1) INTRODUCTION

The general-purpose input and output (GPIO) module communicates to the processor core via a zero wait state interface for maximum pin performance. The GPIO registers support 8-bit, 16-bit or 32-bit accesses.

The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function. The GPIO input data register displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt module for that pin is enabled. Efficient bit manipulation of the general-purpose outputs is supported through the addition of set, clear, and toggle write-only registers for each port output data register.

2) FEATURES

Features of the GPIO module include:
- Pin input data register visible in all digital pin-multiplexing modes
- Pin output data register with corresponding set/clear/toggle registers
- Pin data direction register
- Zero wait state access to GPIO registers through IOPORT

3) MODE OF OPERATION

Modes of operation	Description
Run	The GPIO module operates normally.
Wait	The GPIO module operates normally.
Stop	The GPIO module is disabled.
Debug	The GPIO module operates normally.

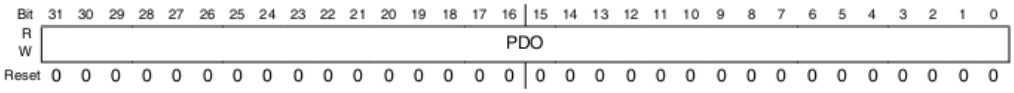
4) GPIO SIGNAL DESCRIPTION

GPIO signal descriptions	Description	I/O
PORTA31–PORTA0	General-purpose input/output	I/O
PORTB31–PORTB0	General-purpose input/output	I/O
PORTC31–PORTC0	General-purpose input/output	I/O
PORTD31–PORTD0	General-purpose input/output	I/O
PORTE31–PORTE0	General-purpose input/output	I/O

5) REGISTER DESCRIPTION

5.1) Port Data Output Register (GPIOx_PDOR)

Address: Base address + 0h offset

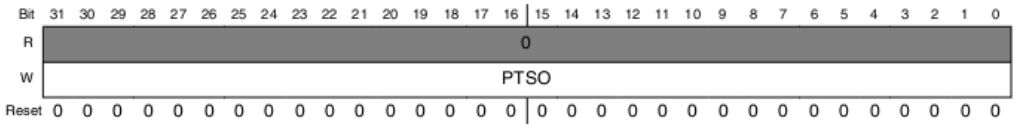


GPIOx_PDOR field descriptions

Field	Description
31–0 PDO	Port Data Output Register bits for un-bonded pins return a undefined value when read. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

5.2) Port Set Output Register (GPIOx_PSOR)

Address: Base address + 4h offset

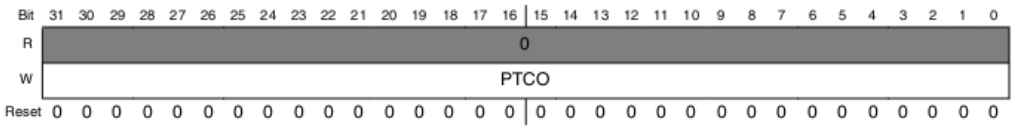


GPIOx_PSOR field descriptions

Field	Description
31–0 PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

5.3) Port Clear Output Register (GPIOx_PCOR)

Address: Base address + 8h offset

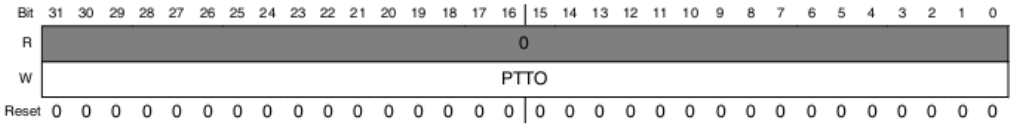


GPIOx_PCOR field descriptions

Field	Description
31–0 PTCO	Port Clear Output Writing to this register will update the contents of the corresponding bit in the Port Data Output Register (PDOR) as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

5.4) Port Toggle Output Register (GPIOx_PTOR)

Address: Base address + Ch offset

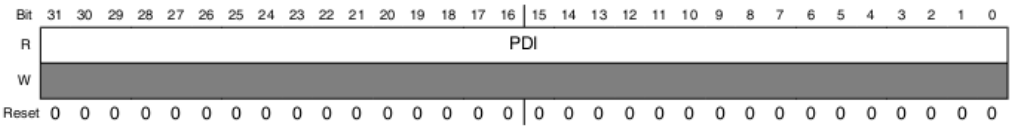


GPIOx_PTOR field descriptions

Field	Description
31–0 PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

5.5) Port Data input Register (GPIOx_PDIR)

Address: Base address + 10h offset

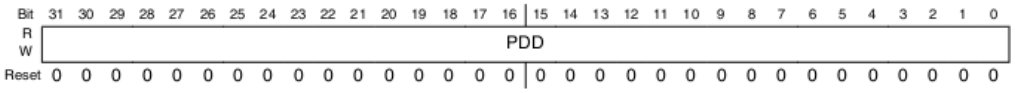


GPIOx_PDIR field descriptions

Field	Description
31–0 PDI	Port Data Input Reads 0 at the unimplemented pins for a particular device. Pins that are not configured for a digital function read 0. If the Port Control and Interrupt module is disabled, then the corresponding bit in PDIR does not update. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

5.6) Port Data Direction Register (GPIOx_PDDR)

Address: Base address + 14h offset



GPIOx_PDDR field descriptions

Field	Description
31–0 PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

6) FGPIO REGISTER DESCRIPTION

Basically FGPIO register set are the same with GPIO register set

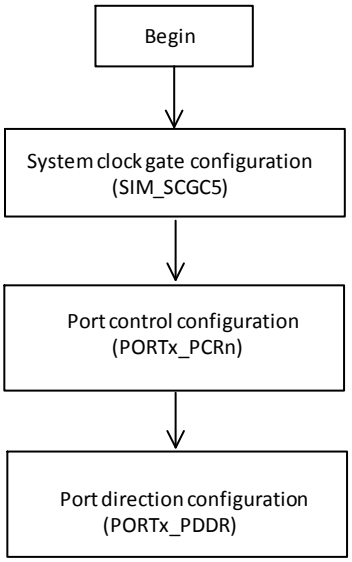
7) KEIL DEFINITION

```
/** GPIO - Register Layout Typedef */
typedef struct {
  __IO uint32_t PDOR;                   /**< Port Data Output Register, offset: 0x0 */
  __O  uint32_t PSOR;                   /**< Port Set Output Register, offset: 0x4 */
  __O  uint32_t PCOR;                   /**< Port Clear Output Register, offset: 0x8 */
  __O  uint32_t PTOR;                   /**< Port Toggle Output Register, offset: 0xC */
  __I  uint32_t PDIR;                   /**< Port Data Input Register, offset: 0x10 */
```

```
  __IO uint32_t PDDR;                    /**< Port Data Direction Register, offset: 0x14 */
} GPIO_Type;

/** FGPIO - Register Layout Typedef */
typedef struct {
  __IO uint32_t PDOR;                    /**< Port Data Output Register, offset: 0x0 */
  __O  uint32_t PSOR;                    /**< Port Set Output Register, offset: 0x4 */
  __O  uint32_t PCOR;                    /**< Port Clear Output Register, offset: 0x8 */
  __O  uint32_t PTOR;                    /**< Port Toggle Output Register, offset: 0xC */
  __I  uint32_t PDIR;                    /**< Port Data Input Register, offset: 0x10 */
  __IO uint32_t PDDR;                    /**< Port Data Direction Register, offset: 0x14 */
} FGPIO_Type;
```

7) GPIO INITIALIZATION



Example : Configure PORTB.18 as gpio output

```
#include <stdio.h>
#include "MKL25Z4.h"

int main (void)
{
      SIM->SCGC5      |= 1UL << 10;   /* Open clock for PORTB  */
      PORTB->PCR[18] |= 1UL <<  8;   /* PORTB.18 as GPIO      */
      PTB->PDDR       |= 1UL << 18;  /* PORTB.18 as output    */
      return 0;
}
```