CP316
Serial Communication-I2C

Terry Sturtevant

Wilfrid Laurier University

February 11, 2019

Serial Communication -I^2C

Serial Communication -I^2C

- Inter-Integrated Circuit Interface

Serial Communication -I^2C

- Inter-Integrated Circuit Interface
- Master/slave communication

Serial Communication -I^2C

- Inter-Integrated Circuit Interface
- Master/slave communication
- Uses 2 signals (and Ground),
  SDA and SCL

Serial Communication -I^2C

- Inter-Integrated Circuit Interface
- Master/slave communication
- Uses 2 signals (and Ground),
  SDA and SCL
- Many slaves can be on the same bus since each has an address

Serial Communication -I^2C

- Inter-Integrated Circuit Interface
- Master/slave communication
- Uses 2 signals (and Ground),

  SDA and SCL
- Many slaves can be on the same bus since each has an address

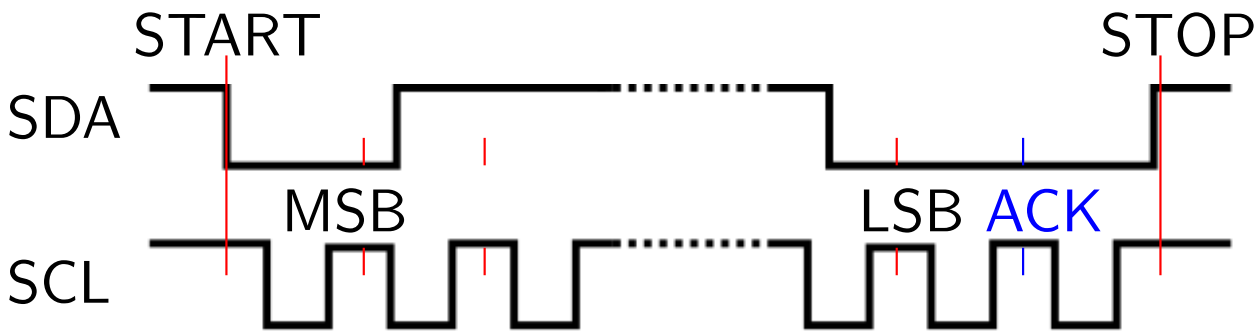
  Device addresses are pre-programmed, but can usually be changed

Serial Communication -I^2C

- Inter-Integrated Circuit Interface
- Master/slave communication
- Uses 2 signals (and Ground),

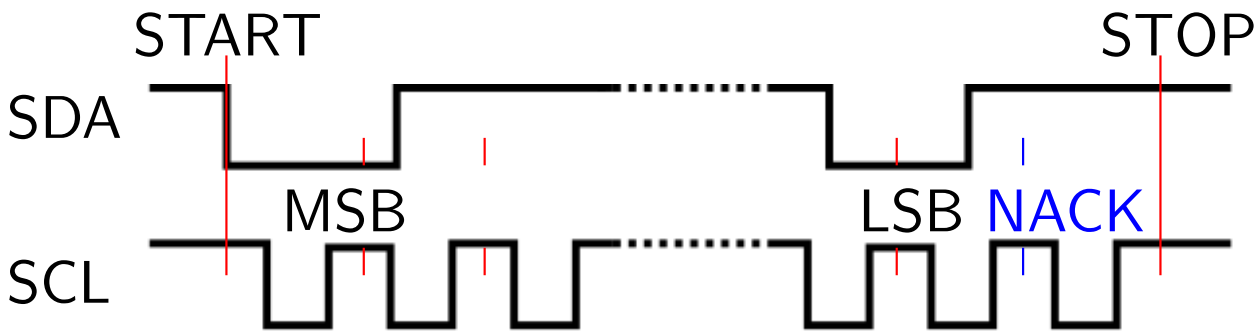
  SDA and SCL
- Many slaves can be on the same bus since each has an address

  Device addresses are pre-programmed, but can usually be changed
- Synchronous, so master controls clock rate



- I^2C ; bits are read when SCL is HIGH
- ACK is sent by receiver if OK

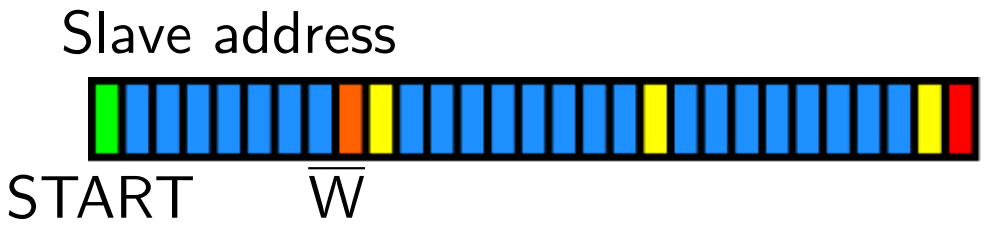
  sender must release SDA after LSB



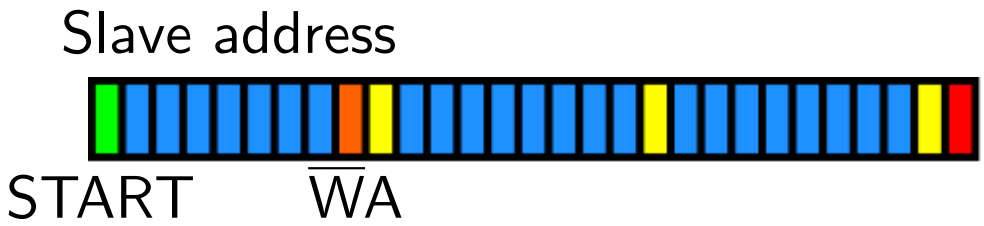
- I^2C ; bits are read when SCL is HIGH
- NACK is sent by master-receiver if OK

  sender must release SDA after LSB

- I^2C write to slave register



- I^2C write to slave register



- I^2C write to slave register

- I^2C write to slave register

- I^2C write to slave register

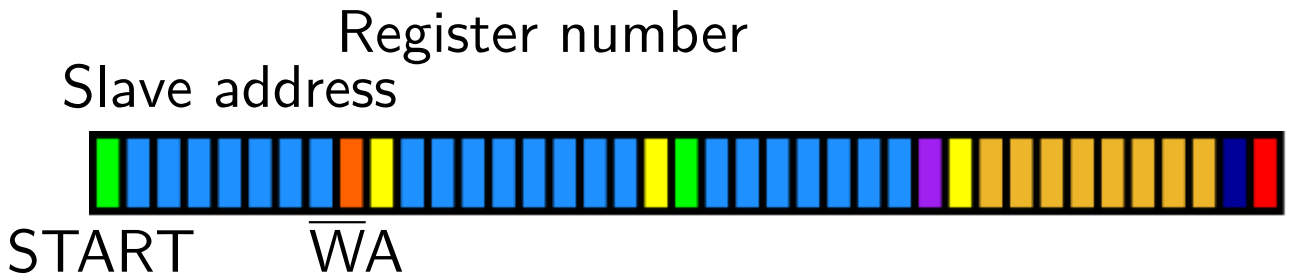
- I^2C write to slave register

- I^2C write to slave register

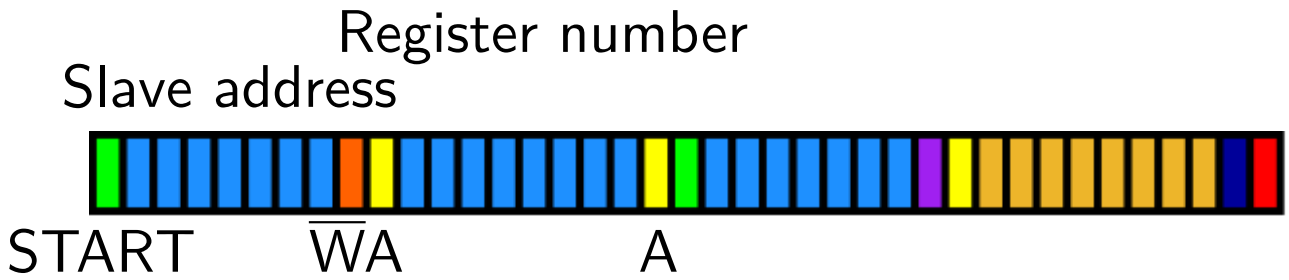
- I^2C read from slave register

- I^2C read from slave register

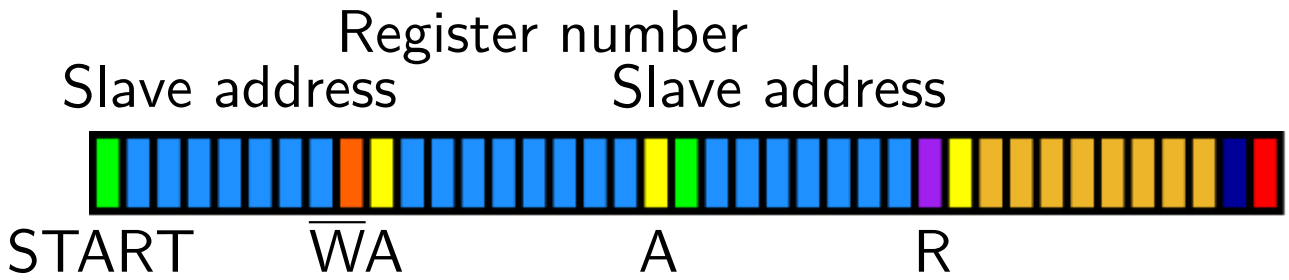
- I^2C read from slave register



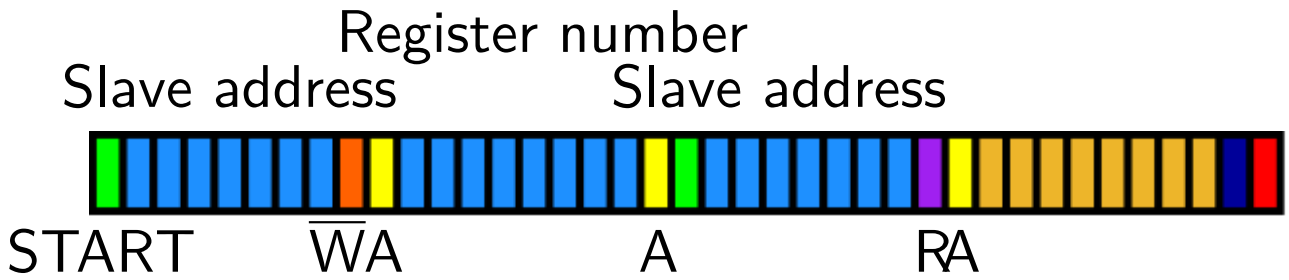
- I^2C read from slave register



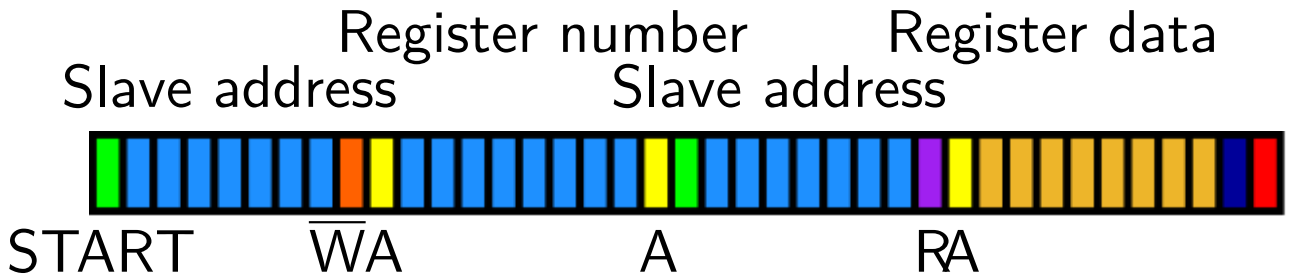
- I^2C read from slave register



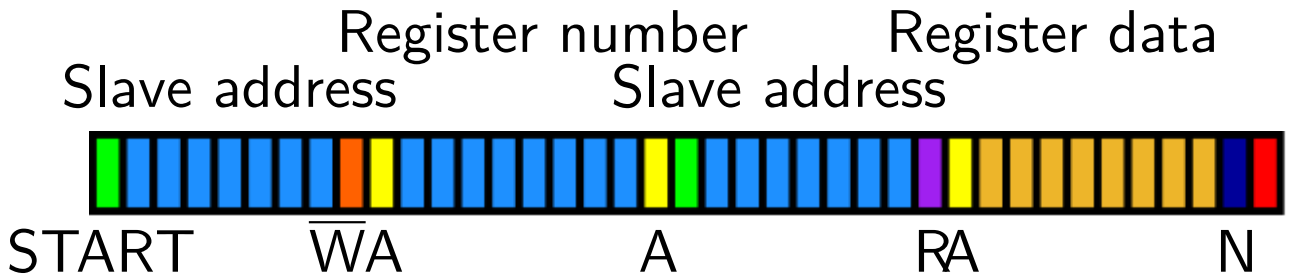
- I^2C read from slave register



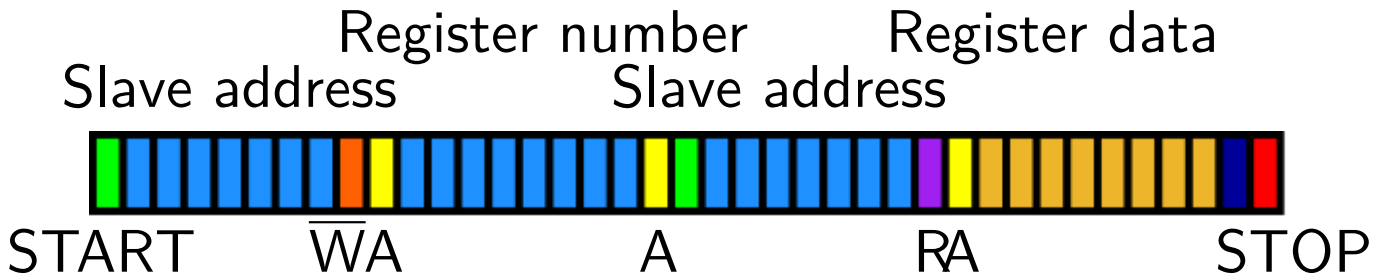
- I^2C read from slave register



- I^2C read from slave register



- I^2C read from slave register



- I^2C read from slave register

Introduction

Introduction

QwikFlash modules

Introduction

QwikFlash modules

ramifications???

Introduction

QwikFlash modules

ramifications???

interrupts; transmit and receive

Introduction

QwikFlash modules

ramifications???

interrupts; transmit and receive

$\rightarrow$ Sections 6.4.5 to 6.4.7

Introduction

QwikFlash modules

ramifications???

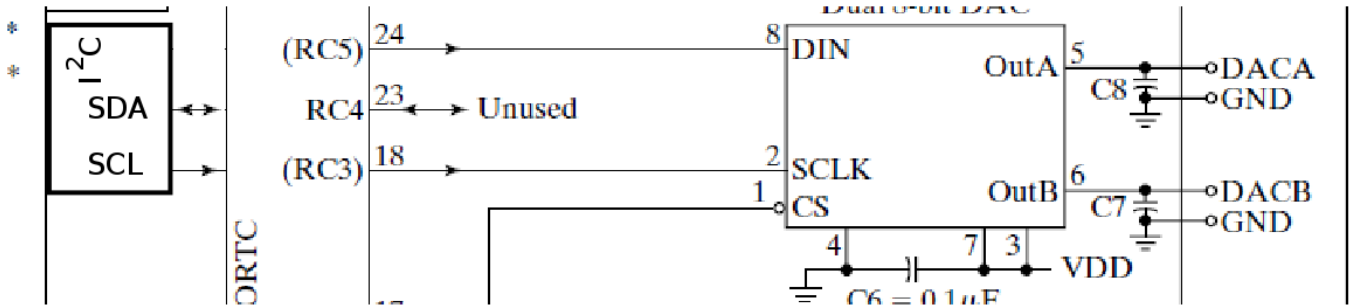
interrupts; transmit and receive

→ Sections 6.4.5 to 6.4.7

→ Section 8.2

QwikFlash I^2C connections

QwikFlash I^2C connections



Master Synchronous Serial Port (MSSP) module

Master Synchronous Serial Port (MSSP) module

2 modes; **SPI** and $\mathrm{I}^2\mathrm{C}$

Master Synchronous Serial Port (MSSP) module

2 modes; **SPI** and I^2C

$\rightarrow$ Section 15.0 -15.2

I^2C

I^2C

2 wires, master-multiple slave

I^2C

2 wires, master-multiple slave

overview

I^2C

2 wires, master-multiple slave

overview

$\rightarrow$ Section 11.2

I^2C

2 wires, master-multiple slave

overview

$\rightarrow$ Section 11.2

I^2C module

I^2C

2 wires, master-multiple slave

overview

$\rightarrow$ Section 11.2

I^2C module

$\rightarrow$ Sections 11.3 to 11.6

I^2C

2 wires, master-multiple slave

overview

→ Section 11.2

I^2C module

→ Sections 11.3 to 11.6

→ Section 15.4

I^2C summary

I^2C summary

2 wires (+ ground), one-to-many

I^2C summary

2 wires (+ ground), one-to-many

SCL (from master)

I^2C summary

2 wires (+ ground), one-to-many

SCL (from master)

SDA (serial data)

I^2C summary

2 wires (+ ground), one-to-many

SCL (from master)

SDA (serial data)

data transmission rate set by SCL

I^2C summary

2 wires (+ ground), one-to-many

SCL (from master)

SDA (serial data)

data transmission rate set by SCL

address for each device, preset (but possibly programmable)

I^2C summary

2 wires (+ ground), one-to-many

SCL (from master)

SDA (serial data)

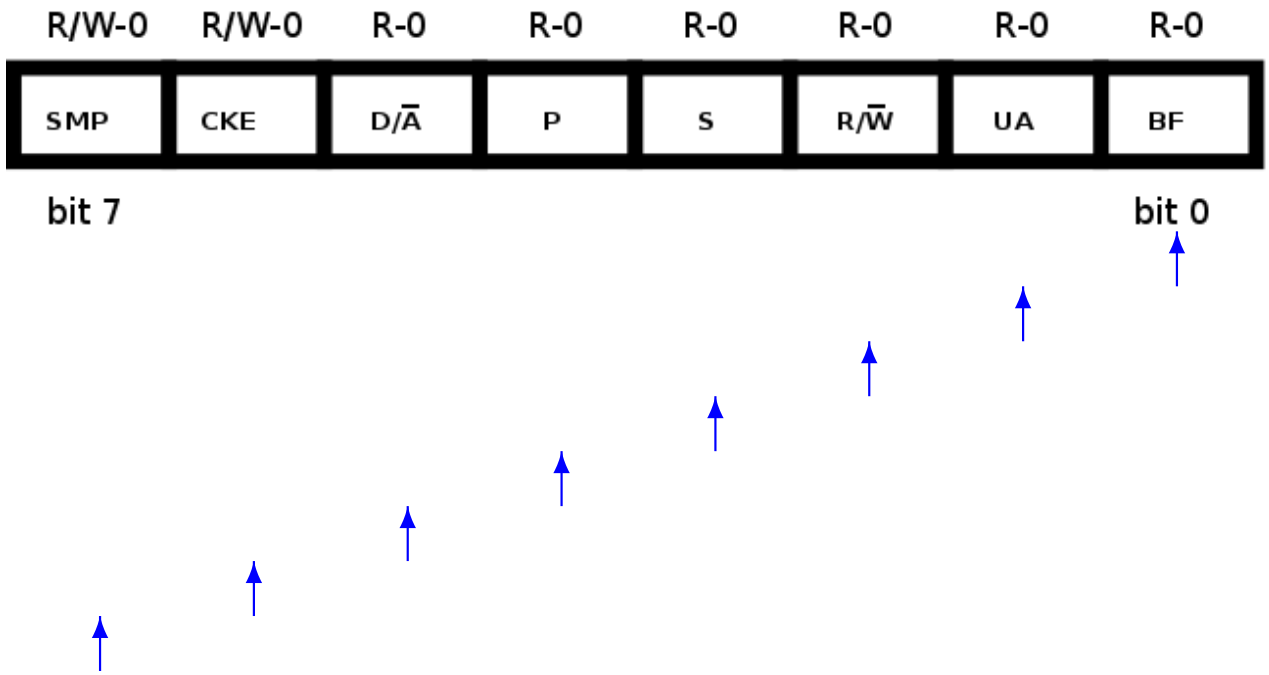
data transmission rate set by SCL

address for each device, preset (but possibly programmable)

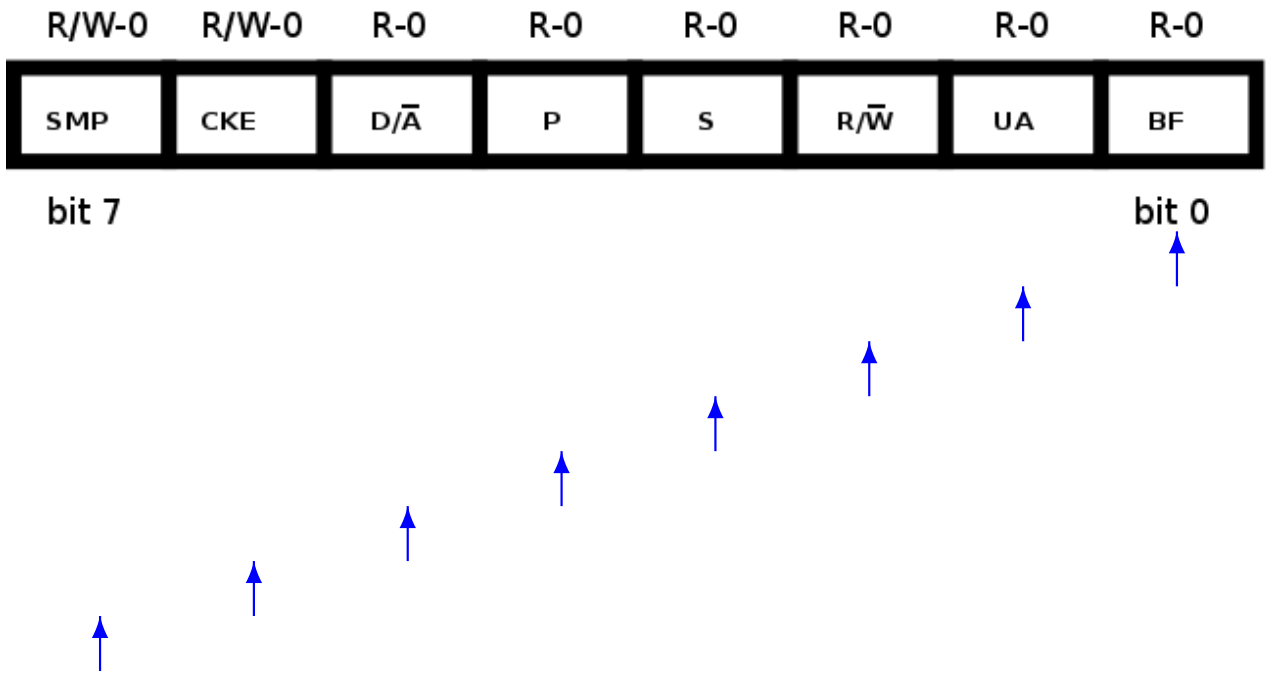
packets are complex; address of recipient, read or write, data (variable number of bytes)

SSPSTAT

SSPSTAT



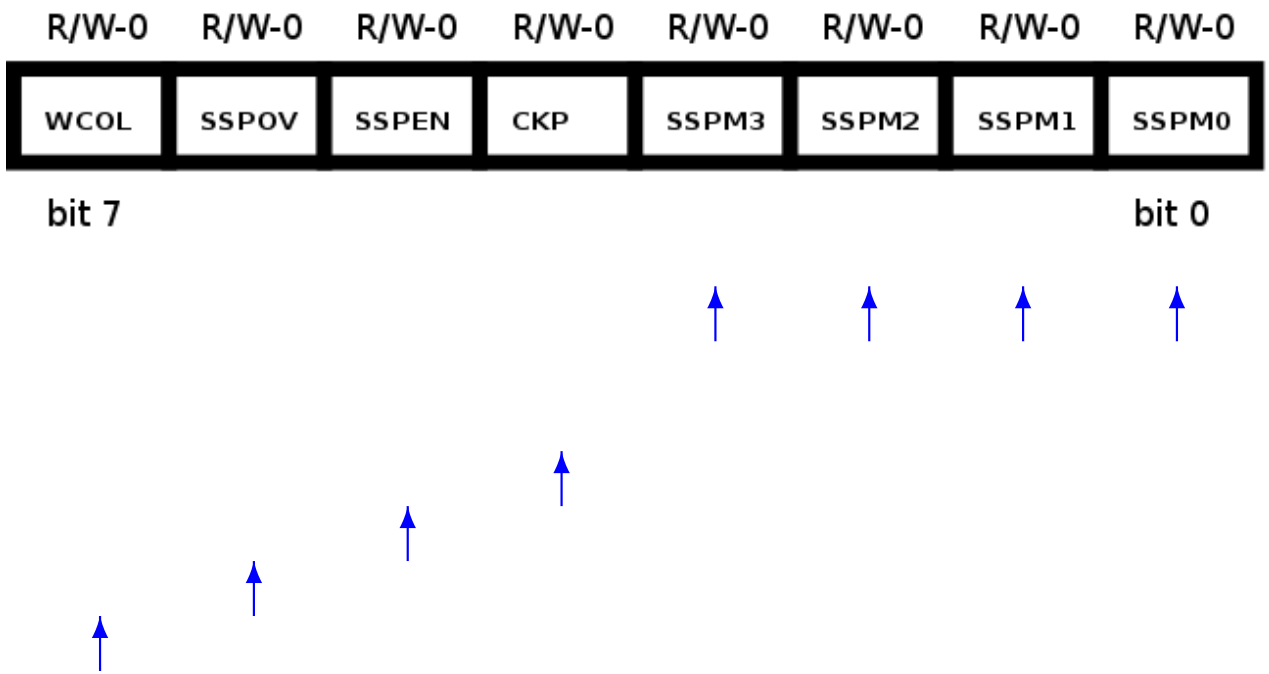
SSPSTAT



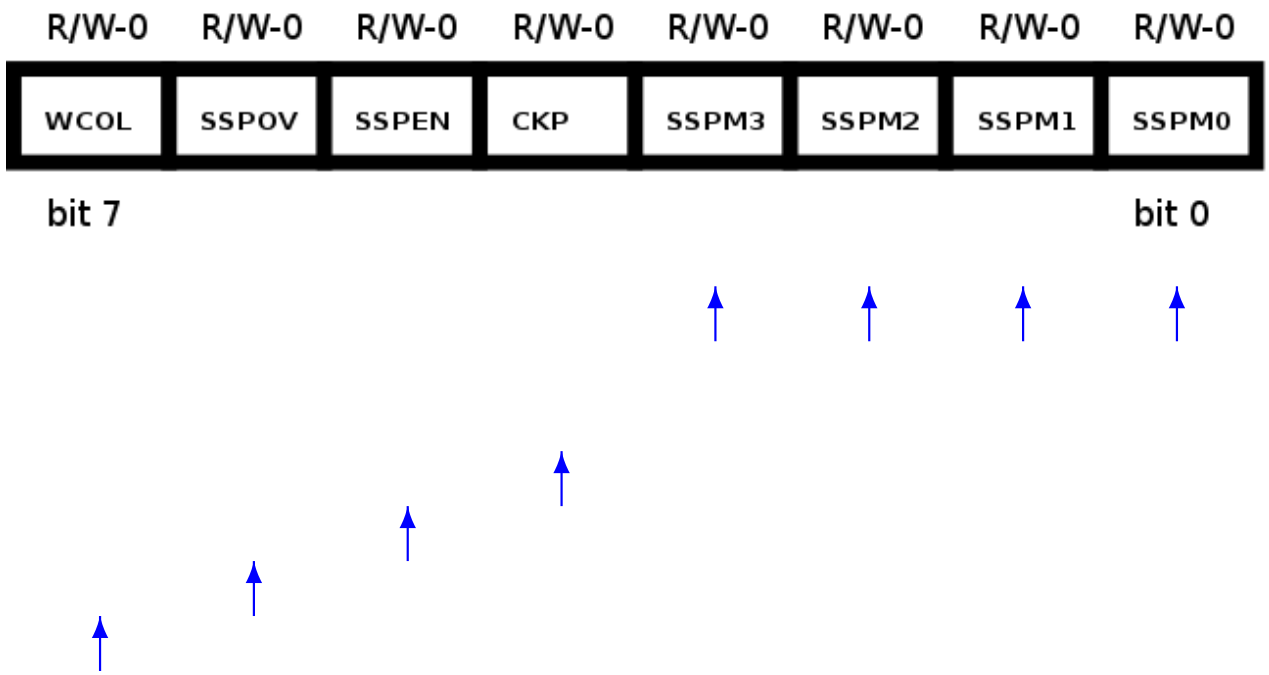
Bits in SSPSTAT register

SSPCON1

SSPCON1


R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
WCOL
SSPOV
SSPEN
CKP
SSPM3
SSPM2
SSPM1
SSPM0
bit 7
bit 0

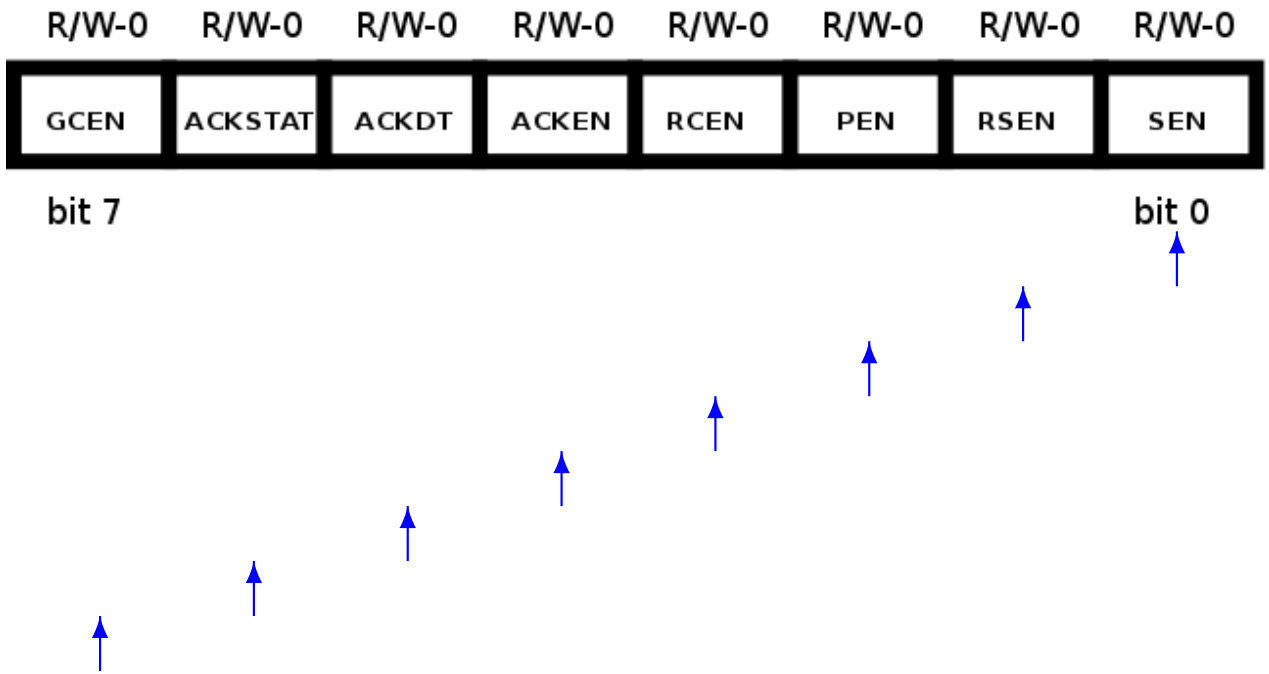
SSPCON1



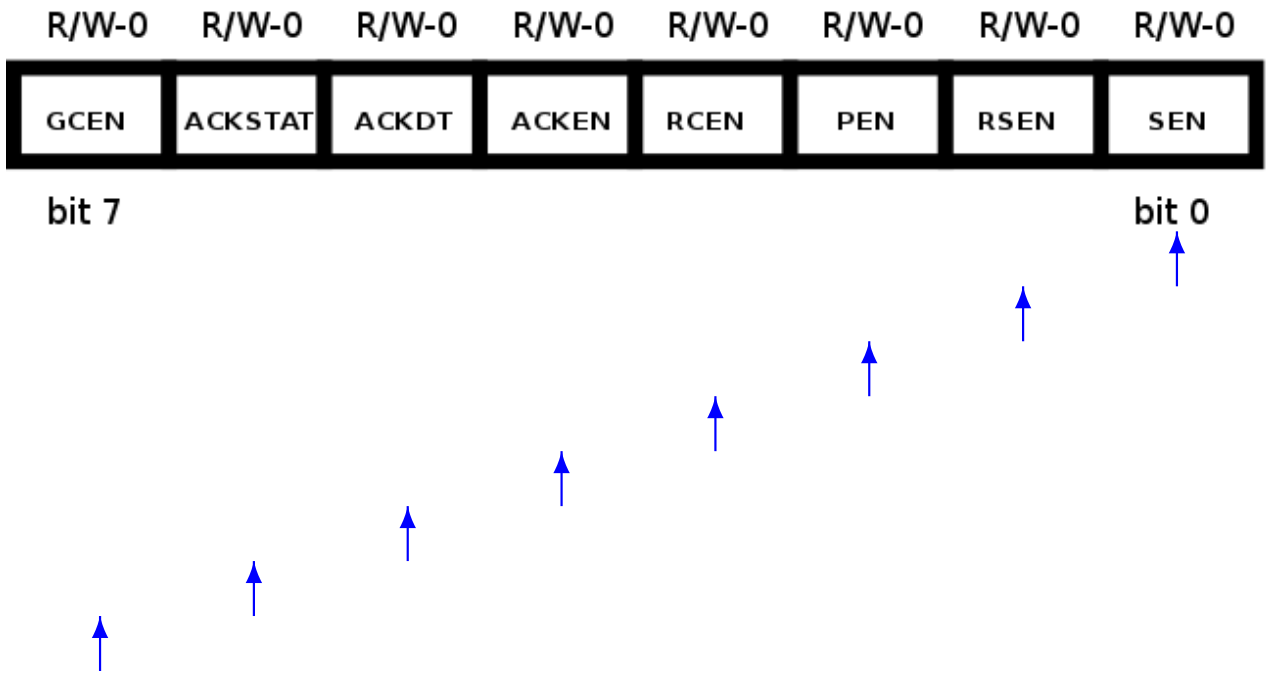
Bits in SSPCON1 register

SSPCON2

SSPCON2


R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
GCEN
ACKSTAT
ACKDT
ACKEN
RCEN
PEN
RSEN
SEN
bit 7
bit 0

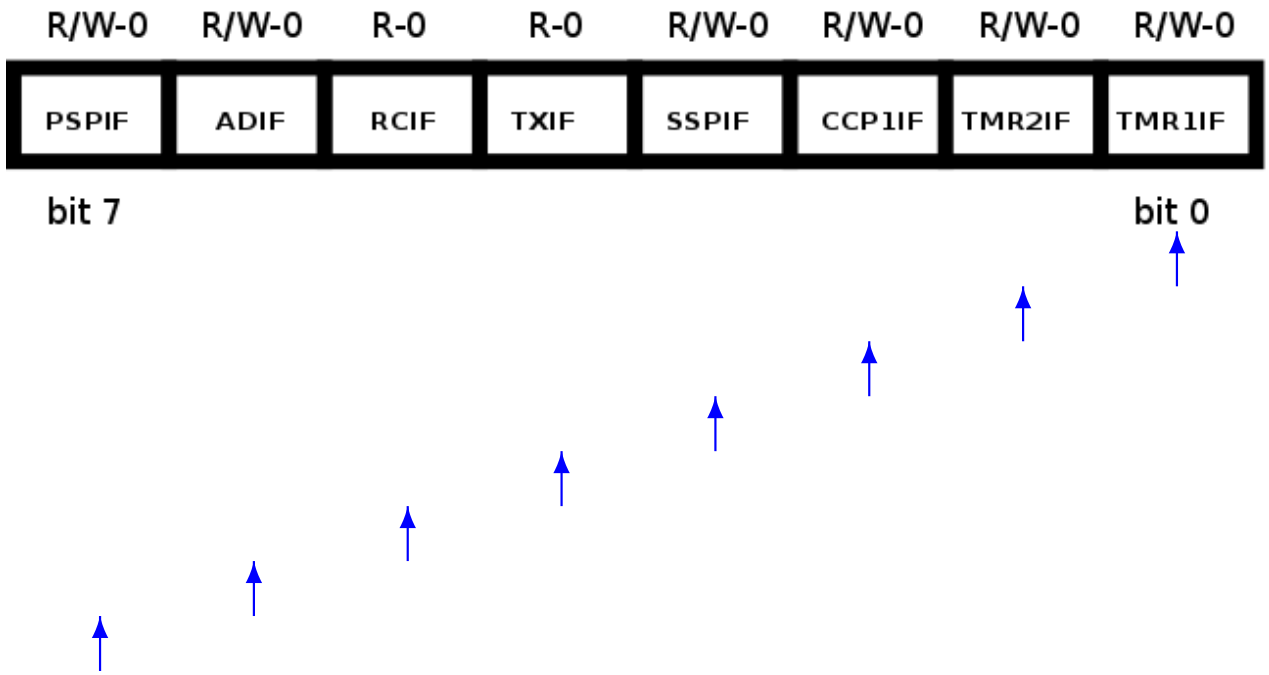
SSPCON2



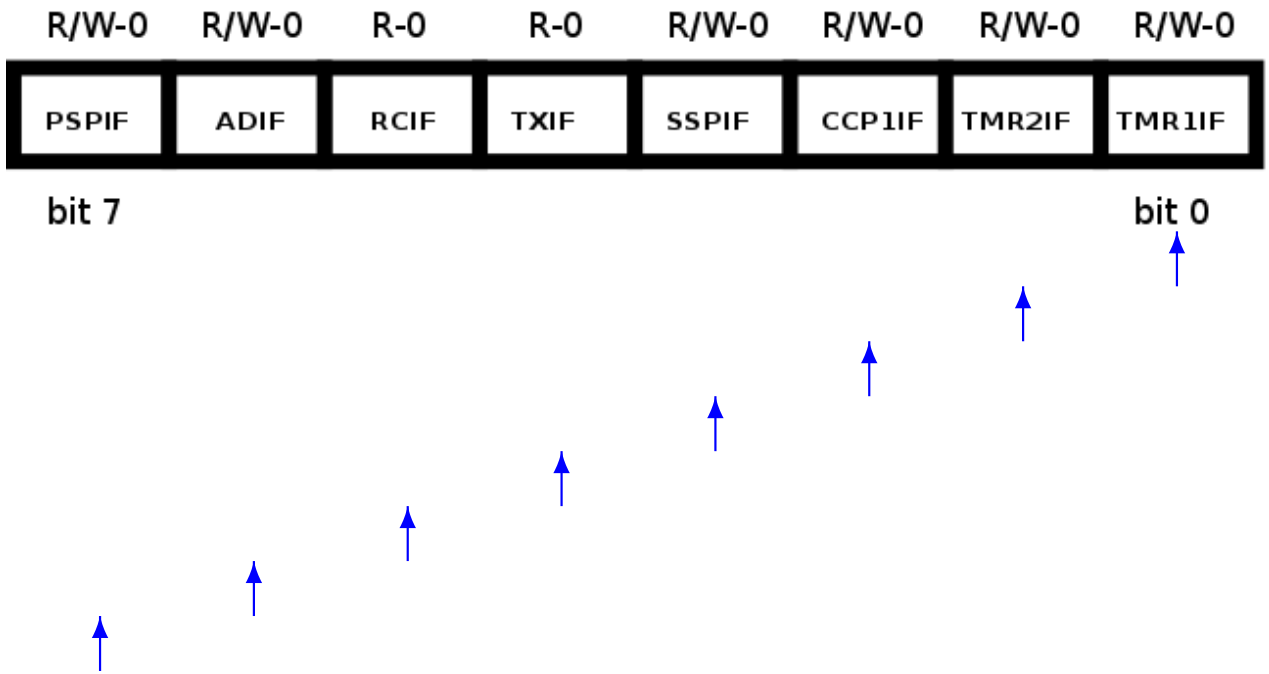
Bits in SSPCON2 register

PIR1

PIR1

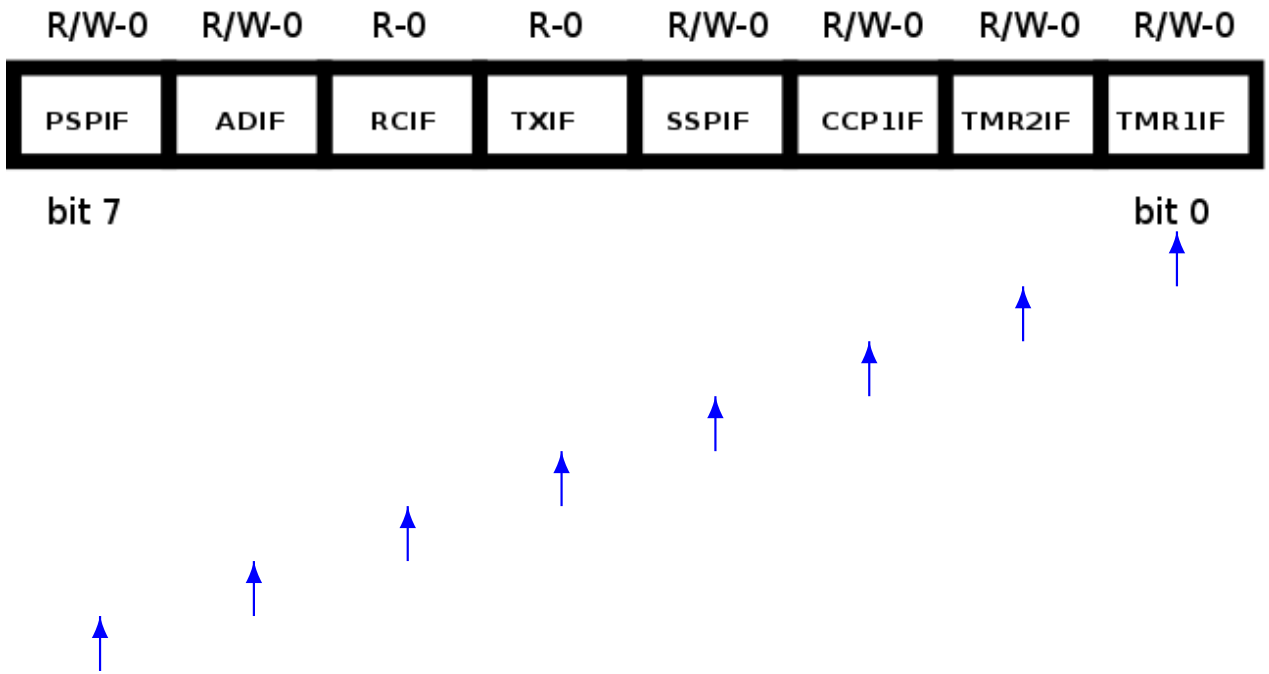

R/W-0
R/W-0
R-0
R-0
R/W-0
R/W-0
R/W-0
R/W-0
PSPIF
ADIF
RCIF
TXIF
SSPIF
CCP1IF
TMR2IF
TMR1IF
bit 7
bit 0

PIR1



Bits in PIR1 register

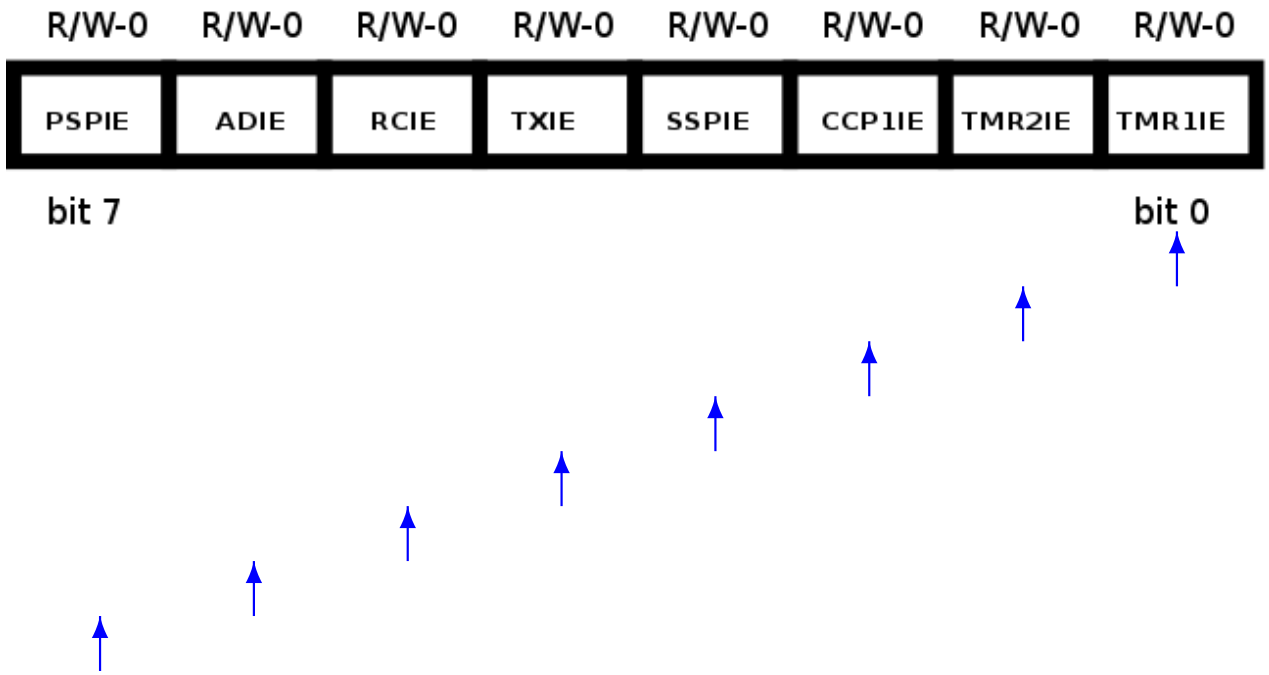
PIR1



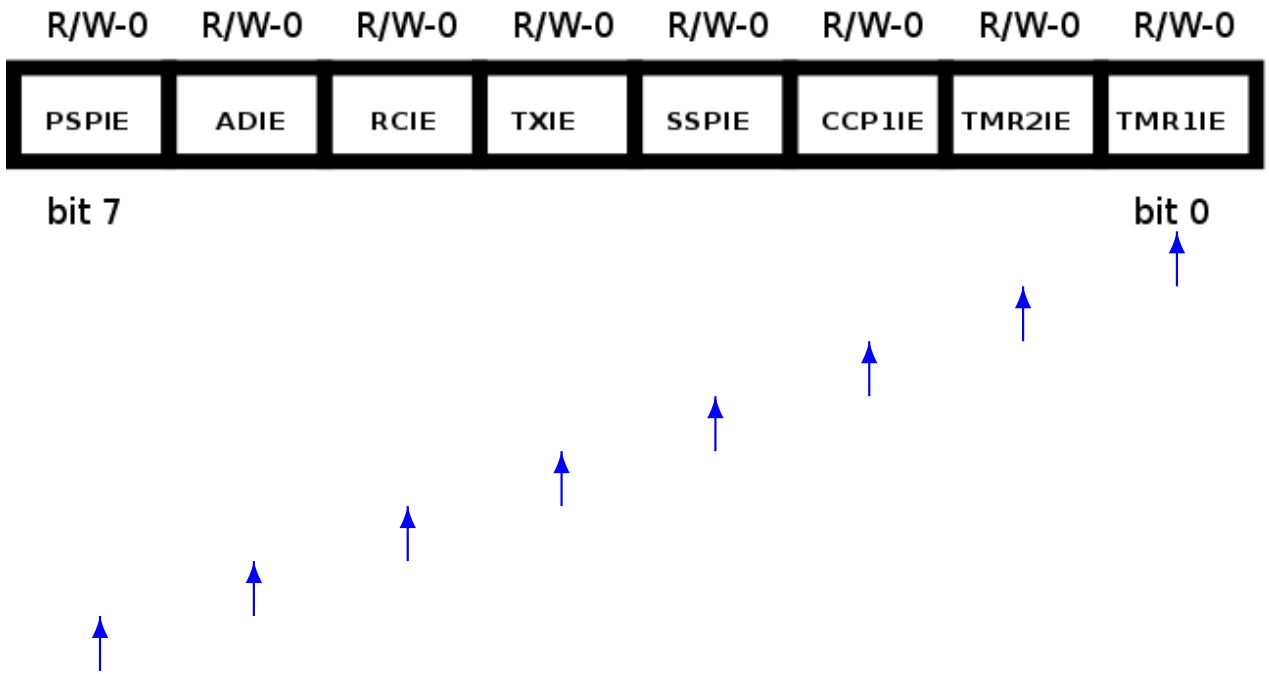
Bits in PIR1 register - Note SSPIF

PIE1

PIE1

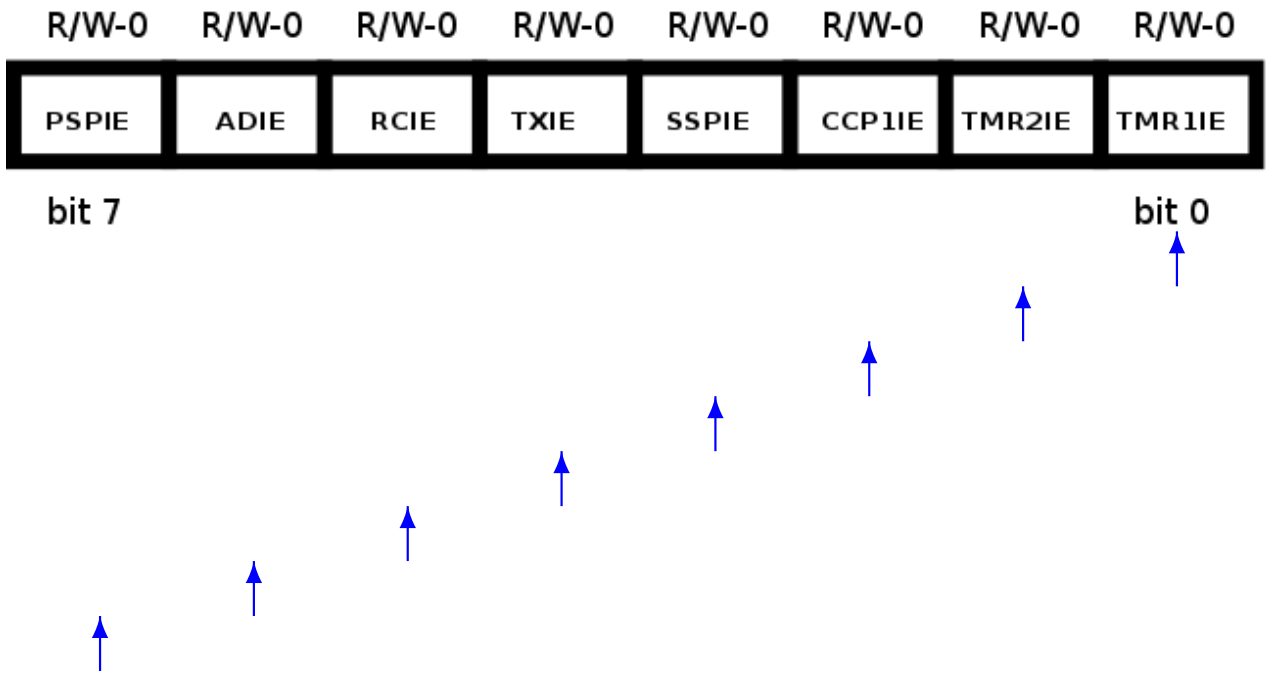

R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
R/W-0
PSPIE
ADIE
RCIE
TXIE
SSPIE
CCP1IE
TMR2IE
TMR1IE
bit 7
bit 0

PIE1



Bits in PIE1 register

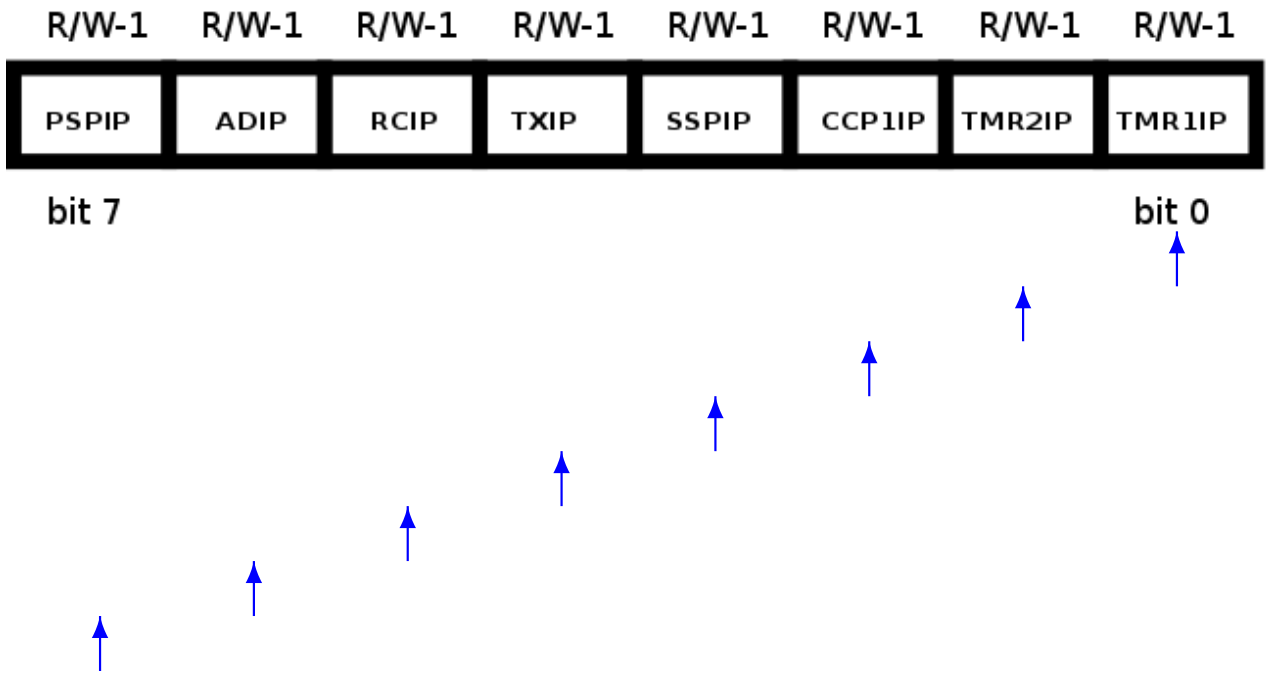
PIE1



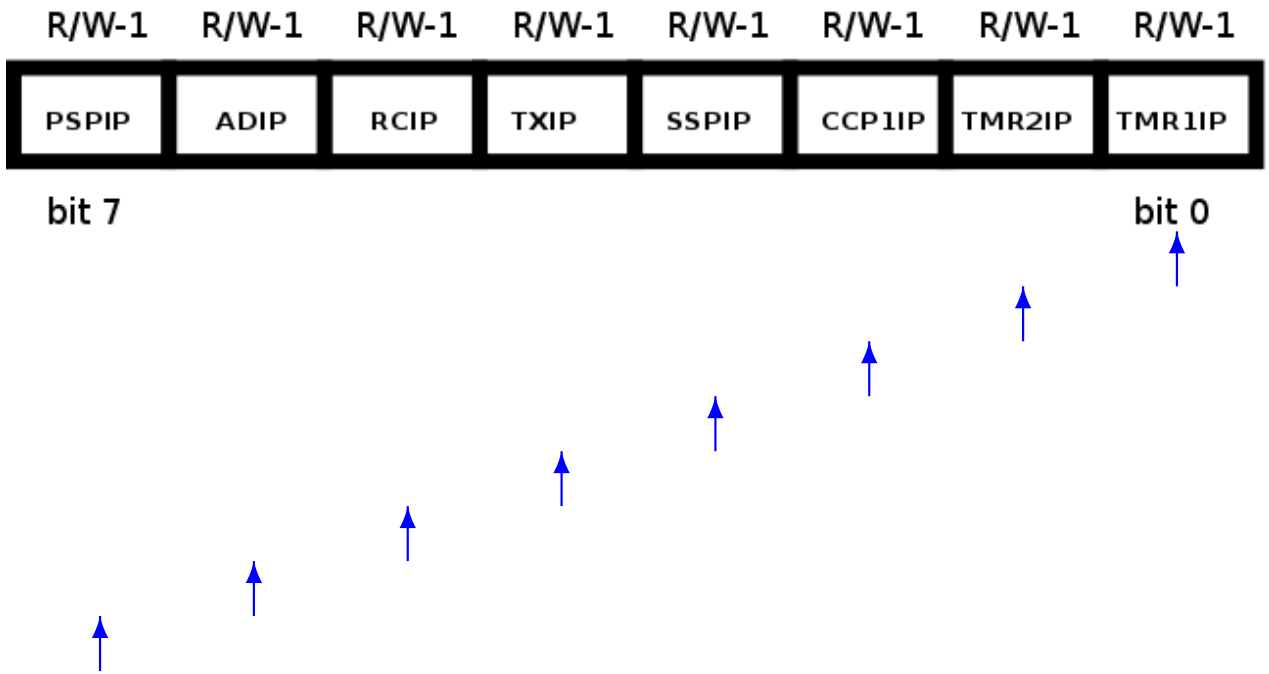
Bits in PIE1 register - Note SSPIE

IPR1

IPR1

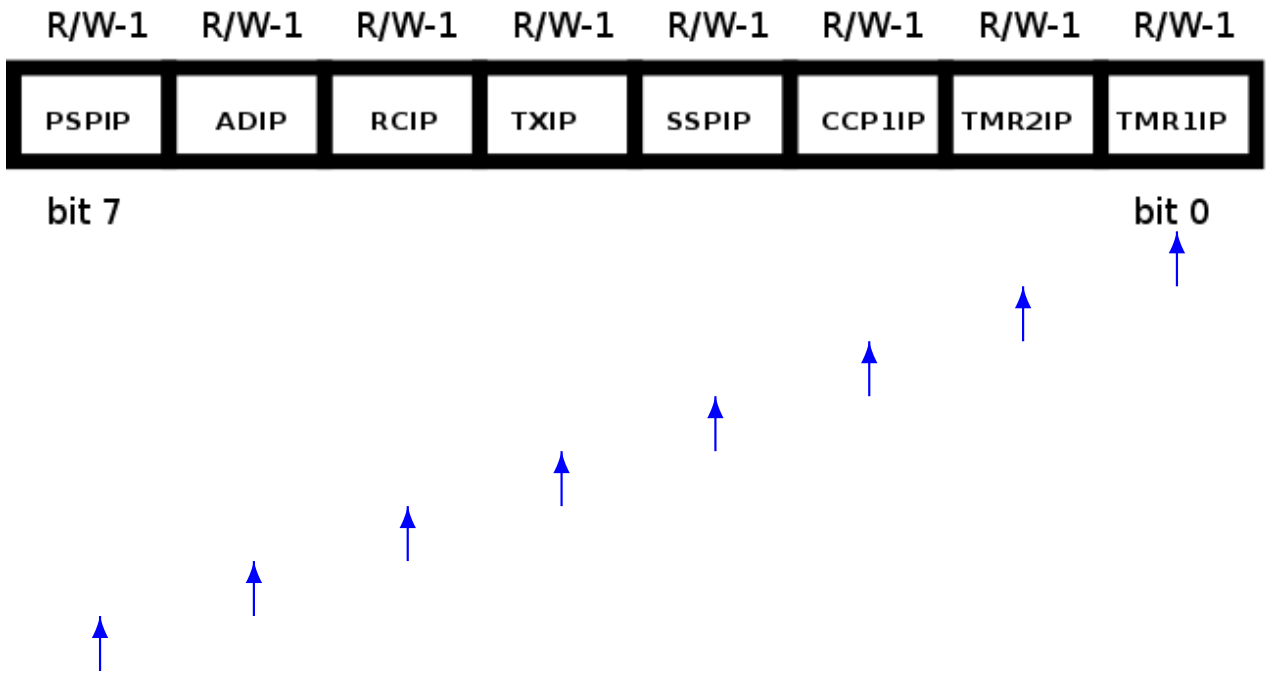

R/W-1
R/W-1
R/W-1
R/W-1
R/W-1
R/W-1
R/W-1
R/W-1
PSPIP
ADIP
RCIP
TXIP
SSPIP
CCP1IP
TMR2IP
TMR1IP
bit 7
bit 0

IPR1



Bits in IPR1 register

IPR1



Bits in IPR1 register - Note SSPIP

Bit-bashing

Bit-bashing

overview

Bit-bashing

overview

reasons

Bit-bashing

overview

reasons

NIB

Code

Code

PORT configuration

Code

PORT configuration

$\rightarrow$ macro or subroutine?

Code

PORT configuration

$\rightarrow$ macro or subroutine?

Initialization

Code

PORT configuration

$\rightarrow$ macro or subroutine?

Initialization

$\rightarrow$ macro or subroutine?

Code

PORT configuration

$\rightarrow$ macro or subroutine?

Initialization

$\rightarrow$ macro or subroutine?

Write to device

Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write to device

→ macro or subroutine?

Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write to device

→ macro or subroutine?

Read from device

Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write to device

→ macro or subroutine?

Read from device

→ macro or subroutine?

Initialization

Initialization

1. configure port (**TRISC**)

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits
2. set MSSP mode and other parameters (**SSPCON1**)

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits
2. set MSSP mode and other parameters (**SSPCON1**)

   I^2C , master, enable, etc.

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits
2. set MSSP mode and other parameters (**SSPCON1**)

   I^2C , master, enable, etc.
3. set other parameters (**SSPSTAT**,**SSPADD**)

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits
2. set MSSP mode and other parameters (**SSPCON1**)

   I^2C , master, enable, etc.
3. set other parameters (**SSPSTAT**,**SSPADD**)

   ?

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits
2. set MSSP mode and other parameters (**SSPCON1**)

   I^2C , master, enable, etc.
3. set other parameters (**SSPSTAT**,**SSPADD**)

   ?
4. enable interrupts (if desired) (**PIE1** , **IPR1**)

Initialization

1. configure port (**TRISC**)

   don't inadvertently alter other bits
2. set MSSP mode and other parameters (**SSPCON1**)

   I^2C , master, enable, etc.
3. set other parameters (**SSPSTAT**,**SSPADD**)

   ?
4. enable interrupts (if desired) (**PIE1** , **IPR1**)

   assuming interrupts are enabled globally

Sending

Sending

Sending data to slave

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *write data*; (**SSPBUF**,**SSPSTAT**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *write data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *write data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)
7. *wait for ACK from receiver*; (**SSPCON2**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *write data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)
7. *wait for ACK from receiver*; (**SSPCON2**)
8. *stop*; (**SSPSTAT**)

Sending

Sending data to slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *write data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)
7. *wait for ACK from receiver*; (**SSPCON2**)
8. *stop*; (**SSPSTAT**)

Receiving

Receiving

Receiving data from slave

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *read data*; (**SSPBUF**,**SSPSTAT**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *read data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *read data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)
7. *wait for NACK from receiver*; (**SSPCON2**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *read data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)
7. *wait for NACK from receiver*; (**SSPCON2**)
8. *stop*; (**SSPSTAT**)

Receiving

Receiving data from slave

1. *start*; (**SSPSTAT**)
2. *write address*; (**SSPBUF**,**SSPSTAT**)
3. *idle*; (**SSPCON2**, **SSPSTAT**)
4. *wait for ACK from receiver*; (**SSPCON2**)
5. *read data*; (**SSPBUF**,**SSPSTAT**)
6. *idle*; (**SSPCON2**, **SSPSTAT**)
7. *wait for NACK from receiver*; (**SSPCON2**)
8. *stop*; (**SSPSTAT**)