Features

- **Compatible with an Embedded 32-bit ARM7TDMI™ Processor**
- **Up to 32 Programmable I/O Lines**
- **Interrupt Generation on Event**
- **Glitch Filter**
- **Fully Scan Testable (up to 98% Fault Coverage)**
- **Can be Directly Connected to the Atmel Implementation of the AMBA™ Peripheral Bus (APB) of the ARM7TDMI Microcontroller**
- **Multi-driver (Open Drain) Option**
- **Certain Options "Parametrizable" on Request:**
- **Number of Programmable Lines**
- **Glitch Filter Option**
- **Multi-driver (Open Drain) Option**
- **Reset State of PIO Status and Glitch Filter Status**

Description

The Parallel Input/Output 1 (PIO1) 32-bit embedded core peripheral features 32 fully-programmable input/output lines, each of which may be dedicated as general purpose I/O or be multiplexed with a signal generated by another embedded peripheral, in order to optimize the use of available package pins in the overall system-on-chip design. The PIO1 controller provides a bit-maskable event driven internal interrupt signal.

The PIO1 and other analog and digital modular embedded peripherals, together with a choice of microprocessor and DSP cores, on-chip RAM, ROM, EEPROM and Flash memory, as well as special purpose analog or digital user-developed blocks, allow rapid and cost-effective design and implementation of an optimized system-on-chip. The large range of functional blocks offers a realistic and efficient design pathway to system-level integration (SLI).

The PIO1 is bus-compatible with the ARM7TDMI 32-bit microcontroller core. It can also be used with other 32-bit MCU or DSP cores.

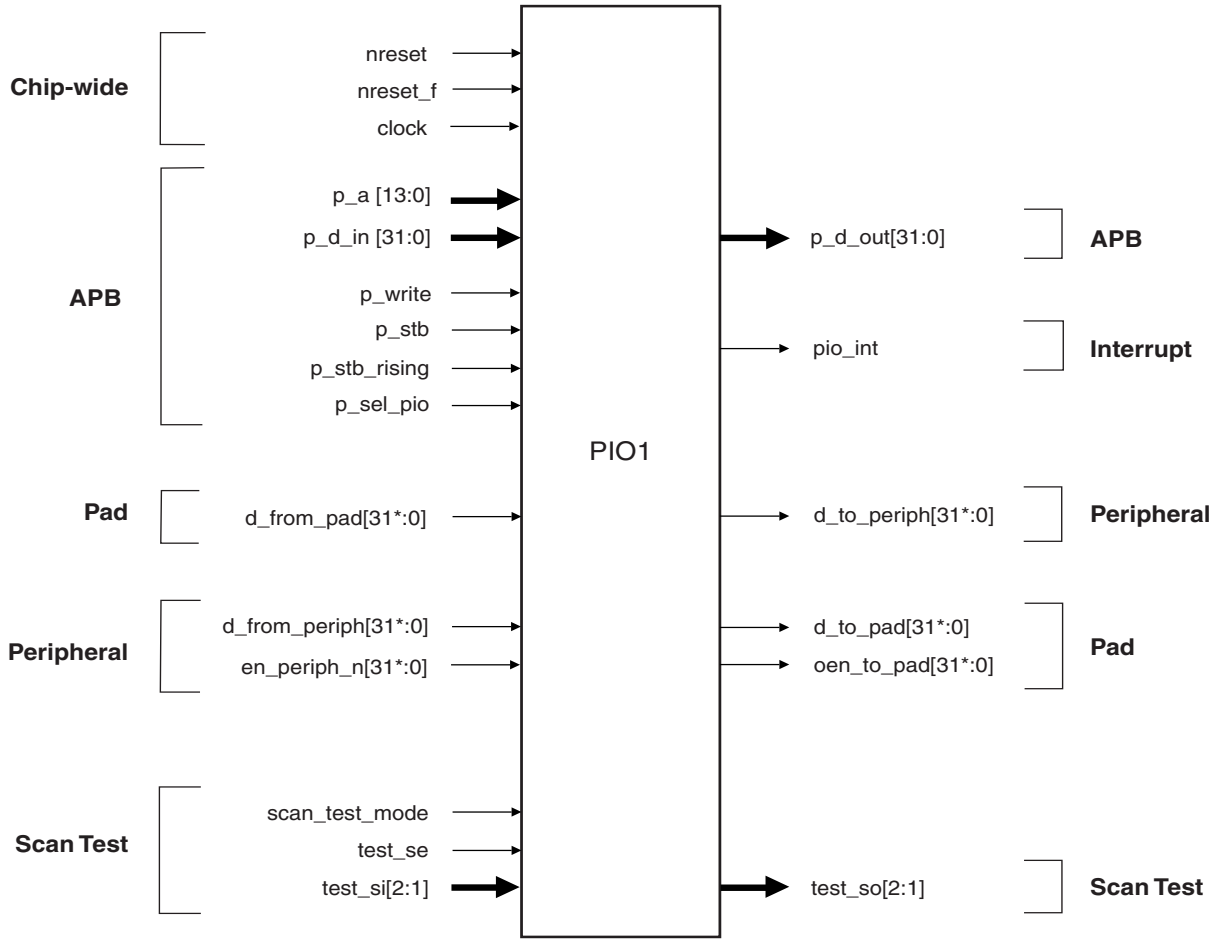
The PIO1 is supplied with comprehensive test vector sets. Atmel's proprietary foundry interface tools ensure a smooth transition from design to fabrication.

32-bit Embedded Core Peripheral

Parallel Input/Output 1 (PIO1)

Rev. 1321C–03/01

Figure 1. PIO1 Terminal Connections



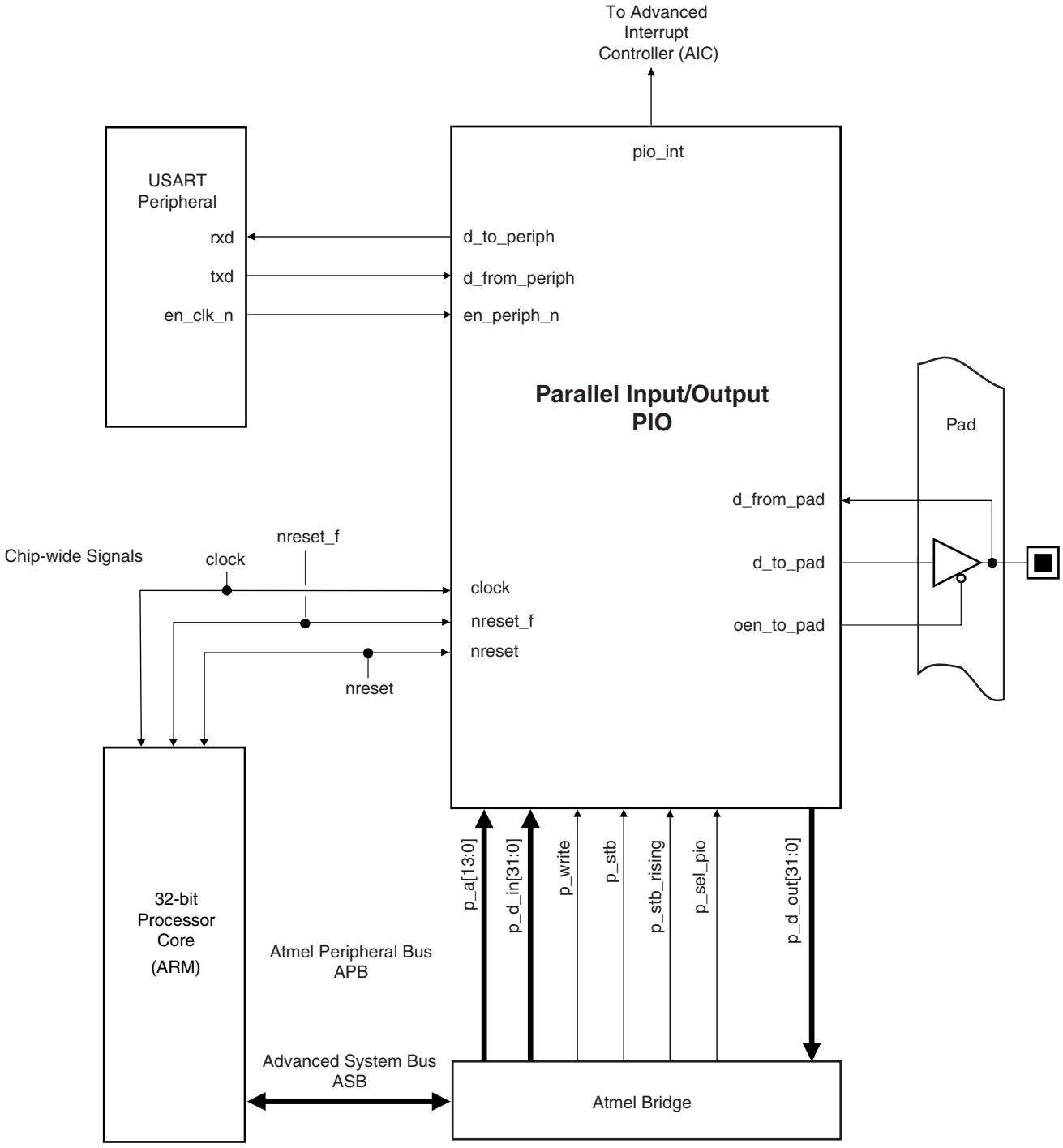
Note: *Depends on "On Request" parameters. For example, if 20 lines are requested, it will be [19:0].

Table 1. PIO Terminal Description

Name	Function	Type	Active Level	Comments
Chip-wide				
nreset	System reset	Input	Low	Resets all counters and signals Clocked on rising edge of clock
nreset_f	System reset	Input	Low	Resets all counters and signals Clocked on falling edge of clock
clock	System clock	Input	–	System clock
Atmel Peripheral Bus (APB)				
p_a [13:0]	Address bus	Input	–	The address takes into account the 2 LSBs [1:0], but the PIO1 macrocell does not decode these bits
p_d_in [31:0]	Input data bus	Input	–	From host (bridge)
p_d_out [31:0]	Output data bus	Output	–	To host (bridge)
p_write	Write enable	Input	High	From host (bridge)
p_stb	Peripheral strobe	Input	High	From host (bridge)
p_stb_rising	User interface clock signal	Input	–	From host (bridge)–Clock for all DFFs controlling the configuration registers
p_sel_pio	Selects the PIO1 block	Input	High	From host (bridge)
Pad				
d_from_pad [31[1]:0]	Pad input data	Input	–	Data from an I/O pad
d_to_pad [31[1]:0]	Pad output data	Output	–	Data to an I/O pad
oen_to_pad [31[1]:0]	Pad output enable	Output	Low	Output enable for a bidirectional pad
Peripheral				
d_from_periph [31[1]:0]	Peripheral data input	Input	–	Data from an on-chip peripheral
d_to_periph [31[1]:0]	Peripheral data output	Output	–	Data to an on-chip peripheral
en_periph_n [31[1]:0]	Peripheral data enable	Input	Low	Enables data from an on-chip peripheral to a pad when a peripheral connection is enabled
Interrupt				
pio_int	PIO1 generated Interrupt	Output	High	Any I/O line may be programmed to generate an event driven interrupt
Test Scan				
scan_test_mode	Clock selection for test purposes	Input	High	
test_se	Scan test enable	Input	High/Low	Scan shift/scan capture
test_si[2:1]	Scan test input	Input	High	Entry of scan chain
test_so [2:1]	Scan test output	Output	–	Ouput of scan chain

Note: 1. Depends on “On Request” parameters. For example, if 20 lines are requested, it will be [19:0].

Figure 2. Interconnecting the PIO1 in an ARM®-based Microcontroller System-on-chip (Example)



Functional Description

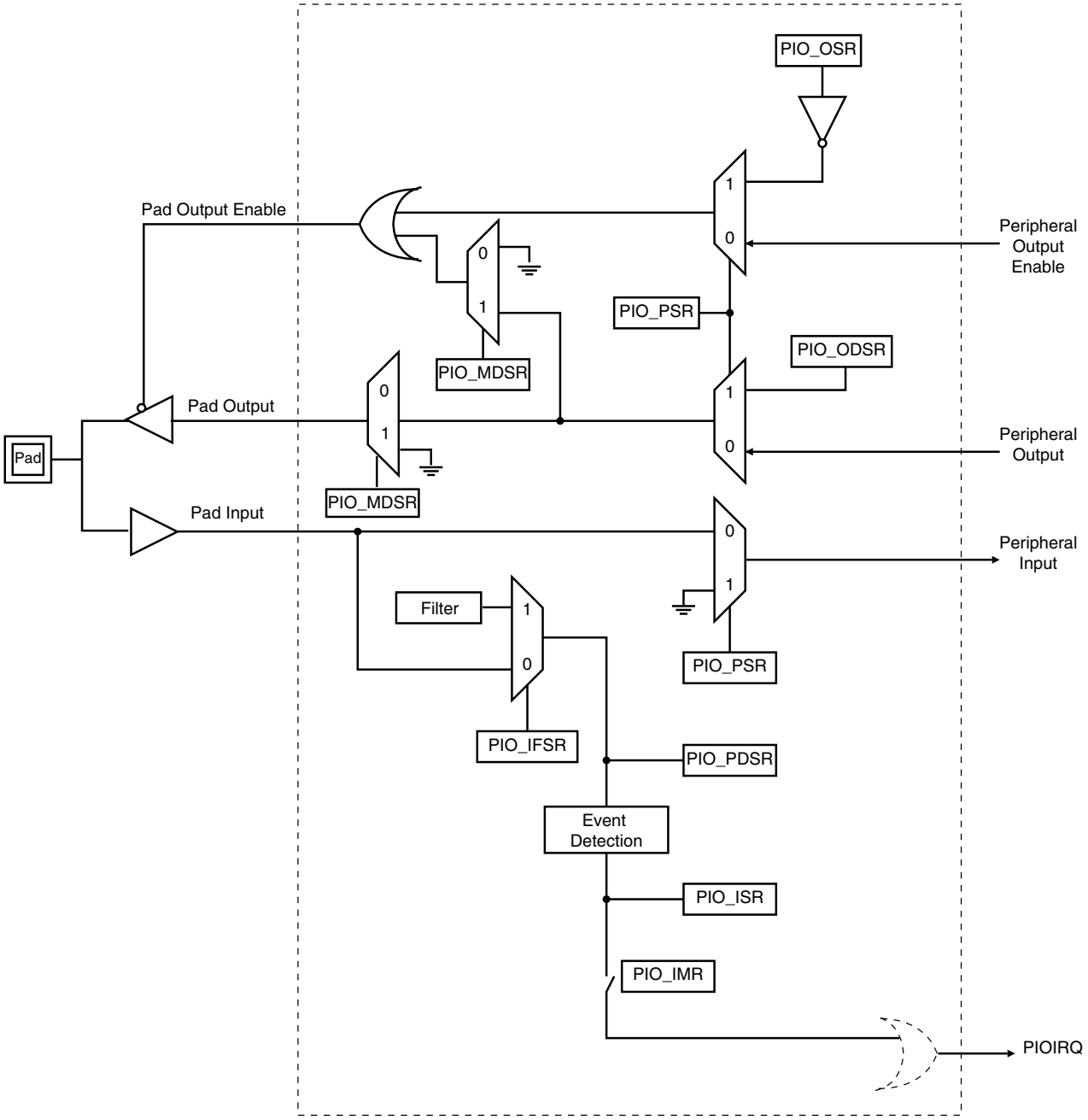
The 32-bit PIO1 peripheral is fully compatible with an embedded ARM7TDMI processor.

The PIO peripheral features 32 fully-programmable I/O lines, each of which may be multiplexed with an on-chip peripheral signal.

The device can also provide a bit-maskable event driven on-chip interrupt signal.

The PIO1 peripheral is fully-controllable via five sets of three 32-bit registers; pin data and interrupt source conditions are available to user software via two 32-bit registers.Figure 3 illustrates PIO1 functionality and the effect of register programming as described in the following sections

Figure 3. .PIO1 Control and Status Register Architecture



Scan Test Configuration

Fault coverage is maximized when all non-scan inputs can be controlled and when all non-scan outputs can be monitored. In order to achieve this, it is preferable that the ATPG vectors be generated on the entire circuit which includes the PIO1 embedded peripheral (top-level); alternatively, all of the inputs and outputs of the PIO1 should be made accessible so that ATPG vectors can be applied to all terminals.

Control Registers

Each individual I/O channel is associated with a bit in the various 32-bit PIO user interface registers (control and status registers) which are listed in Table 2. The interrelationship of functionality with the various status and control registers is also illustrated in Figure 3. If a parallel I/O line is not defined, writing to the corresponding bits has no effect; reading the bit will return zero.

Functionality of the PIO1 is governed by four 32-bit read-only status registers and a read-only interrupt mask register. These registers are each controlled by a pair of user software accessible write-only control registers. Two further read-only 32-bit status registers allow pin data and interrupt source conditions to be monitored by user software.

When writing to a Control Register, only a logic 1 affects the related bit: thus, the user software writes a 1 to the Enable Control Register to enable the desired function, and it writes a 1 to the associated Disable Control Register to disable the function. Writing a logic 0 to a Control Register has no effect. Any attempt at writing to a read-only register has no effect. Any attempt at reading a write-only Control Register returns undefined data.

Bidirectional Multiplexed I/O

After reset, all channels are connected to the PIO1 controller as I/O and are in input mode (unless the reset value of the related status registers is programmed otherwise during system design; this may be the case if other on-chip peripherals are connected to the outside world via the PIO1).

I/O lines may be multiplexed with the input and output signals of another on-chip peripheral. The state of each bit of the PIO Status Register (PIO_PSR) determines whether the related channel is connected to an on-chip peripheral or as processor-addressable Parallel I/O. The PIO Status Register (PIO_PSR) is controlled by writing a logic 1 to the relevant bit of the PIO Enable (PIO_PER) or to the PIO Disable (PIO_PDR) Control Registers. Writing a 0 to a Control Register has no effect.

When the PIO1 is selected, the peripheral input line is connected to zero.

If a pin is a general-purpose parallel I/O pin (not multiplexed with a peripheral), PIO_PER and PIO_PDR has no effect and reading PIO_PSR returns a logic 1 for the bits corresponding to these pins.

Output Enabling

Each channel is effectively bidirectional. A channel output may be disabled by suitably programming the Output Status Register (PIO_OSR) via the PIO_ODR (Output Disable) write-only Control Register. Conversely, the user can enable a channel as an output by writing a logic 1 to the relevant bit of the PIO_OER (Output Enable) Control Register. The status of the various channels is monitored by reading the Output Status Register (PIO_OSR).

Output enabling/disabling may only be controlled by user software when the channel has been configured as parallel I/O (via PIO Status Register, PIO_PSR); when the channel is configured as peripheral-driven, the pad output is enabled by a logic 0 state on the Peripheral Data Enable (EN_PERIPH_N) terminal. This signal originates from another on-chip peripheral.

Pin Signal Levels

Each pin may be actively driven high or low, or its level may be determined by off-chip circuitry. The actual logic level on the pin is governed by the following conditions:

- If a channel is driven by the PIO1 controller and is defined as an output (see "Output Enabling" above), its pin level is governed by the state of the Output Data Status Register (PIO_ODSR) which may be controlled by writing a logic 1 to the Set Output Data Control Register (PIO_SODR) or to the Clear Output Data Control Register (PIO_CODR). The programmed value may be read from the Output Data Status Register (PIO_ODSR).
- If a channel is driven by the PIO1 controller, but is not enabled as an output, the pin level will be determined by the external off-chip circuit, irrespective of the Output Data Status Register contents.
- If a channel is driven by an on-chip peripheral and the PIO Status Register (PIO_PSR) is programmed to enable the peripheral connection, its level will be defined by the peripheral.

In all cases, the actual level on a pin may be monitored by reading the corresponding bit of the Pin Data Status Register (PIO_PDSR).

Interrupts

Each I/O channel can be programmed to generate an interrupt when a voltage level change occurs on the related external pin.

Each individual channel may have its interrupt generation enabled or disabled according to the logic condition of the related bit in the Interrupt Mask Register (PIO_IMR), which is controlled by writing a logic 1 to the relevant bit of the Interrupt Enable (PIO_IER) or to the Interrupt Disable (PIO_IDR) Control Register, in the same fashion as for the other Control Registers described above.

When a change in level occurs, the corresponding bit in the Interrupt Status Register (PIO_ISR) register is set, whether the pin is used for a peripheral function or for processor addressed PIO, and whether the channel is working in input or in output mode. When the Interrupt Status Register (PIO_ISR) is read by user software, its contents are automatically cleared.

If the corresponding interrupt is enabled in the Interrupt Mask Register (PIO_IMR), the relevant interrupt signal is generated. The interrupt signals from all 32 channels are gated together to generate a single PIO Interrupt signal (PIO_INT) that is available for routing to the system interrupt controller.

Glitch Filtering

Optional glitch filtering is available for each channel. It is governed by the state of the Input Filter Status Register (PIO_IFSR) which is controlled by writing a logic 1 to the relevant bit of the Input Filter Enable (PIO_IFER) or to the Input Filter Disable (PIO_IFDR) Control Registers, in the same fashion as for the other control registers described above.

Input glitch filtering may be selected, whether the pin is used for a peripheral function or for processor addressed parallel input.

When the filter is activated, a glitch with a duration of less than 1/2 clock cycle is automatically rejected, while a pulse with a duration of 1 clock cycle or more is accepted. For pulse durations between 1/2 clock cycle and 1 clock cycle the pulse may or may not be taken into account, depending on the precise timing of its occurrence. Thus for a pulse to be visible it must exceed 1 clock cycle, whereas for a glitch to be reliably filtered out, its duration must not exceed 1/2 clock cycle.

Multi-driver (Open Drain)

Each I/O can be programmed for multi-driver option. This means that the I/O is configured as open drain (can only drive a low level) in order to support external drivers on the same pin. An external pull-up is necessary to guarantee a logic level of one when the pin is not being driven.

Registers PIO_MDER (Multi-driver Enable) and PIO_MDDR (Multi-driver Disable) control this option. Multi-driver can be selected whether the I/O pin is controlled by the PIO controller or the peripheral. PIO_MDSR (Multi-driver Status) indicates which pins are configured to support external drivers.

PIO1 User Interface

Table 2. PIO1 Controller Memory Map

Offset	Control Registers	Status Registers	Name	Access	Reset State
0x00	PIO Enable Register		PIO_PER	Write-only	–
0x04	PIO Disable Register		PIO_PDR	Write-only	–
0x08		PIO Status Register	PIO_PSR	Read-only	FFFFFFF (See Notes 1 and 4)
0x0C	**Reserved**				
0x10	Output Enable Register		PIO_OER	Write-only	–
0x14	Output Disable Register		PIO_ODR	Write-only	–
0x18		Output Status Register	PIO_OSR	Read-only	0
0x1C	**Reserved**				
0x20	Glitch Input Filter Enable Register		PIO_IFER	Write-only	–
0x24	Glitch Input Filter Disable Register		PIO_IFDR	Write-only	–
0x28		Glitch Input Filter Status Register	PIO_IFSR	Read-only	(See Note 4)
0x2C	**Reserved**				
0x30	Set Output Data Register		PIO_SODR	Write-only	–
0x34	Clear Output Data Register		PIO_CODR	Write-only	–
0x38		Output Data Status Register	PIO_ODSR	Read-only	0
0x3C		Pin Data Status Register	PIO_PDSR	Read-only	(see Note 2)
0x40	Interrupt Enable Register		PIO_IER	Write-only	–
0x44	Interrupt Disable Register		PIO_IDR	Write-only	–
0x48		Interrupt Mask Register	PIO_IMR	Read-only	0
0x4C		Interrupt Status Register	PIO_ISR	Read-only	0 (see Note 3)
0x50	Multi-driver Enable Register		PIO_MDER	Write-only	-
0x54	Multi-driver Disable Register		PIO_MDDR	Write-only	-
0x58		Multi-driver Status Register	PIO_MDSR	Read-only	(See Note 4)

Notes:
1. Unless otherwise programmed during system-level design according to peripheral mapping and multiplexing.
2. The value of this register will depend on the level of the external pins.
3. This register is cleared on Reset. However, the first read of the register may give a non-zero value if any changes have occurred on any of the external pins between Reset and when the register is read.
4. The value of these registers can be customer-defined. The default value is 0.

Reset Configuration

The Reset state of the following PIO1 registers can be defined by the customer:

PIO Status Filter, Glitch Input Filter Status, Multi-driver.

I User Interface Register Descriptions

All control and status registers are 32-bit: each bit of the User Interface Registers is associated with the corresponding I/O channel, bit 0 with channel 0, bit 1 with channel 1, and so on. This mapping is shown diagrammatically in the table below.

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

LPIO1/Peripheral Multiplexing

For the bit configuration of the register, see "I User Interface Register Descriptions" above.

PIO Enable Register

Register Name: PIO_PER

Access Type: Write-only

This register is used to enable control of individual pins by the PIO controller rather than by an internally connected peripheral. When the PIO is enabled, the Data to Peripheral (D_TO_PERIPH) signal is held at logic zero. The register is programmed as follows:

1 = Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

0 = No effect.

PIO Disable Register

Register Name: PIO_PDR

Access Type: Write-only

This register is used to disable control of individual pins by the PIO controller. When PIO control is disabled, the peripheral function (if any) connected to the I/O channel is enabled to control the corresponding pin. The register is programmed as follows:

1 = Disables the PIO from controlling the corresponding pin (enables peripheral control of the pin).

0 = No effect.

PIO Status Register

Register Name: PIO_PSR

Access Type: Read-only

Reset Value:FFFFFFF (Unless programmed otherwise in system level-design according to peripheral mapping and multiplexing)

This register indicates which pins are enabled for PIO control. This register is updated when PIO lines are enabled or disabled. The register reads as follows:

1 = PIO is active on the corresponding line (peripheral is inactive).

0 = PIO is inactive on the corresponding line (peripheral is active).

PIO Output Enable

For the bit configuration of the register, see “l User Interface Register Descriptions” on page 10

PIO Output Enable Register

Register Name: PIO_OER

Access Type: Write-only

This register is used to enable PIO output drivers. If the pin is driven by an internally connected peripheral, PIO_OER has no effect on the pin, but the information is stored. The register is programmed as follows:

1 = Enables the PIO output on the corresponding pin.

0 = No effect.

PIO Output Disable Register

Register Name: PIO_ODR

Access Type: Write-only

This register is used to disable PIO output drivers. If the pin is driven by an internally connected peripheral, PIO_ODR has no effect on the pin, but the information is stored. The register is programmed as follows:

1 = Disables the PIO output on the corresponding pin.

0 = No effect.

PIO Output Status Register

Register Name: PIO_OSR

Access Type: Read-only

Reset Value: 0

This register shows the PIO pin control (output enable) status which is programmed via PIO_OER and PIO_ODR. The defined value is effective only if the pin is controlled by the PIO. The register reads as follows:

1 = The corresponding PIO is output on this line.

0 = The corresponding PIO is input on this line.

Glitch Filtering

For the bit configuration of the register, see “l User Interface Register Descriptions” on page 10.

PIO Input Filter Enable Register

Register Name: PIO_IFER

Access Type: Write-only

This register is used to enable input glitch filters; it affects the data read from the pin whether or not the PIO is enabled. The register is programmed as follows:

1 = Enables the glitch filter on the corresponding pin.

0 = No effect.

PIO Input Filter Disable Register

Register Name: IO_IFDR

Access Type: Write-only

This register is used to disable input glitch filters. It affects the data read from the pin whether or not the PIO is enabled. The register is programmed as follows:

1 = Disables the glitch filter on the corresponding pin.

0 = No effect.

PIO Input Filter Status Register

Register Name: PIO_IFSR

Access Type: Read-only

Reset Value: 0

This register indicates which pins have glitch filters selected, as programmed via PIO_IFER and PIO_IFDR. The register reads as follows:

1 = Filter is selected on the corresponding input (peripheral and PIO).

0 = Filter is not selected on the corresponding input.

PIO Data Output

For the bit configuration of the register, see “I User Interface Register Descriptions” on page 10.

PIO Set Output DataL Register

Register Name: PIO_SODR

Access Type: Write-only

This register is used to set PIO data output to the corresponding pads. It affects the related pins only if the corresponding PIO outputs are enabled and if the pins are controlled by the PIO controller. Otherwise, the information is simply stored and is acted on if the PIO is later enabled. The register is programmed as follows:

1 = PIO output data on the corresponding pin is set (logic 1).

0 = No effect.

PIO Clear Output Data Register

Register Name: PIO_CODR

Access Type: Write-only

This register is used to clear PIO data output to the corresponding pads. It affects the related pins only if the corresponding PIO outputs are enabled and if the pins are controlled by the PIO controller. Otherwise, the information is simply stored and is acted on if the PIO is later enabled. The register is programmed as follows:

1 = PIO output data on the corresponding pin is cleared (logic 0).

0 = No effect.

PIO Output Data Status Register

Register Name: PIO_ODSR

Access Type: Read-only

Reset Value: 0

This register shows the output data status which is programmed via PIO_SODR or PIO_CODR. The bit pattern is effective only for those pins under the control of the PIO controller and only if the pins are enabled as outputs. Otherwise, the information is simply stored and is acted on if the PIO is later enabled. The register reads as follows:

1 = The output data for the corresponding pin is programmed to 1.

0 = The output data for the corresponding pin is programmed to 0.

Interrupt Masking

For the bit configuration of the register, see “I User Interface Register Descriptions” on page 10.

PIO Interrupt Enable Register

Register Name: PIO_IER

Access Type: Write-only

This register is used to enable PIO interrupts generated by the corresponding pins; logic level changes are detected and stored in the Interrupt Status Register (PIO_ISR). Enabled interrupts will be generated whether the PIO is enabled or not. The register is programmed as follows:

1 = Enables an interrupt when a change of logic level is detected on the corresponding pin.

0 = No effect.

PIO Interrupt Disable Register

Register Name: PIO_IDR

Access Type: Write-only

This register is used to disable PIO interrupts generated by the corresponding pins. Logic level changes are still detected and stored in the Interrupt Status Register (PIO_ISR). Disabled interrupts will be inhibited whether the PIO is enabled or not. The register is programmed as follows:

1 = Disables the interrupt generated by the corresponding pin.

0 = No effect.

PIO Interrupt Mask Register

Register Name: PIO_IMR

Access Type: Read-only

Reset Value: 0

This register shows which pins have interrupts enabled. It is updated when interrupts are enabled or disabled by writing to PIO_IER or PIO_IDR. The register reads as follows:

1 = Interrupt is enabled from the corresponding pin.

0 = Interrupt is disabled from the corresponding input pin.

Interrupt Source

For the bit configuration of the register, see “I User Interface Register Descriptions” on page 10.

PIO Interrupt Status Register

Register Name: PIO_ISR

Access Type: Read-only

Reset Value: 0

This register indicates, for each pin, when a logic level change has been detected (rising or falling edge). This is valid whether the PIO is selected for the pin or not and whether the pin is an input or output.

The register is reset to zero following a read, as well as at reset. The register reads as follows:

1 = At least one change has been detected on the corresponding pin since the register was last read or since reset.

0 = No change has been detected on the corresponding pin since the register was last read or since reset.

Output Pin Level

For the bit configuration of the register, see “I User Interface Register Descriptions” on page 10.

PIO Pin Data Status Register

Register Name: PIO_PDSR

Access Type: Read-only

Reset Value: Undefined

This register shows the logic level of the physical I/O pin. The pin logic levels are always valid, regardless of whether the pins are enabled as PIO, peripheral, input or output.

The value of this register will depend on the level of the external pins. The register reads as follows:

1 = The corresponding pin is at logic 1.

0 = The corresponding pin is at logic 0.

Multi-drive Option

For the bit configuration of the register, see “I User Interface Register Descriptions” on page 10.

PIO Multi-Drive Enable Register

Register Name: PIO_MDER

Access Type: Write-only

This register is used to enable PIO output drivers to be configured as open drain to support external drivers on the same pin.

1 = Enables multi-drive option on the corresponding pin.

0 = No effect.

PIO Multi-drive Disable Register

Register Name: PIO_MDDR

Access Type: Write-only

This register is used to disable the open drain configuration of the output buffer.

1 = Disables multi-drive option on the corresponding pin.

0 = No effect.

PIO Multi-drive Status Register

Register Name: PIO_MDSR

Access Type: Read-only

This register indicates which pins are configured with open drain drivers.

1 = PIO is configured as an open drain.

0 = PIO is not configured as an open drain.

Timing Diagrams

Figure 4. APB Signal Timing

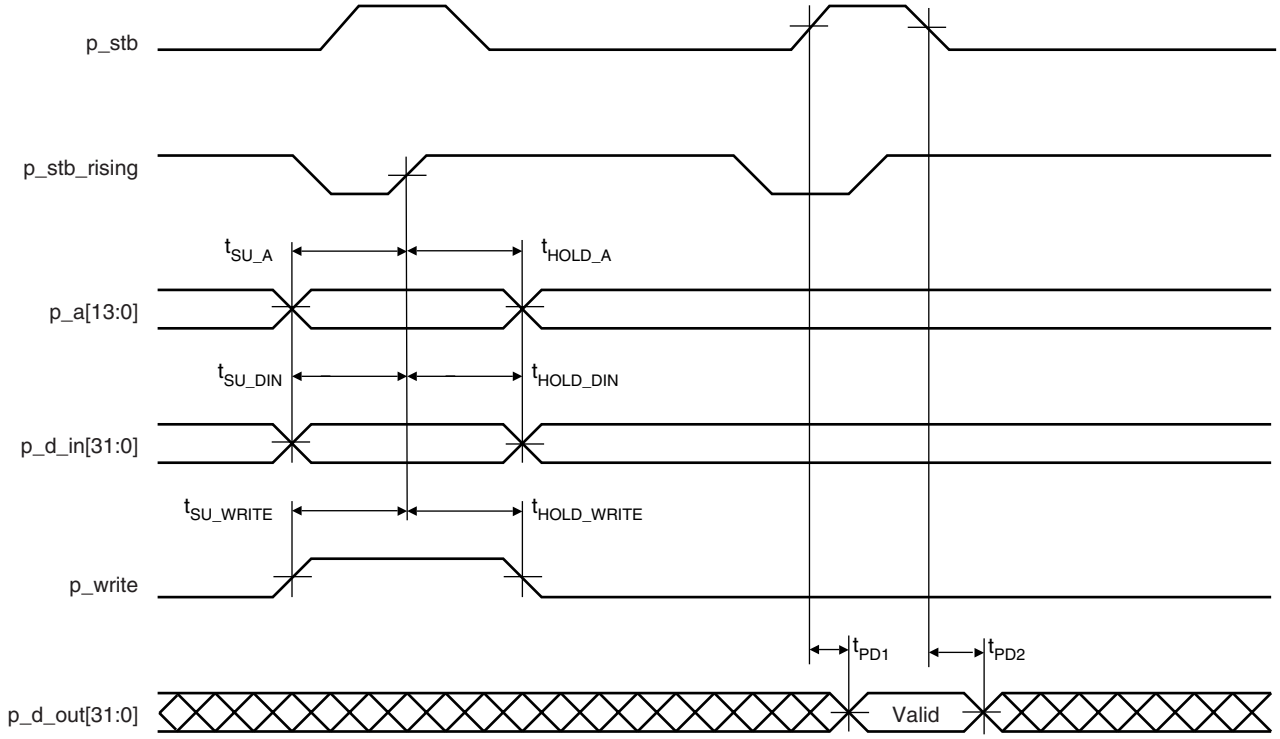
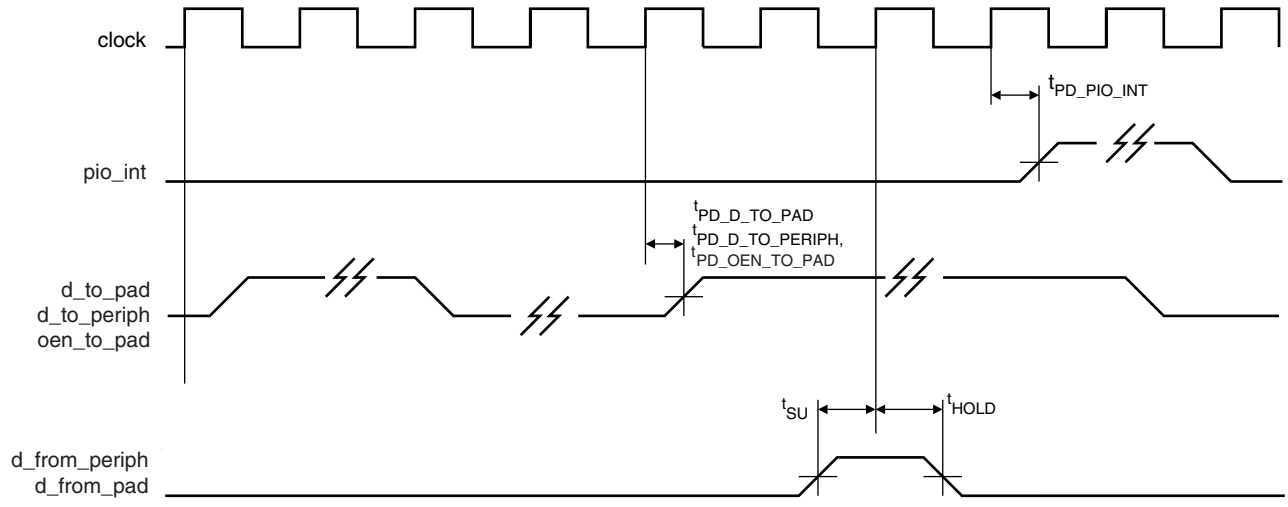


Figure 5. PIO Signal Timing



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

1321C–03/01/xM