selected.

Time nr	t_{min} *[ns]*	t_{typ} *[ns]*	t_{max} *[ns]*	*Comment*
<0>	tbd			Clock period
<1>		tbd		sel setup time
<2>		0		sel hold time
<3>		tbd		rwn setup time
<4>		0		rwn hold time
<5>		tbd		addr setup time
<6>		0		addr hold time
<7>		tbd		data_write setup time
<8>		0		data_write hold time
<9>		tbd		data_read valid after sel, rwn, addr valid
<10>		0		data_read hold time after sel, rwn, addr invalid

About Inicore

- FPGA and ASIC Design
- Easy-to-use IP Cores
- System-on-Chip Solutions
- Consulting Services
- ASIC to FPGA Migration
- Obsolete ASIC Replacements

Inicore is an experienced system design house providing FPGA / ASIC and SoC design services. The company's expertise in architecture, intellectual property, methodology and tool handling provides a complete design environment that helps customers shorten their design cycle and speed time to market. Our offering covers feasibility study, concept analysis, architecture definition, code generation and implementation. When ready, we deliver you a FPGA or take your design to an ASIC provider, whatever is more suitable for your unique solution.

Customer Advantages

We offer one-stop shopping for everything from the specifications to the chip or module solution. Our experience and fast turnaround time reduces your development costs and increases your returns from the market. Your system is not limited by the level of expertise and standard chip solutions you happen to have in-house. Achieve market success by differentiating and optimizing your product. Reusability builds the basis for further developments in the ever-decreasing product life cycle.

Visit us @ www.inicore.com