# 13. I/O Ports

## 13.1 Features

- General purpose input and output pins with individual configuration
- Output driver with configurable driver and pull settings:
  - Totem-pole
  - Wired-AND
  - Wired-OR
  - Bus-keeper
  - Inverted I/O
- Input with synchronous and/or asynchronous sensing with interrupts and events
  - Sense both edges
  - Sense rising edges
  - Sense falling edges
  - Sense low level
- Optional pull-up and pull-down resistor on input and Wired-OR/AND configurations
- Optional slew rate control
- Asynchronous pin change sensing that can wake the device from all sleep modes
- Two port interrupts with pin masking per I/O port
- Efficient and safe access to port pins
  - Hardware read-modify-write through dedicated toggle/clear/set registers
  - Configuration of multiple pins in a single operation
  - Mapping of port registers into bit-accessible I/O memory space
- Peripheral clocks output on port pin
- Real-time counter clock output to port pin
- Event channels can be output on port pin
- Remapping of digital peripheral pin functions
  - Selectable USART, SPI, and timer/counter input/output pin locations

## 13.2 Overview

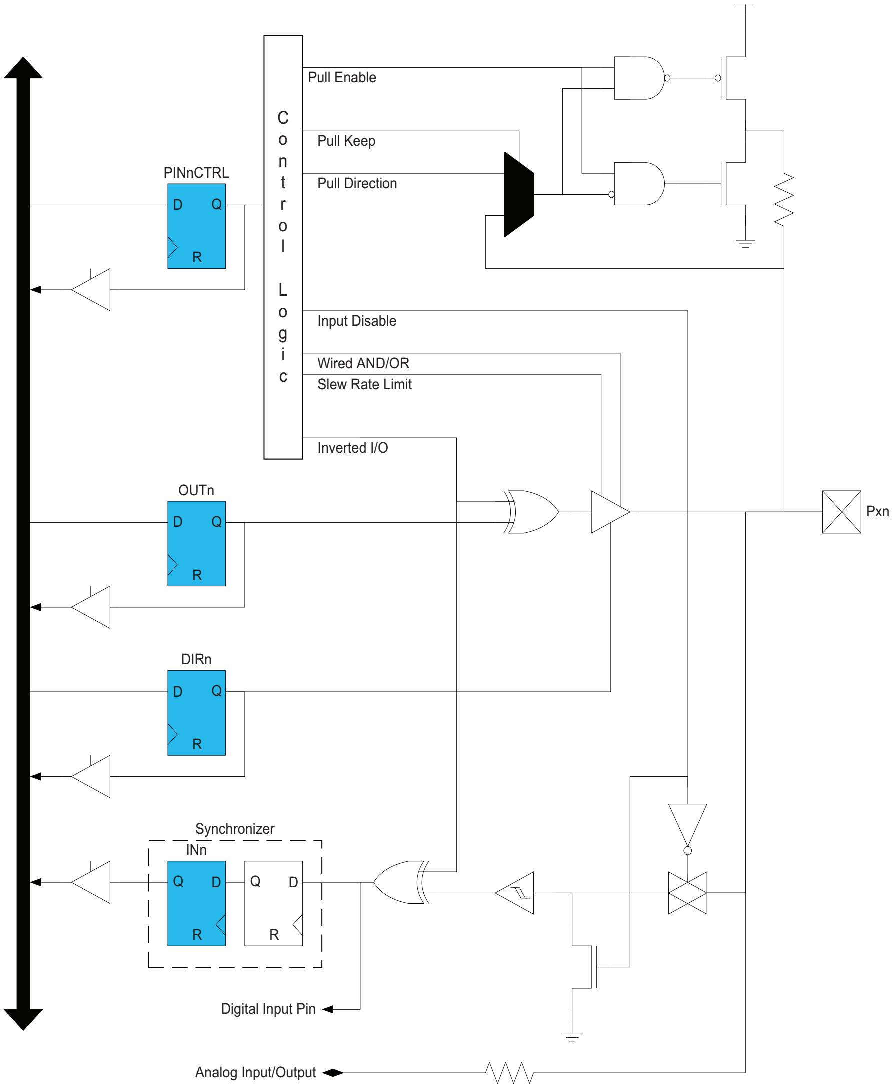
AVR XMEGA microcontrollers have flexible general purpose I/O ports. One port consists of up to eight port pins: pin 0 to 7. Each port pin can be configured as input or output with configurable driver and pull settings. They also implement synchronous and asynchronous input sensing with interrupts and events for selectable pin change conditions. Asynchronous pin-change sensing means that a pin change can wake the device from all sleep modes, included the modes where no clocks are running.

All functions are individual and configurable per pin, but several pins can be configured in a single operation. The pins have hardware read-modify-write (RMW) functionality for safe and correct change of drive value and/or pull resistor configuration. The direction of one port pin can be changed without unintentionally changing the direction of any other pin.

The port pin configuration also controls input and output selection of other device functions. It is possible to have both the peripheral clock and the real-time clock output to a port pin, and available for external use. The same applies to events from the event system that can be used to synchronize and control external functions.  Other digital peripherals, such as USART, SPI, and timer/counters, can be remapped to selectable pin locations in order to optimize pin-out versus application needs.

Figure 13-1 on page 140 shows the I/O pin functionality and the registers that are available for controlling a pin.

**Figure 13-1. General I/O pin functionality.**



## 13.3 I/O Pin Use and Configuration

Each port has one data direction (DIR) register and one data output value (OUT) register that are used for port pin control. The data input value (IN) register is used for reading the port pins. In addition, each pin has a pin configuration (PINnCTRL) register for additional pin configuration.

Direction of the pin is decided by the DIRn bit in the DIR register. If DIRn is written to one, pin n is configured as an output pin. If DIRn is written to zero, pin n is configured as an input pin.

When direction is set as output, the OUTn bit in OUT is used to set the value of the pin. If OUTn is written to one, pin n is driven high. If OUTn is written to zero, pin n is driven low.

The IN register is used for reading pin values. A pin value can always be read regardless of whether the pin is configured as input or output, except if digital input is disabled.

The I/O pins are tri-stated when a reset condition becomes active, even if no clocks are running.

The pin n configuration (PINnCTRL) register is used for additional I/O pin configuration. A pin can be set in a totem-pole, wired-AND, or wired-OR configuration. It is also possible to enable inverted input and output for a pin.

A totem-pole output has four possible pull configurations: totem-pole (push-pull), pull-down, pull-up, and bus-keeper. The bus-keeper is active in both directions. This is to avoid oscillation when disabling the output. The totem-pole configurations with pull-up and pull-down have active resistors only when the pin is set as input. This feature eliminates unnecessary power consumption. For wired-AND and wired-OR configuration, the optional pull-up and pull-down resistors are active in both input and output directions.

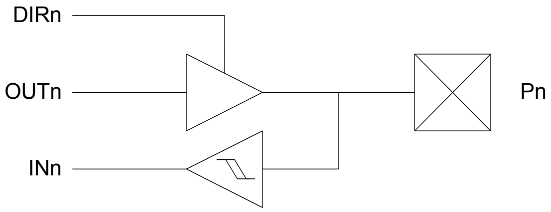
Since pull configuration is configured through the pin configuration register, all intermediate port states during switching of the pin direction and pin values are avoided.

The I/O pin configurations are summarized with simplified schematics in Figure 13-2 on page 141 to Figure 13-7 on page 143.

### 13.3.1 Totem-pole

In the totem-pole (push-pull) configuration, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration, there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull resistor is connected.

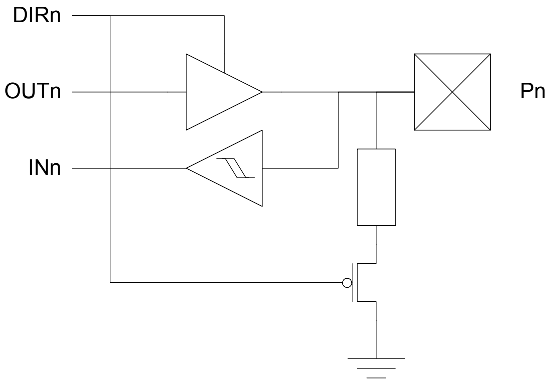
**Figure 13-2. I/O pin configuration - Totem-pole (push-pull).**



#### 13.3.1.1 Totem-pole with Pull-down

In this mode, the configuration is the same as for totem-pole mode, expect the pin is configured with an internal pull-down resistor when set as input.

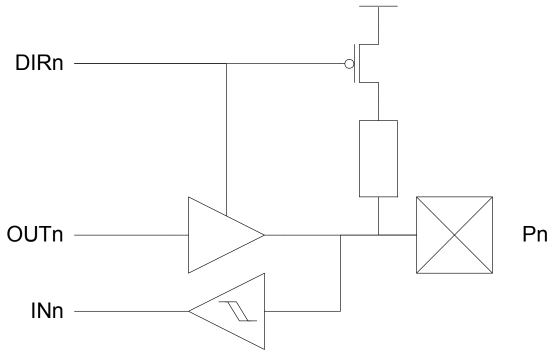
**Figure 13-3. I/O pin configuration - Totem-pole with pull-down (on input).**



#### 13.3.1.2 Totem-pole with Pull-up

In this mode, the configuration is as for totem-pole, expect the pin is configured with internal pull-up when set as input.

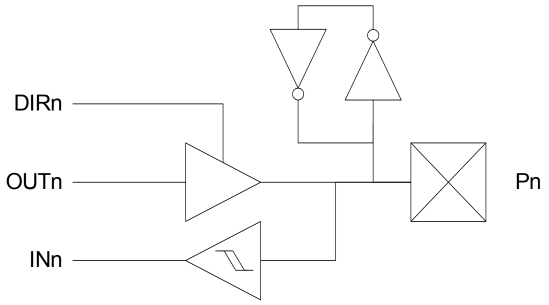
**Figure 13-4. I/O pin configuration - Totem-pole with pull-up (on input).**



### 13.3.2 Bus-keeper

In the bus-keeper configuration, it provides a weak bus-keeper that will keep the pin at its logic level when the pin is no longer driven to high or low. If the last level on the pin/bus was 1, the bus-keeper configuration will use the internal pull resistor to keep the bus high. If the last logic level on the pin/bus was 0, the bus-keeper will use the internal pull resistor to keep the bus low.

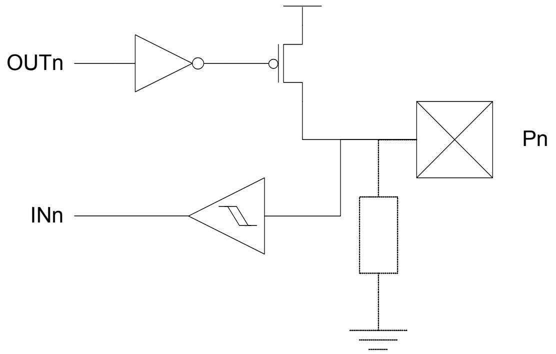
**Figure 13-5. I/O pin configuration - Totem-pole with bus-keeper.**



### 13.3.3 Wired-OR

In the wired-OR configuration, the pin will be driven high when the corresponding bits in the OUT and DIR registers are written to one. When the OUT register is set to zero, the pin is released, allowing the pin to be pulled low with the internal or an external pull-resistor. If internal pull-down is used, this is also active if the pin is set as input.

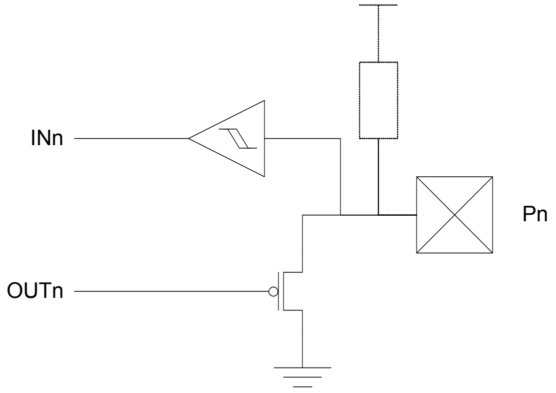
**Figure 13-6. Output configuration - Wired-OR with optional pull-down.**



### 13.3.4 Wired-AND

In the wired-AND configuration, the pin will be driven low when the corresponding bits in the OUT and DIR registers are written to zero. When the OUT register is set to one, the pin is released allowing the pin to be pulled high with the internal or an external pull-resistor. If internal pull-up is used, this is also active if the pin is set as input.

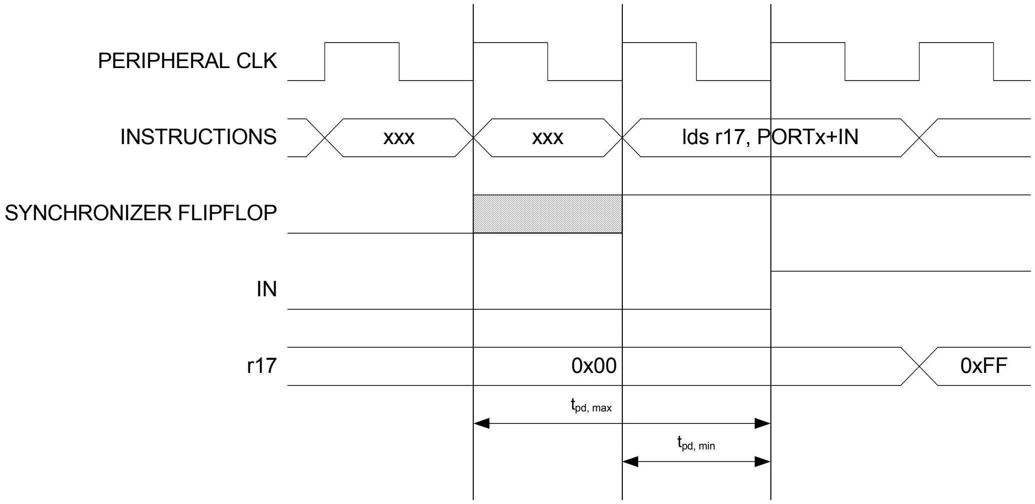
**Figure 13-7. Output configuration - Wired-AND with optional pull-up.**



## 13.4 Reading the Pin Value

Independent of the pin data direction, the pin value can be read from the IN register, as shown in Figure 13-1 on page 140. If the digital input is disabled, the pin value cannot be read. The IN register bit and the preceding flip-flop constitute a synchronizer. The synchronizer introduces a delay on the internal signal line. Figure 13-8 on page 143 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted as $t_{pd,max}$ and $t_{pd,min}$, respectively.

**Figure 13-8. Synchronization when reading a pin value.**

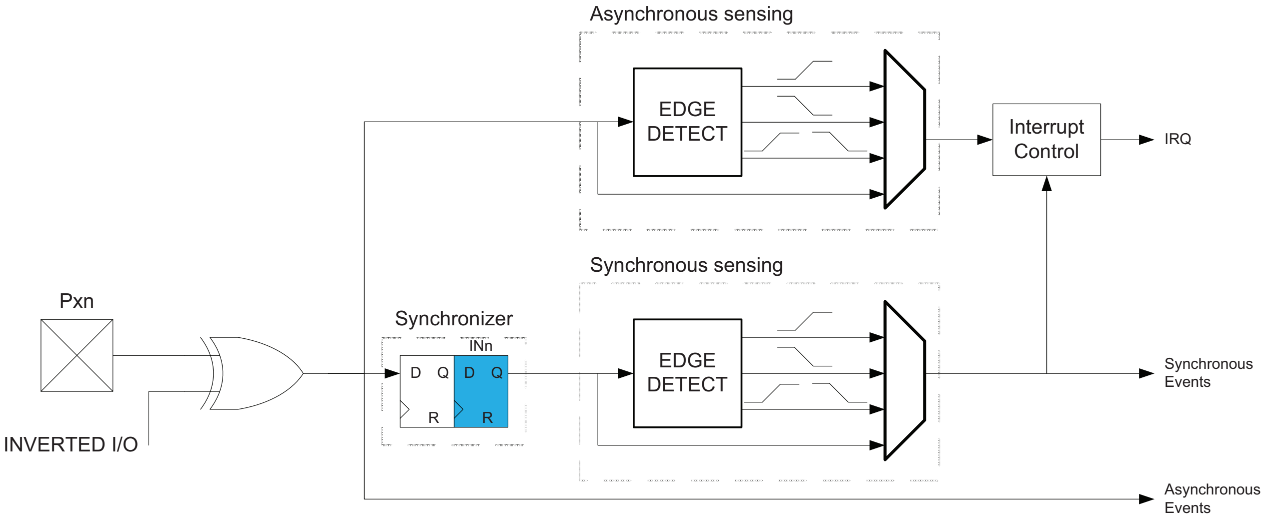


## 13.5 Input Sense Configuration

Input sensing is used to detect an edge or level on the I/O pin input. The different sense configurations that are available for each pin are detection of a rising edge, falling edge, or any edge or detection of a low level. High level can be detected by using the inverted input configuration. Input sensing can be used to trigger interrupt requests (IREQ) or events when there is a change on the pin.

The I/O pins support synchronous and asynchronous input sensing. Synchronous sensing requires the presence of the peripheral clock, while asynchronous sensing does not require any clock.

**Figure 13-9. Input sensing.**



## 13.6 Port Interrupt

Each port has two interrupt vectors, and it is configurable which pins on the port will trigger each interrupt. Port interrupts must be enabled before they can be used. Which sense configurations can be used to generate interrupts is dependent on whether synchronous or asynchronous input sensing is available for the selected pin.

For synchronous sensing, all sense configurations can be used to generate interrupts. For edge detection, the changed pin value must be sampled once by the peripheral clock for an interrupt request to be generated.

For asynchronous sensing, only port pin 2 on each port has full asynchronous sense support. This means that for edge detection, pin 2 will detect and latch any edge and it will always trigger an interrupt request. The other port pins have limited asynchronous sense support. This means that for edge detection, the changed value must be held until the device wakes up and a clock is present. If the pin value returns to its initial value before the end of the device wake-up time, the device will still wake up, but no interrupt request will be generated.

A low level can always be detected by all pins, regardless of a peripheral clock being present or not. If a pin is configured for low-level sensing, the interrupt will trigger as long as the pin is held low. In active mode, the low level must be held until the completion of the currently executing instruction for an interrupt to be generated. In all sleep modes, the low level must be kept until the end of the device wake-up time for an interrupt to be generated. If the low level disappears before the end of the wake-up time, the device will still wake up, but no interrupt will be generated.

Table 13-1, Table 13-2, and Table 13-3 on page 145 summarize when interrupts can be triggered for the various input sense configurations.

**Table 13-1. Synchronous sense support.**

| Sense settings | Supported | Interrupt description |
|---|---|---|
| Rising edge | Yes | Always triggered |
| Falling edge | Yes | Always triggered |
| Any edge | Yes | Always triggered |
| Low level | Yes | Pin level must be kept unchanged during wake up |

Table 13-2. Full asynchronous sense support.

| Sense settings | Supported | Interrupt description |
|---|---|---|
| Rising edge | Yes | Always triggered |
| Falling edge | Yes | Always triggered |
| Both edges | Yes | Always triggered |
| Low level | Yes | Pin level must be kept unchanged during wake up |

Table 13-3. Limited asynchronous sense support.

| Sense settings | Supported | Interrupt description |
|---|---|---|
| Rising edge | No | - |
| Falling edge | No | - |
| Any edge | Yes | Pin value must be kept unchanged during wake up |
| Low level | Yes | Pin level must be kept unchanged during wake up |

## 13.7 Port Event

Port pins can generate an event when there is a change on the pin. The sense configurations decide the conditions for each pin to generate events. Event generation requires the presence of a peripheral clock, and asynchronous event generation is not possible. For edge sensing, the changed pin value must be sampled once by the peripheral clock for an event to be generated.

For level sensing, a low-level pin value will not generate events, and a high-level pin value will continuously generate events. For events to be generated on a low level, the pin configuration must be set to inverted I/O.

Table 13-4. Event sense support.

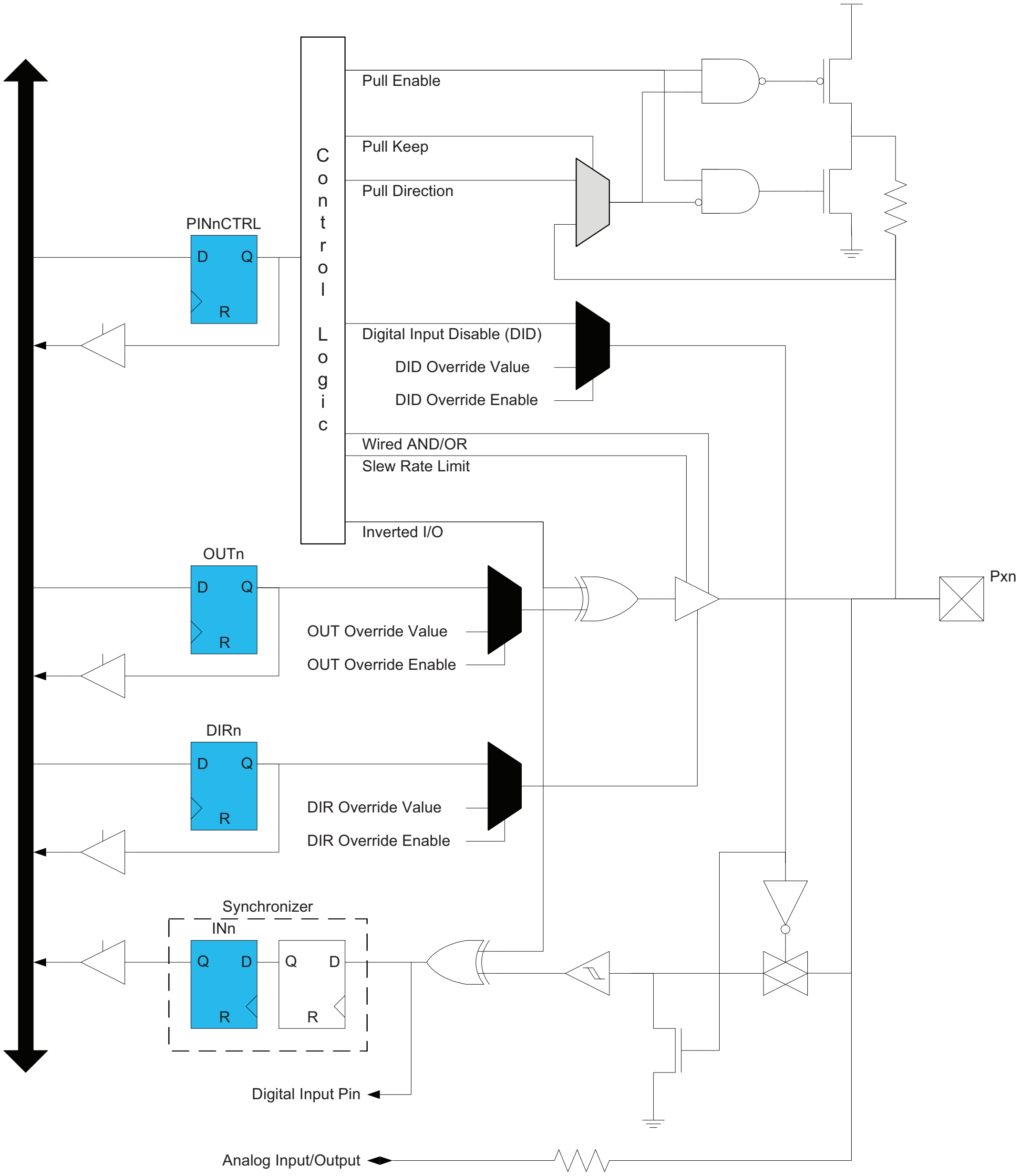
| Sense settings | Signal event | Data event |
|---|---|---|
| Rising edge | Rising edge | Pin value |
| Falling edge | Falling edge | Pin value |
| Both edge | Any edge | Pin value |
| Low level | Pin value | Pin value |

## 13.8 Alternate Port Functions

Most port pins have alternate pin functions in addition to being a general purpose I/O pin. When an alternate function is enabled, it might override the normal port pin function or pin value. This happens when other peripherals that require pins are enabled or configured to use pins. If and how a peripheral will override and use pins is described in the section for that peripheral.

The port override signals and related logic (grey) are shown in Figure 13-10 on page 146. These signals are not accessible from software, but are internal signals between the overriding peripheral and the port pin.

**Figure 13-10. Port override signals and related logic.**



## 13.9 Slew Rate Control

Slew rate control can be enabled for all I/O pins individually. Enabling the slew rate limiter will typically increase the rise/fall time by 50% to 150%, depending on operating conditions and load. For information about the characteristics of the slew rate limiter, please refer to the device datasheet.

## 13.10 Clock and Event Output

It is possible to output the peripheral clock and any of the event channels to the port pins (using EVCTRL register). This can be used to clock, control, and synchronize external functions and hardware to internal device timing. The output port pin is selectable. If an event occurs, it remains visible on the port pin as long as the event lasts; normally one peripheral clock cycle.

## 13.11 Multi-pin configuration

The multi-pin configuration function is used to configure multiple port pins using a single write operation to only one of the port pin configuration registers. A mask register decides which port pin is configured when one port pin register is written, while avoiding several pins being written the same way during identical write operations.

## 13.12 Virtual Ports

Virtual port registers allow the port registers to be mapped virtually in the bit-accessible I/O memory space. When this is done, writing to the virtual port register will be the same as writing to the real port register. This enables the use of I/O memory-specific instructions, such as bit-manipulation instructions, on a port register that normally resides in the extended I/O memory space. There are four virtual ports, and so four ports can be mapped at the same time.

## 13.13 Register Descriptions – Ports

### 13.13.1 DIR – Data Direction register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x00 | DIR[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – DIR[7:0]: Data Direction**
  This register sets the data direction for the individual pins of the port. If DIRn is written to one, pin n is configured as an output pin. If DIRn is written to zero, pin n is configured as an input pin.

### 13.13.2 DIRSET – Data Direction Set register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x01 | DIRSET[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – DIRSET[7:0]: Port Data Direction Set**
  This register can be used instead of a read-modify-write to set individual pins as output. Writing a one to a bit will set the corresponding bit in the DIR register. Reading this register will return the value of the DIR register.

### 13.13.3 DIRCLR – Data Direction Clear register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x02 | DIRCLR[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – DIRCLR[7:0]: Port Data Direction Clear**
  This register can be used instead of a read-modify-write to set individual pins as input. Writing a one to a bit will clear the corresponding bit in the DIR register. Reading this register will return the value of the DIR register.

### 13.13.4 DIRTGL – Data Direction Toggle register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x03 | DIRTGL[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – DIRTGL[7:0]: Port Data Direction Toggle**
  This register can be used instead of a read-modify-write to toggle the direction of individual pins. Writing a one to a bit will toggle the corresponding bit in the DIR register. Reading this register will return the value of the DIR register.

### 13.13.5 OUT – Data Output Value register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x04 | OUT[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – OUT[7:0]: Port Data Output value**
  This register sets the data output value for the individual pins of the port. If OUTn is written to one, pin n is driven high. If OUTn is written to zero, pin n is driven low. For this setting to have any effect, the pin direction must be set as output.

### 13.13.6 OUTSET – Data Output Value Set register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x05 | OUTSET[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – OUTSET[7:0]: Data Output Value Set**
  This register can be used instead of a read-modify-write to set the output value of individual pins to one. Writing a one to a bit will set the corresponding bit in the OUT register. Reading this register will return the value in the OUT register.

### 13.13.7 OUTCLR – Data Output Value Clear Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x06 | OUTCLR[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – OUTCLR[7:0]: Data Output Value Clear**
  This register can be used instead of a read-modify-write to set the output value of individual pins to zero. Writing a one to a bit will clear the corresponding bit in the OUT register. Reading this register will return the value in the OUT register.

### 13.13.8 OUTTGL – Data Output Value Toggle register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x07 | OUTTGL[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – OUTTGL[7:0]: Port Data Output Value Toggle**
  This register can be used instead of a read-modify-write to toggle the output value of individual pins. Writing a one to a bit will toggle the corresponding bit in the OUT register. Reading this register will return the value in the OUT register.

### 13.13.9 IN – Data Input Value register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x08 | IN[7:0] | | | | | | | |
| Read/Write | R | R | R | R | R | R | R | R |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – IN[7:0]: Data Input Value**
  This register shows the value present on the pins if the digital input driver is enabled. INn shows the value of pin n of the port. The input is not sampled and cannot be read if the digital input buffers are disabled.

### 13.13.10 INTCTRL – Interrupt Control register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x09 | – | – | – | – | INT1LVL[1:0] | | INT0LVL[1:0] | |
| Read/Write | R | R | R | R | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:4 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 3:2/1:0 – INTnLVL[1:0]: Interrupt n Level**
  These bits enable port interrupt n and select the interrupt level as described in "Interrupts and Programmable Multilevel Interrupt Controller" on page 131.

### 13.13.11 INT0MASK – Interrupt 0 Mask register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x0A | INT0MSK[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – INT0MSK[7:0]: Interrupt 0 Mask Bits**
  These bits are used to mask which pins can be used as sources for port interrupt 0. If INT0MASKn is written to one, pin n is used as source for port interrupt 0.The input sense configuration for each pin is decided by the PINnCTRL registers.

### 13.13.12 INT1MASK – Interrupt 1 Mask register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x0B | INT1MSK[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – INT1MASK[7:0]: Interrupt 1 Mask Bits**
  These bits are used to mask which pins can be used as sources for port interrupt 1. If INT1MASKn is written to one, pin n is used as source for port interrupt 1.The input sense configuration for each pin is decided by the PINnCTRL registers.

### 13.13.13 INTFLAGS – Interrupt Flag register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x0C | – | – | – | – | – | – | INT1IF | INT0IF |
| Read/Write | R | R | R | R | R | R | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:2 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 1:0 – INTnIF: Interrupt n Flag**
  The INTnIF flag is set when a pin change/state matches the pin's input sense configuration, and the pin is set as source for port interrupt n. Writing a one to this flag's bit location will clear the flag. For enabling and executing the interrupt, refer to the interrupt level description.

### 13.13.14 REMAP – Pin Remap register

The pin remap functionality is available for PORTC - PORTF only.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x0E | – | – | SPI | USART0 | TC0D | TC0C | TC0B | TC0A |
| Read/Write | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:6 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 5 – SPI: SPI Remap**
  Setting this bit to one will swap the pin locations of the SCK and MOSI pins to have pin compatibility between SPI and USART when the USART is operating as a SPI master.

- **Bit 4 – USART0: USART0 Remap**
  Setting this bit to one will move the pin location of USART0 from Px[3:0] to Px[7:4].

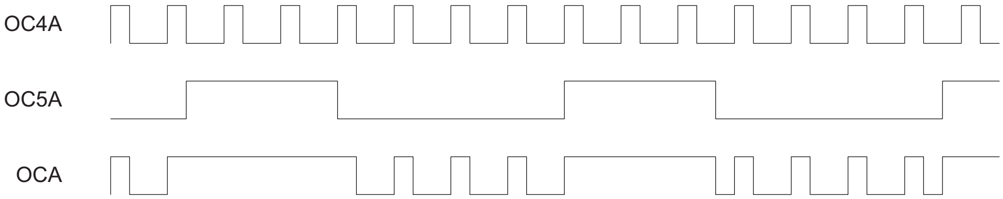
- **Bit 3 – TC0D: Timer/Counter 0 Output Compare D**
  Setting this bit will move the location of OC0D from Px3 to Px7.

- **Bit 2 – TC0C: Timer/Counter 0 Output Compare C**
  Setting this bit will move the location of OC0C from Px2 to Px6.

- **Bit 1 – TC0B: Timer/Counter 0 Output Compare B**
  Setting this bit will move the location of OC0B from Px1 to Px5. If this bit is set and PWM from both timer/counter 0 and timer/counter 1 is enabled, the resulting PWM will be an OR-modulation between the two PWM outputs.

- **Bit 0 – TC0A: Timer/Counter 0 Output Compare A**
  Setting this bit will move the location of OC0A from Px0 to Px4. If this bit is set and PWM from both timer/counter 0 and timer/counter 1 is enabled, the resulting PWM will be an OR-modulation between the two PWM outputs. See Figure 13-11.

Figure 13-11.I/O timer/counter.



### 13.13.15 PINnCTRL – Pin n Configuration register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7 – SRLEN: Slew Rate Limit Enable**
  Setting this bit will enable slew rate limiting on pin n.

- **Bit 6 – INVEN: Inverted I/O Enable**
  Setting this bit will enable inverted output and input data on pin n.

- **Bit 5:3 – OPC: Output and Pull Configuration**
  These bits set the output/pull configuration on pin n according to Table 13-5 on page 152.

Table 13-5. Output/pull configuration.

| | | Description | |
|---|---|---|---|
| OPC[2:0] | Group configuration | Output configuration | Pull configuration |
| 000 | TOTEM | Totem-pole | (N/A) |
| 001 | BUSKEEPER | Totem-pole | Bus-keeper |
| 010 | PULLDOWN | Totem-pole | Pull-down (on input) |
| 011 | PULLUP | Totem-pole | Pull-up (on input) |
| 100 | WIREDOR | Wired-OR | (N/A) |
| 101 | WIREDAND | Wired-AND | (N/A) |
| 110 | WIREDORPULL | Wired-OR | Pull-down |
| 111 | WIREDANDPULL | Wired-AND | Pull-up |

- **Bit 2:0 – ISC[2:0]: Input/Sense Configuration**
  These bits set the input and sense configuration on pin n according to Table 13-6. The sense configuration decides how the pin can trigger port interrupts and events. If the input buffer is disabled, the input cannot be read in the IN register.

**Table 13-6. Input/sense configuration.**

| ISC[2:0] | Group configuration | Description |
|---|---|---|
| 000 | BOTHEDGES | Sense both edges |
| 001 | RISING | Sense rising edge |
| 010 | FALLING | Sense falling edge |
| 011 | LEVEL | Sense low level[1] |
| 100 | | Reserved |
| 101 | | Reserved |
| 110 | | Reserved |
| 111 | INTPUT_DISABLE | Digital input buffer disabled[2] |

Notes: 1. A low-level pin value will not generate events, and a high-level pin value will continuously generate events.

2. Only PORTA - PORTF support the input buffer disable option. If the pin is used for analog functionality, such as AC or ADC, it is recommended to configure the pin to INPUT_DISABLE.

## 13.14 Register Descriptions – Port Configuration

### 13.14.1 MPCMASK – Multi-pin Configuration Mask register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x00 | MPCMASK[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – MPCMASK[7:0]: Multi-pin Configuration Mask**
  The MPCMASK register enables configuration of several pins of a port at the same time. Writing a one to bit n makes pin n part of the multi-pin configuration. When one or more bits in the MPCMASK register is set, writing any of the PINnCTRL registers will update only the PINnCTRL registers matching the mask in the MPCMASK register for that port. The MPCMASK register is automatically cleared after any PINnCTRL register is written.

### 13.14.2 VPCTRLA – Virtual Port-map Control register A

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x02 | VP1MAP[3:0] | | | | VP0MAP[3:0] | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:4 – VP1MAP: Virtual Port 1 Mapping**
  These bits decide which ports should be mapped to Virtual Port 1. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 13-7 on page 154 for configuration.

- **Bit 3:0 – VP0MAP: Virtual Port 0 Mapping**
  These bits decide which ports should be mapped to Virtual Port 0. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 13-7 on page 154 for configuration.

### 13.14.3 VPCTRLB – Virtual Port-map Control register B

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x03 | VP3MAP[3:0] | | | | VP2MAP[3:0] | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:4 – VP3MAP: Virtual Port 3 Mapping**
  These bits decide which ports should be mapped to Virtual Port 3. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 13-7 on page 154 for configuration.
- **Bit 3:0 – VP2MAP: Virtual Port 2 Mapping**
  These bits decide which ports should be mapped to Virtual Port 2. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 13-7 on page 154 for configuration.

**Table 13-7. Virtual port mapping.**

| VPnMAP[3:0] | Group configuration | Description |
|---|---|---|
| 0000 | PORTA | PORTA mapped to Virtual Port n |
| 0001 | PORTB | PORTB mapped to Virtual Port n |
| 0010 | PORTC | PORTC mapped to Virtual Port n |
| 0011 | PORTD | PORTD mapped to Virtual Port n |
| 0100 | PORTE | PORTE mapped to Virtual Port n |
| 0101 | PORTF | PORTF mapped to Virtual Port n |
| 0110 | PORTG | PORTG mapped to Virtual Port n |
| 0111 | PORTH | PORTH mapped to Virtual Port n |
| 1000 | PORTJ | PORTJ mapped to Virtual Port n |
| 1001 | PORTK | PORTK mapped to Virtual Port n |
| 1010 | PORTL | PORTL mapped to Virtual Port n |
| 1011 | PORTM | PORTM mapped to Virtual Port n |
| 1100 | PORTN | PORTN mapped to Virtual Port n |
| 1101 | PORTP | PORTP mapped to Virtual Port n |
| 1110 | PORTQ | PORTQ mapped to Virtual Port n |
| 1111 | PORTR | PORTR mapped to Virtual Port n |

### 13.14.4 CLKEVOUT – Clock and Event Out register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x04 | CLKEVPIN | RTCOUT | EVOUT[1:0] | | CLKOUTSEL[1:0] | | CLKOUT[1:0] | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7 – CLKEVPIN: Clock and Event Output Pin Select**
  Setting this pin enables output of clock and event pins on port pin 4 instead of port pin 7.
- **Bit 6 – RTCOUT: RTC Clock Output Enable**
  Setting this bit enables output of the RTC clock source on PORTC pin 6.
- **Bit 5:4 – EVOUT[1:0]: Event Output Port**
  These bits decide which port event channel 0 from the event system will be output to. Pin 7 on the selected port is the default used, and the CLKOUT bits must be set differently from those of EVOUT. The port pin must be configured as output for the event to be available on the pin.

  Table 13-8 on page 155 shows the possible configurations.

**Table 13-8. Event output pin selection.**

| EVOUT[1:0] | Group configuration | Description |
|---|---|---|
| 00 | OFF | Event output disabled |
| 01 | PC | Event channel 0 output on PORTC |
| 10 | PD | Event channel 0 output on PORTD |
| 11 | PE | Event channel 0 output on PORTE |

- **Bits 3:2 – CLKOUTSEL[1:0]: Clock Output Select**
  These bits are used to select which of the peripheral clocks will be output to the port pin if CLKOUT is configured.

**Table 13-9. Event output clock selection.**

| CLKOUTSEL[1:0] | Group configuration | Description |
|---|---|---|
| 00 | CLK1X | $CLK_{PER}$ output to pin |
| 01 | CLK2X | $CLK_{PER2}$ output to pin |
| 10 | CLK4X | $CLK_{PER4}$ output to pin |
| | | |

- **Bit 1:0 – CLKOUT[1:0]: Clock Output Port**
  These bits decide which port the peripheral clock will be output to. Pin 7 on the selected port is the default used. The CLKOUT setting will override the EVOUT setting. Thus, if both are enabled on the same port pin, the peripheral clock will be visible. The port pin must be configured as output for the clock to be available on the pin.

  Table 13-10 on page 155 shows the possible configurations.

**Table 13-10. Clock output port configurations.**

| CLKOUT[1:0] | Group configuration | Description |
|---|---|---|
| 00 | OFF | Clock output disabled |
| 01 | PC | Clock output on PORTC |
| 10 | PD | Clock output on PORTD |
| 11 | PE | Clock output on PORTE |

### 13.14.5 EBIOUT – EBI Output register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x05 | – | – | – | – | EBIADROUT[1:0] | | EBICSOUT[1:0] | |
| Read/Write | R | R | R | R | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:4 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 3:2 – EBIADROUT[1:0]: EBI Address Output**
  The maximum configuration of the external bus interface (EBI) requires up to 32 dedicated pins. For devices with only 24 EBI pins available, eight additional pins can be enabled and placed on alternate pin locations in order to get a full 32-pin EBI. The port pins must be configured as output for signals to be available on the pins. These bits are available on devices with only three ports dedicated for the EBI interface. The selections are valid only if the EBI is configured to operate in four-port mode.

.

**Table 13-11. EBI address output port selection.**

| EBIADROUT[1:0] | Group configuration | Description |
|---|---|---|
| 00 | PF | EBI port 3 address output on PORTF pins 0 to 7 |
| 01 | PE | EBI port 3 address output on PORTE pins 0 to 7 |
| 10 | PFH | EBI port 3 address output on PORTF pins 4 to 7 |
| 11 | PEH | EBI port 3 address output on PORTE pins 4 to 7 |

**Table 13-12. EBI address output**

| EBIADROUT | SDRAM | SRAM or SRAM LPC<br>(with SDRAM on CS3) | SRAM<br>NOALE or ALE1 |
|---|---|---|---|
| 00 or 01 | 4’h0, A[11:8] | A[23:16] | A[15:8] |
| 10 or 11 | A[11:8] | [19:16] | – |

- **Bit 1:0 – EBICSOUT[1:0]: EBI Chip Select Output**
  These bits decide which port the EBI chip select signals will be output to. The pins must be configured as output pins for signals to be available on the pins. Refer to “Register Description – EBI” on page 329 for chip select configuration.

**Table 13-13. EBI chip select port selection.**

| EBICSOUT[1:0] | Group configuration | Description |
|---|---|---|
| 00 | PH | EBI chip select output to PORTH pin 4 to 7 |
| 01 | PL | EBI chip select output to PORTL pin 4 to 7 |
| 10 | PF | EBI chip select output to PORTF pin 4 to 7 |
| 11 | PE | EBI chip select output to PORTE pin 4 to 7 |

### 13.14.6 EVCTRL – Event Control register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x06 | – | – | – | – | – | EVOUTSEL[2:0] | | |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:3 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 2:0 – EVOUTSEL[2:0]: Event Channel Output Selection**
  These bits define which channel from the event system is output to the port pin. Table 13-14 on page 157 shows the available selections.

**Table 13-14. Event channel output selection.**

| EVOUTSEL[2:0] | Group configuration | Description |
|---|---|---|
| 000 | 0 | Event channel 0 output to pin |
| 001 | 1 | Event channel 1 output to pin |
| 010 | 2 | Event channel 2 output to pin |
| 011 | 3 | Event channel 3 output to pin |
| 100 | 4 | Event channel 4 output to pin |
| 101 | 5 | Event channel 5 output to pin |
| 110 | 6 | Event channel 6 output to pin |
| 111 | 7 | Event channel 7 output to pin |

## 13.15 Register Descriptions – Virtual Port

### 13.15.1 DIR – Data Direction register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x00 | DIR[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – DIR[7:0]: Data Direction**
  This register sets the data direction for the individual pins in the port mapped by VPCTRLA, virtual port-map control register A or VPCTRLB, virtual port-map control register B. When a port is mapped as virtual, accessing this register is identical to accessing the actual DIR register for the port.

### 13.15.2 OUT – Data Output Value register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x01 | OUT[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – OUT[7:0]: Data Output value**
  This register sets the data output value for the individual pins in the port mapped by VPCTRLA, virtual port-map control register A or VPCTRLB, virtual port-map control register B. When a port is mapped as virtual, accessing this register is identical to accessing the actual OUT register for the port.

### 13.15.3 IN – Data Input Value register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x02 | IN[7:0] | | | | | | | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:0 – IN[7:0]: Data Input Value**
  This register shows the value present on the pins if the digital input buffer is enabled. The configuration of VPCTRLA, virtual port-map control register A or VPCTRLB, virtual port-map control register A, decides the value in the register. When a port is mapped as virtual, accessing this register is identical to accessing the actual IN register for the port.

### 13.15.4 INTFLAGS – Interrupt Flag register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| +0x03 | – | – | – | – | – | – | INT1IF | INT0IF |
| Read/Write | R | R | R | R | R | R | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Bit 7:2 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 1:0 – INTnIF: Interrupt n Flag**
  The INTnIF flag is set when a pin change/state matches the pin's input sense configuration, and the pin is set as source for port interrupt n. Writing a one to this flag's bit location will clear the flag. For enabling and executing the interrupt, refer to the interrupt level description. The configuration of VPCTRLA, virtual port-map control register A, or VPCTRLB, Virtual Port-map Control Register B,, decides which flags are mapped. When a port is mapped as virtual, accessing this register is identical to accessing the actual INTFLAGS register for the port.

## 13.16 Register summary – Ports

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| +0x00 | DIR | DIR[7:0] | | | | | | | | 148 |
| +0x01 | DIRSET | DIRSET[7:0] | | | | | | | | 148 |
| +0x02 | DIRCLR | DIRCLR[7:0] | | | | | | | | 148 |
| +0x03 | DIRTGL | DIRTGL[7:0] | | | | | | | | 148 |
| +0x04 | OUT | OUT[7:0] | | | | | | | | 149 |
| +0x05 | OUTSET | OUTSET[7:0] | | | | | | | | 149 |
| +0x06 | OUTCLR | OUTCLR[7:0] | | | | | | | | 149 |
| +0x07 | OUTTGL | OUTTGL[7:0] | | | | | | | | 149 |
| +0x08 | IN | IN[7:0] | | | | | | | | 150 |
| +0x09 | INTCTRL | – | – | – | – | INT1LVL[1:0] | | INT0LVL[1:0] | | 150 |
| +0x0A | INT0MASK | INT0MSK[7:0] | | | | | | | | 150 |
| +0x0B | INT1MASK | INT1MSK[7:0] | | | | | | | | 150 |
| +0x0C | INTFLAGS | – | – | – | – | – | – | INT1IF | INT0IF | 151 |
| +0x0D | Reserved | – | – | – | – | – | – | – | – | |
| +0x0E | REMAP | – | – | SPI | USART0 | TC0D | TC0C | TC0B | TC0A | 151 |
| +0x0F | Reserved | – | – | – | – | – | – | – | – | |
| +0x10 | PIN0CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x11 | PIN1CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x12 | PIN2CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x13 | PIN3CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x14 | PIN4CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x15 | PIN5CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x16 | PIN6CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x17 | PIN7CTRL | SRLEN | INVEN | OPC[2:0] | | | ISC[2:0] | | | 152 |
| +0x18 | Reserved | – | – | – | – | – | – | – | – | |
| +0x19 | Reserved | – | – | – | – | – | – | – | – | |
| +0x1A | Reserved | – | – | – | – | – | – | – | – | |
| +0x1B | Reserved | – | – | – | – | – | – | – | – | |
| +0x1C | Reserved | – | – | – | – | – | – | – | – | |
| +0x1D | Reserved | – | – | – | – | – | – | – | – | |
| +0x1E | Reserved | – | – | – | – | – | – | – | – | |
| +0x1F | Reserved | – | – | – | – | – | – | – | – | |

## 13.17 Register summary – Port Configuration

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| +0x00 | MPCMASK | MPCMASK[7:0] | | | | | | | | 153 |
| +0x01 | Reserved | – | – | – | – | – | – | – | – | |
| +0x02 | VPCTRLA | VP1MAP[3:0] | | | | VP0MAP[3:0] | | | | 153 |
| +0x03 | VPCTRLB | VP3MAP[3:0] | | | | VP2MAP[3:0] | | | | 154 |
| +0x04 | CLKEVOUT | CLKEVPIN | RTCOUT | EVOUT[1:0] | | CLKOUTSEL | | CLKOUT[1:0] | | 154 |
| +0x05 | EBIOUT | – | – | – | – | EBIADROUT[1:0] | | EBICSOUT[1:0] | | 156 |
| +0x06 | EVCTRL | – | – | – | – | – | EVCTRL[2:0] | | | 157 |

## 13.18 Register summary – Virtual Ports

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| +0x00 | DIR | DIR[7:0] | | | | | | | | 158 |
| +0x01 | OUT | OUT[7:0] | | | | | | | | 158 |
| +0x02 | IN | IN[7:0] | | | | | | | | 158 |
| +0x03 | INTFLAGS | – | – | – | – | – | – | INT1IF | INT0IF | 158 |

## 13.19 Interrupt vector summary – Ports

**Table 13-15. Port interrupt vectors and their word offset address.**

| Offset | Source | Interrupt description |
|---|---|---|
| 0x00 | INT0_vect | Port interrupt vector 0 offset |
| 0x02 | INT1_vect | Port interrupt vector 1 offset |