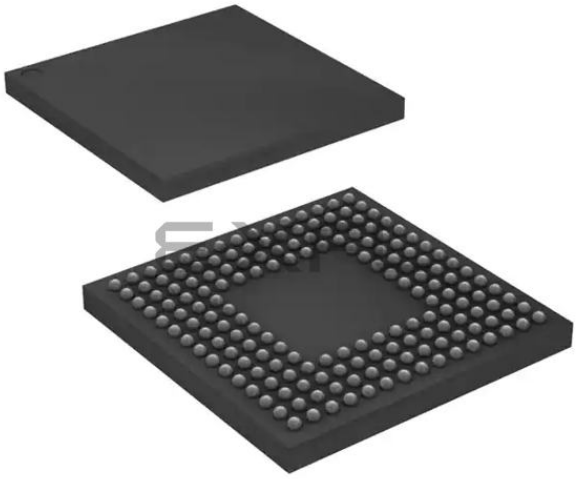


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | ARM7® |
| Core Size | 16/32-Bit |
| Speed | 33MHz |
| Connectivity | EBI/EMI, SPI, UART/USART |
| Peripherals | POR, WDT |
| Number of I/O | 58 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 8x10b; D/A 2x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-LFBGA |
| Supplier Device Package | 176-BGA (13x13) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at91m55800a-33cj-999 |

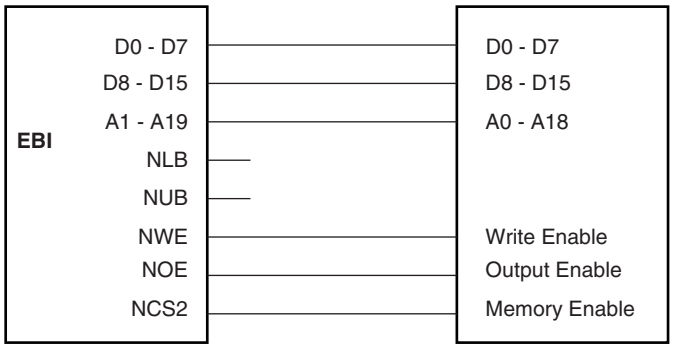
# 3. Pin Description

**Table 3-1.** Pin Description

| Module | Name | Function | Type | Active Level | Comments |
|---|---|---|---|---|---|
| EBI | A0 - A23 | Address bus | Output | – | |
| | D0 - D15 | Data bus | I/O | – | |
| | NCS0 - NCS7 | Chip select | Output | Low | |
| | NWR0 | Lower byte 0 write signal | Output | Low | Used in Byte-write option |
| | NWR1 | Lower byte 1 write signal | Output | Low | Used in Byte-write option |
| | NRD | Read signal | Output | Low | Used in Byte-write option |
| | NWE | Write enable | Output | Low | Used in Byte-select option |
| | NOE | Output enable | Output | Low | Used in Byte-select option |
| | NUB | Upper byte-select | Output | Low | Used in Byte-select option |
| | NLB | Lower byte-select | Output | Low | Used in Byte-select option |
| | NWAIT | Wait input | Input | Low | |
| | BMS | Boot mode select | Input | – | Sampled during reset |
| AIC | IRQ0 - IRQ5 | External interrupt request | Input | – | PIO-controlled after reset |
| | FIQ | Fast external interrupt request | Input | – | PIO-controlled after reset |
| Timer | TCLK0 - TCLK5 | Timer external clock | Input | – | PIO-controlled after reset |
| | TIOA0 - TIOA5 | Multipurpose timer I/O pin A | I/O | – | PIO-controlled after reset |
| | TIOB0 - TIOB5 | Multipurpose timer I/O pin B | I/O | – | PIO-controlled after reset |
| USART | SCK0 - SCK2 | External serial clock | I/O | – | PIO-controlled after reset |
| | TXD0 - TXD2 | Transmit data output | Output | – | PIO-controlled after reset |
| | RXD0 - RXD2 | Receive data input | Input | – | PIO-controlled after reset |
| SPI | SPCK | SPI clock | I/O | – | PIO-controlled after reset |
| | MISO | Master in slave out | I/O | – | PIO-controlled after reset |
| | MOSI | Master out slave in | I/O | – | PIO-controlled after reset |
| | NSS | Slave select | Input | Low | PIO-controlled after reset |
| | NPCS0 - NPCS3 | Peripheral chip select | Output | Low | PIO-controlled after reset |
| PIO | PA0 - PA29 | Parallel I/O port A | I/O | – | Input after reset |
| | PB0 - PB27 | Parallel I/O port B | I/O | – | Input after reset |
| WD | NWDOVF | Watchdog timer overflow | Output | Low | Open drain |
| ADC | AD0-AD7 | Analog input channels 0 - 7 | Analog in | – | |
| | AD0TRIG | ADC0 external trigger | Input | – | PIO-controlled after reset |
| | AD1TRIG | ADC1 external trigger | Input | – | PIO-controlled after reset |
| | ADVREF | Analog reference | Analog ref | – | |

Figure 11-6 shows how to connect a 16-bit device without byte access (e.g. Flash) on NCS2.

**Figure 11-6.** Connection for a 16-bit Data Bus Without Byte-write Capability.




## 11.5 Boot on NCS0

Depending on the device and the BMS pin level during the reset, the user can select either an 8-bit or 16-bit external memory device connected on NCS0 as the Boot Memory. In this case, EBI_CSR0 (Chip-select Register 0) is reset at the following configuration for chip select 0:

- 8 wait states (WSE = 1, NWS = 7)
- 8-bit or 16-bit data bus width, depending on BMS

Byte access type and number of data float time are respectively set to Byte-write Access and 0. With a nonvolatile memory interface, any value can be programmed for these parameters.

Before the remap command, the user can modify the chip select 0 configuration, programming the EBI_CSR0 with exact boot memory characteristics. The base address becomes effective after the remap command, but the new number of wait states can be changed immediately. This is useful if a boot sequence needs to be faster.

### 12.9.7 APMC Clock Generator Mode Register

**Register Name:** APMC_CGMR

**Access Type:** Read/Write

**Reset Value:** 0x0

**Offset:** 0x20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | PLLCOUNT | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| OSCOUNT | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CSS | | MUL | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | PRES | | | – | MCKODS | MOSCEN | MOSCBYP |

- **MOSCBYP: Main Oscillator Bypass (Code Label `APMC_MOSC_BYP`)**

0 = Crystal must be connected between XIN and XOUT.

1 = External clock must be provided on XIN.

- **MOSCEN: Main Oscillator Enable (Code Label `APMC_MOSC_EN`)**

0 = Main Oscillator is disabled.

1 = Main Oscillator is enabled.

Note: When operating in Bypass Mode, the Main Oscillator must be disabled. MOSCEN and MOSCBYP bits must never be set together.

- **MCKODS: Master Clock Output Disable (Code Label `APMC_MCKO_DIS`)**

0 = The MCKO pin is driven with the Master Clock (MCK).

1 = The MCKO pin is tri-stated.

- **PRES: Prescaler Selection**

| PRES | | | Prescaler Selected | Code Label |
|---|---|---|---|---|
| 0 | 0 | 0 | None. Prescaler Output is the selected clock. | APMC_PRES_NONE |
| 0 | 0 | 1 | Selected clock is divided by 2 | APMC_PRES_DIV2 |
| 0 | 1 | 0 | Selected clock is divided by 4 | APMC_PRES_DIV4 |
| 0 | 1 | 1 | Selected clock is divided by 8 | APMC_PRES_DIV8 |
| 1 | 0 | 0 | Selected clock is divided by 16 | APMC_PRES_DIV16 |
| 1 | 0 | 1 | Selected clock is divided by 32 | APMC_PRES_DIV32 |
| 1 | 1 | 0 | Selected clock is divided by 64 | APMC_PRES_DIV64 |
| 1 | 1 | 1 | Reserved | – |

### 13.3.5 RTC Time Alarm Register

**Register Name:** RTC_TAR

**Access Type:** Read/Write

**Reset State:** 0x0

**Offset:** 0x10

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| HOUREN | AMPM | HOUR | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MINEN | MIN | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SECEN | SEC | | | | | | |

- **SEC: Second Alarm**

This field is the alarm field corresponding to the BCD-coded second counter.

- **SECEN: Second Alarm Enable**

| SECEN | Selected SECEN | Code Label |
|---|---|---|
| 0 | The second matching alarm is disabled. | RTC_SEC_ALARM_DIS |
| 1 | The second matching alarm is enabled. | RTC_SEC_ALARM_EN |

- **MIN: Minute Alarm**

This field is the alarm field corresponding to the BCD-coded minute counter.

- **MINEN: Minute Alarm Enable**

| MINEN | Selected MINEN | Code Label |
|---|---|---|
| 0 | The minute matching alarm is disabled. | RTC_MIN_ALARM_DIS |
| 1 | The minute matching alarm is enabled. | RTC_MIN_ALARM_EN |

- **HOUR: Hour Alarm**

This field is the alarm field corresponding to the BCD-coded hour counter.

- **AMPM: AM/PM Indicator**

This bit is the AM/PM indicator in 12-Hour mode. It must be written at 0 if HRMOD in RTC_HMR defines 24-Hour mode.

- **HOUREN: Hour Alarm Enable**

| HOUREN | Selected HOUREN | Code Label |
|---|---|---|
| 0 | The hour matching alarm is disabled. | RTC_HOUR_ALARM_DIS |
| 1 | The hour matching alarm is enabled. | RTC_HOUR_ALARM_EN |

# 16. PIO: Parallel I/O Controller

The AT91M55800A has 58 programmable I/O lines. 13 pins are dedicated as general-purpose I/O pins. The other I/O lines are multiplexed with an external signal of a peripheral to optimize the use of available package pins. The PIO lines are controlled by two separate and identical PIO Controllers called PIOA and PIOB. The PIO controller enables the generation of an interrupt on input change and insertion of a simple input glitch filter on any of the PIO pins.

## 16.1 Multiplexed I/O Lines

Some I/O lines are multiplexed with an I/O signal of a peripheral. After reset, the pin is controlled by the PIO Controller and is in input mode.

When a peripheral signal is not used in an application, the corresponding pin can be used as a parallel I/O. Each parallel I/O line is bi-directional, whether the peripheral defines the signal as input or output. Figure 16-1 shows the multiplexing of the peripheral signals with Parallel I/O signals.

If a pin is multiplexed between the PIO Controller and a peripheral, the pin is controlled by the registers PIO_PER (PIO Enable) and PIO_PDR (PIO Disable). The register PIO_PSR (PIO Status) indicates whether the pin is controlled by the corresponding peripheral or by the PIO Controller.

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

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

## 16.2 Output Selection

The user can enable each individual I/O signal as an output with the registers PIO_OER (Output Enable) and PIO_ODR (Output Disable). The output status of the I/O signals can be read in the register PIO_OSR (Output Status). The direction defined has effect only if the pin is configured to be controlled by the PIO Controller.

## 16.3 I/O Levels

Each pin can be configured to be driven high or low. The level is defined in four different ways, according to the following conditions.

If a pin is controlled by the PIO Controller and is defined as an output (see Output Selection above), the level is programmed using the registers PIO_SODR (Set Output Data) and PIO_CODR (Clear Output Data). In this case, the programmed value can be read in PIO_ODSR (Output Data Status).

If a pin is controlled by the PIO Controller and is not defined as an output, the level is determined by the external circuit.

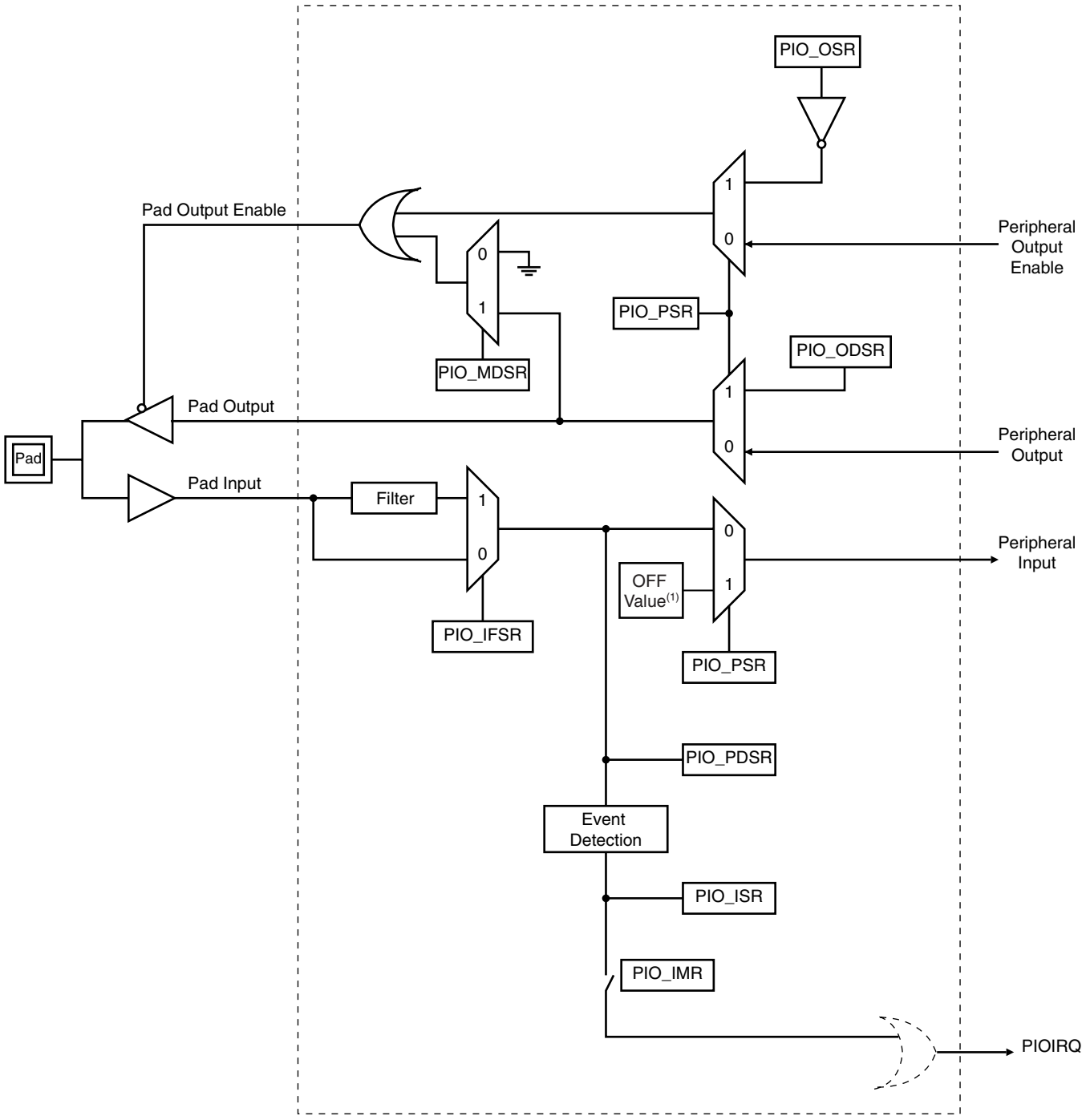
If a pin is not controlled by the PIO Controller, the state of the pin is defined by the peripheral (see peripheral datasheets).

In all cases, the level on the pin can be read in the register PIO_PDSR (Pin Data Status).

## 16.4 Filters

Optional input glitch filtering is available on each pin and is controlled by the registers PIO_IFER (Input Filter Enable) and PIO_IFDR (Input Filter Disable). The input glitch filtering can be

**Figure 16-1.** Parallel I/O Multiplexed with a Bi-directional Signal




Note: 1. See “Section 16.8 ”PIO Connection Tables” .”

### 16.9.1 PIO Enable Register

**Register Name:** PIO_PER

**Access Type:** Write-only

**Offset:** 0x00

| 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 |

This register is used to enable individual pins to be controlled by the PIO Controller instead of the associated peripheral. When the PIO is enabled, the associated peripheral (if any) is held at logic zero.

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

0 = No effect.

### 16.9.2 PIO Disable Register

**Register Name:** PIO_PDR

**Access Type:** Write-only

**Offset:** 0x04

| 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 |

This register is used to disable PIO control of individual pins. When the PIO control is disabled, the normal peripheral function is enabled on the corresponding pin.

1 = Disables PIO control (enables peripheral control) on the corresponding pin.

0 = No effect.

### 16.9.13 PIO Pin Data Status Register

**Register Name:** PIO_PDSR

**Access Type:** Read-only

**Offset:** 0x3C

**Reset Value:** Undefined

| 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 |

This register shows the state of the physical pin of the chip. The pin values are always valid, regardless of whether the pins are enabled as PIO, peripheral, input or output. The register reads as follows:

1 = The corresponding pin is at logic 1.

0 = The corresponding pin is at logic 0.

### 16.9.14 PIO Interrupt Enable Register

**Register Name:** PIO_IER

**Access Type:** Write-only

**Offset:** 0x40

| 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 |

This register is used to enable PIO interrupts on the corresponding pin. It has effect whether PIO is enabled or not.

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

0 = No effect.

### 17.2.4 SF Protect Mode Register

**Register Name:** SF_PMR

**Access Type:** Read/Write

**Reset Value:** 0x0

**Offset:** 0x18

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PMRKEY | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PMRKEY | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | AIC | – | – | – | – | – |

- **PMRKEY: Protect Mode Register Key**

Used only when writing SF_PMR. PMRKEY is reads 0.

0x27A8: Write access in SF_PMR is allowed.

Other value: Write access in SF_PMR is prohibited.

- **AIC: AIC Protect Mode Enable (Code Label `SF_AIC`)**

0 = The Advanced Interrupt Controller runs in Normal Mode.

1 = The Advanced Interrupt Controller runs in Protect Mode.

## 18.10 USART User Interface

**Base Address USART0:** 0xFFFC0000 (Code Label `USART0_BASE`)
**Base Address USART1:** 0xFFFC4000 (Code Label `USART1_BASE`)
**Base Address USART2:** 0xFFFC8000 (Code Label `USART2_BASE`)

**Table 18-2.** Register Mapping

| Offset | Register | Name | Access | Reset |
|---|---|---|---|---|
| 0x00 | Control Register | US_CR | Write-only | – |
| 0x04 | Mode Register | US_MR | Read/Write | 0 |
| 0x08 | Interrupt Enable Register | US_IER | Write-only | – |
| 0x0C | Interrupt Disable Register | US_IDR | Write-only | – |
| 0x10 | Interrupt Mask Register | US_IMR | Read-only | 0 |
| 0x14 | Channel Status Register | US_CSR | Read-only | 0x18 |
| 0x18 | Receiver Holding Register | US_RHR | Read-only | 0 |
| 0x1C | Transmitter Holding Register | US_THR | Write-only | – |
| 0x20 | Baud Rate Generator Register | US_BRGR | Read/Write | 0 |
| 0x24 | Receiver Time-out Register | US_RTOR | Read/Write | 0 |
| 0x28 | Transmitter Time-guard Register | US_TTGR | Read/Write | 0 |
| 0x2C | Reserved | – | – | – |
| 0x30 | Receive Pointer Register | US_RPR | Read/Write | 0 |
| 0x34 | Receive Counter Register | US_RCR | Read/Write | 0 |
| 0x38 | Transmit Pointer Register | US_TPR | Read/Write | 0 |
| 0x3C | Transmit Counter Register | US_TCR | Read/Write | 0 |

- **TIMEOUT: Enable Time-out Interrupt (Code Label `US_TIMEOUT`)**

0 = No effect.

1 = Enables Reception Time-out Interrupt.

- **TXEMPTY: Enable TXEMPTY Interrupt (Code Label `US_TXEMPTY`)**

0 = No effect.

1 = Enables TXEMPTY Interrupt.

### 18.10.7 USART Receiver Holding Register

**Name:** US_RHR
**Access Type:** Read-only
**Reset State:** 0
**Offset:** 0x18

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | RXCHR |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXCHR | | | | | | | |

- **RXCHR: Received Character**

Last character received if RXRDY is set. When number of data bits is less than 9 bits, the bits are right-aligned.

All unused bits read zero.

### 18.10.8 USART Transmitter Holding Register

**Name:** US_THR
**Access Type:** Write-only
**Offset:** 0x1C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | TXCHR |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXCHR | | | | | | | |

- **TXCHR: Character to be Transmitted**

Next character to be transmitted after the current character if TXRDY is not set. When number of data bits is less than 9 bits, the bits are right-aligned.

### 19.5.10 TC Status Register

**Register Name:** TC_SR
**Access Type:** Read/Write
**Offset:** 0x20

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | MTIOB | MTIOA | CLKSTA |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ETRGS | LDRBS | LDRAS | CPCS | CPBS | CPAS | LOVRS | COVFS |

- **COVFS: Counter Overflow Status (Code Label `TC_COVFS`)**

0 = No counter overflow has occurred since the last read of the Status Register.

1 = A counter overflow has occurred since the last read of the Status Register.

- **LOVRS: Load Overrun Status (Code Label `TC_LOVRS`)**

0 = Load overrun has not occurred since the last read of the Status Register or WAVE = 1.

1 = RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status Register, if WAVE = 0.

- **CPAS: RA Compare Status (Code Label `TC_CPAS`)**

0 = RA Compare has not occurred since the last read of the Status Register or WAVE = 0.

1 = RA Compare has occurred since the last read of the Status Register, if WAVE = 1.

- **CPBS: RB Compare Status (Code Label `TC_CPBS`)**

0 = RB Compare has not occurred since the last read of the Status Register or WAVE = 0.

1 = RB Compare has occurred since the last read of the Status Register, if WAVE = 1.

- **CPCS: RC Compare Status (Code Label `TC_CPCS`)**

0 = RC Compare has not occurred since the last read of the Status Register.

1 = RC Compare has occurred since the last read of the Status Register.

- **LDRAS: RA Loading Status (Code Label `TC_LDRAS`)**

0 = RA Load has not occurred since the last read of the Status Register or WAVE = 1.

1 = RA Load has occurred since the last read of the Status Register, if WAVE = 0.

- **LDRBS: RB Loading Status (Code Label `TC_LDRBS`)**

0 = RB Load has not occurred since the last read of the Status Register or WAVE = 1.

1 = RB Load has occurred since the last read of the Status Register, if WAVE = 0.

- **ETRGS: External Trigger Status (Code Label `TC_ETRGS`)**

0 = External trigger has not occurred since the last read of the Status Register.

1 = External trigger has occurred since the last read of the Status Register.

# 20. SPI: Serial Peripheral Interface

The AT91M55800A includes an SPI which provides communication with external devices in master or slave mode.

The SPI has four external chip selects which can be connected to up to 15 devices. The data length is programmable, from 8- to 16-bit.

As for the USART, a 2-channel PDC can be used to move data between memory and the SPI without CPU intervention.

## 20.1 Pin Description

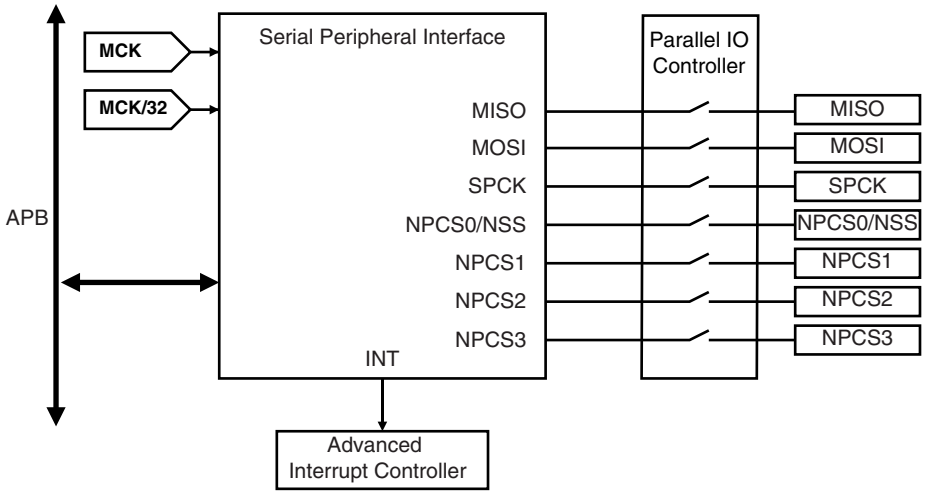
Seven pins are associated with the SPI Interface. When not needed for the SPI function, each of these pins can be configured as a PIO.

Support for an external master is provided by the PIO Controller Multi-driver option. To configure an SPI pin as open-drain to support external drivers, set the corresponding bits in the PIO_MDSR register.

An input filter can be enabled on the SPI input pins by setting the corresponding bits in the PIO_IFSR.

The NPCS0/NSS pin can function as a peripheral chip select output or slave select input. Refer to Table 20-1 for a description of the SPI pins.

**Figure 20-1.** SPI Block Diagram

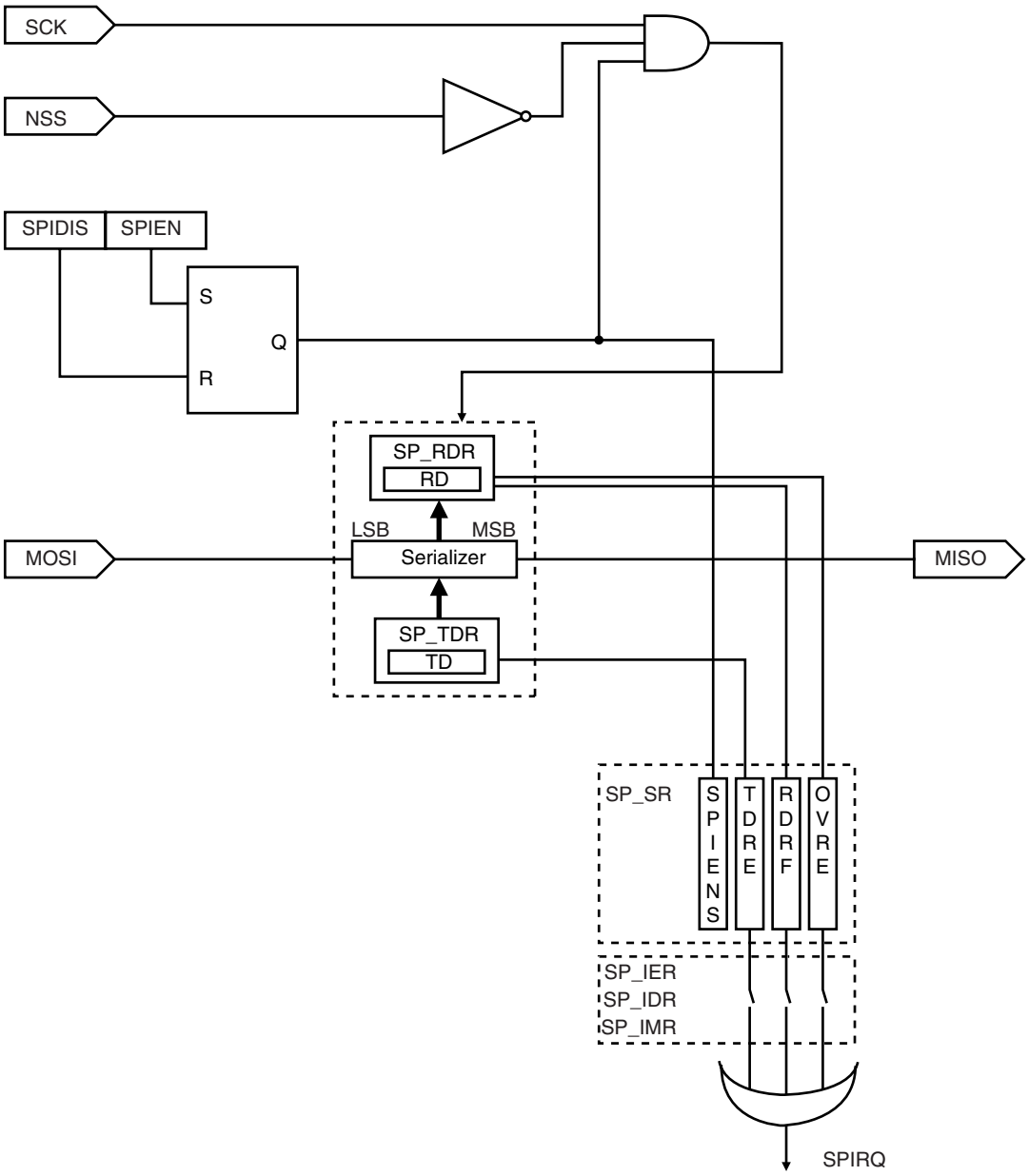


## 20.3 Slave Mode

In Slave Mode, the SPI waits for NSS to go active low before receiving the serial clock from an external master.

In slave mode CPOL, NCPHA and BITS fields of SP_CSR0 are used to define the transfer characteristics. The other Chip Select Registers are not used in slave mode.

**Figure 2.** SPI in Slave Mode



### 20.7.2 SPI Mode Register

**Register Name:** SP_MR
**Access Type:** Read/Write
**Reset State:** 0
**Offset:** 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DLYBCS | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | PCS | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LLB | – | – | – | MCK32 | PCSDEC | PS | MSTR |

- **MSTR: Master/Slave Mode (Code Label `SP_MSTR`)**

0 = SPI is in Slave mode.

1 = SPI is in Master mode.

MSTR configures the SPI Interface for either master or slave mode operation.

- **PS: Peripheral Select**

| PS | Selected PS | Code Label: `SP_PS` |
|---|---|---|
| 0 | Fixed Peripheral Select | SP_PS_FIXED |
| 1 | Variable Peripheral Select | SP_PS_VARIABLE |

- **PCSDEC: Chip Select Decode (Code Label `SP_PCSDEC`)**

0 = The chip selects are directly connected to a peripheral device.

1 = The four chip select lines are connected to a 4- to 16-bit decoder.

When PCSDEC equals one, up to 16 Chip Select signals can be generated with the four lines using an external 4- to 16-bit decoder.

The Chip Select Registers define the characteristics of the 16 chip selects according to the following rules:

SP_CSR0defines peripheral chip select signals 0 to 3.

SP_CSR1defines peripheral chip select signals 4 to 7.

SP_CSR2defines peripheral chip select signals 8 to 11.

SP_CSR3defines peripheral chip select signals 12 to 15[(1)].

Note: 1. The 16th state corresponds to a state in which all chip selects are inactive. This allows a different clock configuration to be defined by each chip select register.

- **MCK32: Clock Selection (Code Label `SP_DIV32`)**

0 = SPI Master Clock equals MCK.

1 = SPI Master Clock equals MCK/32.

- **SCBR: Serial Clock Baud Rate (Code Label `SP_SCBR`)**

In Master Mode, the SPI Interface uses a modulus counter to derive the SPCK baud rate from the SPI Master Clock (selected between MCK and MCK/32). The Baud rate is selected by writing a value from 2 to 255 in the field SCBR. The following equation determines the SPCK baud rate:

$$\text{SPCK\_Baud\_Rate} = \frac{\text{SPI\_Master\_Clock\_frequency}}{2 \times \text{SCBR}}$$

Giving SCBR a value of zero or one disables the baud rate generator. SPCK is disabled and assumes its inactive state value. No serial transfers may occur. At reset, baud rate is disabled.

- **DLYBS: Delay Before SPCK (Code Label `SP_DLYBS`)**

This field defines the delay from NPCS valid to the first valid SPCK transition.

When DLYBS equals zero, the NPCS valid to SPCK transition is 1/2 the SPCK clock period.

Otherwise, the following equation determines the delay:

NPCS_to_SPCK_Delay = DLYBS * SPI_Master_Clock_period

- **DLYBCT: Delay Between Consecutive Transfers (Code Label `SP_DLYBCT`)**

This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.

When DLYBCT equals zero, a delay of four SPI Master Clock periods are inserted.

Otherwise, the following equation determines the delay:

Delay_After_Transfer = 32 * DLYBCT * SPI_Master_Clock_period

### 21.0.14 ADC Interrupt Mask Register

**Register Name:** ADC_IMR
**Access Type:** Read-only
**Reset State:** 0
**Offset:** 0x2C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | OVRE3 | OVRE2 | OVRE1 | OVRE0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | EOC3 | EOC2 | EOC1 | EOC0 |

- **EOC: End of Conversion Interrupt Mask (Code Label `ADC_EOCx`)**

0 = End of Conversion Interrupt is disabled.

1 = End of Conversion Interrupt is enabled.

- **OVRE: Overrun Error Interrupt Mask (Code Label `ADC_OVREx`)**

0 = Overrun Error Interrupt is disabled.

1 = Overrun Error Interrupt is enabled.

### 21.0.15 ADC Convert Data Register

**Register Name:** ADC_CDR0 to ADC_CDR3
**Access Type:** Read-only
**Reset State:** 0
**Offset:** 0x30 to 0x3C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | DATA | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |

- **DATA: Converted Data**

The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed. The Convert Data Register (CDR) is only loaded if the corresponding analog channel is enabled.

| DATA | Selected DATA | Code Label: `ADC_CDRx` |
|---|---|---|
| 0 or 1 | 10-bits Data | ADC_DATA_10BITS |
| 0 or 1 | 8-bits Data | ADC_DATA_8BITS |