ARM PrimeCell™ General Purpose Input/Output (PL061)

Technical Reference Manual

ARM®

ARM DDI 0190B

ARM PrimeCell General Purpose Input/Output (PL061)
Technical Reference Manual

Release Information

The following changes have been made to this document.

Change history

Date	Issue	Change
August 2000	A	First release
September 2000	B	Only confidentiality and product status changed.

Proprietary Notice

ARM, the ARM Powered logo, Thumb, and StrongARM are registered trademarks of ARM Limited.

The ARM logo, AMBA, Angel, ARMulator, EmbeddedICE, ModelGen, Multi-ICE, PrimeCell, ARM7TDMI, ARM7TDMI-S, ARM9TDMI, ARM9E-S, ARM966E-S, ETM7, ETM9, TDMI and STRONG are trademarks of ARM Limited.

All other products or services mentioned herein may be trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Contents
ARM PrimeCell General Purpose Input/Output (PL061) Technical Reference Manual

Preface

Chapter 1 Introduction

Chapter 2 Functional Overview

Chapter 3 Programmer's Model

Chapter 4 Programmer's Model for Test

List of Tables
ARM PrimeCell General Purpose Input/Output (PL061) Technical Reference Manual

List of Figures
ARM PrimeCell General Purpose Input/Output (PL061) Technical Reference Manual

Preface

This preface introduces the ARM PrimeCell GPIO (PL061) and its reference documentation. It contains the following sections:

- *About this document* on page x
- *Further reading* on page xii
- *Feedback* on page xiii.

About this document

This document is a technical reference manual for the ARM PrimeCell GPIO (PL061).

Intended audience

This document has been written for hardware and software engineers implementing System-on-Chip designs. It provides information to enable designers to integrate the peripheral into a target system as quickly as possible.

Using this manual

This document is organized into the following chapters:

Chapter 1 ***Introduction*** Read this chapter for an introduction to the ARM PrimeCell GPIO (PL061).

Chapter 2 ***Functional Overview*** Read this chapter for a description of the major functional blocks of the PrimeCell GPIO.

Chapter 3 ***Programmer's Model*** Read this chapter for a description of the PrimeCell GPIO registers and programming details.

Chapter 4 ***Programmer's Model for Test*** Read this chapter for a description of the logic in the PrimeCell GPIO for functional verification and production testing.

Chapter 5 ***Signal Descriptions*** Read this appendix for details of the PrimeCell GPIO signals.

Typographical conventions

The following typographical conventions are used in this document:

bold Highlights ARM processor signal names, and interface elements such as menu names. Also used for terms in descriptive lists, where appropriate.

italic Highlights special terminology, cross-references, and citations.

`typewriter` Denotes text that can be entered at the keyboard, such as commands, file names and program names, and source code.

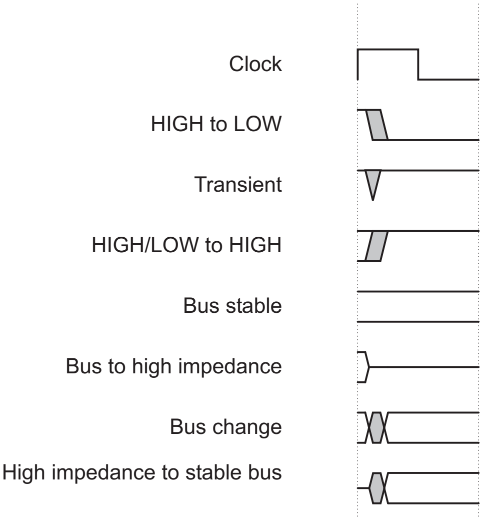
`typewriter` Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.

`typewriter italic` Denotes arguments to commands or functions where the argument is to be replaced by a specific value.

typewriter bold Denotes language keywords when used outside example code.

Timing diagram conventions

This manual contains one or more timing diagrams. The following key explains the components used in these diagrams. Any variations are clearly labeled when they occur. Therefore, no additional meaning should be attached unless specifically stated.



Key to timing diagram conventions

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Further reading

This section lists publications by ARM Limited, and by third parties.

ARM periodically provides updates and corrections to its documentation. See `http://www.arm.com` for current errata sheets and addenda.

See also the ARM Frequently Asked Questions list at: `http://www.arm.com/DevSupp/Sales+Support/faq.html`

ARM publications

This document contains information that is specific to the ARM PrimeCell GPIO (PL061). Refer to the following documents for other relevant information:

- *AMBA Specification (Rev 2.0)* (ARM IHI 0011)
- *ARM PrimeCell GPIO (PL061) Design Manual* (PL061 DDES 0000)
- *ARM PrimeCell GPIO (PL061) Integration Manual* (PL061 INTM 0000).

Feedback

ARM Limited welcomes feedback both on the ARM PrimeCell GPIO (PL061), and on the documentation.

Feedback on the ARM PrimeCell GPIO (PL061)

If you have any comments or suggestions about this product, please contact your supplier giving:

- the product name
- a concise explanation of your comments.

Feedback on this document

If you have any comments on about this document, please send email to `errata@arm.com` giving:

- the document title
- the document number
- the page number(s) to which your comments refer
- a concise explanation of your comments.

General suggestions for additions and improvements are also welcome.

Chapter 1
Introduction

This chapter introduces the ARM PrimeCell GPIO (PL061). It contains the following section:

- *About the ARM PrimeCell GPIO (PL061)* on page 1-2.

1.1 About the ARM PrimeCell GPIO (PL061)

The PrimeCell GPIO is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant *System-on-Chip* (SoC) peripheral that is developed, tested, and licensed by ARM.

The PrimeCell GPIO is an AMBA slave module that connects to the *Advanced Peripheral Bus* (APB). The PrimeCell GPIO provides eight programmable inputs or outputs that you can control in two modes:

- software mode through an APB bus interface
- hardware mode through a hardware control interface.

You can create ports of different widths (for example 16, 24, 32, and 40 bits) by multiple instantiation. An interrupt interface is provided to configure any number of pins as interrupt sources. You can generate interrupts depending on a level, or a transitional value of a pin. At system reset, PrimeCell GPIO lines default to inputs. The PrimeCell GPIO interfaces with input and output pad cells using a data input, data output, and output enable per pad.

Figure 1-1 on page 1-3 shows the PrimeCell GPIO interfaces.

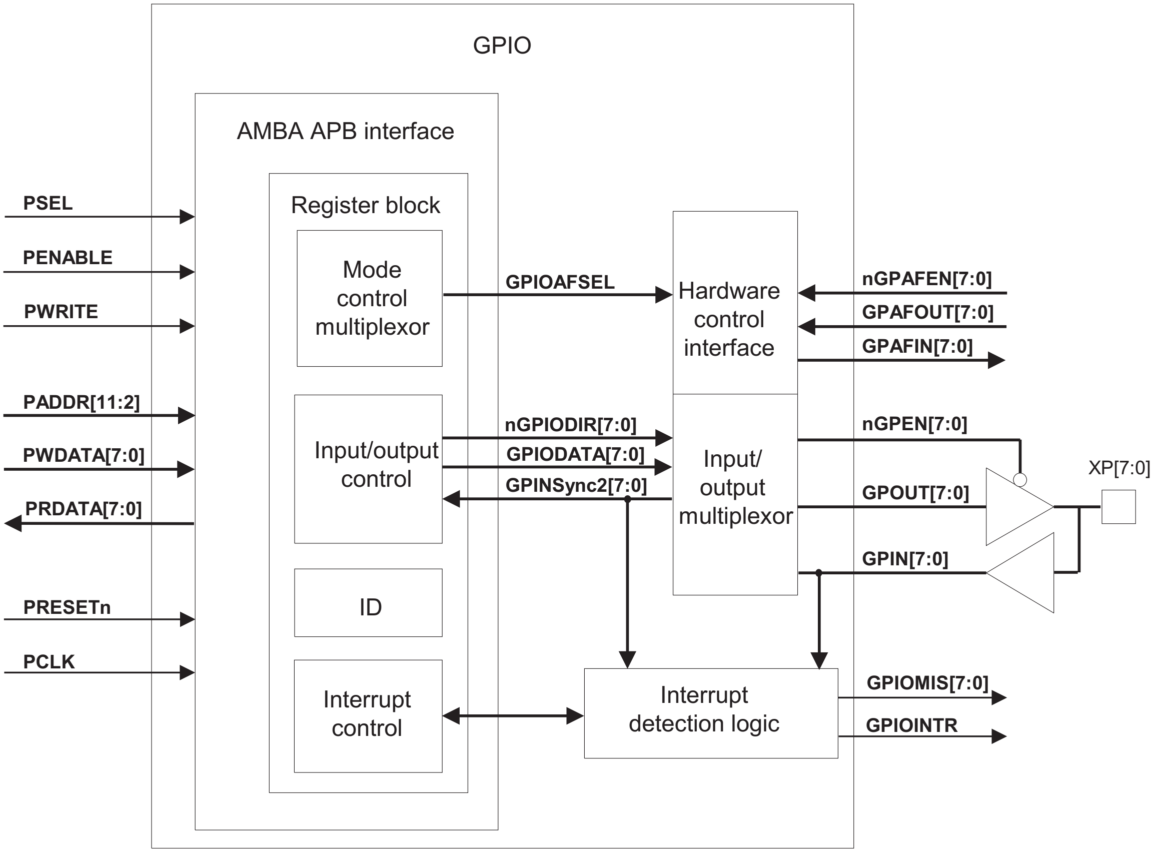


Figure 1-1 PrimeCell GPIO block diagram and pads connections

1.1.1 Features of the PrimeCell GPIO

The PrimeCell GPIO offers:

- Compliance to the AMBA Specification (Rev 2.0) onwards for easy integration into SoC implementation.
- Eight individually programmable input/output pins, default to input at reset.
- Scalability by multiple instantiation to 16, 24, 32, 40, or more bits.
- Programmable interrupt generation capability, from a transition or a level condition, on any number of pins.

- Hardware control capability of PrimeCell GPIO lines for different system configurations.
- Bit masking in both read and write operations through address lines.
- Identification registers that uniquely identify the PrimeCell GPIO.

Chapter 2
Functional Overview

This chapter describes the major functional blocks of the ARM PrimeCell GPIO. It contains the following sections:

2.1 PrimeCell GPIO overview

The PrimeCell GPIO is an *Advanced Microcontroller Bus Architecture* (AMBA) bus slave that connects to the AMBA *Advanced Peripheral Bus* (APB). It provides eight programmable inputs or outputs that you can control in two modes:

- software mode through an APB bus interface
- hardware mode through a hardware control interface.

The CPU accesses data, control, and status information of the PrimeCell GPIO through the APB bus interface.

The PrimeCell GPIO peripheral includes the following registers:

- *Data direction register*
- *Data register*
- *Interrupt control registers*
- *Mode control select register*
- *Identification registers* on page 2-3.

2.1.1 Data direction register

The data direction register is eight bits wide and configures each pin as an input or output.

2.1.2 Data register

The data register is eight bits wide and is used to:

- read the value input on those PrimeCell GPIO lines that are configured as inputs
- program the value on those PrimeCell GPIO lines that are configured as outputs.

The same data register appears at 256 locations in the memory map. This allows you to use the address bus [9:2] as an additional bit masking feature.

2.1.3 Interrupt control registers

The PrimeCell GPIO has interrupt generation capability. You can configure any number of the external PrimeCell GPIO lines independently to trigger an interrupt through its corresponding bit in the seven interrupt registers.

2.1.4 Mode control select register

The PrimeCell GPIO lines can be controlled by software through the APB bus or by hardware through the hardware control interface. The mode of each PrimeCell GPIO line is selected by the mode control select register.

When you enable hardware control in a particular line, control and data transfer over this line is provided from an auxiliary source.

2.1.5 Identification registers

The PrimeCell GPIO identification registers contain peripheral and BIOS information that uniquely identifies the peripheral. An automatically configuring BIOS can scan through memory searching for the BIOS ID number. Once found, the BIOS can read the peripheral ID numbers and automatically configure itself.

2.2 PrimeCell GPIO functional description

Figure 2-1 shows a block diagram of the PrimeCell GPIO.

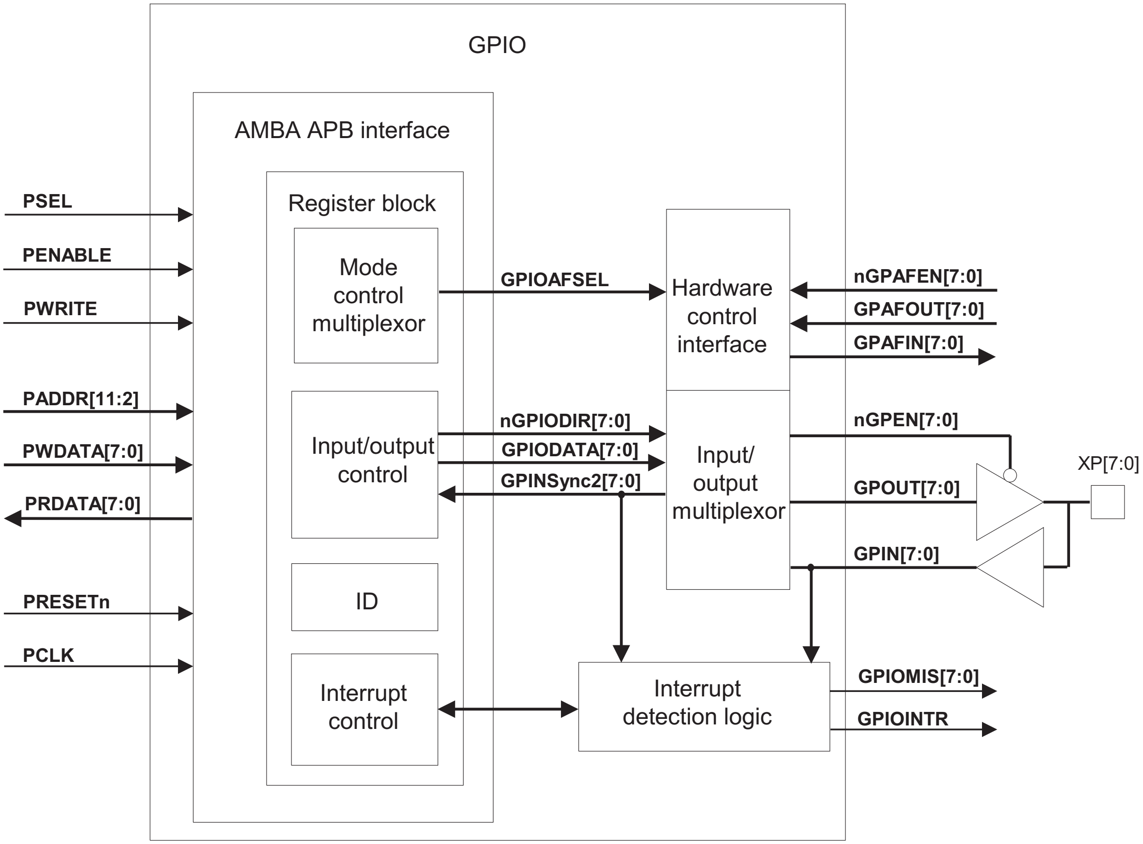


Figure 2-1 PrimeCell GPIO block diagram

Note

In Figure 2-1, for clarity, test logic is not shown.

The functions of the PrimeCell GPIO are described in the following sections:

- *AMBA APB interface* on page 2-5
- *Interrupt detection logic* on page 2-5
- *Mode control* on page 2-6.

2.2.1 AMBA APB interface

The AMBA APB is a local secondary bus that provides a low-power extension to the higher bandwidth AMBA *Advanced High-performance Bus* (AHB), within the AMBA system hierarchy. The AMBA APB groups narrow-bus peripherals to avoid loading the system bus, and provides an interface using memory-mapped registers that are accessed under programmed control.

The AMBA APB interface generates read and write decodes for accesses to control, interrupt, and data registers within the PrimeCell GPIO. A read-only decode is provided to access the ID codes.

The AMBA APB interface implements the storage elements for the data, data direction, mode control, interrupt interface, and identification registers.

2.2.2 Interrupt detection logic

The PrimeCell GPIO has the ability to generate mask-programmable interrupts based on the level, or transitional value of any of its PrimeCell GPIO lines.

The *General Purpose Input Output Interrupt* (GPIOINTR) indicates to an interrupt controller that an interrupt occurred in one or more of the PrimeCell GPIO lines.You can configure interrupts so that they are generated either on a change in the level, or on an edge of the PrimeCell GPIO line. The edge and level on which the interrupt must be generated is programmable.Seven registers in the AMBA APB interface, each controlling a different feature or condition in the interrupt triggering chain, allow the following functionality:

- interrupt generation either on a change in the level, one edge, or both edges of the PrimeCell GPIO line
- reading raw and masked interrupt status
- reading from and writing to the interrupt enable
- interrupt clear (write-only).

Each input/output line has a corresponding masked interrupt output line. Setting the appropriate mask bit HIGH enables the interrupt. GPIOINTR is the combined interrupt output of the PrimeCell GPIO masked interrupt status lines. It indicates to the interrupt controller that this block is requesting service from one or more of its interrupt sources.

Provision of individual outputs as well as a combined interrupt output, allows you to use either a global interrupt service routine or modular device drivers to handle interrupts.

2.2.3 Mode control

You can control the PrimeCell GPIO lines by software through the APB bus, or by hardware through the hardware control interface. Select the mode of each PrimeCell GPIO line using the mode control select register (GPIOAFSEL).

When software control mode is enabled (default) the data direction is controlled by the data direction register. Data writes and reads are then performed through the APB interface.

When hardware control mode is enabled, data direction is controlled through the auxiliary port direction control pins. Similarly, data is written and read through this port, but pin status can also be read through the APB interface.

2.3 PrimeCell GPIO operation

The operation of the PrimeCell GPIO is described in the following sections:

- *Interface reset*
- *Interface configuration*
- *Operation of the input/output lines* on page 2-8
- *Interrupt operation* on page 2-10
- *Mode control* on page 2-12.

2.3.1 Interface reset

All block registers are cleared during power-on-reset (LOW). This disables the output drivers for the Primecell GPIO lines, so that the pins are configured as inputs.

2.3.2 Interface configuration

On application of **PRESETn** as LOW:

- interrupts in the desired line are disabled by clearing the corresponding bit in GPIOIE
- all registers are cleared to zero
- input and output pins are configured as inputs
- interrupts to the external world are all masked as disabled
- raw interrupts are cleared to zero
- edge triggered interrupts are selected as source.

Recommendations

If you want to generate edge-triggered interrupts you must perform the following initialization sequence to avoid spurious interrupts being interpreted by the system:

- program GPIOIBE appropriately as individual or both-edge detection
- program GPIOIEV, if you have selected individual edge transactions previously
- program GPIOIS to select edge-triggered path
- apply three clock pulses to clean interrupt pipeline
- ensure GPIN[7:0] bus remains stable throughout this operation
- clear all interrupts by writing `0xFF` to GPIOIC
- program GPIOIE to enable interrupts.

2.3.3 Operation of the input/output lines

The Primecell GPIO block comprises eight programmable input/output lines. When the software control mode is enabled, data and control for these lines are provided by a data register and a data direction register. On reads, the data register contains the current status of the Primecell GPIO pins, whether they are configured as input or output. Writing to the data register only affects the pins that are configured as outputs.

Data register

So that independent software drivers can set their GPIO bits without affecting any other pins in a single write operation, the address bus is used as a mask on read/write operations. The data register effectively covers 256 locations in the address space. The eight address lines used are **PADDR [9: 2]**. During a write, if the address bit associated with that data bit is HIGH, the value of the GPIODATA register is altered. If it is LOW, it is left unchanged. For example:

Writing to address GPIODATA + 0x098 = 0b000010011000

PADDR[9:2] = 0b0000100110. When a value of `0xFB` is written to the address `0x098` then:

- bits 5, and 1 of the PrimeCell GPIO pins are set to 1, and bit 2 is set to 0
- the other bits are not changed.

Figure 2-2 shows the above effect of the address value of `0x098` operating on the data value of `0xFB`.

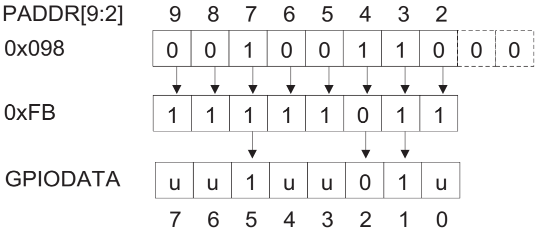


Figure 2-2 Example 2 Write to address 0x098

Note

In Figure 2-2 u indicates that the bit value is unchanged.

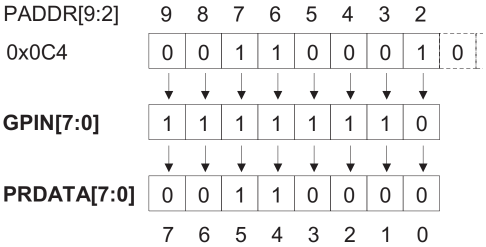
During a read if an address bit associated with data is HIGH the value is read, if it is LOW it is as zero. For example:

Read from address GPIODATA + 0x0C4 = 0b000011000100

PADDR[9:2] = 0b0000110001. When reading from `0x0C4` then:

- bits 5, 4, and 0 of the PrimeCell GPIO pins are returned
- the value of bits 7, 6, 3, 2, and 1 are returned as zero, regardless of their state.

Figure 2-3 shows a read from the address `0x0C4` and the output on the **PRDATA[7:0]** lines.



PADDR[9:2]	9	8	7	6	5	4	3	2	
0x0C4	0	0	1	1	0	0	0	1	0
GPIN[7:0]	1	1	1	1	1	1	1	0	
PRDATA[7:0]	0	0	1	1	0	0	0	0	
	7	6	5	4	3	2	1	0	

Figure 2-3 Example 2 Read from address 0x0C4

Data direction registers

The data direction registers operate in the following manner:

- 0 indicates the corresponding output pin is defined as an input
- 1 indicates the corresponding output pin is defined as an output.

Figure 2-4 on page 2-10 shows a typical write, in this case to the data direction register.

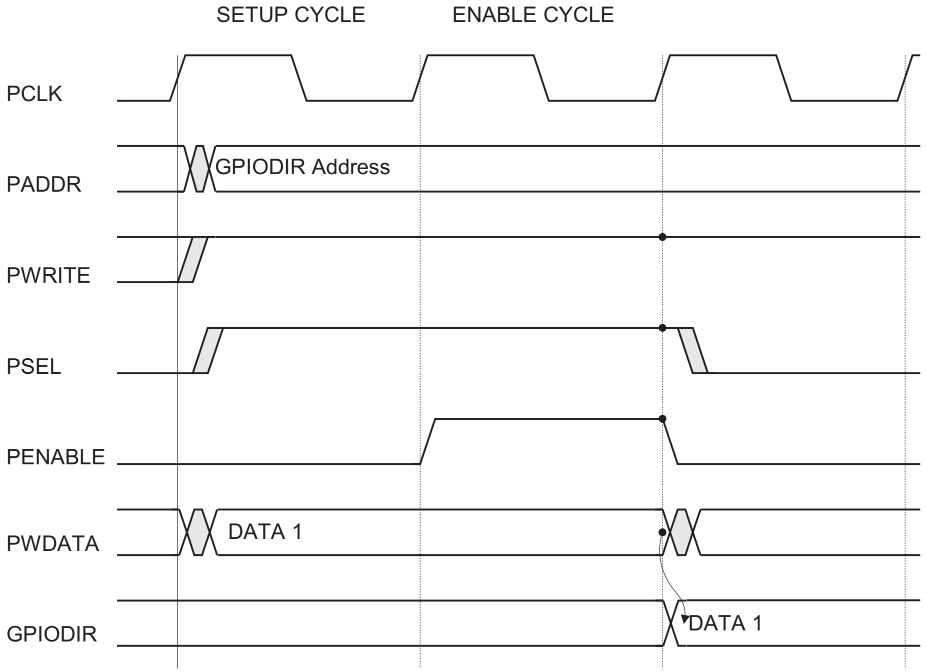


Figure 2-4 Write to data direction register

2.3.4 Interrupt operation

The interrupt section of the PrimeCell GPIO is controlled by a set of seven registers. You can select the source of the interrupt, its polarity, and edge properties. When one or more PrimeCell GPIO lines causes an interrupt, a single interrupt output **GPIOINTR** and/or the individual interrupts can be sent to the interrupt controller. For edge-triggered interrupts, software must clear the interrupt to enable any further interrupts. For a level case, it is assumed that the external source holds the level constant for the interrupt to be recognized by the processor.

Three registers are required to define the edge or sense that causes an interrupt:

- GPIOIS
- GPIOIBE
- GPIOIEV.

Figure 2-5 on page 2-11 shows how the bits of the three registers combine to select an interrupt source event.

Note

Each bit of the interrupt registers corresponds to a PrimeCell GPIO pin.

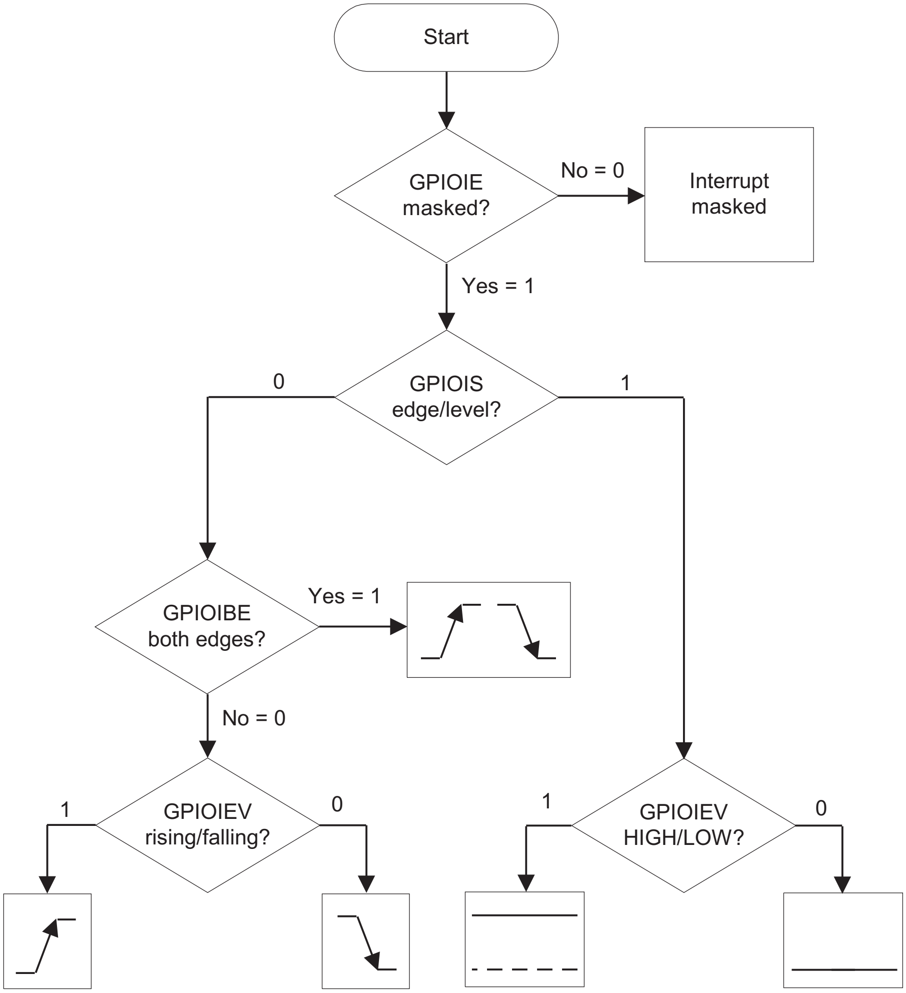


Figure 2-5 PrimeCell GPIO interrupt registers

Registers to be programmed

Table 2-1 shows how an interrupt is triggered by a rising edge detected on input pin 2.

Table 2-1 Triggering an interrupt from pin 2

Register	Desired trigger	7	6	5	4	3	2	1	0
GPIOIS	0 = edge 1 = level	x	x	x	x	x	0	x	x
GPIOIBE	0 = single edge 1 = both edges	x	x	x	x	x	0	x	x
GPIOIEV	LOW level, or negative edge HIGH level, or positive edge	x	x	x	x	x	1	x	x
GPIOIE	0 = masked 1 = not masked	0	0	0	0	0	1	0	0

Note

If any GPIOIE register bit is 0, the interrupt triggering on the associated line is disabled. In Table 2-1 an x indicates that the value of the associated bit is irrelevant, a consequence of the bit being masked by the GPIOIE register setting.

You must perform programming of the interrupt control registers when the respective interrupts are not enabled. Writing to interrupt control registers can generate spurious interrupts if the corresponding bits are enabled.

2.3.5 Mode control

You can control the PrimeCell GPIO lines through the hardware control interface. The mode of each line is selected by the mode control register. When hardware control is enabled in a particular line, the data and data direction control signals of the corresponding PrimeCell GPIO pins are ignored. Data and control transfers over these PrimeCell GPIO pins are then driven or read by some external auxiliary control block.

See Figure 2-6 on page 2-13 for a block diagram showing the mode control multiplexor.

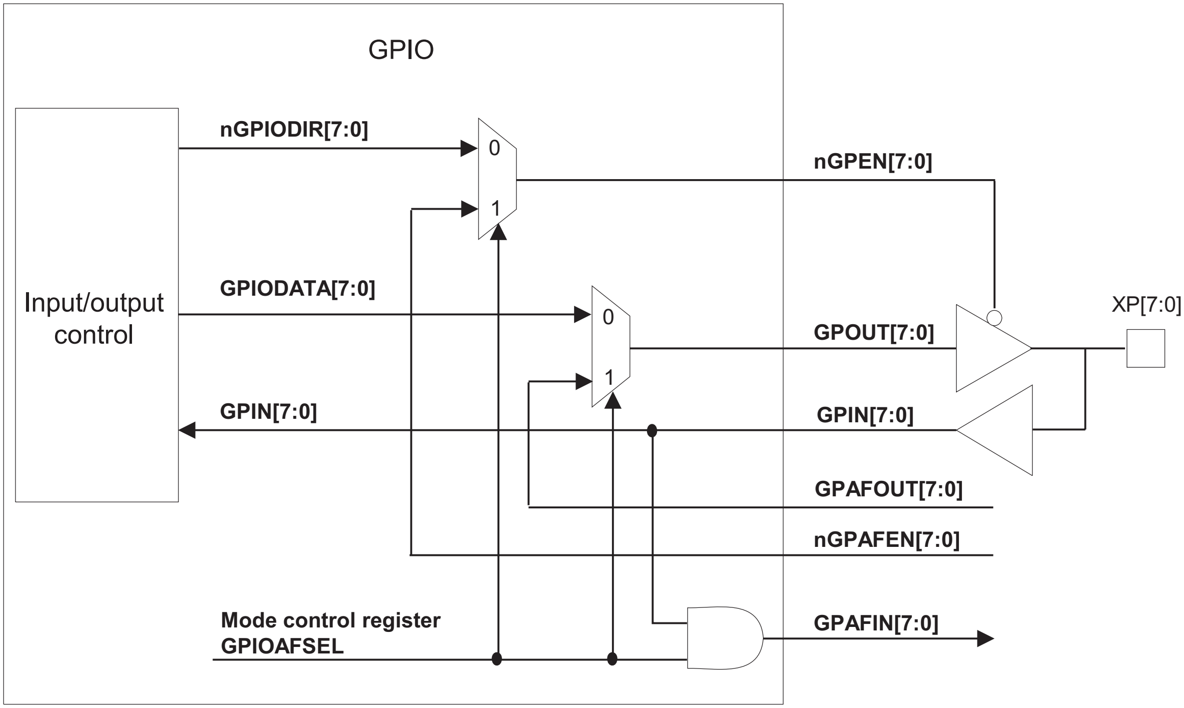


Figure 2-6 Mode control multiplexor

Table 2-2 shows how the pads are configured for hardware and software control.

Table 2-2 Pad configuration

	Hardware				Software			
	Output		Input		Output		Input	
	7	6	5	4	3	2	1	0
Mode control register, GPIOAFSEL	1	1	1	1	0	0	0	0
Hardware enable input, nGPAFEN	0	0	1	1	x	x	x	x
Hardware data input, GPAFOUT	A	B	x	x	x	x	x	x

Table 2-2 Pad configuration (continued)

	Hardware				Software			
	Output		Input		Output		Input	
	7	6	5	4	3	2	1	0
Hardware data output, GPAFIN	A	B	c	d	0	0	0	0
Software enable output, GPIODIR	x	x	x	x	1	1	0	0
Software data output, GPIODATA	x	x	x	x	E	F	g	h
Software data input, GPIN	A	B	c	d	E	F	g	h
Pad enable, nGPEN	0	0	1	1	0	0	1	1
Pad output, GPOUT	A	B	x	x	E	F	x	x
Pad input, GPIN	A	B	c	d	E	F	g	h
Bidirectional pad, XP	A	B	c	d	E	F	g	h

Note

A, B, E, F, are sources of output data. c, d, g, h, are sources of input data.

In hardware control mode the GPIODATA register can also read the values of lines configured as inputs.

Table 2-2 on page 2-13 is a concise truth table of operation when the GPIO PL061 has its data port pins configured as both software and hardware controlled. The values that exist on the bidirectional XP[7:0] lines are shown as the last row within the table, but it is realized that they have different sources as described below.

Table 2-2 on page 2-13 is best explained by considering each mode separately.

Hardware control mode

The left hand four columns of Table 2-2 relate to the hardware mode of operation.XP[7:4] pins have been configured as being under hardware control by setting the respective bits to 1 within the GPIOAFSEL register.Pins configured as outputs:

- XP[7:6] pins are configured as outputs by applying a 0 value to the respective nGPAFEN port signals.
- XP[7:6] data values are sourced from the GPAFOUT[7:6] port input signals, shown as (A,B). These values are propagated and driven out onto the XP[7:6] pins. These values are also transferred back to the GPAFIN[7:6] port output signals through the GPIN[7:6] pins. This feature also allows the XP[7:0] values to be read through the APB interface GPIODATA register.

Pins configured as inputs:

- XP[5:4] pins are configured as inputs by applying a 1 value to the respective nGPAFEN port signals.
- XP[5:4] values are driven from an external source, and as above, these values (c, d), are transferred to GPAFIN[5:4] signals. Again, the XP[7:0] can be read through the APB interface GPIODATA register.

Software control

The right hand four columns of Table 2-2 relate to the software mode of operation. In software mode the GPAFIN port signals are forced LOW as a power saving feature.XP[3:0] pins have been configured as being under software control by setting the respective bits to 0 within the GPIOAFSEL register.Pins configured as outputs:

- XP[3:2] pins are configured as outputs by setting the respective bits to 1 within the GPIODIR data direction register.
- XP[3:2] data values are sourced from the GPIODATA[3:2] register bits, shown as (E,F). These values are propagated and driven out onto the XP[3:2] pins. These values are transferred back to the GPIN[3:2] pins, but not to the GPAFIN pins as this route is disabled when in software mode.
- The XP[3:2] pin values can be read through the APB interface GPIODATA register.

Pins configured as inputs:

- XP[1:0] pins have been configured as inputs by setting the respective bits to 0 within the GPIODIR data direction register.

- XP[1:0] values are driven from an external source, shown as (g, h). The XP[1:0] pin values can be read through the APB interface GPIODATA register.

Chapter 3
Programmer's Model

This chapter describes the ARM PrimeCell GPIO (PL061) registers and provides details needed when programming the peripheral. It contains the following sections:

3.1 About the programmer's model

The base address of the PrimeCell GPIO is not fixed, and can be different for any particular system implementation. However, the offset of any particular register from the base address is fixed.The following locations are reserved and must not be used during normal operation:

- locations at offsets `0x424` to `0xFCC` are reserved for possible future extensions and test purposes
- locations at offsets `+0xFD0` to `+0xFDC` are reserved for future ID expansion.

3.2 Summary of PrimeCell GPIO registers

The PrimeCell GPIO registers are shown in Table 3-1.

Table 3-1 PrimeCell GPIO register summary

Address	Type	Width	Reset value	Name	Description
GPIO base + 0x000 to GPIO base + 0x3FC	Read/write	8	0x00	GPIODATA	PrimeCell GPIO data register
GPIO base + 0x400	Read/write	8	0x00	GPIODIR	PrimeCell GPIO data direction register
GPIO base + 0x404	Read/write	8	0x00	GPIOIS	PrimeCell GPIO interrupt sense register
GPIO base + 0x408	Read/write	8	0x00	GPIOIBE	PrimeCell GPIO interrupt both edges register
GPIO base + 0x40C	Read/write	8	0x00	GPIOIEV	PrimeCell GPIO interrupt event register
GPIO base + 0x410	Read/write	8	0x00	GPIOIE	PrimeCell GPIO interrupt mask
GPIO base + 0x414	Read	8	0x00	GPIORIS	PrimeCell GPIO raw interrupt status
GPIO base + 0x418	Read	8	0x00	GPIOMIS	PrimeCell GPIO masked interrupt status
GPIO base + 0x41C	Write	8	0x00	GPIOIC	PrimeCell GPIO interrupt clear
GPIO base + 0x420	Read/write	8	0x00	GPIOAFSEL	PrimeCell GPIO mode control select
GPIO base + 0x424-0xFCC	-	-	-	-	Reserved for future use and test purposes
GPIO base + 0xFD0-0xFDC	-	-	-	-	Reserved for future ID expansion
GPIO base + 0xFE0	Read	8	0x61	GPIOPeriphID0	Peripheral identification register bits 7:0

Table 3-1 PrimeCell GPIO register summary (continued)

Address	Type	Width	Reset value	Name	Description
GPIO base + `0xFE4`	Read	8	`0x10`	GPIOPeriphID1	Peripheral identification register bits 15:8
GPIO base + `0xFE8`	Read	8	`0x04`	GPIOPeriphID2	Peripheral identification register bits 23:16
GPIO base + `0xFEC`	Read	8	`0x00`	GPIOPeriphID3	Peripheral identification register bits 31:24
GPIO base + `0xFF0`	Read	8	`0x0D`	GPIOPCellID0	PrimeCell identification register bits 7:0
GPIO base + `0xFF4`	Read	8	`0xF0`	GPIOPCellID1	PrimeCell identification register bits 15:8
GPIO base + `0xFF8`	Read	8	`0x05`	GPIOPCellID2	PrimeCell identification register bits 23:16
GPIO base + `0xFFC`	Read	8	`0xB1`	GPIOPCellID3	PrimeCell identification register bits 31:24

3.3 Register descriptions

The following PrimeCell GPIO registers are described in this section:

- *Data register, GPIODATA*
- *Data direction register, GPIODIR* on page 3-6
- *Interrupt sense register, GPIOIS* on page 3-6
- *Interrupt both-edges register, GPIOIBE* on page 3-7
- *Interrupt event register, GPIOIEV* on page 3-7
- *Raw interrupt status register, GPIORIS* on page 3-8
- *Masked interrupt status register, GPIOMIS* on page 3-8
- *Interrupt clear register, GPIOIC* on page 3-9
- *Mode control select register, GPIOAFSEL* on page 3-9
- *Peripheral identification registers, GPIOPeriphID0-3* on page 3-10
- *PrimeCell identification registers, GPIOPCellID0-3* on page 3-12.

3.3.1 Data register, GPIODATA

The GPIODATA register is the data register. In software control mode, values written in the GPIODATA register are transferred onto the **GPOUT** pins if the respective pins have been configured as outputs through the GPIODIR register.

In order to write to GPIODATA, the corresponding bits in the mask, resulting from the address bus, **PADDR[9:2]**, must be HIGH. Otherwise the bit values remain unchanged by the write.

Similarly, the values read from this register are determined for each bit, by the mask bit derived from the address used to access the data register, **PADDR[9:2]**. Bits that are 1 in the address mask cause the corresponding bits in GPIODATA to be read, and bits that are 0 in the address mask cause the corresponding bits in GPIODATA to be read as 0, regardless of their value.

A read from GPIODATA returns the last bit value written if the respective pins are configured as output, or it returns the value on the corresponding input **GPIN** bit when these are configured as inputs. All bits are cleared by a reset.

Table 3-2 shows the bit assignment of the GPIODATA register. For examples of address masking on reads and writes, see *Data register* on page 2-2.

Table 3-2 GPIODATA register

Bits	Name	Type	Function
7:0	Data register	Read/ write	Input data Output data

3.3.2 Data direction register, GPIODIR

The GPIODIR register is the data direction register. Bits set to HIGH in the GPIODIR configure corresponding pin to be an output. Clearing a bit configures the pin to be input. All bits are cleared by a reset. Therefore, the GPIO pins are input by default.

Table 3-3 shows the bit assignment of the GPIODIR register.

Table 3-3 GPIODIR register

Bits	Name	Type	Function
7:0	Data direction register	Read/ write	Bits set, pins output Bits cleared, pins output

3.3.3 Interrupt sense register, GPIOIS

The GPIOIS register is the interrupt sense register. Bits set to HIGH in GPIOIS configure the corresponding pins to detect levels. Clearing a bit configures the pin to detect edges. All bits are cleared by a reset.

Table 3-4 shows the bit assignment of the GPIOIS register.

Table 3-4 GPIOIS register

Bits	Name	Type	Function
7:0	Interrupt sense register	Read/ write	Bits clear, edge on corresponding pin is detected Bits set, level on corresponding pin is detected

3.3.4 Interrupt both-edges register, GPIOIBE

The GPIOIBE register is the interrupt both-edges register. When the corresponding bit in GPIOIS is set to detect edges, bits set to HIGH in GPIOIBE configure the corresponding pin to detect both rising and falling edges, regardless of the corresponding bit in the GPIOIEV (interrupt event register). Clearing a bit configures the pin to be controlled by GPIOIEV. All bits are cleared by a reset.

Table 3-5 shows the bit assignment of the GPIOIBE register.

Table 3-5 GPIOIBE register

Bits	Name	Type	Function
7:0	Interrupt both edges	Read/write	Bits set, both edges on corresponding pin trigger an interrupt.Bits cleared, interrupt generation event is controlled by GPIOIEV. Single edge, determined by corresponding bit in GPIOIEV register.

3.3.5 Interrupt event register, GPIOIEV

The GPIOIEV register is the interrupt event register. Bits set to HIGH in GPIOIEV configure the corresponding pin to detect rising edges or high levels, depending on the corresponding bit value in GPIOIS. Clearing a bit configures the pin to detect falling edges or low levels, depending on the corresponding bit value in GPIOIS. All bits are cleared by a reset.

Table 3-6 shows the bit assignment of the GPIOIEV register.

Table 3-6 GPIOIEV register

Bits	Name	Type	Function
7:0	Interrupt event register	Read/write	Bits set, rising edges, or high levels on corresponding pins trigger interrupts. Bits cleared, falling edges, or low levels on corresponding pin trigger interrupts.

3.3.6 Interrupt mask register, GPIOIE

The GPIOIE register is the interrupt mask register. Bits set to HIGH in GPIOIE allow the corresponding pins to trigger their individual interrupts and the combined GPIOINTR line. Clearing a bit disables interrupt triggering on that pin. All bits are cleared by a reset.

Table 3-7 shows the bit assignment of the GPIOIE register.

Table 3-7 GPIOIE register

Bits	Name	Type	Function
7:0	Interrupt mask register	Read/write	Bits set, corresponding pin is not masked. Bits cleared, corresponding pin interrupt is masked.

3.3.7 Raw interrupt status register, GPIORIS

The GPIORIS register is the raw interrupt status register. Bits read HIGH in GPIORIS reflect the status of interrupts trigger conditions detected (raw, prior to masking), indicating that all the requirements have been met, before they are finally allowed to trigger by GPIOIE. Bits read as zero indicate that corresponding input pins have not initiated an interrupt. This register is read only, and bits are cleared by a reset.

Table 3-8 shows the bit assignment of the GPIORIS register.

Table 3-8 GPIORIS register

Bits	Name	Type	Function
7:0	Raw interrupt status	Read	Reflect the status of interrupts trigger conditions detection on pins (raw, prior to masking). Bits set, requirements met by corresponding pins.Bits clear, requirements not met.

3.3.8 Masked interrupt status register, GPIOMIS

The GPIOMIS register is the masked interrupt status register. Bits read HIGH in GPIOMIS reflect the status of input lines triggering an interrupt. Bits read as LOW indicate that either no interrupt has been generated, or the interrupt is masked. GPIOMIS is the state of the interrupt after masking. This register is read-only, and all bits are cleared by a reset.

The contents of this register are made available externally through the intra-chip (or on-chip) **GPIOMIS[7:0]** signals.

Table 3-9 shows the bit assignment of the GPIOMIS register.

Table 3-9 GPIOMIS register

Bits	Name	Type	Function
7:0	Masked interrupt status	Read	Masked value of interrupt due to corresponding pin. Bits clear, PrimeCell GPIO line interrupt not active. Bits set, PrimeCell GPIO line asserting interrupt.

3.3.9 Interrupt clear register, GPIOIC

The GPIOIC register is the interrupt clear register. Writing a 1 to a bit in this register clears the corresponding interrupt edge detection logic register. Writing a 0 has no effect. This register is write-only and all bits are cleared by a reset.

Table 3-10 shows the bit assignment of the GPIOIC register.

Table 3-10 GPIOIC register

Bits	Name	Type	Function
7:0	Interrupt clear register	Write	Bit written as 1, clears edge detection logic. Bit written as 0, has no effect.

3.3.10 Mode control select register, GPIOAFSEL

The GPIOAFSEL register is the mode control select register. Writing a 1 to any bit in this register selects the hardware control for the corresponding PrimeCell GPIO line. All bits are cleared by a reset, therefore no PrimeCell GPIO line is set to hardware control by default.

Table 3-11 shows the bit assignment of the GPIOAFSEL register.

Table 3-11 GPIOAFSEL register

Bits	Name	Type	Function
7:0	Mode control select register	Read/ write	Bit set, enables hardware control mode. Bit cleared, enables software control mode.

3.3.11 Peripheral identification registers, GPIOPeriphID0-3

The GPIOPeriphID0-3 registers are four 8-bit registers, that span address locations `0xFE0` to `0xFEC`. The registers can conceptually be treated as a 32-bit register. The read only registers provide the following options of the peripheral:

PartNumber[11:0] This is used to identify the peripheral. The three digits product code `0x061` is used.

Designer ID[19:12] This is the identification of the designer. ARM Ltd is `0x41` (ASCII A).

Revision[23:20] This is the revision number of the peripheral. The revision number starts from 0.

Configuration[31:24] This is the configuration option of the peripheral. The configuration value is 0.

Figure 3-1 shows the bit assignment for the GPIOPeriphID0-3 registers.

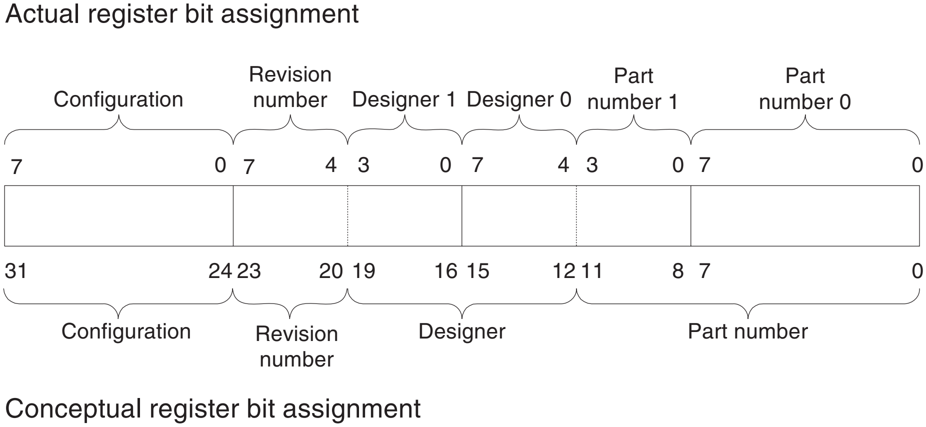


Figure 3-1 Peripheral identification register bit assignment

Note

When you design a systems memory map you must remember that the PrimeCell GPIO has a 4KB memory footprint. All memory accesses to the peripheral identification registers must be 32-bit, using the LDR and STR instructions.

The four, 8-bit peripheral identification registers are described in the following subsections:

- *GPIOPeriphID0 register*
- *GPIOPeriphID1 register*
- *GPIOPeriphID2 register* on page 3-12
- *GPIOPeriphID3 register* on page 3-12.

GPIOPeriphID0 register

The GPIOPeriphID0 register is hard coded and the fields within the register determine the reset value. Table 3-12 shows the bit assignment of the GPIOPeriphID0 register.

Table 3-12 GPIOPeriphID0 register

Bits	Name	Description
15:8	-	Reserved, read undefined must read as zeros
7:0	PartNumber0	These bits read back as `0x61`

GPIOPeriphID1 register

The GPIOPeriphID1 register is hard coded and the fields within the register determine the reset value. Table 3-13 shows the bit assignment of the GPIOPeriphID1 register.

Table 3-13 GPIOPeriphID1 register

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:4	Designer0	These bits read back as `0x1`
3:0	PartNumber1	These bits read back as `0x0`

GPIOPeriphID2 register

The GPIOPeriphID2 register is hard coded and the fields within the register determine the reset value. Table 3-14 shows the bit assignment of the GPIOPeriphID2 register.

Table 3-14 GPIOPeriphID2 register

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:4	Revision	These bits read back as `0x0`
3:0	Designer1	These bits read back as `0x4`

GPIOPeriphID3 register

The GPIOPeriphID3 register is hard coded and the fields within the register determine the reset value. Table 3-15 shows the bit assignment of the GPIOPeriphID3 register.

Table 3-15 GPIOPeriphID3 register

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:0	Configuration	These bits read back as `0x00`

3.3.12 PrimeCell identification registers, GPIOPCellID0-3

The GPIOPCellID0-3 registers are four 8-bit wide registers, that span address locations `0xFF0-0xFFC`. The registers can conceptually be treated as a 32-bit register. The register is used as a standard cross-peripheral identification system. The GPIOPCellID register is set to `0xB105F00D`. Figure 3-2 on page 3-13 shows the bit assignment for the GPIOPCellID0-3 registers.

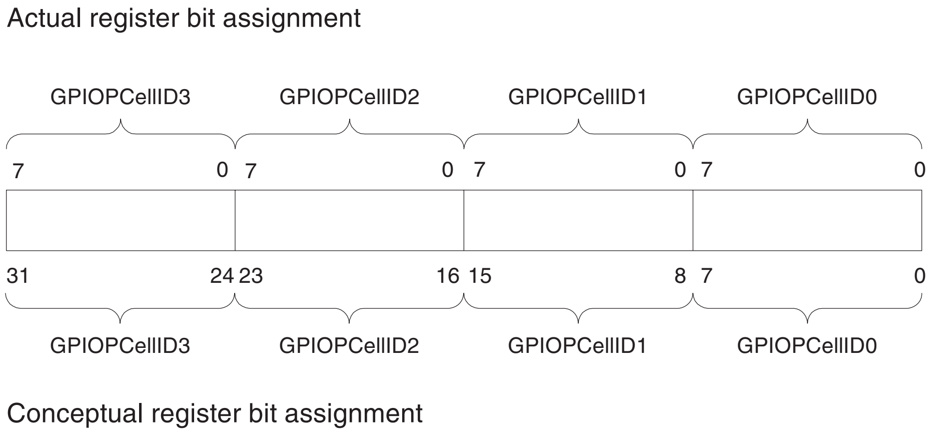


Figure 3-2 PrimeCell identification register bit assignment

The four 8-bit PrimeCell identification registers are described in the following subsections:

- *GPIOPCellID0 register*
- *GPIOPCellID1 register* on page 3-14
- *GPIOPCellID2 register* on page 3-14
- *GPIOPCellID3 register* on page 3-14.

GPIOPCellID0 register

The GPIOPCellID0 register is hard coded and the fields within the register determine the reset value. Table 3-16 shows the bit assignment of the GPIOPCellID0 register.

Table 3-16 GPIOPCellID0 register

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:0	GPIOPCellID0	These bits read back as `0x0D`

GPIOPCellID1 register

The GPIOPCellID1 register is hard coded and the fields within the register determine the reset value. Table 3-17 shows the bit assignment of the GPIOPCellID1 register.

Table 3-17 GPIOPCellID1 register read bits

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:0	GPIOPCellID1	These bits read back as `0xF0`

GPIOPCellID2 register

The GPIOPCellID2 register is hard coded and the fields within the register determine the reset value. Table 3-18 shows the bit assignment of the GPIOPCellID2 register.

Table 3-18 GPIOPCellID2 register read bits

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:0	GPIOPCellID2	These bits read back as `0x05`

GPIOPCellID3 register

The GPIOPCellID3 register is hard coded and the fields within the register determine the reset value. Table 3-19 shows the bit assignment of the GPIOPCellID3 register.

Table 3-19 GPIOPCellID3 register read bits

Bits	Name	Description
15:8	-	Reserved, read undefined, must read as zeros
7:0	GPIOPCellID3	These bits read back as `0xB1`

Chapter 4
Programmer's Model for Test

This chapter describes the additional logic for functional verification and production testing. It contains the following sections:

4.1 PrimeCell GPIO test harness overview

The additional logic for functional verification and integration vectors allows:

- capture of input signals to the block
- stimulation of the output signals.

The integration vectors provide a way of verifying that the PrimeCell GPIO is correctly wired into a system. This is done by separately testing three groups of signals:

AMBA signals

These are tested by checking the connections of all the address and data bits.

Primary input/output signals

These are tested using a simple trickbox that can demonstrate the correct connection of the input/output signals to external pads.

Intra-chip signals (such as interrupt sources)

The tests for these signals are system-specific, and enable you to write the necessary tests. Additional logic is implemented allowing you to read and write to each intra-chip input/output signal.

These test features are controlled by test registers. This allows you to test the PrimeCell GPIO in isolation from the rest of the system using only transfers from the AMBA APB.

Off-chip test vectors are supplied using a 32-bit parallel *External Bus Interface* (EBI) and converted to internal AMBA bus transfers. The application of test vectors is controlled through the *Test Interface Controller* (TIC) AMBA bus master module.

4.2 Scan testing

The PrimeCell GPIO has been designed to simplify:

- insertion of scan test cells
- use of *Automatic Test Pattern Generation* (ATPG).

This provides the recommended method of manufacturing test.

4.3 Integration test registers

The PrimeCell GPIO test registers are memory-mapped as shown in Table 4-1.

Table 4-1 Test registers memory map

Address	Type	Width	Reset value	Name	Description
GPIOBase + `0x600`	Read/write	1	`0x0`	GPIOITCR	Integration test control register.
GPIOBase + `0x604`	Read/write	8	`0x00`	GPIOITIP1	Integration test input read/set register.
GPIOBase + `0x608`	Read/write	8	`0x00`	GPIOITIP2	Integration test input read/set register.
GPIOBase + `0x60C`	Write	8	`0x00`	GPIOITOP1	Integration test output set register. It is possible to read the value of this register by setting GPIOITCR to HIGH and reading GPIOMIS.
GPIOBase + `0x610`	Read	1	`0x0`	GPIOITOP2	Integration test output read register. This bit reflects any of the bits within GPIOTOP1 being set when GPIOITCR is set to HIGH.
GPIOBase + `0x614`	Read/write	8	0x00	GPIOITOP3	Integration test output read/set register.

Each register shown in Table 4-1 is described in the following sections:

- *Integration test control register, GPIOITCR* on page 4-5
- *Integration test input read/set register, GPIOITIP1* on page 4-5
- *Integration test input read/set register, GPIOITIP2* on page 4-5
- *Integration test output set register, GPIOITOP1* on page 4-6
- *Integration test output read register, GPIOITOP2* on page 4-6
- *Integration test output read/set register, GPIOITOP3* on page 4-7.

4.3.1 Integration test control register, GPIOITCR

GPIOITCR is the integration test control register. This test register controls operation of the PrimeCell GPIO under integration test conditions. Table 4-2 shows the bit assignments for the GPIOITCR.

Table 4-2 GPIOITCR register bits

Bits	Name	Description
7:1	-	Reserved, unpredictable when read.
0	ITEN	Integration test enable. When this bit is 1, the PrimeCell GPIO is placed in integration test mode, otherwise it is in normal mode.

4.3.2 Integration test input read/set register, GPIOITIP1

GPIOITIP1 is the integration test input read/set register. It is a a read/write register. In integration test mode it allows intra-chip input signals **GPAFOUT[7:0]** to be both written to and read from. Table 4-3 shows the bit assignments for the GPIOITIP1.

Table 4-3 GPIOITIP1 register bits

Bits	Name	Description
7:0	GPAFOUT	Writes specify the value to be driven on the intra-chip input, **GPAFOUT[7:0]**, in the integration test mode. Reads return the value of **GPAFOUT[7:0]** at the output of the test multiplexor.

4.3.3 Integration test input read/set register, GPIOITIP2

GPIOITIP2 is the integration test input read/set register. It is a a read/write register. In integration test mode it allows intra-chip input signals **nGPAFEN[7:0]** to be both written to and read from. Table 4-4 shows the bit assignments for the GPIOITIP2.

Table 4-4 GPIOITIP2 register bits

Bits	Name	Description
7:0	nGPAFEN	Writes specify the value to be driven on the intra-chip input, **nGPAFEN[7:0]**, in the integration test mode. Reads return the value of **nGPAFEN[7:0]** at the output of the test multiplexor.

4.3.4 Integration test output set register, GPIOITOP1

GPIOITOP1 is the integration test output set register for **GPIOMIS[7:0]**. In integration test mode it allows intra-chip outputs **GPIOMIS[7:0]** to be written to. Table 4-5 shows the bit assignments for the GPIOITOP1.

Table 4-5 GPIOITOP1 register bits

Bits	Name	Description
7:0	GPIOMIS	Intra-chip output. Writes specify the value to be driven on the **GPIOMIS[7:0]** lines in the integration test mode. Reads must be performed in GPIOMIS.

4.3.5 Integration test output read register, GPIOITOP2

GPIOITOP2 is the integration test output read register for GPIOINTR. Table 4-6 shows the bit assignments for the GPIOITOP2.

Table 4-6 GPIOITOP2 register bits

Bits	Name	Description
7:1	-	Reserved, unpredictable when read
0	GPIOITOP2	Reads return the value of GPIOINTR

Note

The intra-chip output **GPIOINTR** is read only. In order to drive this signal, writes to GPIOMIS must be performed using the integration test output, set-only register GPIOITOP1. GPIOITOP1 is write-only. The status of GPIOITOP1 is accessible in integration test mode by reading from the GPIOMIS register.

4.3.6 Integration test output read/set register, GPIOITOP3

GPIOITOP3 is the integration test output set register for **GPAFIN[7:0]**. In integration test mode it allows outputs to be both written to, and read from. Table 4-7 shows the bit assignments for the GPIOITOP3.

Table 4-7 GPIOITOP3 register bits

Bits	Name	Description
7:0	GPIOITOP3	Intra-chip output. Writes specify the value to be driven on the **GPAFIN[7:0]** lines in the integration test mode. Reads must be performed of GPIOAFIN at the output of the test multiplexor.

4.4 Integration testing of block inputs

The following sections describe the integration testing for the block inputs:

- *Intra-chip inputs*
- *Primary inputs* on page 4-10.

4.4.1 Intra-chip inputs

Figure 4-1 on page 4-9 explains the implementation details of the input integration test harness. The ITEN bit is used as the control bit for the multiplexor, which is used in the read path of the **nGPAFEN[7:0]** and **GPAFOUT** intra-chip inputs. If the ITEN control bit is asserted, the stored values in GPIOTIP2 and GPIOTIP1 are driven on the **nGPAFEN[7:0]** and **GPAFOUT[7:0]** internal line respectively. Deasserted, the **nGPAFEN[7:0]** and **GPAFOUT** intra-chip inputs are routed as the internal **nGPAFEN[7:0]** and **GPAFOUT** inputs respectively.

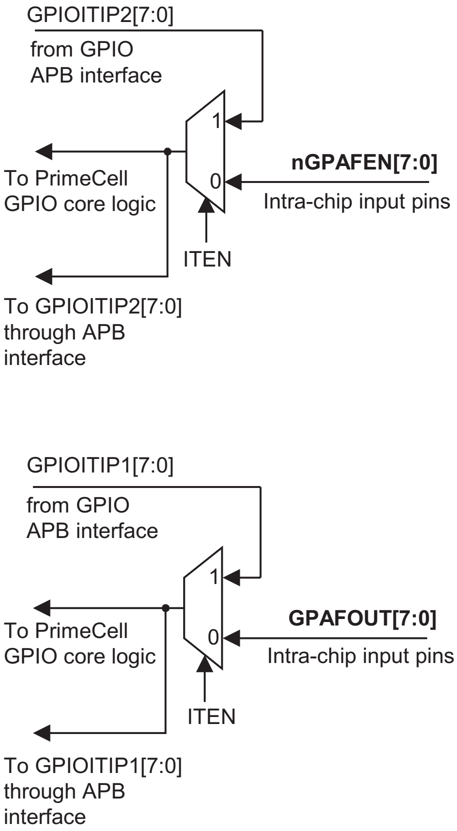


Figure 4-1 Input integration test harness

In order to run integration tests with the PrimeCell GPIO:

- Write a 1 to the ITEN bit in the integration test control register GPIOTCR. This selects the test path from the GPIOITIP1 register to **GPAFOUT,** and from the GPIOITIP2 register to the **GPAFEN** internal signals.
- Write a 1 and then a 0 to each of the GPIOTIP1[7:0] and GPIOTIP2[7:0] register bits, and read the same register bits to ensure that the value written is read out.

When you run integration tests with the PrimeCell GPIO as part of an integrated system:

- Write a 0 to the ITEN bit in the control register. This selects the normal path from the external **nGPAFEN** pins to the internal **nGPAFEN** signals, and the path from the external **GPAFOUT** pins to the internal **GPAFOUT** signals.

- Write a 1 and then a 0 to toggle the **GPAFEN[7:0]** and **GPAFOUT[7:0]** intra-chip input signals connection using the device connected to the hardware control lines. Read from the GPIOITIP2[7:0] and GPIOITIP1[7:0] register bits to verify that the value in the device connected to the hardware control lines is read out through the PrimeCell GPIO.

4.4.2 Primary inputs

The primary inputs are tested using the integration vector trickbox by looping back primary input **GPIN[7:0]** as XOR **nGPEN[7:0]** and **GPOUT[7:0]** pairs.

Write a 1 to the ITEN bit in the control register. Select hardware control writing `0xFF` in the GPIOAFSEL register. All 1s and 0s, and a walking 1 are driven onto the primary output lines **nGPEN[7:0]** and **GPOUT[7:0]** through the GPIOITIP2[7:0] and GPIOITIP1[7:0] registers respectively. The result of the logical XOR operation performed in the integration vector trickbox can be read back through the GPIODATA[7:0] register.

Note

When you are working with GPIODATA remember the address masking characteristics of this register. Using address offset + `0x3FC` all bits in GPIODATA can be read or written. Data entering **GPIN[7:0]** is considered potentially asynchronous to **PCLK**. Therefore, before GPIN data can be read from GPIODATA a time period of twice **PCLK** must be allowed.

4.5 Integration testing of block outputs

The following sections describe the integration testing for the block outputs:

- *Intra-chip outputs*
- *Primary outputs* on page 4-12.

4.5.1 Intra-chip outputs

Use this test for the following outputs:

- **GPAFIN[7:0]**
- **GPIOMIS[7:0]**
- **GPIOINTR**.

When you run integration tests with the PrimeCell GPIO in a standalone test setup:

- Write a 1 to the ITEN bit in the integration test control register. This selects the test path from the GPIOITOP1[7:0] register bits to the intra-chip output signals **GPIOMIS[7:0]**. **GPIOINTR** is the result of a logical OR over the GPIOMIS[7:0] bits. The test path from GPIOITOP3[7:0] register bits to the intra-chip output signals **GPAFIN[7:0]** is also selected.

- Write a 1 and then a 0 to the GPIOITOP1[7:0] register bits, and read the same register bits in the GPIOMIS[7:0] register to verify that the value written is read out. **GPIOINTR** can be read to check its status independently of GPIOITCR.

- Write a 1s and 0s to the GPIOITOP3[7:0] register bits, and read the same register bits to verify that the value written is read out.

In order to run integration tests with the PrimeCell GPIO as part of an integrated system:

- Write a 1 to the ITEN bit in the control register. This selects the test path from the internal GPIOITOP1[7:0] lines to the external **GPIOMIS[7:0]** intra-chip output signals.

- The test path from GPIOITOP3[7:0] register bits to the intra-chip output signals **GPAFIN[7:0]** is also selected.

- Write a 1 and then a 0 to the GPIOITOP1[7:0] register bits to toggle the signal connections between the interrupt controller and the PrimeCell GPIO. Read from the internal test registers of the interrupt controller to verify that the value written into the GPIOITOP1[7:0] register is read out through the PrimeCell GPIO.

- Write a 1s and 0s to the GPIOITOP3[7:0] register bits, and read the same register bits from the internal register of the interrupt controller to toggle the signal connections between to verify that the value written into the GPIOITOP3[7:0] register is read out through the PrimeCell GPIO.

Figure 4-2 explains the implementation details of the output integration test harness for intra-chip outputs.

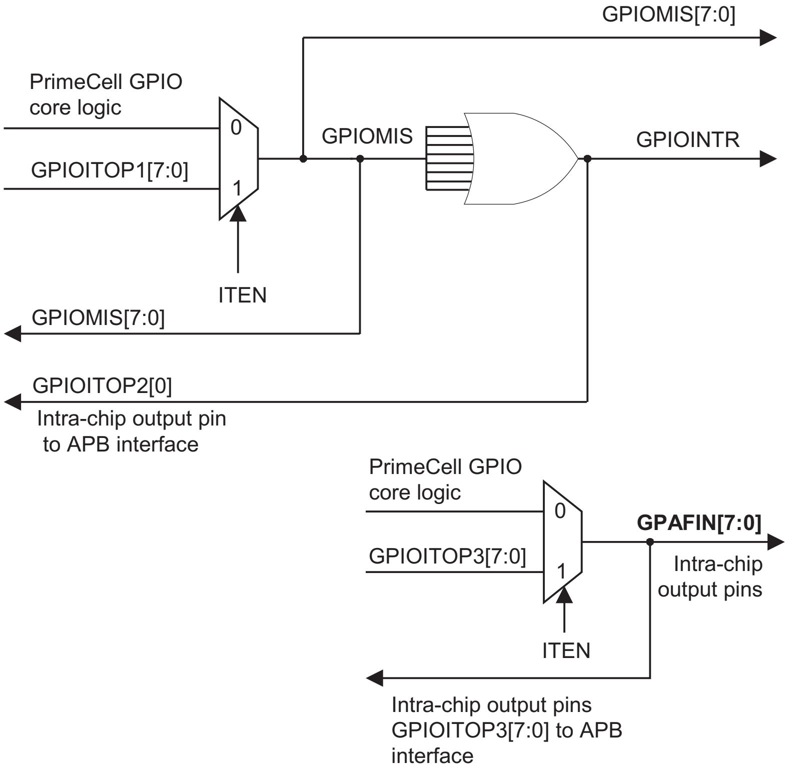

GPIOMIS[7:0]
PrimeCell GPIO core logic
0
GPIOITOP1[7:0]
1
GPIOMIS
GPIOINTR
ITEN
GPIOMIS[7:0]
GPIOITOP2[0]
Intra-chip output pin to APB interface
PrimeCell GPIO core logic
0
GPIOITOP3[7:0]
1
GPAFIN[7:0]
Intra-chip output pins
ITEN
Intra-chip output pins GPIOITOP3[7:0] to APB interface

Figure 4-2 Output integration test harness, intra-chip outputs

4.5.2 Primary outputs

Integration testing of primary outputs and primary inputs is carried out using the integration vector trickbox. Use this test for the following outputs:

- **nGPEN[7:0]**
- **GPOUT[7:0]**.

The primary output pins (listed above) are looped back to the primary input pins **GPIN[7:0]** as an XOR logical operation of primary outputs **nGPEN[7:0]** and **GPOUT[7:0]** through the integration vector trickbox.

Verify the primary input and output pin connections as follows:

- Primary outputs **nGPEN** and **GPOUT** can be accessed through the GPIOITIP2[7:0] and GPIOITIP1[7:0] registers respectively by selecting integration test enable and hardware control mode. Different data patterns are written to the output pins using the GPIOITIP2 and GPIOITIP1 registers.
- The looped back data is read back through the GPIODATA register.

——— **Note** ———

Data entering **GPIN[7:0]** is considered potentially asynchronous to **PCLK**. Therefore, before **GPIN[7:0]** can be read from GPIODATA two **PCLK** rising edges must be applied.

———

Figure 4-3 explains the implementation details of the output integration test harness in the case of primary outputs.

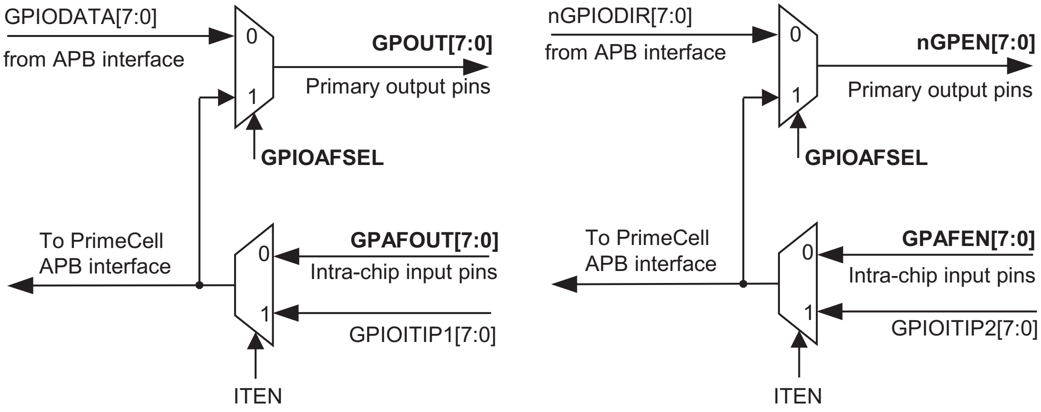


Figure 4-3 Output integration test harness, primary outputs

4.6 Integration test summary

Table 4-8 summarizes the integration test strategy for all PrimeCell GPIO pins.

Table 4-8 PrimeCell GPIO integration test strategy

Name	Type	Source/ destination	Test strategy
PRESETn	Input	Reset controller	Not tested using integration test vectors
PADDR [11:2]	Input	APB	Register read/write
PCLK	Input	APB	Register read/write
PENABLE	Input	APB	Register read/write
PRDATA [7:0]	Output	APB	Register read/write
PSEL	Input	APB	Register read/write
PWDATA [7:0]	Input	APB	Register read/write
PWRITE	Input	APB	Register read/write
GPIOMIS[7:0]	Output	Interrupt controller	Using GPIOITOP1 and GPIOITOP2 registers
GPIOINTR	Output	Interrupt controller	Using GPIOITOP1 and GPIOITOP2 registers
nGPEN[7:0]	Output	PAD	Using integration vector trickbox and GPIOITIP2 and GPIOAFSEL registers
GPOUT[7:0]	Output	PAD	Using integration vector trickbox and GPIOITIP1 and GPIOAFSEL registers
GPIN[7:0]	Input	PAD	Using integration vector trickbox and GPIODATA and GPIOAFSEL registers[a]
nGPAFEN[7:0]	Output	Generic	Using GPIOITIP2 register
GPAFOUT[7:0]	Output	Generic	Using GPIOITIP1 register
GPAFIN[7:0]	Input	Generic	Using GPIOIT0P3 register

[a] Data entering **GPIN[7:0]** is considered potentially asynchronous to **PCLK**. Therefore, before **GPIN** can be read from GPIODATA two **PCLK** rising edges must be applied.

Chapter 5
Signal Descriptions

This appendix describes the signals that interface with the ARM PrimeCell GPIO (PL061) block. It contains the following sections:

- *AMBA APB signals* on page 5-2
- *On-chip signals* on page 5-3
- *Signals to pads* on page 5-4.

5.1 AMBA APB signals

The PrimeCell GPIO module is connected to the AMBA APB as a bus slave. AMBA APB signals have a **P** prefix and are active HIGH. Active LOW signals contain a lower case **n**. The AMBA APB signals are described in Table 5-1.

Table 5-1 AMBA APB signal descriptions

Name	Type	Source/ destination	Description
PRESETn	Input	Reset controller	Bus reset signal, active LOW.
PADDR [11:2]	Input	APB	Subset of AMBA APB address bus.
PCLK	Input	APB	AMBA APB clock, used to time all bus transfers.
PENABLE	Input	APB	AMBA APB enable signal. **PENABLE** is asserted HIGH for one cycle of **PCLK** to enable a bus transfer.
PRDATA [7:0]	Output	APB	Subset of unidirectional AMBA APB read data bus.
PSEL	Input	APB	PrimeCell GPIO select signal from decoder. When set to 1 this signal indicates the slave device is selected by the AMBA APB bridge, and that a data transfer is required.
PWDATA [7:0]	Input	APB	Subset of unidirectional AMBA APB write data bus.
PWRITE	Input	APB	AMBA APB transfer direction signal, indicates a write access when HIGH, read access when LOW.

5.2 On-chip signals

The on-chip signals required in addition to the AMBA APB signals are shown in Table 5-2.

The signals marked as generic in the table can be connected to any peripheral, and the pads can be shared between the GPIO and something else.

Table 5-2 On-chip signal descriptions

Name	Type	Source/ destination	Description
nGPAFEN[7:0]	Input	Generic	Hardware control output enable, active LOW. If not utilized these pins must be tied HIGH.
GPAFOUT[7:0]	Input	Generic	Hardware control data input. If not utilized these pins can be tied LOW or HIGH.
GPAFIN[7:0]	Output	Generic	Hardware control data output.
GPIOMIS[7:0]	Output	Interrupt controller	Interrupt signals to the interrupt module. A HIGH on one of the lines indicates that a valid match has occurred between the interrupt set of registers and the signals on the corresponding pads.
GPIOINTR	Output	Interrupt controller	Combined OR version of GPIOMIS. When HIGH, this signal indicates that a valid match has occurred between any of the interrupt sets of registers and signals on the corresponding pad.
SCANENABLE	Input	Scan controller	Scan enable.
SCANINPCLK	Input	Scan controller	Scan data input for PCLK domain.
SCANOUTPCLK	Output	Scan controller	Scan data output for PCLK domain.

5.3 Signals to pads

Table 5-3 describes the signals from the PrimeCell GPIO to input/output pads of the chip. You must make proper use of the peripheral pins to meet the exact interface requirements.

Table 5-3 Pad signal descriptions

Name	Type	Pad type	Description
nGPEN[7:0]	Output	PAD	PrimeCell GPIO output pad enable signal, active LOW. This pin is driven by the GPIODIR register when software control mode is selected, or through auxiliary pins when hardware control mode is selected.
GPOUT[7:0]	Output	PAD	PrimeCell GPIO output pad data signal driver. This pin is driven by the GPIODATA register when software control mode is selected, or through auxiliary pins when hardware control mode is selected.
GPIN[7:0]	Input	PAD	PrimeCell GPIO input data from pad. Values on these pins can be read through the APB interface path when software control mode is selected, or transferred to auxiliary pins when hardware control mode is selected. It is also possible to read the values through the APB interface when in hardware control mode.

Index