MCP19214/5

Digitally Enhanced Power Analog, Dual Channel, Low-Side PWM Controller

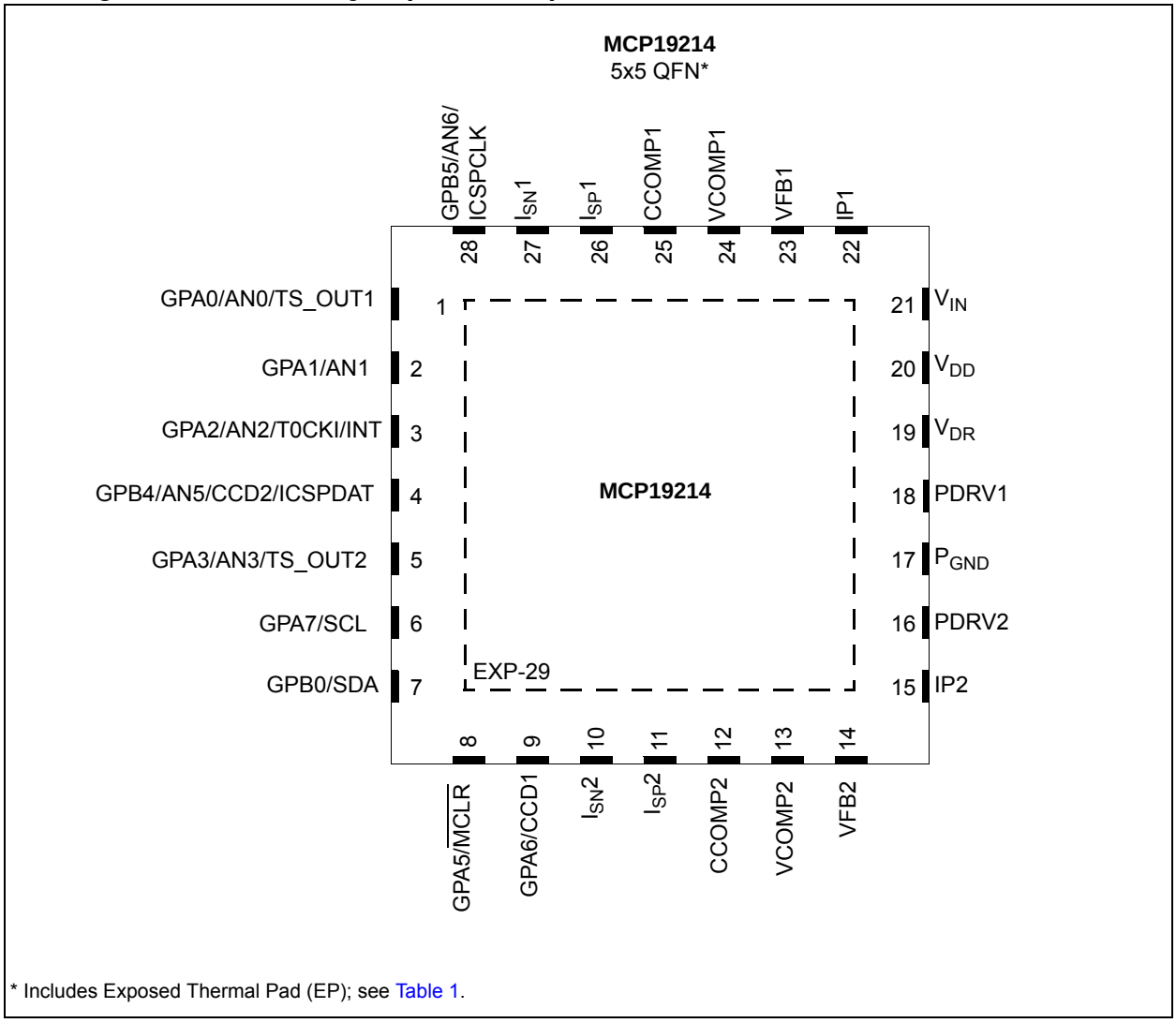
Features

- AEC Q100 Qualified and PPAP Capable
- Input Voltage Range: 4.5V-42V
- Two Independent High-Performance PWM Controllers
- Two Independent Control Loops per Channel allowing the User to Simultaneously Control the Output Voltage and Current
- Can Be Configured to Control Multiple Topologies including but Not Limited To:
  - Boost
  - Flyback
  - Ćuk
  - Single-Ended Primary-Inductor Converter (SEPIC)
- Peak Current Mode Control
- Master/Slave Operation of the PWM Controllers with Adjustable Phase Shift
- Differential Output Current Sense Capability
- Integrated Low-Side Current Sense Differential Amplifier (10X)
- Integrated Low-Side Gate Drivers:
  - 0.5A Sink/Source Current Capability at 5V Supply Voltage
  - 1A Sink/Source Current Capability at 10V Supply Voltage
- Special Events Generator (the state of regulating loops can be monitored without firmware overhead)
- Configurable Parameters:
  - Reference Voltages for Regulating Loops (four internal DACs)
  - Input Undervoltage Lockout (UVLO)
  - Input Overvoltage Lockout (OVLO)
  - Primary Current Leading Edge Blanking: 0 ns, 50 ns, 100 ns and 200 ns
  - Fixed Switching Frequency Range: 31.25 kHz-2.0 MHz
  - Slope Compensation
  - Primary Current Sense Offset Adjustment
  - Configurable GPIO Pin Options
- Low Quiescent Current: 10 mA Typical
- Low Sleep Current: 120 µA Typical
- Thermal Shutdown

Microcontroller Features

- Precision 8 MHz Internal Oscillator Block:
  - Factory-Calibrated to ±1%, Typical
- Interrupt-Capable
  - Firmware
  - Interrupt-on-Change Pins
- Only 35 Instructions to Learn
- 8192 Words On-Chip Program Memory
- 336 bytes of Internal RAM
- High-Endurance Flash:
  - 100,000 Write Flash Endurance
  - Flash Retention: > 40 Years
- Watchdog Timer (WDT) with Independent Oscillator for Reliable Operation
- Programmable Code Protection
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- Up to 12 I/O Pins and One Input-Only Pin
- Analog-to-Digital Converter (ADC):
  - 10-Bit Resolution
  - Internal 4096 mV Precision Reference Generator
  - Up to 8 External Channels
- Timer0: 8-Bit Timer/Counter with 8-Bit Prescaler
- Enhanced Timer1:
  - 16-bit Timer with Prescaler
  - Two Selectable Clock Sources
- Timer2: 8-Bit Timer with Prescaler
  - 8-Bit Period Register
- I^2C Communication:
  - 7-bit Address Masking
  - Two Dedicated Address Registers
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) (only MCP19215):
  - 8 and 9-Bit Data Operations
  - Address Detect
  - Asynchronous and Synchronous Operating Modes

Pin Diagram – 28-Pin 5x5 QFN (MCP19214)



* Includes Exposed Thermal Pad (EP); see Table 1.

TABLE 1: 28-PIN QFN (MCP19214) SUMMARY

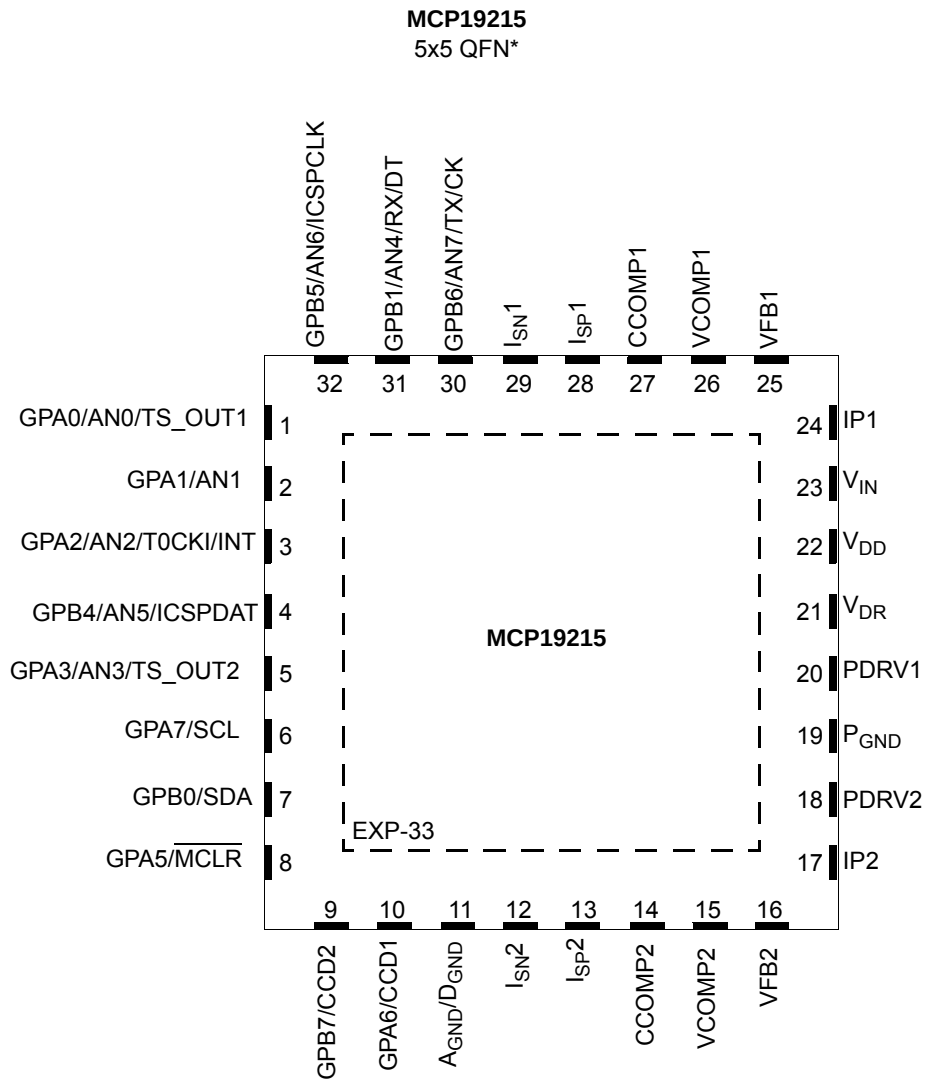
I/O	28-Pin QFN	ANSEL	A/D	Timers	MSSP	AUSART	Interrupt	Pull-Up	Basic	Additional
GPA0	1	Y	AN0	—	—	—	IOC	Y	—	Analog Circuitry Debug Out(1)
GPA1	2	Y	AN1	—	—	—	IOC	Y	—	—
GPA2	3	Y	AN2	T0CKI	—	—	IOC INT	Y	—	—
GPB4	4	Y	AN5	—	—	—	IOC	Y	—	Dual Capture/Compare Input 2
GPA3	5	Y	AN3	—	—	—	IOC	Y		Digital Circuitry Debug Out(1)
GPA7	6	N	—	—	SCL	—	IOC	Y	ICSPDAT	—
GPB0	7	N	—	—	SDA	—	IOC	Y		—
GPA5	8	N	—	—	—	—	IOC(2)	Y(3)	$\overline{MCLR}$	Test Enable Input
GPA6	9	N	—	—	—	—	IOC	Y	—	Dual Capture/Compare Input 1
$I_{SN}2$	10	—	—	—	—	—	—	—	—	Current Sense Amplifier Negative Input for PWM Channel 2
$I_{SP}2$	11	—	—	—	—	—	—	—	—	Current Sense Amplifier Positive Input for PWM Channel 2
CCOMP2	12	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Current Loop of PWM Channel 2
VCOMP2	13	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Voltage Loop of PWM Channel 2
VFB2	14	—	—	—	—	—	—	—	—	Feedback Input of the Voltage Loop of PWM Channel 2
IP2	15	—	—	—	—	—	—	—	—	Primary Input Current Sense of PWM Channel 2
PDRV2	16	—	—	—	—	—	—	—	—	Gate Drive Output of PWM Channel 2
P_{GND}	17	—	—	—	—	—	—	—	—	Power Ground
PDRV1	18	—	—	—	—	—	—	—	—	Gate Drive Output of PWM Channel 1
V_{DR}	19	—	—	—	—	—	—	—	—	Gate Drive Supply Voltage
V_{DD}	20	—	—	—	—	—	—	—	—	V_{DD} Output (+5V)
V_{IN}	21	—	—	—	—	—	—	—	—	Input Supply Voltage
IP1	22	—	—	—	—	—	—	—	—	Primary Input Current Sense of PWM Channel 1
VFB1	23	—	—	—	—	—	—	—	—	Feedback Input of the Voltage Loop of PWM Channel 1
VCOMP1	24	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Voltage Loop of PWM Channel 1
CCOMP1	25	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Current Loop of PWM Channel 1
$I_{SP}1$	26	—	—	—	—	—	—	—	—	Current Sense Amplifier Positive Input for PWM Channel 1
$I_{SN}1$	27	—	—	—	—	—	—	—	—	Current Sense Amplifier Negative Input for PWM Channel 1
GPB5	28	Y	AN6	—		—	IOC	Y	ICSPCLK	—
EP	29	—	—	—	—		—	—	A_{GND}	Small Signal Ground and Digital Ground

Note 1: The Analog/Digital Debug Output is selected through the control of the ABECON1 register.

2: The IOC is disabled when $\overline{MCLR}$ is enabled.

3: Weak pull-up always enabled when $\overline{MCLR}$ is enabled, otherwise the pull-up is under user control.

Pin Diagram – 32-PIN 5X5 QFN (MCP19215)



* Includes Exposed Thermal Pad (EP); see Table 2.

TABLE 2: 32-PIN QFN (MCP19215) SUMMARY

I/O	32-Pin QFN	ANSEL	A/D	Timers	MSSP	AUSART	Interrupt	Pull-Up	Basic	Additional
GPA0	1	Y	AN0	—	—	—	IOC	Y	—	Analog Circuitry Debug Out(1)
GPA1	2	Y	AN1	—	—	—	IOC	Y	—	—
GPA2	3	Y	AN2	T0CKI	—	—	IOC INT	Y	—	—
GPB4	4	Y	AN5	—	—	—	IOC	Y	ICSPDAT	—
GPA3	5	Y	AN3	—	—	—	IOC	Y	—	Digital Circuitry Debug Out(1)
GPA7	6	N	—	—	SCL	—	IOC	Y	—	Dual Capture/Single Compare1 Input
GPB0	7	N	—	—	SDA	—	IOC	Y	—	—
GPA5	8	N	—	—	—	—	IOC(2)	Y(3)	$\overline{MCLR}$	—
GPB7	9	N	—	—	—	—	IOC	Y	—	—
GPA6	10	N	—	—	—	—	IOC	Y	—	—
A_{GND}/D_{GND}	11	—	—	—	—	—	—	—	—	—
$I_{SN}2$	12	—	—	—	—	—	—	—	—	Current Sense Amplifier Negative Input for PWM Channel 2
$I_{SP}2$	13	—	—	—	—	—	—	—	—	Current Sense Amplifier Positive Input for PWM Channel 2
CCOMP2	14	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Current Loop of PWM Channel 2
VCOMP2	15	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Voltage Loop of PWM Channel 2
VFB2	16	—	—	—	—	—	—	—	—	Feedback Input of the Voltage Loop of PWM Channel 2
IP2	17	—	—	—	—	—	—	—	—	Primary Input Current Sense of PWM Channel 2
PDRV2	18	—	—	—	—	—	—	—	—	Gate Drive Output of PWM Channel 2
P_{GND}	19	—	—	—	—	—	—	—	—	Power Ground
PDRV1	20	—	—	—	—	—	—	—	—	Gate Drive Output of PWM Channel 1
V_{DR}	21	—	—	—	—	—	—	—	—	Gate Drives Supply Voltage
V_{DD}	22	—	—	—	—	—	—	—	—	V_{DD} Output (+5V)
V_{IN}	23	—	—	—	—	—	—	—	—	Input Supply Voltage
IP1	24	—	—	—	—	—	—	—	—	Primary Input Current Sense of PWM Channel 1
VFB1	25	—	—	—	—	—	—	—	—	Feedback Input of the Voltage Loop of PWM Channel 1

Note 1: The Analog/Digital Debug Output is selected through the control of the ABECON1 register.

2: The IOC is disabled when $\overline{MCLR}$ is enabled.

3: Weak pull-up always enabled when $\overline{MCLR}$ is enabled, otherwise the pull-up is under user control.

TABLE 2: 32-PIN QFN (MCP19215) SUMMARY (CONTINUED)

I/O	32-Pin QFN	ANSEL	A/D	Timers	MSSP	AUSART	Interrupt	Pull-Up	Basic	Additional
VCOMP1	26	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Voltage Loop of PWM Channel 1
CCOMP1	27	—	—	—	—	—	—	—	—	The Output of the Error Amplifier of the Current Loop of PWM Channel 1
$I_{SP}1$	28	—	—	—	—	—	—	—	—	Current Sense Amplifier Positive Input for PWM Channel 1
$I_{SN}1$	29	—	—	—	—	—	—	—	—	Current Sense Amplifier Negative Input for PWM Channel 1
GPB6	30	Y	AN7	—	—	TX/CK	IOC	Y	—	—
GPB1	31	Y	AN4	—	—	RX/DT	IOC	Y	—	—
GPB5	32	Y	AN6	—	—	—	IOC	Y	ICSPCLK	—
EP	33	—	—	—	—	—	—	—	—	—

Note 1: The Analog/Digital Debug Output is selected through the control of the ABECON1 register.

2: The IOC is disabled when $\overline{\text{MCLR}}$ is enabled.

3: Weak pull-up always enabled when $\overline{\text{MCLR}}$ is enabled, otherwise the pull-up is under user control.

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via e-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our web site at:

www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's web site; **www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

The MCP19214/5 devices are highly integrated, digitally enhanced PWM controllers, used for battery chargers, bidirectional converters, LED lighting systems and other low-side switch PWM applications.

These devices feature two independent analog PWM controllers, and the internal architecture is optimized for applications that require precise control of the output parameters. Like the other members of the digitally enhanced PWM controllers family, MCP19214/5 includes a fully programmable microcontroller core and a 10-bit analog-to-digital converter.

Each PWM channel includes two error amplifiers with independent adjustable reference voltage generators, current sense input with programmable leading edge blanking, programmable slope compensation ramp generator, integrated internal programmable oscillator, current sense differential amplifier and an integrated MOSFET driver.

An internal LDO (+5V) is used to power the PIC core, the analog circuitry and to provide 5V externally. This 5V external output can also be used to supply the internal MOSFET drivers. The internal MOSFET drivers have the option to be powered from an external voltage source (up to 10V) in order to accommodate applications that require higher voltages for gate driving.

The MCP19214/5 controllers offer a very high degree of integration, allowing the user to develop complex applications without additional circuitry. Some unique features, like simultaneous control of the converter's output current and voltage, make MCP19214/5 devices ideally suited for battery chargers, LED drivers and bidirectional converters. Additionally, the General Purpose Inputs/Outputs (GPIOs) can be used to drive various switches, to enable/disable additional circuitry, or to indicate a typical state.

The MCP19214 is packaged in a 28-lead 5 mm x 5 mm QFN, and the MCP19215 in a 32-lead 5 mm x 5 mm QFN. The operating input voltage for normal device operation is 4.5V-42V, with an absolute maximum of 44V. The maximum transient voltage is 48V for 500 mS.

Power trains supported by this architecture include, but are not limited to, Boost, Buck-Boost, Flyback, SEPIC and Cuk.

MCP19214/5 integrates an I^2C controller and an Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) module (only for MCP19215). The user can develop specific communication protocols using the internal interfaces. A PMBus compatible protocol, specific for power converters, can be implemented using the provided I^2C serial bus.

Complete customization of the device operating parameters, start-up or shutdown profiles, protection levels and fault handling procedures are accomplished by firmware that can be developed using Microchip's MPLAB® X Integrated Development Environment. Programming the MCP19214/5 is done using one of Microchip's many in-circuit debugger and device programmers.

FIGURE 1-1: MCP19214/5 SIMPLIFIED INTERNAL BLOCK DIAGRAM

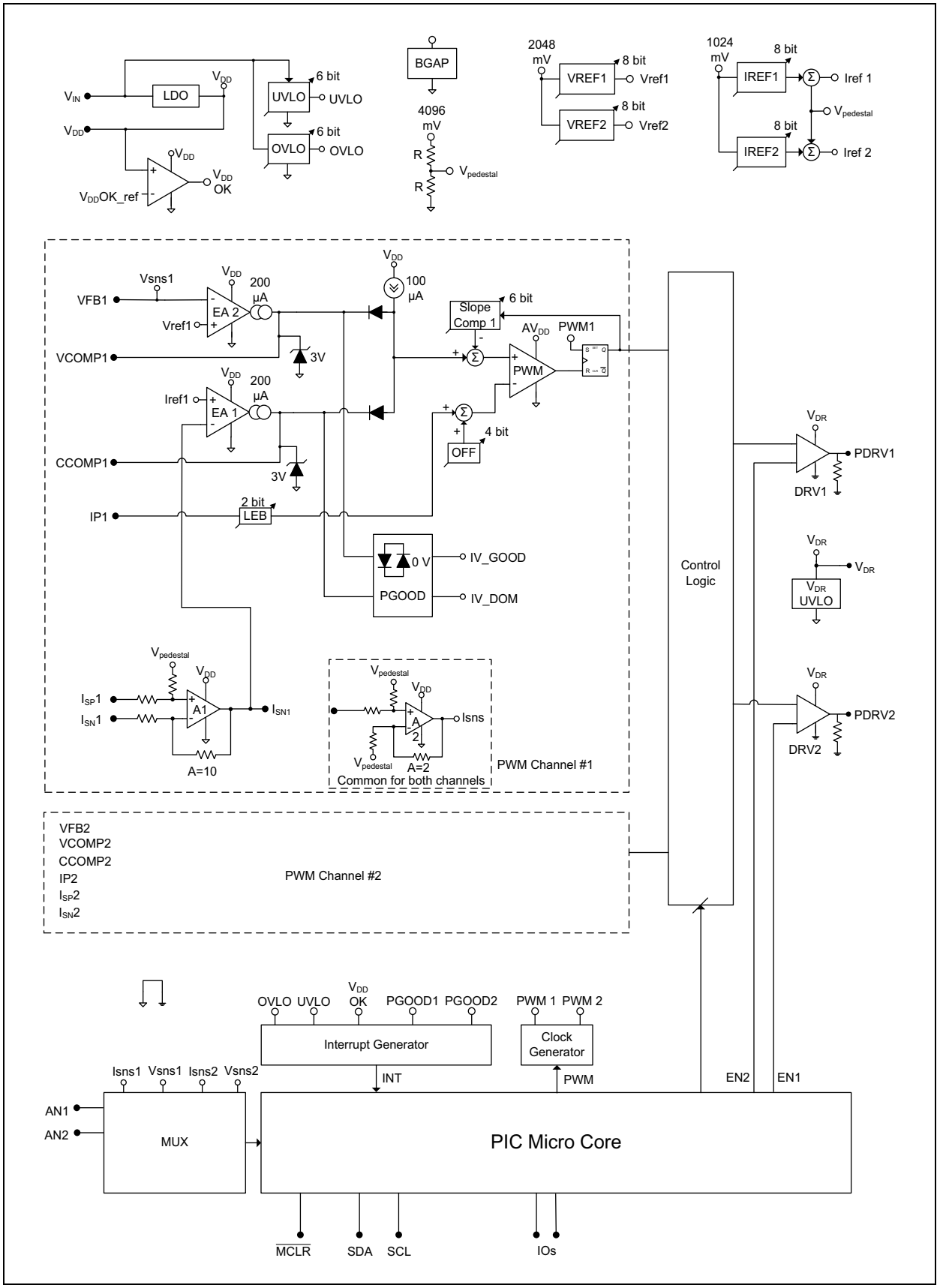


FIGURE 1-2: MCP19214 DUAL LED STRING APPLICATION DIAGRAM

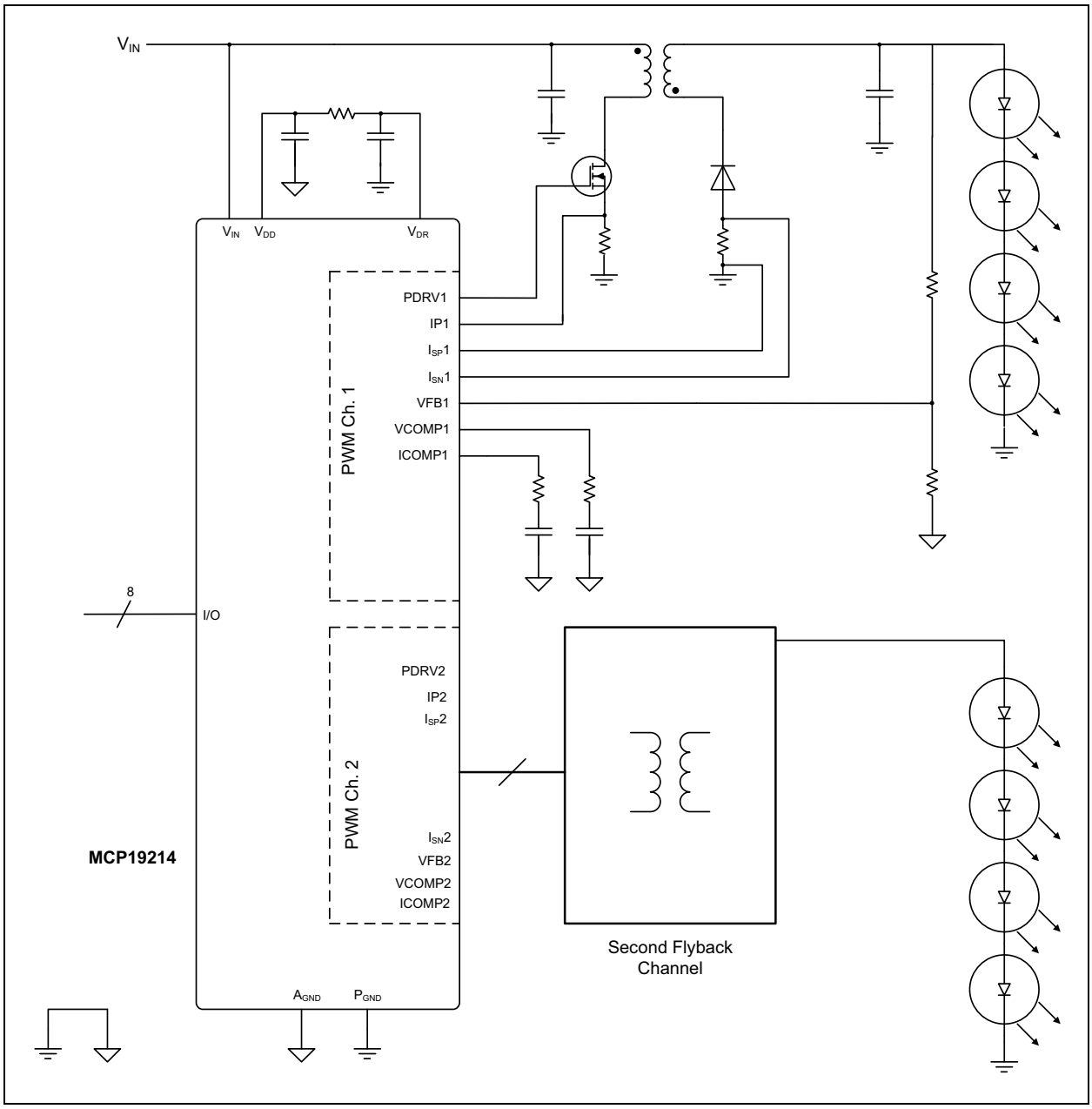


FIGURE 1-3: MCP19214 BIDIRECTIONAL CONVERTER APPLICATION DIAGRAM

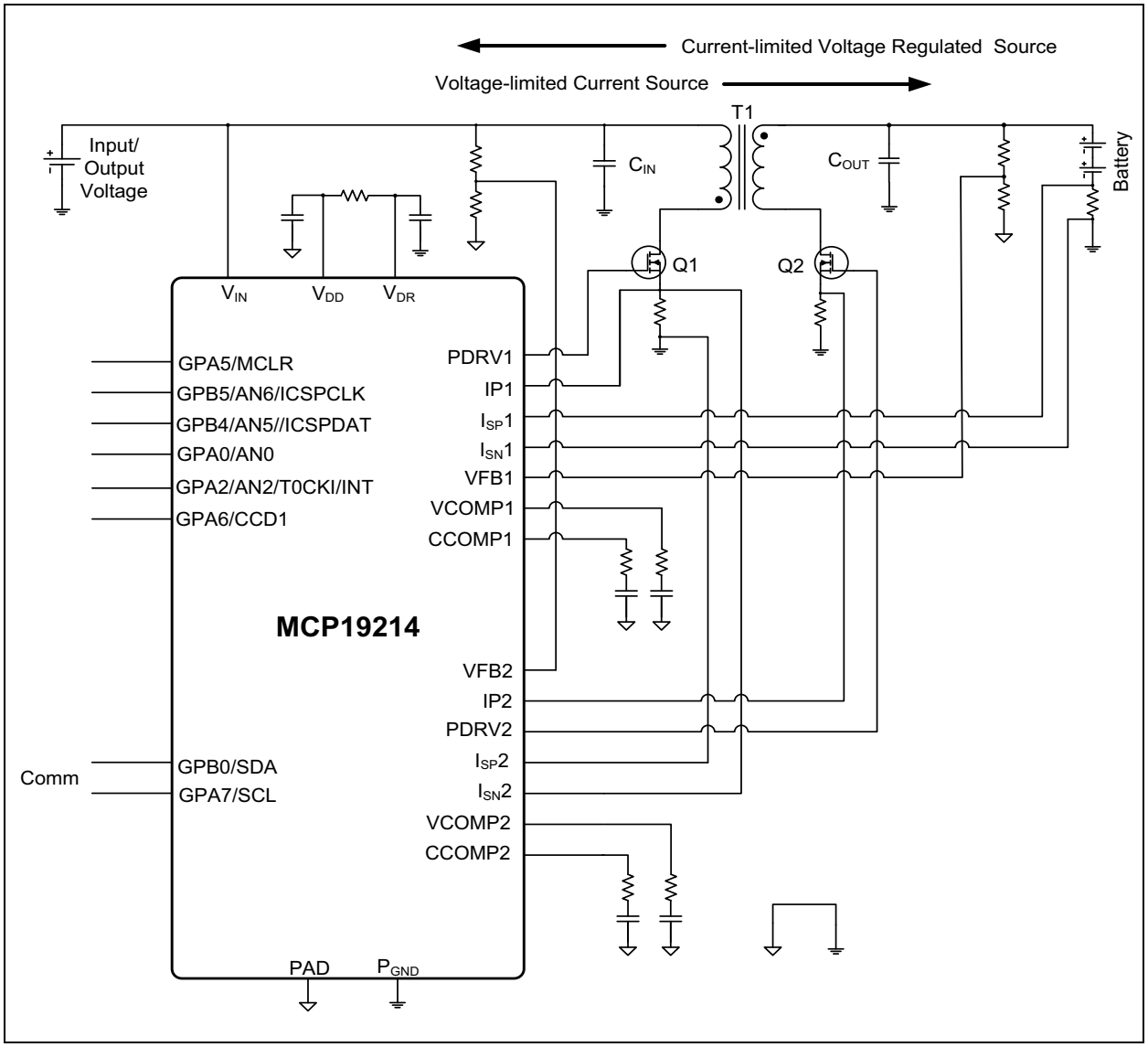
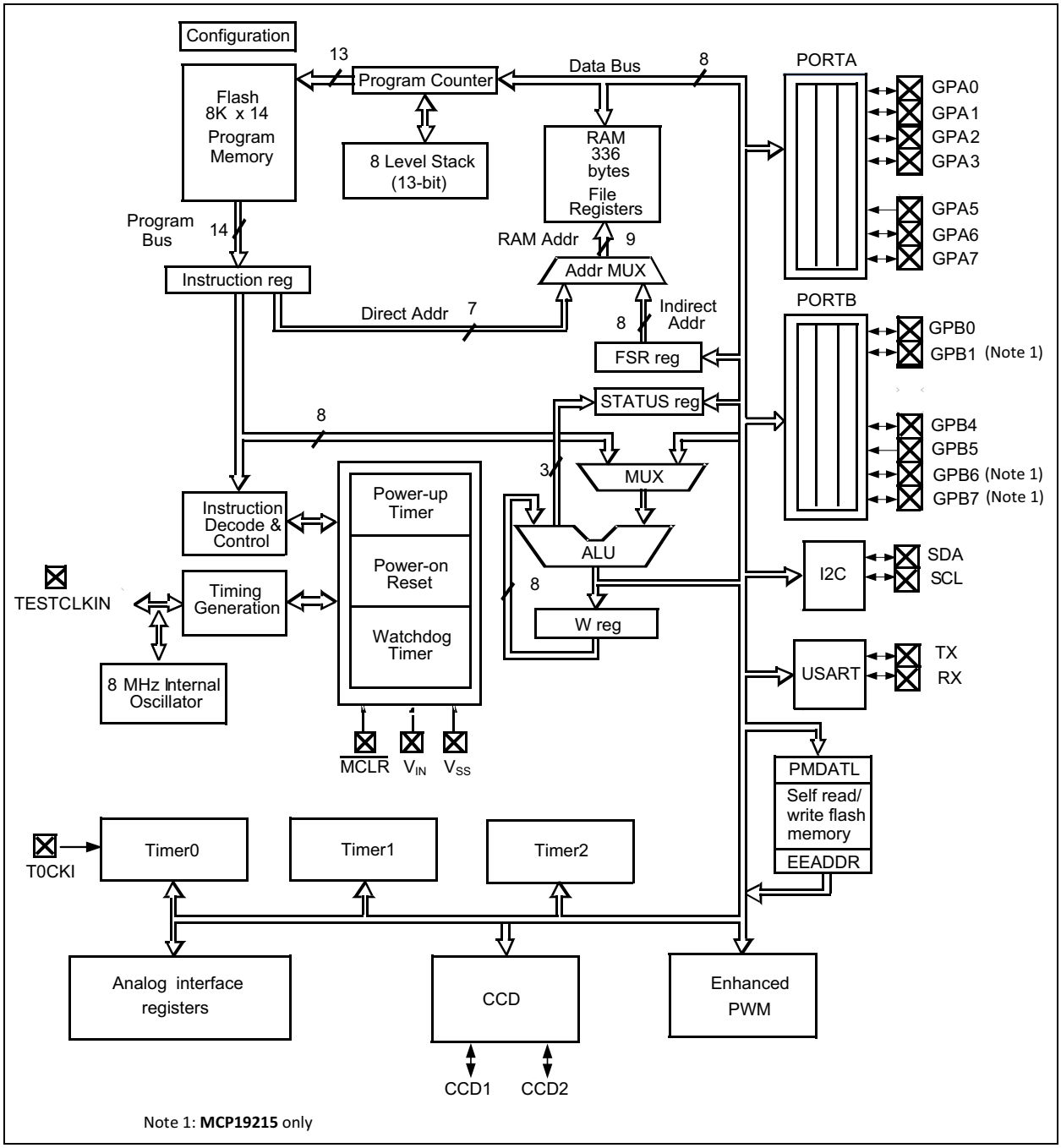


FIGURE 1-4: MICROCONTROLLER CORE BLOCK DIAGRAM



2.0 PIN DESCRIPTION

The 28-lead MCP19214 and 32-lead MCP19215 devices feature pins that have multiple functions associated with each pin. Table 2-1 provides a description of the different functions. Refer to **Section 2.1 "Detailed Pin Functional Description"** for more detailed information.

TABLE 2-1: MCP19214/5 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
GPA0/AN0/TS_OUT1	GPA0	TTL	CMOS	General-purpose I/O
	AN0	AN	—	A/D Channel 0 input
	TS_OUT1	—	—	Internal analog signal multiplexer output(1)
GPA1/AN1	GPA1	TTL	CMOS	General-purpose I/O
	AN1	AN	—	A/D Channel 1 input
GPA2/AN2/T0CKI/INT	GPA2	ST	CMOS	General-purpose I/O
	AN2	AN	—	A/D Channel 2 input
	T0CKI	ST	—	Timer0 clock input
	INT	ST	—	External interrupt
GPA3/AN3/TS_OUT2	GPA3	TTL	CMOS	General-purpose I/O
	AN3	AN	—	A/D Channel 3 input
	TS_OUT2			Internal digital signals multiplexer output(1)
GPA5/$\overline{\text{MCLR}}$	GPA5	TTL	—	General-purpose input only
	$\overline{\text{MCLR}}$	ST	—	Master Clear with internal pull-up
GPA6/CCD1	GPA6	ST	CMOS	General-purpose I/O
	CCD1	ST	CMOS	Single Compare output of PWM channel 1. Dual Capture input of PWM channel 1.
GPA7/SCL	GPA7	ST	CMOS	General-purpose open drain I/O
	SCL	I^2C	OD	I^2C clock
GPB0/SDA	GPB0	TTL	CMOS	General-purpose I/O
	SDA	I^2C	OD	I^2C data input/output
GPB1/AN4/RX/DT (**MCP19215** Only)	GPB1	TTL	CMOS	General-purpose I/O
	AN4	AN	—	A/D Channel 4 input
	RX	ST	—	AUSART asynchronous receive input
	DT	TTL	CMOS	AUSART synchronous data input/output
GPB4/AN5/ICSPDAT	GPB4	TTL	CMOS	General-purpose I/O
	AN5	AN	—	A/D Channel 5 input
	ICSPDAT	ST	CMOS	In-Circuit Debugger and ICSP programming data
GPB5/AN6/ICSPCLK	GPB5	TTL	CMOS	General-purpose I/O
	AN6	AN	—	A/D Channel 6 input
	ISCPCLK	ST	—	In-Circuit Debugger and ICSP programming clock

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C

Note 1: The Analog/Digital Debug Output is selected through the control of the ABECON1 register.

TABLE 2-1: MCP19214/5 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
GPB6/AN7/TX/CK (**MCP19215** Only)	GPB6	TTL	CMOS	General-purpose I/O
	AN7	AN	—	A/D Channel 7 input
	TX	—	CMOS	AUSART asynchronous transmit output
	CK	TTL	CMOS	AUSART synchronous clock input/output
GPB7/CCD2 (**MCP19215** Only)	GPB7	TTL	CMOS	General-purpose I/O
	CCD2	ST	CMOS	Single Compare output of PWM channel 2 Dual Capture input of PWM channel 2
V_{IN}	V_{IN}	—	—	Device input supply voltage
V_{DD}	V_{DD}	—	—	Internal +5V LDO output pin
V_{DR}	V_{DR}	—	—	Gate drivers supply voltage
A_{GND}/D_{GND}	A_{GND}	—	—	Small signal quiet ground
P_{GND}	P_{GND}	—	—	Large signal power ground
PDRV1	PDRV1	—	—	Primary PWM channel MOSFET gate drive
PDRV2	PDRV2	—	—	Secondary PWM channel MOSFET gate drive
IP1	IP1	—	—	Primary input current sense for PWM channel 1
IP2	IP2	—	—	Primary input current sense for PWM channel 2
$I_{SN}1$	$I_{SN}1$	—	—	Differential current sense amplifier negative input of PWM channel 1
$I_{SN}2$	$I_{SN}2$	—	—	Differential current sense amplifier negative input of PWM channel 2
$I_{SP}1$	$I_{SP}1$	—	—	Differential current sense amplifier positive input of PWM channel 1
$I_{SP}2$	$I_{SP}2$	—	—	Differential current sense amplifier positive input of PWM channel 2
CCOMP1	CCOMP1	—	—	Output of the current loop error amplifier of PWM channel 1
CCOMP2	CCOMP2	—	—	Output of the current loop error amplifier of PWM channel 2
VCOMP1	VCOMP1	—	—	Output of the voltage loop error amplifier of PWM channel 1
VCOMP2	VCOMP2	—	—	Output of the voltage loop error amplifier of PWM channel 2
VFB1	VFB1	—	—	Feedback input of the voltage loop of PWM channel 1
VFB2	VFB2	—	—	Feedback input of the voltage loop of PWM channel 2
EP				Exposed Pad. The A_{GND} and D_{GND} internal nodes close here. Connect this to the PCBs ground plane using multiple vias.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C

Note 1: The Analog/Digital Debug Output is selected through the control of the ABECON1 register.

2.1 Detailed Pin Functional Description

2.1.1 GPA0 PIN

GPA0 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPA. An internal weak pull-up and interrupt-on-change are also available.

AN0 is an input to the A/D. To configure this pin to be read by the A/D on channel 0, bits TRISA0 and ANSA0 must be set.

The ABECON1/2 registers can be configured to set this pin to the TS_OUT1 function. It is a buffered output of the internal analog signals multiplexer. Analog signals present on this pin are controlled by the ADCON0 register.

2.1.2 GPA1 PIN

GPA1 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPA. An internal weak pull-up and interrupt-on-change are also available.

AN1 is an input to the A/D. To configure this pin to be read by the A/D on channel 1, bits TRISA1 and ANSA1 must be set.

2.1.3 GPA2 PIN

GPA2 is a general-purpose ST input or CMOS output pin whose data direction is controlled in TRISGPA. An internal weak pull-up and interrupt-on-change are also available.

AN2 is an input to the A/D. To configure this pin to be read by the A/D on channel 2, bits TRISA2 and ANSA2 must be set.

When bit T0CS is set in the OPTION_REG register, the T0CKI function is enabled. Refer to **Section 21.0 "Timer0 Module"** for more information.

GPA2 can also be configured as an external interrupt by setting the INTE bit. Refer to **Section 13.2 "GPA2/INT Interrupt"** for more information.

2.1.4 GPA3 PIN

GPA3 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPA. An internal weak pull-up and interrupt-on-change are also available.

AN3 is an input to the A/D. To configure this pin to be read by the A/D on channel 3, bits TRISA3 and ANSA3 must be set.

The ABECON1/2 registers can be configured to set this pin to the TS_OUT2 function. It is a buffered output of the internal digital signals multiplexer. Digital signals present on this pin are controlled by the ABECON2 register.

2.1.5 GPA5 PIN

GPA5 is a general-purpose TTL input only pin. An internal weak pull-up and interrupt-on-change are also available.

For programming purposes, this pin is to be connected to the $\overline{\text{MCLR}}$ pin of the serial programmer. Refer to **Section 32.0 "In-Circuit Serial Programming™ (ICSP™)"** for more information.

This pin is $\overline{\text{MCLR}}$ when the $\overline{\text{MCLR}}$E bit is set in the CONFIG register. When the $\overline{\text{MCLR}}$ is active, the interrupt-on-change is disabled and the weak pull-up is always enabled.

2.1.6 GPA6 PIN

GPA6 is a general-purpose CMOS output ST input pin whose data direction is controlled in TRISGPA.

GPA6 is part of the CCD Module. For more information, refer to **Section 31.0 "Dual Capture/Compare (CCD) Module"**.

2.1.7 GPA7 PIN

GPA7 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPA. An internal weak pull-up and interrupt-on-change are also available.

When the MCP19214/5 is configured for I^2C communication, **Section 27.2 "I^2C Mode Overview"**, GPA7 functions as the I^2C clock (SCL). This pin must be configured as an input to allow proper operation.

2.1.8 GPB0 PIN

GPB0 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPB. An internal weak pull-up and interrupt-on-change are also available.

When the MCP19214/5 are configured for I^2C communication, **Section 27.2 "I^2C Mode Overview"**, GPB0 functions as the I^2C clock (SDA). This pin must be configured as an input to allow proper operation.

2.1.9 GPB1 PIN **(MCP19215 ONLY)**

GPB1 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPB. An internal weak pull-up and interrupt-on-change are also available.

AN4 is an input to the A/D. To configure this pin to be read by the A/D on channel 4, bits TRISB1 and ANSB1 must be set.

RX is the receiver's input of the AUSART block for asynchronous operation.

DT is the input/output pin of the AUSART block for synchronous operation

2.1.10 GPB4 PIN

GPB4 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPB. An internal weak pull-up and interrupt-on-change are also available.

AN5 is an input to the A/D. To configure this pin to be read by the A/D on channel 5, bits TRISB4 and ANSB4 must be set.

ICSPDAT is the serial programming data I/O function. This is used in conjunction with ICSPCLK to serial program the device. Refer to **Section 32.0 "In-Circuit Serial Programming™ (ICSP™)"** for more information.

2.1.11 GPB5 PIN

GPB5 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPB. An internal weak pull-up and interrupt-on-change are also available.

AN6 is an input to the A/D. To configure this pin to be read by the A/D on channel 6, bits TRISB5 and ANSB5 must be set.

ICSPCLK is the serial programming/debugging clock function. This is used in conjunction with ICSPDAT to serial program the device. Refer to **Section 32.0 "In-Circuit Serial Programming™ (ICSP™)"** for more information.

2.1.12 GPB6 PIN **(MCP19215 ONLY)**

GPB6 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPB. An internal weak pull-up and interrupt-on-change are also available.

AN7 is an input to the A/D. To configure this pin to be read by the A/D on channel 7, bits TRISB6 and ANSB6 must be set.

TX is the transmitter's output of the AUSART block during asynchronous operation.

CK is the input/output clock of the AUSART block during synchronous operation.

2.1.13 GPB7 PIN **(MCP19215 ONLY)**

GPB7 is a general-purpose TTL input or CMOS output pin whose data direction is controlled in TRISGPB. An internal weak pull-up and interrupt-on-change are also available.

GPB7 is part of the CCD Module. For more information, refer to **Section 31.0 "Dual Capture/Compare (CCD) Module"**.

2.1.14 V_{IN} PIN

V_{IN} is the input voltage pin of the MCP19214/5 controller. This pin is connected to the input of the +5V internal voltage regulator. A bypass capacitor of minimum 100 nF must be connected between this pin and GND. This capacitor should be physically placed close to the device.

2.1.15 V_{DD} PIN

The output of the internal +5.0V regulator is connected to this pin. It is recommended that a minimum 4.7 µF ceramic bypass capacitor be connected between this pin and the GND pin of the device. The bypass capacitor should be physically placed close to the device.

2.1.16 V_{DR} PIN

The supply for the MOSFET drivers is connected to this pin and has an absolute maximum rating of +13.5V. A decoupling capacitor of 1.0 µF should be placed between this pin and P_{GND} pin. This pin can be connected by an RC filter to the V_{DD} pin.

2.1.17 A_{GND}/D_{GND} PIN **(MCP19215 ONLY)**

A_{GND}/D_{GND} is the small signal ground connection pin. This pin should be connected to a low noise ground.

2.1.18 P_{GND} PIN

This is the large signal ground pin. P_{GND} is the return path for the internal MOSFET drivers. This pin should be connected to the power train ground using short, low impedance connection.

2.1.19 PDRV1 PIN

The output of the internal MOSFET driver of PWM channel 1. Connect this pin to the gate of the power MOSFET using short, low-impedance trace.

2.1.20 PDRV2 PIN

The output of the internal MOSFET driver of PWM channel 2. Connect this pin to the gate of the power MOSFET using short, low-impedance trace.

2.1.21 IP1 PIN

This pin is the primary current sense input of the PWM channel 1. IP1 is connected to the main PWM comparator via a blanking circuit.This input is sensitive to high-frequency noise. Keep the associated trace away from noise sources like the main switch node of the converter or the MOSFET's gate drive signals. It is recommended to insert an RC low-pass filter between this input and the shunt resistor used to sense the inductor's current.

2.1.22 IP2 PIN

This pin is the primary current sense input of the PWM channel 2. IP2 is connected to the main PWM comparator via a blanking circuit.This input is sensitive to high-frequency noise. Keep the associated trace away from noise sources like the main switch node of the converter or the MOSFETs gate drive signals. It is recommended to insert an RC low-pass filter between this input and the shunt resistor used to sense the inductor's current.

2.1.23 $I_{SN}1$ PIN

This pin is the inverting input of the internal differential current sense amplifier of PWM channel 1.

2.1.24 $I_{SN}2$ PIN

This pin is the inverting input of the internal differential current sense amplifier of PWM channel 2.

2.1.25 $I_{SP}1$ PIN

This pin is the noninverting input of the internal differential current sense amplifier of PWM channel 1.

2.1.26 $I_{SP}2$ PIN

This pin is the noninverting input of the internal differential current sense amplifier of PWM channel 2.

2.1.27 CCOMP1

The CCOMP1 pin is the output of the current loop error amplifier of PWM channel 1. The loop's compensation network is connected between this pin and GND. This is a high-impedance node, and the traces associated with this pin must be kept far from the noise sources like the main switch node of the converter or the MOSFET's gate drive signals.

2.1.28 CCOMP2

The CCOMP2 pin is the output of the current loop error amplifier of PWM channel 2. The loop's compensation network is connected between this pin and GND. This is a high-impedance node, and the traces associated with this pin must be kept far from the noise sources like the main switch node of the converter or the MOSFETs gate drive signals.

2.1.29 VCOMP1

The VCOMP1 pin is the output of the voltage loop error amplifier of PWM channel 1. The compensation network is connected between this pin and GND. This is a high-impedance node, and the traces associated with this pin must be kept far from the noise sources like the main switch node of the converter or the MOSFET's gate drive signals.

2.1.30 VCOMP2

The VCOMP2 pin is the output of the voltage loop error amplifier of PWM channel 2. The compensation network is connected between this pin and GND. This is a high-impedance node, and the traces associated with this pin must be kept far from the noise sources like the main switch node of the converter or the MOSFET's gate drive signals.

2.1.31 VFB1 PIN

The inverting input of the error amplifier of the voltage loop for PWM channel 1. This is a high-impedance input and is sensitive to noise. Keep the trace associated with this pin far from noise sources like the main switch node of the converter or MOSFETs gate drive signals.

2.1.32 VFB2 PIN

The inverting input of the error amplifier of the voltage loop for PWM channel 2. This is a high-impedance input and is sensitive to noise. Keep the trace associated with this pin far from noise sources like the main switch node of the converter or MOSFET's gate drive signals.

2.1.33 EXPOSED PAD (EP)

This pad should be connected to a solid ground plane using multiple vias.The connection must provide low thermal impedance as well as low electrical noise. In case of MCP19214, this pad is associated with analog and digital internal grounds (A_{GND}/D_{GND}).

3.0 FUNCTIONAL DESCRIPTION

3.1 Linear Regulator

The operating input voltage for the MCP19214/5 devices ranges from 4.5V-42V. The internal 5V LDO provides bias voltage for the microcontroller core and for the analog circuitry. The output of this LDO can be used for bias additional external low-power circuitry, as well as for the integrated MOSFET drivers. Care should be exercised to avoid an overload of the LDO. The output of this LDO is monitored using a comparator and a specific interrupt is generated in case of malfunction. The thresholds of this comparator are adjustable. Bits that control the functionality of the V_{DD} UVLO comparator are located in the VDDCON register.

In order to minimize the current consumption during Sleep mode, the output voltage of the LDO can be adjusted in two steps: 3V and 5V (typical). There are also two operating modes during sleep: Normal mode and Low-Power mode. Bits that control the functionality of the LDO during Sleep mode are located in PE1 register.

The MCP19214/5 also incorporate a brown-out protection. Refer to **Section 12.3 "Brown-out Reset (BOR)"** for details. The PIC core will reset at 2.0V V_{DD}.

3.2 Output Drive Circuitry

The MCP19214/5 integrates two low-side drivers, one per PWM channel used to drive the external low-side N-Channel power MOSFETs. MCP19214/5 directly controls only topologies that involve low-side MOSFET drivers like boost, buck-boost, flyback, SEPIC or Cuk. For topologies that require high-side MOSFET drive (e.g., buck or synchronous buck) an external, specialized MOSFET driver can be used.

The gate drive (V_{DR}) can be supplied from 5V-10V. The drive strength is capable of up to 1A sink/source with 10V gate drive and down to 0.5A sink/source with 5V gate drive. The supply voltage of the MOSFET drivers is monitored by a UVLO circuit in order to prevent damage to the external power switches. The MOSFET driver's UVLO circuit has two thresholds: 2.7V and 5.4V (typical).

Each driver has its own enable bit, controlled by the microcontroller core. These bits are located in PE1 register.

3.3 PWM Controller

MCP19214/5 integrates two independent PWM controllers. Each PWM channel comprises a set of two error amplifiers with independent reference voltage generators, a PWM comparator with latched output, a current sense input with adjustable leading-edge blanking time, and a ramp generator for slope compensation. See Figure 3-1 for details.

The error amplifiers are of transconductance type (OTA) with separate, external compensation network connected between the output and the ground (A_{GND}). The outputs of the OTAs are tied together through diodes in order to allow the simultaneous control of the output voltage and current. The typical sink current is 200 uA. This type of amplifier makes possible the usage of high-valued resistors for the feedback divider without affecting the frequency response of the amplifier, as these resistors are out of the compensation loop. Only one inverting input is accessible outside the chip (VFBx), the other being internally connected to the output of the 10X differential amplifier. The error signal is clamped at a certain level (3V, typical) in order to prevent overcurrent in the main switching MOSFET of the converter.

The output of the error amplifiers (CCOMP and VCOMP) are equipped with switches (S1 and S2 in Figure 3-1) in order to reset the compensation networks before the soft start or before the activation of a certain loop. These switches are under the software control (VLRES and CLRES bits), and it is the responsibility of the user to ensure proper operation.

The internal 10X differential amplifier is used to improve the accuracy of the current regulated loops and reduce the power dissipation in the external current sensing element (shunt).

A second amplifier (common for both PWM channels) with a gain of 2X is connected between the ADC input and the output of the 10X amplifier. This second amplifier increases the dynamic range of the current measurement circuitry.

In order to prevent any inherent errors that may occur when the output of the amplifier goes near ground or input rail, a special circuit centers the common mode voltage at a specified level (pedestal voltage, typical 2.048V). This technique allows the microcontroller core to read the current in both directions (as with the bidirectional converters): the forward direction when the converter charges the battery, and the reverse direction when the converter delivers constant output voltage from the battery. The output of the 10X differential amplifier as well as the output of the second 2X amplifier will be at the pedestal voltage if the sensed current is zero. The same pedestal voltage is added tothe value of the reference voltage generator of the current loop in order to compensate the DC offset introduced by the pedestal voltage. The 2X amplifier output can be connected via the analog multiplexer to the ADC input for current monitoring purposes (Isnsx signal). The pedestal voltage is fixed at 2.048V.

The loop implements the Peak Current mode control. The IPx input is used to sense the inductor's current. A leading-edge blanking circuit (LEB) is used to prevent false reset of the PWM circuitry. The LEB can be set to four steps (0 ns, 50 ns, 100 ns and 200 ns). The blank time is controlled from the ICLEBCON register. External resistor capacitor filtering techniques can still be used to filter the leading edge if desired. An adjustable offset voltage generator is used to add a certain amount of DC offset (programmable) on top of the IPx signal. This offset prevents the effects of the nonlinearities associated with the error amplifiers outputs at very low-voltage levels, and can also be used for overcurrent protection purposes (cycle-by-cycle current limit). This offset adjustment is controlled by the ICOACON register.

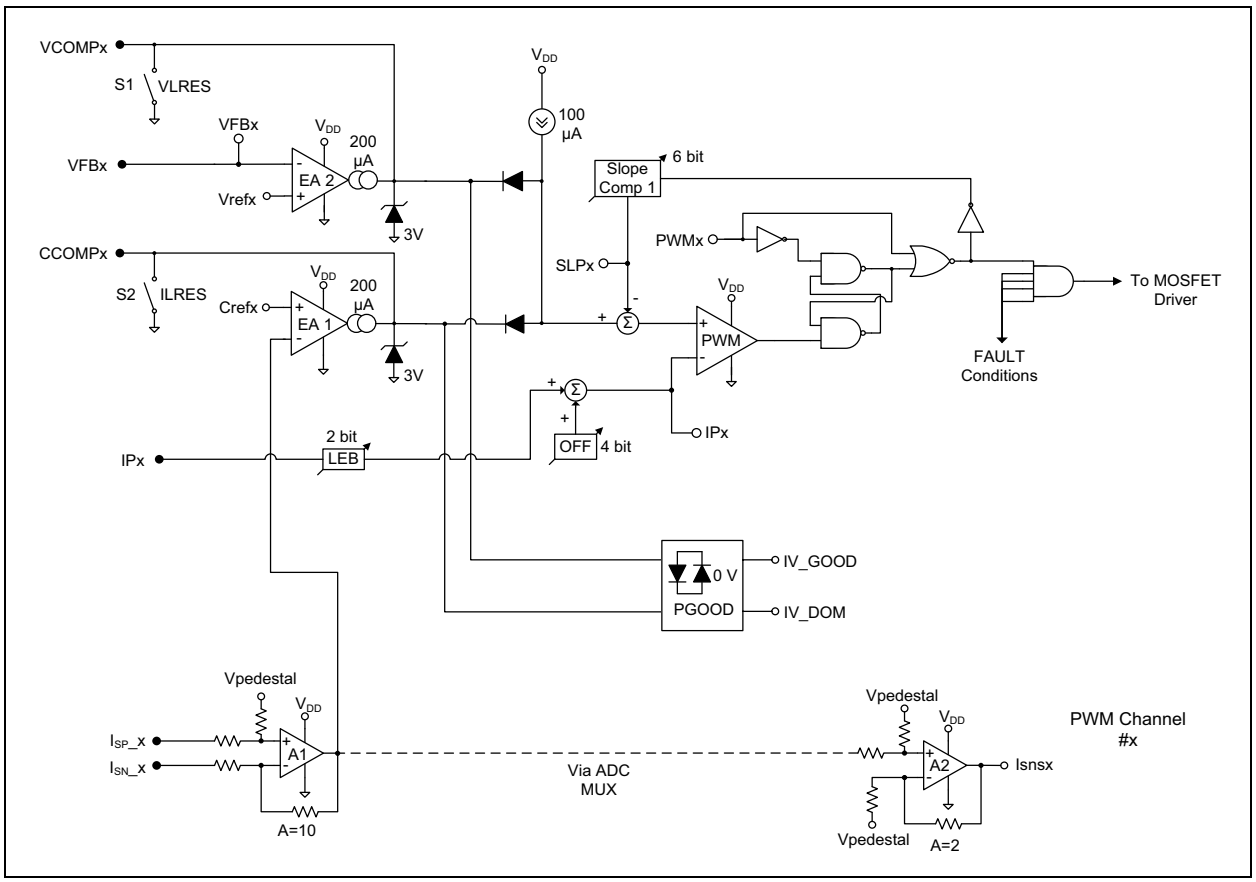
The Peak Current mode control requires slope compensation in order to avoid subharmonic oscillations. A programmable (6-bit) ramp generator is provided and its output is subtracted from the error signal.

When the current sense signal reaches the level of the control voltage minus slope compensation, the on cycle is terminated and the external switch is latched off until the beginning of the next cycle, which begins at the next clock cycle.

A S-R Latch (Set-Reset Latch) is used to prevent the PWM circuitry from turning the external switch on until the beginning of the next clock cycle.

In order to avoid severe overshoots of the controlled parameter (current or voltage) when the loop switches between operating modes (constant current or constant voltage), the outputs of the error amplifiers are clamped together.

FIGURE 3-1: THE PWM CONTROLLER BLOCK DIAGRAM



3.4 Current Sense

The output current is differentially sensed by the MCP19214/5. In high-current applications, this helps to maintain high-system efficiency by minimizing power dissipation in current sense resistors. Differential current sensing also minimizes external ground shift errors. The internal differential amplifier has a typical gain of 10 V/V (typical).

3.5 Power Good Circuitry

The Power Good circuitry monitors the outputs of the error amplifiers to detect an open-loop condition. The IV_GOOD signal will generate an interrupt and the PIC core will read the status of the IV_GOOD and IV_DOM bits. The IV_GOOD signal indicates if one of the loops is active, and IV_DOM indicates which loop is dominant. The associated bits are located in LOOPCON1 and LOOPCON2.

3.6 PWM Frequency

The MCP19214/5 device uses a fixed frequency PWM control strategy. Both PWM channels will have the same switching frequency, but the phase difference between them is adjustable. The first PWM channel is considered the master channel, while the second is the slave channel. The user sets the MCP19214/5 switching frequency by configuring the PR2 register. The maximum allowable PDRVx duty cycle is adjustable and is controlled by the PWMRL register. The programmable range of the switching frequency will be 31.25 kHz to 2 MHz. The available switching frequency below 2 MHz is defined as F_{SW} = 8 MHz/N, where N is a whole number between $4 \leq N \leq 256$. Refer to **Section 24.0 "Enhanced PWM Module"** for details.

3.7 Reference Voltage Generators

There are four internal reference generators, two for each PWM channel. The digital-to-analog converters that control these reference have 8-bit resolution and are controlled by firmware.

The output voltage range of the DACs that set the reference voltage for the voltage loops is 0 mV to 2.048 mV (typical). Thus, the reference voltage of the voltage loops can be adjusted with a step of 8 mV. The associated registers that hold the DAC value are VREFCON1 and VREFCON2.

The output voltage range of the DACs that set the reference voltage for the current loops is 0 mV to 1.024 mV (typical). Thus, the reference voltage of the current loops can be adjusted with a step of 4 mV. The associated registers that hold the DAC value are CREFCON1 and CREFCON2.

3.8 Start-Up

To control the output current during start-up, the MCP19214/5 has the capability to monotonically increase system voltage or current at the user's discretion. This is accomplished through the control of the reference voltage DACs for each control loop. The entire start-up profile is under user control via software.

3.9 Temperature Management

3.9.1 THERMAL SHUTDOWN

To protect the MCP19214/5 from overtemperature conditions, a 150°C (typical) junction temperature thermal shutdown has been implemented. When the junction temperature reaches this limit, the device disables the output drivers. In Shutdown mode, both PDRV1 and PDRV1 outputs are disabled and the overtemperature flag (OTIF) is set in the PIR2 register. The internal LDO is also disabled during thermal shutdown phase. When the junction temperature is reduced by 20°C to 130°C (typical), the MCP19214/5 can resume normal output drive switching.

3.9.2 TEMPERATURE REPORTING

The MCP19214/5 has a second on-chip temperature monitoring circuit that can be read by the ADC through the analog test MUX. Refer to **Section 25.0 "Internal Temperature Indicator Module"** for details on this internal temperature monitoring circuit.

4.0 ELECTRICAL CHARACTERISTICS

4.1 ABSOLUTE MAXIMUM RATINGS †

V_{IN} - V_{GND} (operating) ..... –0.3V to +44V
V_{IN} (transient < 500 ms) ..... +48V
PDRVx ..... (GND - 0.3V) to (V_{DR} + 0.3V)
V_{DD} Internally Generated ..... +6.5V
V_{DR} Externally Generated ..... +13.5V
Voltage on $\overline{MCLR}$ with respect to GND ..... –0.3V to +13.5V
Maximum voltage: any other pin ..... +(V_{GND} - 0.3V) to (V_{DD} + 0.3V)
Maximum output current sunk by any single I/O pin ..... 25 mA
Maximum output current sourced by any single I/O pin ..... 25 mA
Maximum current sunk by all GPIO ..... 90 mA
Maximum current sourced by all GPIO ..... 35 mA
Storage Temperature ..... –65°C to +150°C
Maximum Junction Temperature ..... +150°C
Operating Junction Temperature ..... –40°C to +125°C
ESD protection on all pins (CDM) ..... 2.0 kV
ESD protection on all pins (HBM) ..... 1.0 kV
ESD protection on all pins (MM) 100V

† **Notice:** Stresses above those listed under “Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

4.2 Electrical Characteristics

Electrical Specifications: Unless otherwise noted, V_{IN} = 12V, F_{SW} = 300 kHz, T_A = +25°C, **Boldface** specifications apply over the T_A range of –40°C to +125°C

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Input						
Input Voltage	V_{IN}	**4.5**	—	**42**	V	Steady State
Input Quiescent Current	I_Q	—	6	**10**	mA	Not Switching, Analog circuitry disabled
Shutdown Current	I_{SHDN}	—	110	**400**	µA	Depends on the selected mode; see **Section 19.0 “Power-Down Mode (Sleep)”**
Linear Regulator V_{DD}						
Internal Circuitry Bias Voltage	V_{DD}	**4.75**	5.0	**5.25**	V	V_{IN} = 6.0V to 42V
Maximum External V_{DD} Output Current	I_{DD_OUT}	**35**	—	—	mA	V_{IN} = 6.0V to 42V, V_{DD} = 5.0V, (**Note 1**)
Line Regulation	ΔV_{DD-OUT}/ (V_{DD-OUT}*ΔV_{IN})	—	0.02	**0.1**	%/V	(V_{DD} + 1.0V) ≤ V_{IN} ≤ 20V (**Note 1**)
Load Regulation	ΔV_{DD-OUT}/ V_{DD-OUT}	**-1**	±0.1	**+1**	%	I_{DD_OUT} = 1 mA to 35 mA (**Note 1**)
Output Short Circuit Current	I_{DD_SC}	—	60	**90**	mA	V_{IN} = (V_{DD} + 1.0V) (**Note 1**)

Note 1: V_{DD} is the voltage present at the V_{DD} pin.

2: Dropout voltage is defined as the input-to-output voltage differential at which the output voltage drops 2% below its nominal value measured at a 1V differential between V_{IN} and V_{DD}.

3: Characterized during validation phase, not production tested.

4: The V_{DD} LDO will limit the total source current to less than 90 mA. Each pin individually can source a maximum of 15 mA.

4.2 Electrical Characteristics (Continued)

Electrical Specifications: Unless otherwise noted, V_{IN} = 12V, F_{SW} = 300 kHz, T_A = +25°C, **Boldface** specifications apply over the T_A range of –40°C to +125°C

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Dropout Voltage	V_{IN} - V_{DD}	—	0.3	**0.6**	V	I_{DD_OUT} = 35 mA, (**Notes 1** and **2**)
Power Supply Rejection Ratio	$PSRR_{LDO}$	40	60	—	dB	f ≤ 1000 Hz, I_{DD_OUT} = 25 mA C_{IN} = 0 µF, C_{DD} = 4.7 µF
Band Gap Voltage	BG	**1.215**	1.23	**1.245**	V	Trimmed at 1.0% tolerance
Input UVLO Voltage DAC						
UVLO Range (***VIN_ON***)	$UVLO_{ON}$	4.0	—	20	V	V_{IN} Falling
UVLO Hysteresis (***VIN_OFF***)	$UVLO_{HYS}$	—	5	—	%	Hysteresis is based upon $UVLO_{ON}$ setting
Resolution	nbits	—	6	—	Bits	Logarithmic DAC
Input UVLO Comparator						
Input Offset Voltage	V_{OS}	—	+/-10	20	mV	Ensure by design
Input-to-Output Delay	TD	—	5	—	µs	100 ns rise time to 1V overdrive on V_{IN}. V_{IN} > UVLO to flag set.
Input OVLO Voltage DAC						
OVLO (V_{IN} Rising) Range (***VIN_ON***)	$OVLO_{ON}$	**8.8**	—	**44**	V	V_{IN} Rising
OVLO Hysteresis (***VIN_OFF***)	$OVLO_{HYS}$	—	5	—	%	Hysteresis is based upon $OVLO_{ON}$ setting
Resolution	nbits	—	6	—	Bits	Logarithmic DAC
Input OVLO Comparator						
Input Offset Voltage	V_{OS}	—	+/-10	20	mV	
Input-to-Output Delay	TD	—	5	—	µs	100 nS rise time to 1V overdrive on V_{IN} V_{IN} > OVLO to flag set
V_{DD} UVLO Comparator						
Input Offset Voltage	V_{OS}	—	20	50	mV	
Input-to-Output Delay	TD	—	5	—	µs	**Note 3**
Voltage Loop Reference DAC(PWM #1/2)						
Resolution	nbits	—	8	—	Bits	Linear DAC
Full Scale Range	FSR	—	2048	—	mV	
Tolerance	$VVREF_{TOL}$	**-2**	+/-1	**+2**	%	Trimmed
Current Loop Reference DAC(PWM #1/2)						
Resolution	nbits	—	8	—	Bits	Linear DAC
Full Scale Range	FSR	—	1024	—	mV	The pedestal voltage is added on top of this voltage
Tolerance	$CVREF_{TOL}$	**-2**	+/-1	**+2**	%	Trimmed

Note 1: V_{DD} is the voltage present at the V_{DD} pin.

2: Dropout voltage is defined as the input-to-output voltage differential at which the output voltage drops 2% below its nominal value measured at a 1V differential between V_{IN} and V_{DD}.

3: Characterized during validation phase, not production tested.

4: The V_{DD} LDO will limit the total source current to less than 90 mA. Each pin individually can source a maximum of 15 mA.

4.2 Electrical Characteristics (Continued)

Electrical Specifications: Unless otherwise noted, V_{IN} = 12V, F_{SW} = 300 kHz, T_A = +25°C, **Boldface** specifications apply over the T_A range of –40°C to +125°C

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Differential Current Sense Amplifiers (A1)						
Input Offset Voltage	V_{OS}	—	+/-1	+/-2	mV	Trimmed, 6 bits adjustable
Amplifier PSRR	PSRR	60	—	—	dB	V_{CM} = 2V
Voltage Gain	$A1_{VCL}$	**9.5**	10	**10.5**	V/V	V_{CM} = ±0.1V
Low-Level Output	V_{OL}	—	50	—	mV	
Gain Bandwidth Product	GBWP	7	10		MHz	V_{VDD} = 5V
Input Impedance	R_{IN}		20		kΩ	
Common Mode Range	V_{CMR}	GND-0.3	—	2	V	
Common Mode Rejection Ratio	CMRR	30	—	—	dB	**Note 3**
Auxiliary Measuring Amplifier						
Input Offset Voltage	V_{OS}	—	1	**5**	mV	Trimmed
Voltage Gain	V_G	—	1/2	—	V/V	Programmable
Pedestal Voltage						
Pedestal Voltage Level	V_{PD}	**2028**	2048	**2068**	mV	Trimmed, 6 bits
Current loop Error Amplifiers (EA1)						
Input Current Offset	I_{OS}	**–4**	0	**+4**	µA	Trimmed, 4 bits
Error Amplifier PSRR	PSRR	80	—	—	dB	V_{CM} = 2.048V
Common Mode Input Range	V_{CM}	0.8	—	3	V	
Common Mode Rejection Ratio	CMRR	60	—		dB	V_{CM} = 0.8V-2.5V
Transconductance	g_m	**180**	200	**220**	µS	V_{CM} = 0.8V-3V, Trimmed, 4 bits
Gain Bandwidth Product	GBWP	—	3.5	—	MHz	
Low-Level Output	V_{OL}	—	100	—	mV	
Voltage Loop Error Amplifiers (EA2)						
Input Current Offset	I_{OS}	**–4**	0	**+4**	µA	Trimmed, 4 bits
Error Amplifier PSRR	PSRR	80	—	—	dB	V_{CM} = 1.024V
Common Mode Input Range	V_{CM}	0.8	—	3	V	
Common Mode Rejection Ratio	CMRR	60	—	—	dB	V_{CM} = 0.8V to 3V (**Note 3**)
Transconductance	g_m	**180**	200	**220**	µS	V_{CM} = 0.8V to 3.0V, Trimmed, 4 bits
Gain Bandwidth Product	GBWP	—	3.5	—	MHz	
Low-Level Output	V_{OL}	—	100	—	mV	
Peak Current Sense Input						
Maximum Current Sense Signal Voltage	V_{IP_MAX}	—	1.2	1.5	V	

Note 1: V_{DD} is the voltage present at the V_{DD} pin.

2: Dropout voltage is defined as the input-to-output voltage differential at which the output voltage drops 2% below its nominal value measured at a 1V differential between V_{IN} and V_{DD}.

3: Characterized during validation phase, not production tested.

4: The V_{DD} LDO will limit the total source current to less than 90 mA. Each pin individually can source a maximum of 15 mA.

4.2 Electrical Characteristics (Continued)

Electrical Specifications: Unless otherwise noted, V_{IN} = 12V, F_{SW} = 300 kHz, T_A = +25°C, **Boldface** specifications apply over the T_A range of –40°C to +125°C

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
PWM Comparator						
Common-Mode Input Voltage Range	V_{CMR}	GND-0.3	—	3	V	
Input-to-Output Delay	TD	—	20	**50**	ns	**Note 3**
Peak Current Leading Edge Blanking						
Resolution	LEB	—	2	—	Bits	
Blanking Time Adjustable Range	LEB_{RANGE}	0	—	200	ns	4 Step Programmable Range (0, 50,100 and 200 ns)
Offset Adjustment (I_P Sense)						
Resolution	OS_{ADJ}	—	4	—	Bits	
Offset Adjustment Range	OS_{ADJ_RANGE}	0		750	mV	
Offset Adjustment Step Size	OS_{ADJ_STEP}	—	50	—	mV	Linear Steps
Adjustable Slope Compensation						
Resolution	SC_{RES}		6		Bits	Log steps
Slope	m	4		437	mV/µs	
Slope Step Size	SC_{STEP}	—	8	—	%	Log Steps
Ramp Set Point Tol	m_{TOL}	—	+/-10	+/-32	%	
V_{DR} UVLO						
V_{DR} UVLO (2.7V V_{DR} Falling)	$V_{DR\_UVLO\_2.7}$	2.6	2.7	2.8	V	
V_{DR} UVLO (2.7 V_{DR} Rising)	$V_{DR\_UVLO\_2.7}$	2.9	3.05	3.2	V	
V_{DR} UVLO (2.7V) Hysteresis	$V_{DR\_UVLO\ 2.7\ HYS}$	300	350	400	mV	
V_{DR} UVLO (5.4V V_{DR} Falling)	$V_{DR\_UVLO\_5.4}$	5.2	5.4	5.6	V	
V_{DR} UVLO (5.4V V_{DR} Rising)	$V_{DR\_UVLO\_5.4}$	5.8	6.1	6.4	V	
V_{DR} UVLO (5.4V) Hysteresis	$V_{DR\_UVLO\ 5.4\ HYS}$	600	700	800	mV	
Output Driver (PDRV 1/2)						
PDRV Gate Drive Source Resistance	$R_{DR\text{-}SCR}$	—	—	**12**	Ω	V_{DR} = 4.5V (**Note 3**)
PDRV Gate Drive Sink Resistance	$R_{DR\text{-}SINK}$	—	—	**12**	Ω	V_{DR} = 4.5V (**Note 3**)
PDRV Gate Drive Source Current	$I_{DR\text{-}SCR}$	— —	0.5 1.0	— —	A	V_{DR} = 5V V_{DR} = 1 0V (**Note 3**)
PDRV Gate Drive Sink Current	$I_{DR\text{-}SINK}$	— —	0.5 1.0	— —	A	V_{DR} = 5V V_{DR} = 10V (**Note 3**)

Note 1: V_{DD} is the voltage present at the V_{DD} pin.

2: Dropout voltage is defined as the input-to-output voltage differential at which the output voltage drops 2% below its nominal value measured at a 1V differential between V_{IN} and V_{DD}.

3: Characterized during validation phase, not production tested.

4: The V_{DD} LDO will limit the total source current to less than 90 mA. Each pin individually can source a maximum of 15 mA.

4.2 Electrical Characteristics (Continued)

Electrical Specifications: Unless otherwise noted, V_{IN} = 12V, F_{SW} = 300 kHz, T_A = +25°C, **Boldface** specifications apply over the T_A range of –40°C to +125°C

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
Oscillator/PWM						
Internal Oscillator Frequency	F_{OSC}	**7.60**	8.00	**8.40**	MHz	
Switching Frequency	F_{SW}	—	F_{OSC}/N	—	MHz	
Switching Frequency Range Select	N	4	—	255		F_{MAX} = 2 MHz
A/D Converter (ADC) Characteristics						
Resolution	N_R	—	—	10	Bits	
Integral Error	E_{IL}	—	—	**±1**	LSb	V_{REF_ADC} = 4.096V (**Note 3**)
Differential Error	E_{DL}	—	—	**±1**	LSb	No missing code in 10 bits, V_{REF_ADC} = 4.096V (**Note 3**)
Offset Error	E_{OFF}	—	+1.5	**+7**	LSb	V_{REF_ADC} = 4.096V (**Note 3**)
Gain Error	E_{GN}	—	—	**±6**	LSb	V_{REF_ADC} = 4.096V (**Note 3**)
Reference Voltage	V_{REF_ADC}	**4.055**	4.096	**4.137**	V	Trimmed
Full-Scale Range	$FSR_{A/D}$	GND	—	V_{REF_ADC}	V	
GPIO Pins						
Maximum GPIO Sink Current (all pin ports combined)	I_{SINK_GPIO}	—	—	90	mA	**Note 4**
Maximum GPIO Sink Current (all pin ports combined)	I_{SOURCE_GPIO}	—	—	90	mA	**Note 4**
GPIO Weak Pull-up Current	$I_{PULL-UP\_GPIO}$	50	250	400	µA	
GPIO Input Low Voltage	V_{GPIO_IL}	**GND**	—	**0.8**	V	I/O Port with TTL buffer, V_{DD} = 5V
		GND	—	**$0.2V_{DD}$**	V	I/O Port with Schmitt Trigger buffer, V_{DD} = 5V
		GND	—	**$0.2V_{DD}$**	V	$\overline{MCLR}$
GPIO Input High Voltage	V_{GPIO_IH}	**2.0**	—	**V_{DD}**	V	I/O Port with TTL buffer, V_{DD} = 5V
		$0.8V_{DD}$	—	**V_{DD}**	V	I/O Port with Schmitt Trigger buffer, V_{DD} = 5V
		$0.8V_{DD}$	—	**V_{DD}**	V	$\overline{MCLR}$

Note 1: V_{DD} is the voltage present at the V_{DD} pin.

2: Dropout voltage is defined as the input-to-output voltage differential at which the output voltage drops 2% below its nominal value measured at a 1V differential between V_{IN} and V_{DD}.

3: Characterized during validation phase, not production tested.

4: The V_{DD} LDO will limit the total source current to less than 90 mA. Each pin individually can source a maximum of 15 mA.

4.2 Electrical Characteristics (Continued)

Electrical Specifications: Unless otherwise noted, V_{IN} = 12V, F_{SW} = 300 kHz, T_A = +25°C, **Boldface** specifications apply over the T_A range of –40°C to +125°C

Parameters	Sym.	Min.	Typ.	Max.	Units	Conditions
GPIO Output Low Voltage	V_{GPIO_OL}	—	—	**0.6**	V	I_{OL} = 7 mA, V_{DD} = 5V
GPIO Output High Voltage	V_{GPIO_OH}	V_{DD}-0.7	—	—	V	I_{OH} = 2.5 mA, V_{DD} = 5V
GPIO Input Leakage Current	GPIO_I_{IL}	—	±0.1	**±1**	µA	Negative current is defined as current sourced by the pin.
Thermal Shutdown						
Thermal Shutdown	T_{SHD}	—	150	—	°C	
Thermal Shutdown Hysteresis	T_{SHD_HYS}	—	20	—	°C	

Note 1: V_{DD} is the voltage present at the V_{DD} pin.

2: Dropout voltage is defined as the input-to-output voltage differential at which the output voltage drops 2% below its nominal value measured at a 1V differential between V_{IN} and V_{DD}.

3: Characterized during validation phase, not production tested.

4: The V_{DD} LDO will limit the total source current to less than 90 mA. Each pin individually can source a maximum of 15 mA.

4.3 Thermal Specifications

Parameters	Sym.	Min.	Typ.	Max.	Units
Temperature Ranges					
Specified Temperature Range	T_A	–40	—	+125	°C
Operating Junction Temperature Range	T_J	–40	—	+125	°C
Maximum Junction Temperature	T_J	—	—	+150	°C
Storage Temperature Range	T_A	–65	—	+150	°C
Thermal Package Resistances					
Thermal Resistance, 28L-QFN 5x5	θ_{JA}	—	26	—	°C/W
Thermal Resistance, 32L-QFN 5x5	θ_{JA}	—	25.8	—	°C/W

NOTES:

5.0 DIGITAL ELECTRICAL CHARACTERISTICS

5.1 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS
3. $T_{CC:ST}$ (I^2C specifications only)
4. Ts (I^2C specifications only)

T			
F	Frequency	T	Time

Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	wr	$\overline{WR}$
mc	$\overline{MCLR}$		

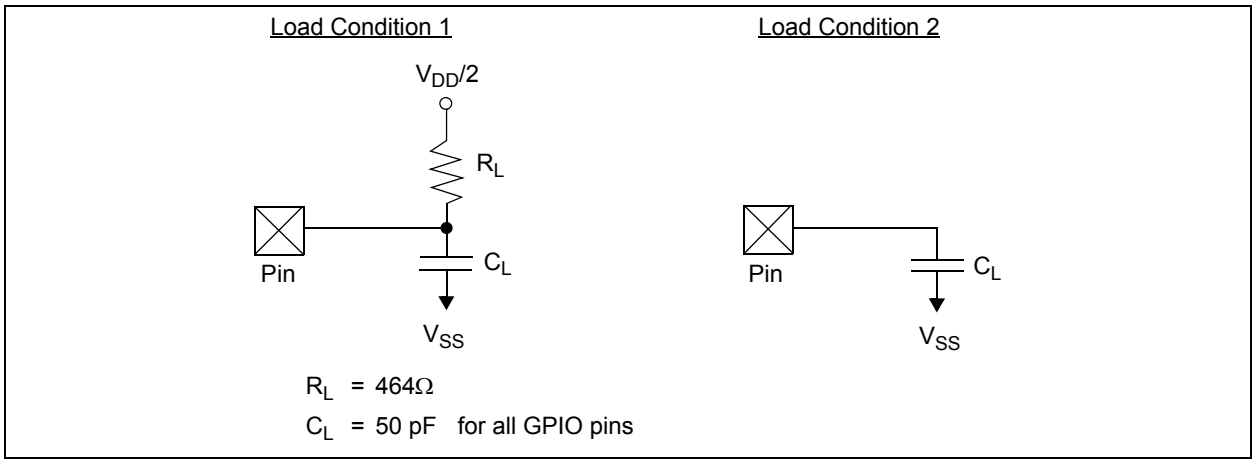
Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (high-impedance)	V	Valid
L	Low	Z	High-Impedance
I^2C only			
AA	output access	High	High
BUF	Bus free	Low	Low

$T_{CC:ST}$ (I^2C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

FIGURE 5-1: LOAD CONDITIONS



5.2 AC Characteristics: MCP19214 (Industrial, Extended)

FIGURE 5-2: I/O TIMING

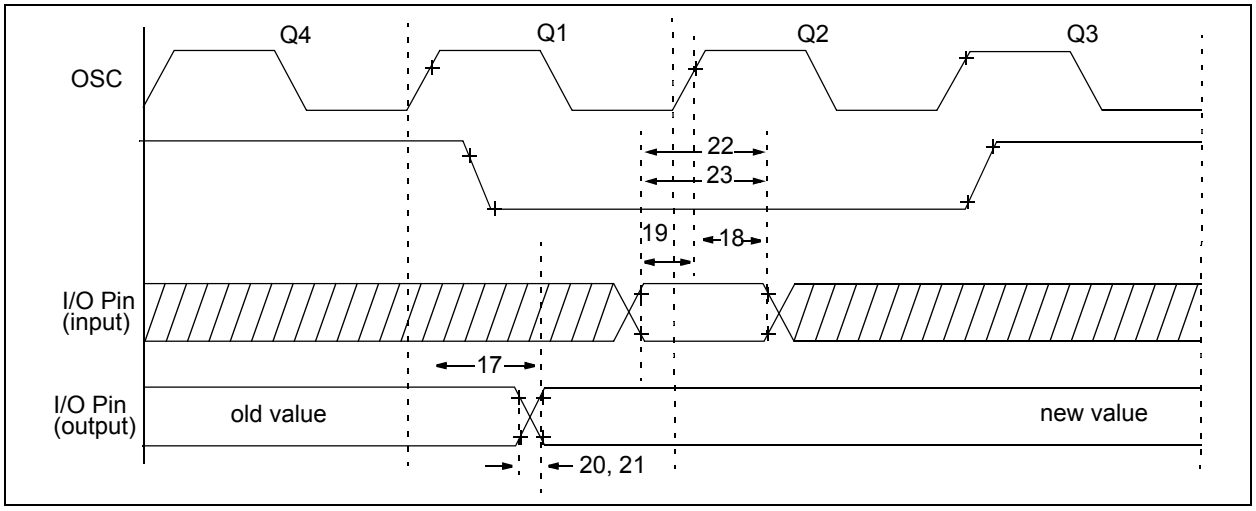


TABLE 5-1: I/O TIMING REQUIREMENTS

Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	70*	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	20	—	—	ns	
20	TioR	Port output rise time	—	32	40	ns	
21	TioF	Port output fall time	—	15	30	ns	
22*	Tinp	INT pin high or low time	25	—	—	ns	
23*	T_{RABP}	GPIO interrupt-on-change new input level time	T_{CY}	—	—	ns	

† Data in "Typ" column is at V_{IN} = 12V (V_{DD} = 5V), 25°C unless otherwise stated.

* These parameters are characterized but not tested.

FIGURE 5-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

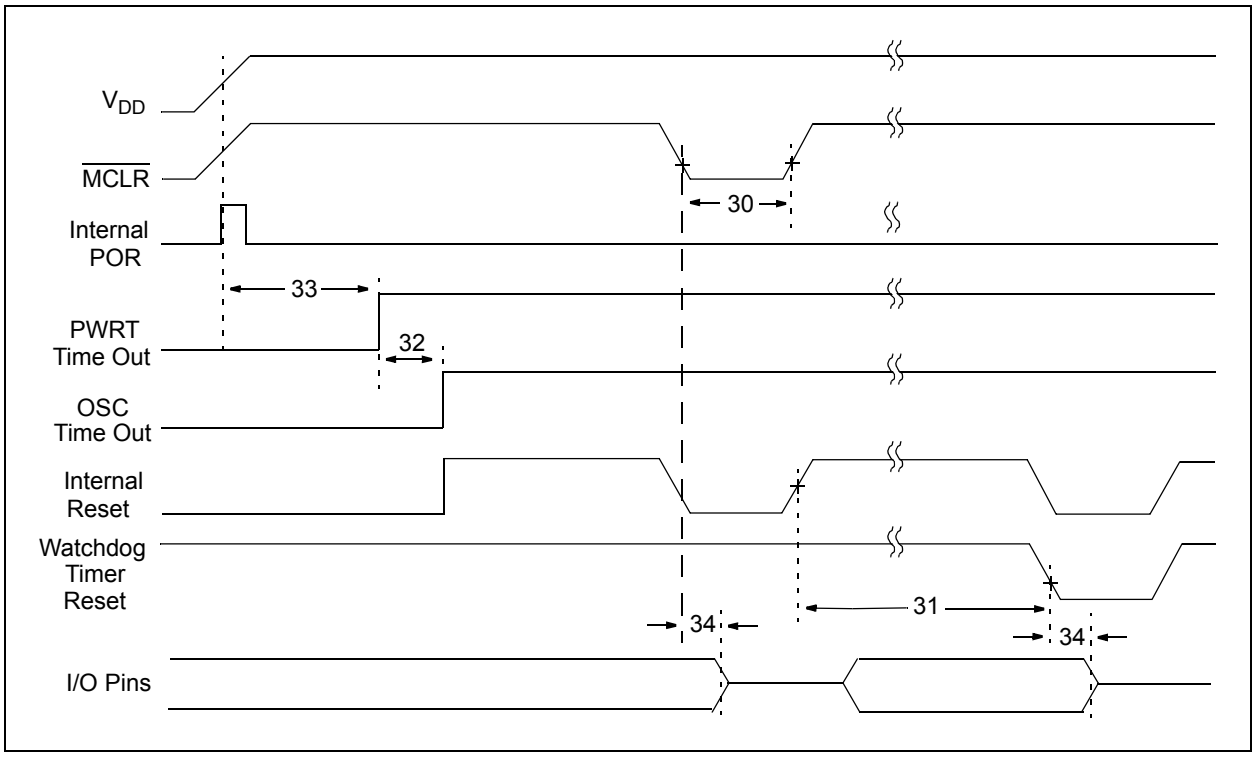


FIGURE 5-4: BROWN-OUT RESET TIMING AND CHARACTERISTICS

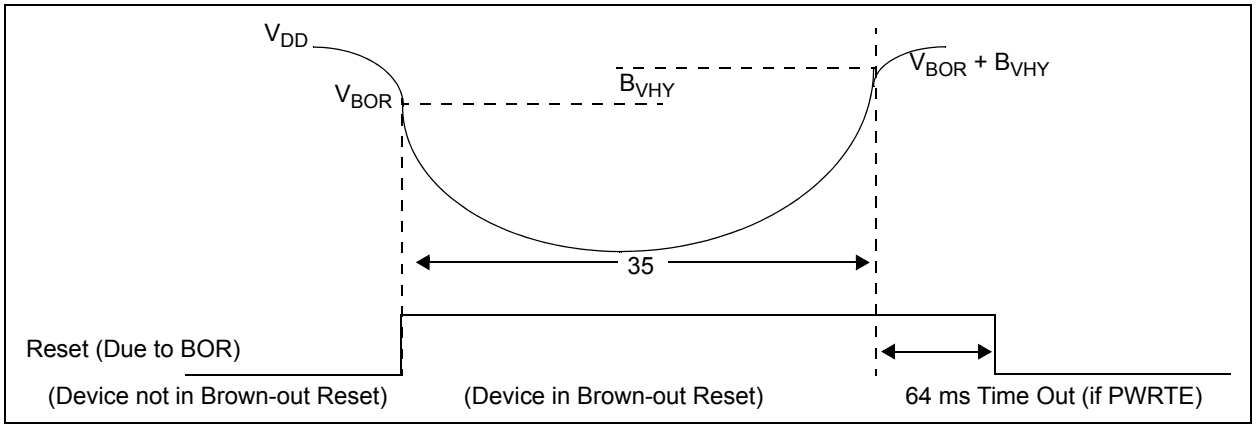


TABLE 5-2: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
30	T_{MCL}	$\overline{MCLR}$ Pulse Width (low)	2	—	—	µs	V_{DD} = 5 V, –40°C to +85°C
31	T_{WDT}	Watchdog Timer Time-Out Period (No Prescaler)	7	18	33	ms	V_{DD} = 5 V, –40°C to +85°C
32	T_{OST}	Oscillation Start-Up Timer Period	—	$1024T_{OSC}$	—	—	T_{OSC} = OSC1 period
33*	T_{PWRT}	Power-up Timer Period (4 x T_{WDT})	28	72	132	ms	V_{DD} = 5 V, –40°C to +85°C
34	T_{IOZ}	I/O high impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0	µs	
	V_{BOR}	Brown-out Reset voltage	2.0	2.13	2.3	V	
	B_{VHY}	Brown-out Hysteresis	—	100	—	mV	
35	T_{BCR}	Brown-out Reset pulse width	100*	—	—	µs	$V_{DD} \leq V_{BOR}$ (D005)
48	$T_{CKEZ-TMR}$	Delay from clock edge to timer increment	$2T_{OSC}$	—	$7T_{OSC}$		

* These parameters are characterized but not tested.

† Data in "Typ." column is at V_{IN} = 12V (V_{DD} = AV_{DD} = 5V), 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 5-5: TIMER0 EXTERNAL CLOCK TIMING

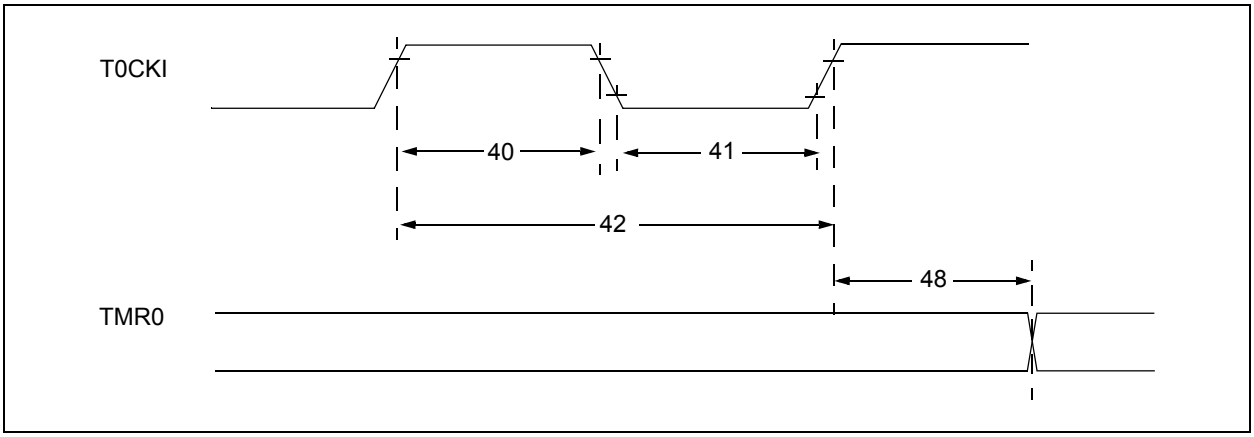


TABLE 5-3: TIMER0 EXTERNAL CLOCK REQUIREMENTS

Param. No.	Sym.	Characteristic		Min.	Typ.†	Max.	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		Greater of: 20 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in “Typ.” column is at V_{IN} = 12V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 5-4: MCP19214/5 A/D CONVERTER (ADC) CHARACTERISTICS

Electrical Specifications: Unless otherwise noted, operating temperature = –40°C ≤ T_A ≤ +125°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
AD01	N_R	Resolution	—	—	10 bits	bit	
AD02	E_{IL}	Integral Error	—	—	**±1**	LSb	AV_{DD} = 5V
AD03	E_{DL}	Differential Error	—	—	**±1**	LSb	No missing codes to 10 bits AV_{DD} = 5V
AD04	E_{OFF}	Offset Error	—	+3.0	**+7**	LSb	AV_{DD} = 5V
AD07	E_{GN}	Gain Error	—	±2	**±6**	LSb	AV_{DD} = 5V
AD07	V_{AIN}	Full-Scale Range	A_{GND}	—	4.096	V	
AD08	Z_{AIN}	Recommended Impedance of Analog Voltage Source	—	—	10	kΩ	

* These parameters are characterized but not tested.

† Data in 'Typ.' column is at V_{IN} = 12V (AV_{DD} = 5V), 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 5-5: MCP19214/5 A/D CONVERSION REQUIREMENTS

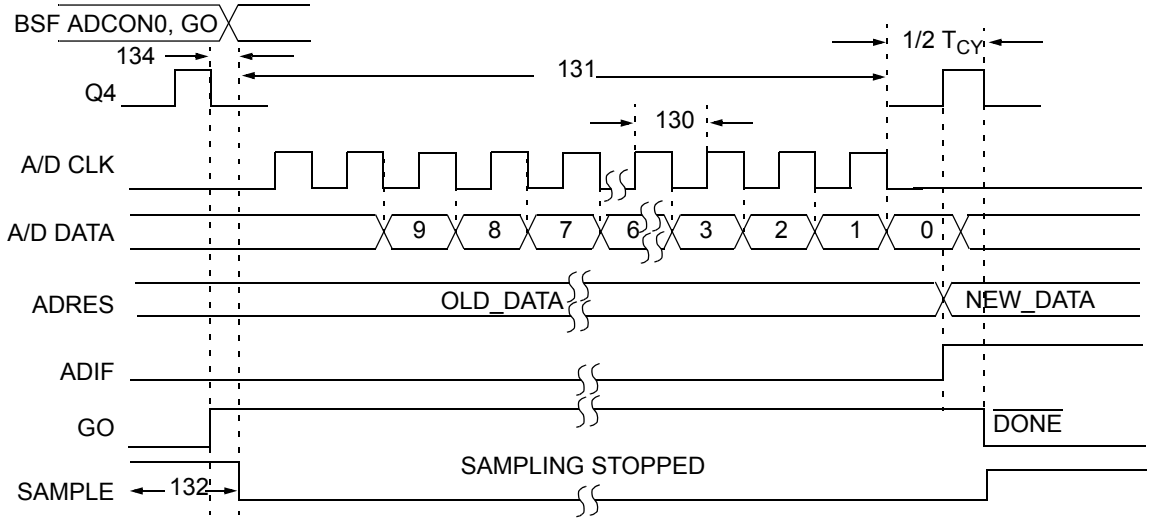
Electrical Specifications: Unless otherwise noted, operating temperature = 40°C ≤ T_A ≤ +125°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
AD130*	T_{AD}	A/D Clock Period	1.6	—	9.0	µs	T_{OSC}-based
		A/D Internal RC Oscillator Period	1.6	4.0	6.0	µs	ADCS<1:0> = 11 (ADRC mode)
AD131	T_{CNV}	Conversion Time (not including Acquisition Time)(1)	—	11	—	T_{AD}	Set GO/$\overline{\text{DONE}}$ bit to new data in A/D Result registers
AD132*	T_{ACQ}	Acquisition Time	—	11.5	—	µs	
AD133*	T_{AMP}	Amplifier Settling Time	—	—	5	µs	
AD134	T_{GO}	Q4 to A/D Clock Start	—	T_{OSC}/2	—	—	

* These parameters are characterized but not tested.

† Data in 'Typ.' column is at V_{IN} = 12V (V_{DD} = AV_{DD} = 5V), 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: ADRESH and ADRESL registers may be read on the following T_{CY} cycle.

FIGURE 5-6: A/D CONVERSION TIMING



Note: If the A/D clock source is selected as RC, a time of T_{CY} is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

6.0 CONFIGURING THE MCP19214/5

The MCP19214/5 devices are digitally enhanced analog controllers. This means that most of the device configuration is handled through register settings instead of adding external components. There are several internal configurable modules used to interface the analog circuitry to digital core. These modules are very similar to a standard comparator module found in many PIC microcontrollers. The following sections detail how to set the analog control registers for all the configurable parameters.

6.1 Input Undervoltage and Overvoltage Lockout (UVLO and OVLO)

VINCON is the comparator control register for both the VINUVLO and VINOVLO registers. It contains the enable bits, the polarity edge detection bits and the status output bits for both protection circuits. The interrupt flags <UVLOIF> and <OVLOIF> in the PIR2 register are independent of the enable <UVLOEN> and <OVLOEN> bits in the VINCON register. The <UVLOOUT> Undervoltage Lockout Status Output bit in the VINCON register indicates if an UVLO event has occurred. The <OVLOOUT> Overvoltage Lockout Status Output bit in the VINCON register indicates if an OVLO event has occurred.

The VINUVLO register contains the digital value that sets the input undervoltage lockout. UVLO has a range of 4V-20V. For V_{IN} values below this range and above processor come-alive (V_{DD} = 2V), the UVLO comparator and the UVLOOUT Status bit will indicate an undervoltage condition. If using UVLO to determine power-up V_{IN}, it is recommended to poll the UVLOOUT bit for status. When the input voltage on the V_{IN} pin to the MCP19214/5 is below this programmed level and the <UVLOEN> bit in the VINCON register is set, both PDRV1 and PDRV2 gate drivers are disabled. This bit is automatically cleared when the MCP19214/5 V_{IN} voltage rises above this programmed level.

The VINOVLO register contains the digital value that sets the input overvoltage lockout. OVLO has a range of 8.8V-44V. When the input voltage on the V_{IN} pin to the MCP19214/5 is above this programmed level and the <OVLOEN> bit in the VINCON register is set, both PDRV1 and PDRV2 gate drivers are disabled. This bit is automatically cleared when the MCP19214/5 V_{IN} voltage drops below this programmed level. Refer to Figure 26-1.

Note:	The UVLOIF and OVLOIF interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit (GIE) in the INTCON register.

REGISTER 6-1: VINCON: UVLO AND OVLO COMPARATOR CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0
UVLOEN	UVLOOUT	UVLOINTP	UVLOINTN	OVLOEN	OVLOOUT	OVLOINTP	OVLOINTN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **UVLOEN:** UVLO Comparator Module Logic Enable bit
- 1 = UVLO Comparator Module Logic enabled
- 0 = UVLO Comparator Module Logic disabled

bit 6 **UVLOOUT:** Undervoltage Lockout Status Output
- 1 = UVLO event has occurred
- 0 = No UVLO event has occurred

bit 5 **UVLOINTP:** UVLO Comparator Interrupt on Positive Going Edge Enable bit
- 1 = The UVLOIF interrupt flag will be set upon a positive going edge of the UVLO
- 0 = No UVLOIF interrupt flag will be set upon a positive going edge of the UVLO

bit 4 **UVLOINTN:** UVLO Comparator Interrupt on Negative Going Edge Enable bit
- 1 = The UVLOIF interrupt flag will be set upon a negative going edge of the UVLO
- 0 = No UVLOIF interrupt flag will be set upon a negative going edge of the UVLO

REGISTER 6-1: VINCON: UVLO AND OVLO COMPARATOR CONTROL REGISTER (CONTINUED)

bit 3 **OVLOEN:** OVLO Comparator Module Logic enable bit
1 = OVLO Comparator Module Logic enabled
0 = OVLO Comparator Module Logic disabled

bit 2 **OVLOOUT:** Overvoltage Lockout Status Output bit
1 = OVLO event has occurred
0 = No OVLO event has occurred

bit 1 **OVLOINTP:** OVLO Comparator Interrupt on Positive Going Edge Enable bit
1 = The OVLOIF interrupt flag will be set upon a positive going edge of the OVLO
0 = No OVLOIF interrupt flag will be set upon a positive going edge of the OVLO

bit 0 **OVLOINTN:** OVLO Comparator Interrupt on Negative Going Edge Enable bit
1 = The OVLOIF interrupt flag will be set upon a negative going edge of the OVLO
0 = No OVLOIF interrupt flag will be set upon a negative going edge of the OVLO

REGISTER 6-2: VINUVLO: INPUT UNDERVOLTAGE LOCKOUT REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	UVLO5	UVLO4	UVLO3	UVLO2	UVLO1	UVLO0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **UVLO<5:0>:** Undervoltage Lockout Configuration bits
UVLO(V) = 3.5472 * (1.0285^N) where N = the decimal value written to the VINUVLO register from 0 to 63

REGISTER 6-3: VINOVLO: INPUT OVERVOLTAGE LOCKOUT REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	OVLO5	OVLO4	OVLO3	OVLO2	OVLO1	OVLO0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **OVLO<5:0>:** Overvoltage Lockout Configuration bits
OVLO(V) = 7.4847 * (1.0286^N) where N = the decimal value written to the VINOVLO register from 0 to 63

6.2 V_{DD} VOLTAGE UVLO Control Register

The VDDCON register holds the set-up control bits for the UVLO circuits of the internal voltage regulator (V_{DD} LDO).

REGISTER 6-4: VDDCON: V_{DD} UVLO CONTROL REGISTER (ADDRESS 98H)

R/W-0	R-x	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
VDDUVEN	VDDUVOUT	VDDUVINTP	VDDUVINTN	—	—	VDDUV1	VDDUV0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **VDDUVEN:** V_{DD} UVLO Comparator Module Logic enable bit
1 = V_{DD} UVLO Comparator Module Logic enabled
0 = V_{DD} UVLO Comparator Module Logic disabled

bit 6 **VDDUVOUT**: V_{DD} Undervoltage Lockout Status Output
1 = V_{DD} UVLO event has occurred
0 = No V_{DD} UVLO event has occurred

bit 5 **VDDUVINTP:** V_{DD} UVLO Comparator Interrupt on Positive Going Edge Enable bit
1 = The VDDUVIF interrupt flag will be set upon a positive going edge of the V_{DD} UVLO
0 = No VDDUVIF interrupt flag will be set upon a positive going edge of the V_{DD} UVLO

bit 4 **VDDUVINTN:** V_{DD} UVLO Comparator Interrupt on Negative Going Edge Enable bit
1 = The VDDUVIF interrupt flag will be set upon a negative going edge of the V_{DD} UVLO
0 = No VDDUVIF interrupt flag will be set upon a negative going edge of the V_{DD} UVLO

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **VDDUV<1:0>:** V_{DD} UVLO Thresholds Configuration bits
00 = 3V
01 = 3.5V
10 = 4V
11 = 4.5V

6.3 Slope Compensation

This register defines the slope compensation ramp that is added to the error amplifier output. The six SLPS <5:0> bits control the slew rate of the ramp. The SLPBY bit controls the slope compensation circuitry. Setting this bit bypasses the slope compensation circuitry. No slope compensation will be added to the error signal.

REGISTER 6-5: SLPCRCON1: SLOPE COMPENSATION RAMP REGISTER FOR PWM CHANNEL #1 (ADDRESS 9Ch)

U-0	R/W-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SLPBY	SLPS5	SLPS4	SLPS3	SLPS2	SLPS1	SLPS0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **SLPBY**: Slope Compensation Bypass Control bit
1 = Slope compensation is Bypassed
0 = Slope compensation is not Bypassed

bit 5-0 **SLPS<5:0>**: Slope Compensation Slew Rate Control bits
SLPS[mV/us] = 4.4683 * 1.077^N where N is the decimal value written to the SLPCRCON1 Register <SLPS5:SLPS0> from 0 to 63

REGISTER 6-6: SLPCRCON2: SLOPE COMPENSATION RAMP REGISTER FOR PWM CHANNEL #2 (ADDRESS 9Dh)

U-0	R/W-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SLPBY	SLPS5	SLPS4	SLPS3	SLPS2	SLPS1	SLPS0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **SLPBY**: Slope Compensation Bypass Control bit
1 = Slope compensation is Bypassed
0 = Slope compensation is not Bypassed

bit 5-0 **SLPS<5:0>**: Slope Compensation Slew Rate Control bits
SLPS[mV/us] = 4.4683 * 1.077^N where N is the decimal value written to the SLPCRCON2 Register <SLPS5:SLPS0> from 0 to 63.

6.4 Input Current Offset Adjust

This register contains the four bits that set the Input Current Offset Adjustment. This Offset Adjustment gets added to the Primary Input Current voltage measurement.

REGISTER 6-7: ICOACON: INPUT CURRENT OFFSET ADJUST CONTROL REGISTER (ADDRESS 9Eh)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
IC1OAC3	IC1OAC2	IC1OAC1	IC1OAC0	IC2OAC3	IC2OAC2	IC2OAC1	IC2OAC0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-4 **IC1OAC<7:4>:** Input current offset adjustment Configuration bits for PWM channel #1

0000 = 0 mV
0001 = 50 mV
0010 = 100 mV
0011 = 150 mV
0100 = 200 mV
0101 = 250 mV
0110 = 300 mV
0111 = 350 mV
1000 = 400 mV
1001 = 450 mV
1010 = 500 mV
1011 = 550 mV
1100 = 600 mV
1101 = 650 mV
1110 = 700 mV
1111 = 750 mV

bit 3-0 **IC2OAC1<3:0>:** Input current offset adjustment Configuration bits for PWM channel #2

0000 = 0 mV
0001 = 50 mV
0010 = 100 mV
0011 = 150 mV
0100 = 200 mV
0101 = 250 mV
0110 = 300 mV
0111 = 350 mV
1000 = 400 mV
1001 = 450 mV
1010 = 500 mV
1011 = 550 mV
1100 = 600 mV
1101 = 650 mV
1110 = 700 mV
1111 = 750 mV

6.5 Leading Edge Blanking

This register contains the two bits that set the Input Peak Leading Edge Blanking. Leading Edge Blanking is applied to the Primary Input Current measurement. The amount of leading edge blanking time is controlled by ICLEB<1:0> bits in the ICLEBCON register.

REGISTER 6-8: ICLEBCON: INPUT CURRENT LEADING EDGE BLANKING CONTROL REGISTER (ADDRESS 9Fh)

U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	IC1LEBC1	IC1LEBC0	IC2LEBC1	IC2LEBC0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-2 **IC1LEBC<1:0>:** Input current Leading Edge Blanking Configuration bits for PWM channel #1
00 = 0 nS
01 = 50 nS
10 = 100 nS
11 = 200 nS

bit 1-0 **IC2LEBC<1:0>:** Input current Leading Edge Blanking Configuration bits for PWM channel #2
00 = 0 nS
01 = 50 nS
10 = 100 nS
11 = 200 nS

6.6 PWM channel #1 I/V Good Comparators Control Register

This register contains the configuration bits that reset the compensation networks and set the behaviour of the interrupt generated by the IV_GOOD signal. The outputs of the IV_GOOD and IV_DOM comparators can be read from this register.

REGISTER 6-9: LOOPCON1: I/V LOOPS CONTROL REGISTER FOR PWM #1 (ADDRESS 99h)

R/W-1	U-0	R-x	R/W-0	R/W-0	R-x	U-0	U-0
IVLRES	—	IVGOOD	IGDINTP	IGDINTN	IV_DOM	—	—
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IVLRES: C**urrent and **V**oltage **L**oop **RES**et bit PWM channel #1
1 = The CCOMP1 and ICCOMP1 pins are connected to ground
0 = The CCOMP1 and ICCOMP1 pins are disconnected from ground (normal operation)

bit 6 **Unimplemented:**

bit 5 **IV_GOOD:** Output of the IV_GOOD Comparator of the PWM channel #1
1 = The current or voltage loop is in regulation
0 = The current/voltage loops are out of regulation

bit 4 **IVGDINTP:** IV_GOOD Comparator Interrupt on Positive Going Edge Enable bit (PWM #1)
1 = The IVGD1IF interrupt flag will be set upon a positive going edge of the IGOOD signal
0 = No IVGD1IF interrupt flag will be set upon a positive going edge of the IGOOD signal

bit 3 **IVGDINTP:** IV_GOOD Comparator Interrupt on Negative Going Edge Enable bit (PWM #1)
1 = The IVGD1IF interrupt flag will be set upon a negative going edge of the IGOOD signal
0 = No IVGD1IF interrupt flag will be set upon a negative going edge of the IGOOD signal

bit 2 **IV_DOM:** Output of the Dominant Loop Comparator for PWM channel #1
1 = The voltage loop controls the PWM
0 = The current loop controls the PWM

bit 1 **Unimplemented:** Read as '0'

bit 0 **Unimplemented:** Read as '0'

6.7 PWM Channel #2 I/V Good Comparators Control Register

This register contains the configuration bits that reset the compensation networks and set the behaviour of the interrupt generated by the IV_GOOD signal. The outputs of the IV_GOOD and IV_DOM comparators can be read from this register.

REGISTER 6-10: LOOPCON2: I/V LOOPS CONTROL REGISTER FOR PWM #2 (ADDRESS 9Ah)

R/W-1	U-0	R-x	R/W-0	R/W-0	R-x	U-0	U-0
IVLRES	—	IV_GOOD	IGDINTP	IGDINTN	IV_DOM	—	—
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IVLRES: C**urrent and **V**oltage **L**oop **RES**et bit PWM channel #2
1 = the CCOMP2 and VCOMP2 pins are connected to ground
0 = the CCOMP2 and VCOMP2 pins are disconnected from ground (normal operation)

bit 6 **Unimplemented:** Read as '0'

bit 5 **IV_GOOD:** Output of the IV_GOOD Comparator of the PWM channel #2
1 = the current or voltage loop is in regulation
0 = the current/voltage loops are out of regulation

bit 4 **IVGDINTP:** IV_GOOD Comparator Interrupt on Positive Going Edge Enable bit (PWM #2)
1 = The IVGD2IF interrupt flag will be set upon a positive going edge of the IV_GOOD signal
0 = No IVGD2IF interrupt flag will be set upon a positive going edge of the IV_GOOD signal

bit 3 **IVGDINTP:** IV_GOOD Comparator Interrupt on Negative Going Edge Enable bit (PWM #2)
1 = The IVGD2IF interrupt flag will be set upon a negative going edge of the IV_GOOD signal
0 = No IVGD2IF interrupt flag will be set upon a negative going edge of the IV_GOOD signal

bit 2 **IV_DOM:** Output of the Dominant Loop Comparator for PWM channel #2
1 = the voltage loop controls the PWM
0 = the current loop controls the PWM

bit 1 **Unimplemented:** Read as '0'

bit 0 **Unimplemented:** Read as '0'

6.8 Reference Voltage Control Registers for the Voltage Regulation Loops

These registers hold the digital value that controls the DAC used to set the output voltage regulation set point.

REGISTER 6-11: VREFCON1: VOLTAGE REGULATION SET-POINT REGISTER FOR PWM CHANNEL #1 (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VREF7	VREF6	VREF5	VREF4	VREF3	VREF2	VREF1	VREF0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **VREF<7:0>:** Reference voltage generator for voltage loop of PWM channel #1 bits

V_{REF} [mV] = 2048*(VREF(dec)/2^8)

REGISTER 6-12: VREFCON2: VOLTAGE REGULATION SET-POINT REGISTER FOR PWM CHANNEL #2 (ADDRESS 92h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VREF7	VREF6	VREF5	VREF4	VREF3	VREF2	VREF1	VREF0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **VREF<7:0>:** Reference voltage generator for voltage loop of PWM channel #2 bits

V_{REF} [mV] = 2048*(VREF(dec)/2^8)

6.9 Reference Voltage Control Registers for the Current Regulation Loops

These registers hold the digital value that controls the DAC used to set the output current regulation set point.

REGISTER 6-13: CREFCON1: CURRENT REGULATION SET-POINT REGISTER FOR PWM CHANNEL #1 (ADDRESS 8Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CREF7	CREF6	CREF5	CREF4	CREF3	CREF2	CREF1	CREF0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **CREF<7:0>:** Reference voltage generator for current loop of PWM channel #1 bits
CREF[mV] = 1024*(CREF(dec)/2^8)

REGISTER 6-14: CREFCON2: CURRENT REGULATION SET-POINT REGISTER FOR PWM CHANNEL #2 (ADDRESS 90h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CREF7	CREF6	CREF5	CREF4	CREF3	CREF2	CREF1	CREF0
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **CREF<7:0>:** Reference voltage generator for current loop of PWM channel #2 bits
CREF[mV] = 1024*(CREF(dec)/2^8)

6.10 Peripheral Control

This register contains the bits that control the various peripheral settings in the MCP19214/5.

The PDRVxEN bits enable the gates drivers ouputs for each PWM channel.

The ISxPUEN bit determines if the IP inputs are pulled up to V_{DD}.

The LDO_LP bit enables the low-power operating mode of LDO.

REGISTER 6-15: PE1: PERIPHERAL ENABLE REGISTER 1 (ADDRESS 107h)

R/W-0	R/W-0	U-0	U-0	R/W-1	R/W-1	R/W-0	R/W-0
PDR1VEN	PDR2VEN	—	—	IS1PUEN	IS2PUEN	LDO_LV	LDO_LP
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **PDR1VEN:** PDRV1 Gate Drive Enable bit
1 = ENABLED
0 = DISABLED

bit 6 **PDR2VEN:** PDRV2 Gate Drive Enable bit
1 = ENABLED
0 = DISABLED

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **IS1PUEN:** I_{SP} Weak Pull Up Enable bit
1 = I_{SP} weak pull up is enabled
0 = I_{SP} weak pull up is disabled

bit 2 **IS2PUEN:** I_{SP} Weak Pull Up Enable bit
1 = I_{SP} weak pull up is enabled
0 = I_{SP} weak pull up is disabled

bit 1 **LDO_LV:** LDO Voltage During Sleep
1 = LDO voltage is 5V
0 = LDO voltage is 3V

bit 0 **LDO_LP:** LDO low power mode Enable bit
1 = Low Power Mode Enabled
0 = Low Power Mode Disabled

6.10.1 ENABLE CONTROLS

Various analog circuit blocks can be enabled or disabled to reduce current consumption. The ABECON1/2 registers contain the bits that control certain analog circuit blocks. These registers also contain control bits to send analog and digital test signals to a GPIO pin.

The DIGOEN bit enables the output of the Digital Test MUX to be connected to the GPA3 pin. The DCHSEL0 and DCHSEL1 bits select the digital channels for these signals. Refer to Figure 8-1 for details about the configuration of the digital circuitry test MUX.

The ANAOEN bit enables the Analog Mux to be connected to the GPA0 pin. When ANAEON is active, the channel select line controls the analog signals to GPA0.

REGISTER 6-16: ABECON1: ANALOG BLOCK CONTROL REGISTER FOR PWM CHANNEL #1 (ADDRESS 10Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DIGOEN	GCTRL	DCHSEL1	DCHSEL0	DRUVSEL	EA1DIS1	EA2DIS1	ANAOEN
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **DIGOEN:** DIG Test MUX to GPA3 connection control
1 = Digital Test MUX output is connected to external pin GPA3
0 = Digital Test MUX output is not connected to external pin GPA3

bit 6 **GCTRL:** Auxiliary AmplifierGain Control Bit
1 = Gain is set to 2 (6 dB)
0 = Gain is set to 1 (0 dB)

bit 5-4 **DCHSEL<1:0>**: Digital MUX selection bits (Refer to Figure 8-1 for details)

bit 3 **DRUVSEL**: Selects Gate Drive Undervoltage Lockout level
1 = Gate Drive UVLO set to 5.4V
0 = Gate Drive UVLO set to 2.7V

bit 2 **EA1DIS1:** Current Loop Error Amplifier Disable bit (PWM #1)
1 = Disables the Current Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 1 **EA2DIS1:** Voltage Loop Error Amplifier Disable bit (PWM #1)
1 = Disables the Voltage Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 0 **ANAOEN:** Analog Mux Output Control bit
1 = Analog Mux output is connected to external pin GPA0
0 = Analog Mux output is not connected to external pin GPA0

REGISTER 6-17: ABECON2: ANALOG BLOCK CONTROL REGISTER FOR PWM CHANNEL #2 (ADDRESS 10Dh)

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	DSEL2	DSEL1	DSEL0	—	EA1DIS2	EA2DIS2	—
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **DSEL<2:0>** :MUX selection bits (Refer to Figure 8-1 for details)

bit 3 **Unimplemented:** Read as '0'

bit 2 **EA1DIS2:** Current Loop Error Amplifier Disable bit (PWM #2)

1 = Disables the Current Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 1 **EA2DIS2:** Voltage Loop Error Amplifier Disable bit (PWM #2)

1 = Disables the Voltage Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 0 **Unimplemented:** Read as '0'

NOTES:

7.0 TYPICAL PERFORMANCE CURVES

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

Note: Unless otherwise indicated, V_{IN} = 12V, F_{SW} = 150 kHz, T_A = +25°C.

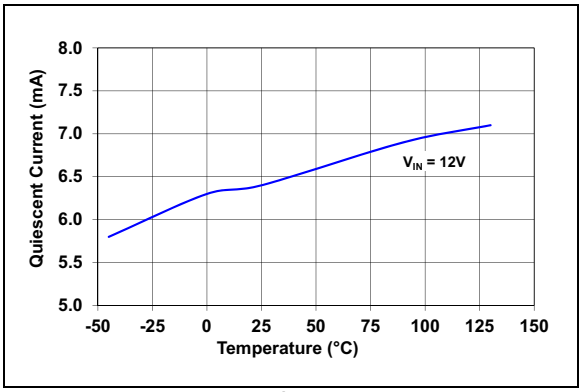


FIGURE 7-1: *Quiescent Current vs. Temperature.*

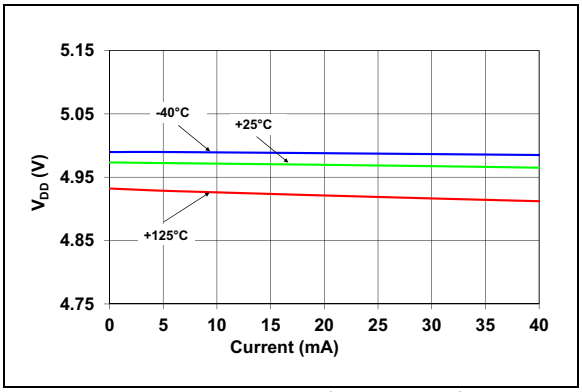


FIGURE 7-4: *Internal LDO Load Regulation.*

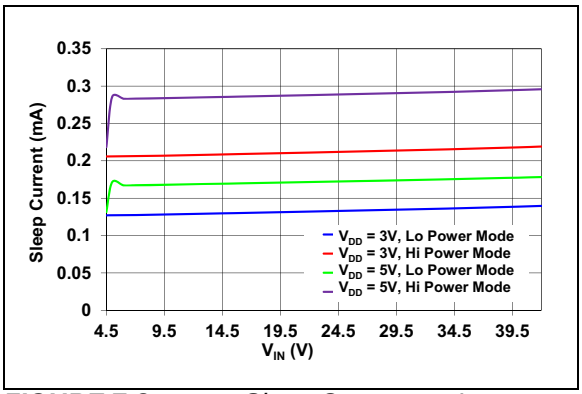


FIGURE 7-2: *Sleep Current vs. Input Voltage.*

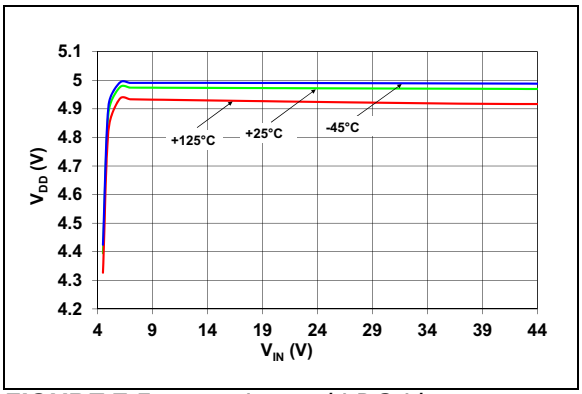


FIGURE 7-5: *Internal LDO Line Regulation.*

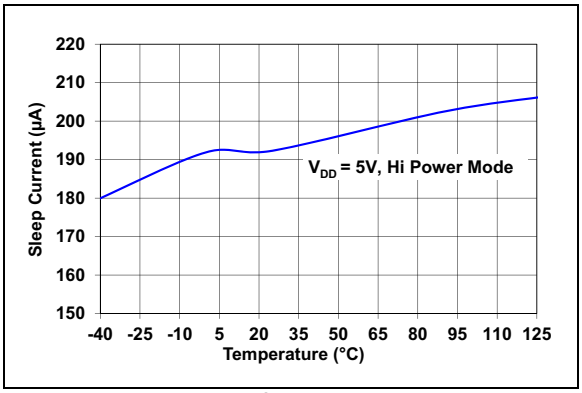


FIGURE 7-3: *Sleep Current vs. Temperature.*

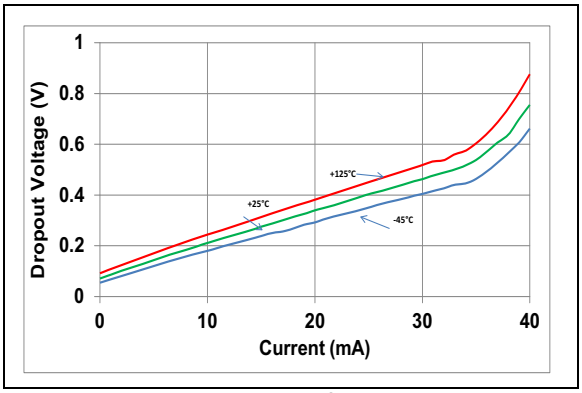


FIGURE 7-6: *Internal LDO Dropout Voltage vs. Load Current.*

Note: Unless otherwise indicated, V_{IN} = 12V, F_{SW} = 150 kHz, T_A = +25°C.

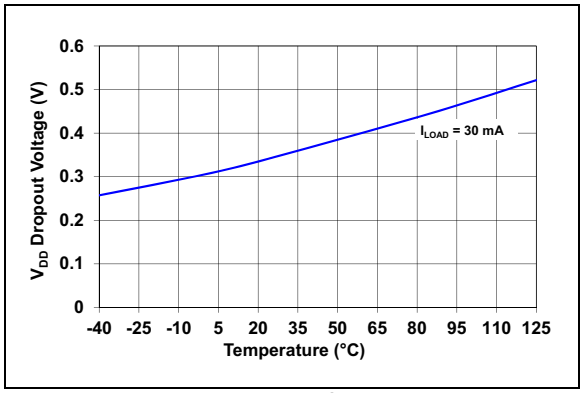


FIGURE 7-7: *Internal LDO Dropout Voltage vs. Temperature.*

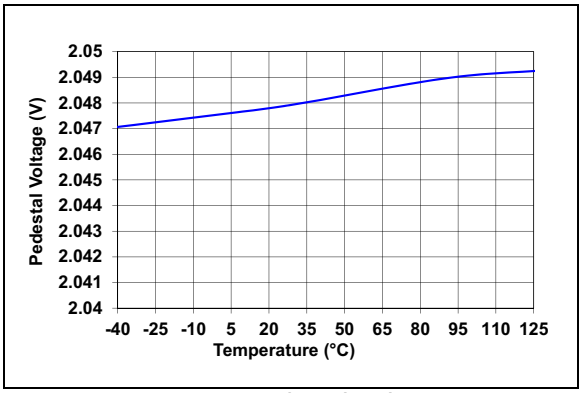


FIGURE 7-10: *Pedestal Voltage vs. Temperature.*

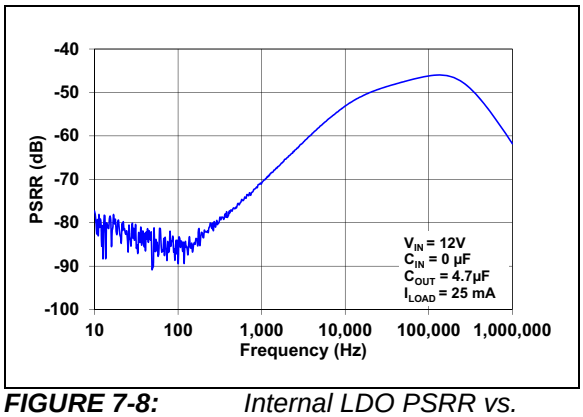


FIGURE 7-8: *Internal LDO PSRR vs. Frequency.*

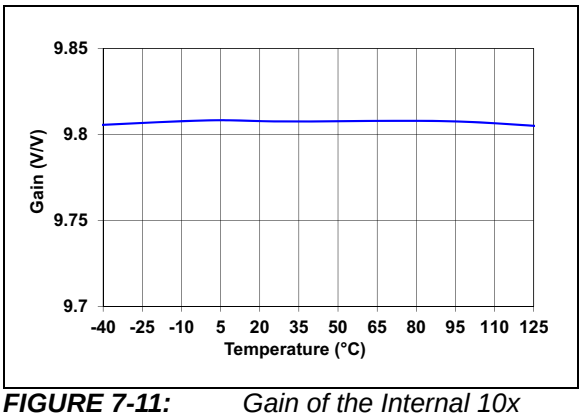


FIGURE 7-11: *Gain of the Internal 10x Differential Amplifier vs. Temperature.*

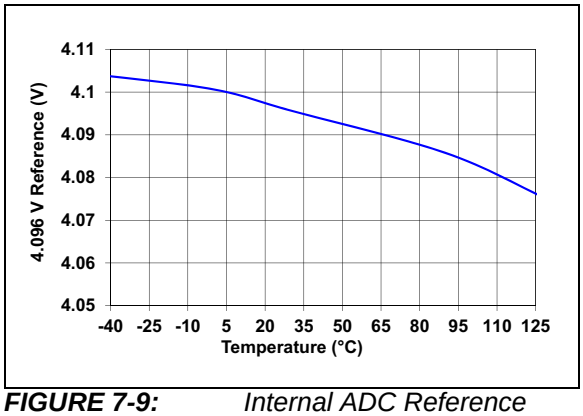


FIGURE 7-9: *Internal ADC Reference Voltage vs. Temperature.*

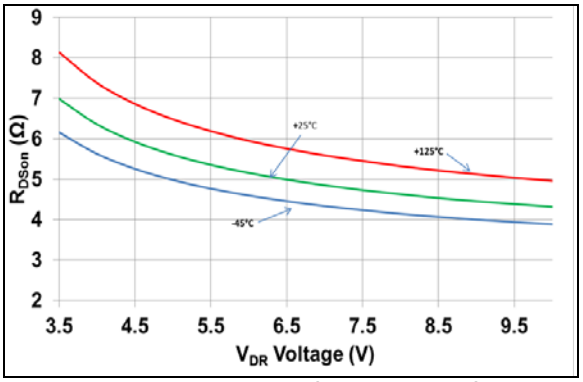


FIGURE 7-12: *Sourcing Output Driver R_{DSon} vs. V_{DR}.*

Note: Unless otherwise indicated, V_{IN} = 12V, F_{SW} = 150 kHz, T_A = +25°C.

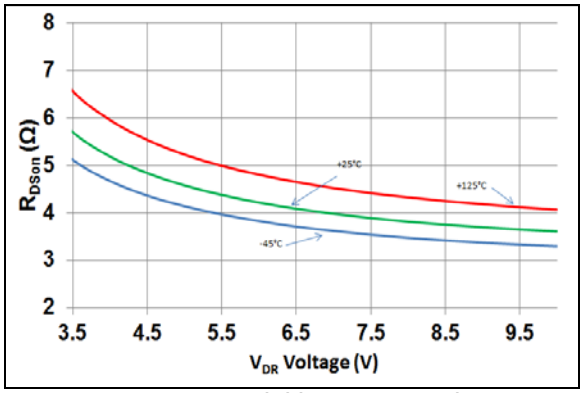


FIGURE 7-13: *Sinking Output Driver R_{DSon} vs. V_{DR}.*

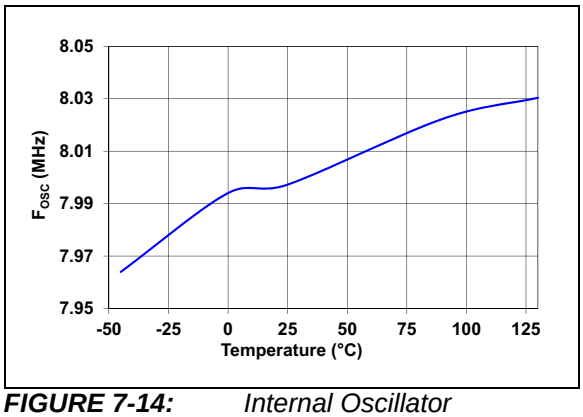


FIGURE 7-14: *Internal Oscillator Frequency vs. Temperature.*

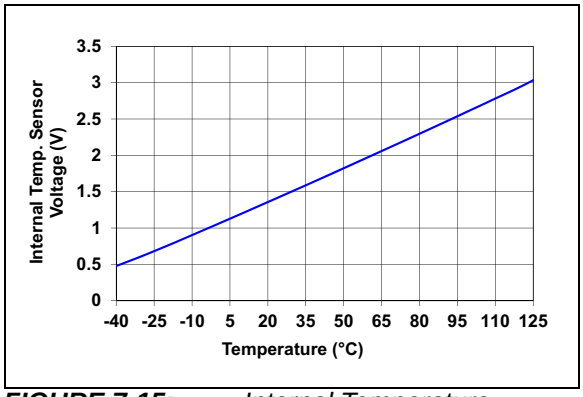


FIGURE 7-15: *Internal Temperature Sensor Voltage vs. Temperature.*

NOTES:

8.0 SYSTEM BENCH TESTING

To allow for easier system design and bench testing, the MCP19214/5 devices feature two multiplexers used to output various internal analog and digital signals.

8.0.1 THE ANALOG TEST MUX

These signals can be measured on the GPA0/AN0/TS_OUT1 pin through a unity gain buffer. The ANAOEN bit from the ABECON1 register controls the functionality of GPA0/AN0/TS_OUT1 pin.

The analog signals MUX is controlled by the CHS<4:0> bits of the ADCON0 register. Refer to Register 8-2 for details.

8.0.2 THE DIGITAL TEST MUX

These signals can be measured on the GPA3/AN3/TS_OUT2 pin. The DIGOEN bit from the ABECON1 register controls the functionality of GPA3/AN3/TS_OUT2 pin. The digital signals MUX is controlled by the DCHSEL<1:0> bits of the ABECON1 register and the DSEL<2:0> bits of the ABECON2 register. Refer to Register 8-1 and Register 8-3 for details. Figure 8-1 presents the internal diagram of the digital signals MUX.

REGISTER 8-1: ABECON1: ANALOG BLOCK CONTROL REGISTER FOR PWM CHANNEL #1 (ADDRESS 10Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DIGOEN	GCTRL	DCHSEL1	DCHSEL0	DRUVSEL	EA1DIS1	EA2DIS1	ANAOEN
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **DIGOEN:** DIG Test MUX to GPA3 connection control
1 = Digital Test MUX output is connected to external pin GPA3
0 = Digital Test MUX output is not connected to external pin GPA3

bit 6 **GCTRL:** Auxiliary AmplifierGain Control Bit
1 = Gain is set to 2 (6 dB)
0 = Gain is set to 1 (0 dB)

bit 5-4 **DCHSEL<1:0>**: Digital MUX selection bits (Refer to Figure 8-1 for details)

bit 3 **DRUVSEL**: Selects Gate Drive Undervoltage Lockout level
1 = Gate Drive UVLO set to 5.4V
0 = Gate Drive UVLO set to 2.7V

bit 2 **EA1DIS1:** Current Loop Error Amplifier Disable bit (PWM #1)
1 = Disables the Current Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 1 **EA2DIS1:** Voltage Loop Error Amplifier Disable bit (PWM #1)
1 = Disables the Voltage Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 0 **ANAOEN:** Analog Mux Output Control bit
1 = Analog Mux output is connected to external pin GPA0
0 = Analog Mux output is not connected to external pin GPA0

REGISTER 8-2: ADCON0: ANALOG-TO-DIGITAL CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHS5	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-2 **CHS<5:0>:** Analog Channel Select bits

000000 = PWM #1 EA1 (Current loop) reference voltage
000001 = PWM #1 EA2 (Voltage loop) reference voltage
000010 = PWM #1 Error Amplifier output voltage
000011 = PWM #1 VFB1 pin voltage
000100 = PWM #1 Slope Compensation reference voltage
000101 = PWM #1 IP1 signal offset reference voltage
000110 = PWM #1 PWM Comparator negative input
000111 = PWM #1 PWM Comparator positive input
001000 = PWM #2 EA1 (Current loop) reference voltage
001001 = PWM #2 EA2 (Voltage loop) reference voltage
001010 = PWM #2 Error Amplifier output voltage
001011 = PWM #2 VFB2 pin voltage
001100 = PWM #2 Slope Compensation reference voltage
001101 = PWM #2 IP2 signal offset reference voltage
001110 = PWM #2 PWM Comparator negative input
001111 = PWM #2 PWM Comparator positive input
010000 = PWM #1 A1 Current Sense Amplifier output
010001 = 1024 mV reference adjust
010010 = PWM #2 A1 Current Sense Amplifier output
010011 = Internal V_{DD} for analog circuitry
010100 = Internal V_{DD} for digital circuitry
010101 = 4096 mV Reference Voltage
010110 = 2048 mV Reference Voltage
010111 = PWM #2 ISN2 pin voltage
011000 = 1024 mV Reference Voltage
011001 = Bandgap Reference Voltage
011010 = V_{IN}/n voltage
011011 = V_{IN} UVLO Threshold
011100 = V_{IN} OVLO Threshold
011101 = V_{DD} UVLO voltage
011110 = V_{DR}/n (MOSFET drivers supply voltage)
011111 = TEMP_SNS temperature sensor voltage measurement
100000 = Internal GND node
111000 = AN0 analog input
111001 = AN1 analog input
111010 = AN2 analog input
111011 = AN3 analog input
111100 = AN4 analog input
111101 = AN5 analog input
111110 = AN6 analog input
111111 = AN7 analog input

REGISTER 8-2: ADCON0: ANALOG-TO-DIGITAL CONTROL REGISTER (CONTINUED)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Conversion Status bit
1 = A/D converter module is operating
0 = A/D converter is shut off and consumes no operating current

REGISTER 8-3: ABECON2: ANALOG BLOCK CONTROL REGISTER FOR PWM CHANNEL #2 (ADDRESS 10Dh)

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	DSEL2	DSEL1	DSEL0	—	EA1DIS2	EA2DIS2	—
bit7							bit0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **DSEL<2:0>**: Digital MUX selection bits Refer to Figure 8-1 for details

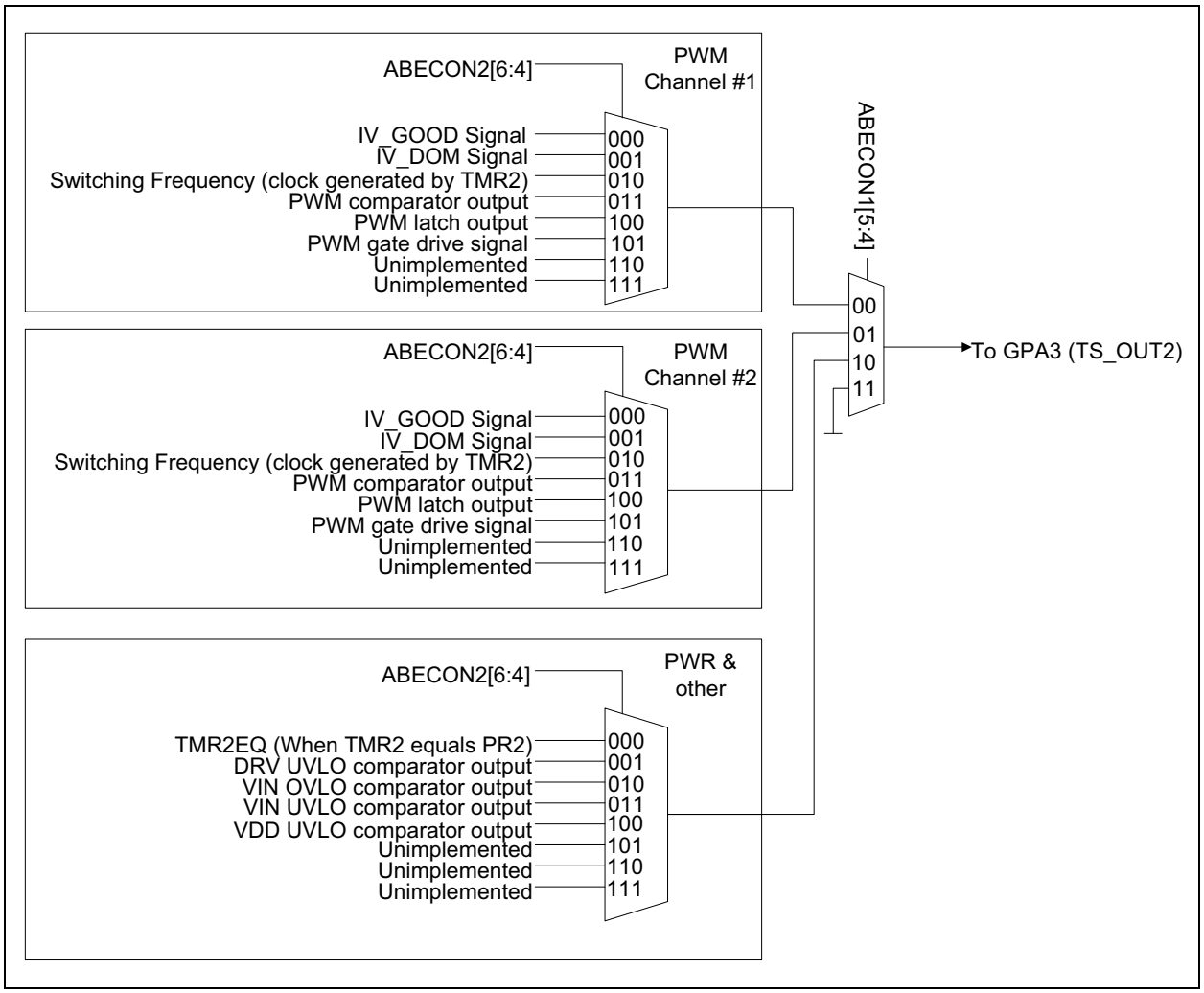
bit 3 **Unimplemented:** Read as 0

bit 2 **EA1DIS2:** Current Loop Error Amplifier Disable bit (PWM #2)
1 = Disables the Current Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 1 **EA2DIS2:** Voltage Loop Error Amplifier Disable bit (PWM #2)
1 = Disables the Voltage Loop Error Amplifier (Output is clamped to V_{DD})
0 = Enables the Error Amplifier (Normal operation)

bit 0 **Unimplemented:** Read as '0'

FIGURE 8-1: DIGITAL SIGNALS TEST MUX



9.0 DEVICE CALIBRATION

Read-only memory locations 2080h through 208Fh contain factory calibration data. Refer to **Section 17.0 "Flash Program Memory Control"** for information on how to read from these memory locations.

9.1 Calibration Word 1

Calibration Word 1 at memory location 2080h contains the calibration bits for the current sense differential amplifier, voltage loop EA and current loop EA of PWM channel #1.

The DCSCAL<5:0> bits set the offset calibration for the current sense differential amplifier (10X) of the first PWM channel. The IGMCAL<3:0> bits trim the transconductance of the current loop EA of PWM channel #1. The VGMCAL<3:0> bits trim the transconductance of the voltage loop EA of PWM channel #1.

Firmware must load these values into the appropriate registers (DCSCAL1 and GMCAL1 registers).

REGISTER 9-1: CONFIG: CALIBRATION WORD 1 (ADDRESS 2080H)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DCSCAL5	DCSCAL4	DCSCAL3	DCSCAL2	DCSCAL1	DCSCAL0
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
IGMCAL3	IGMCAL2	IGMCAL1	IGMCAL0	VGMCAL3	VGMCAL2	VGMCAL1	VGMCAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13-8 **DCSCAL<5:0>:** Differential Current Sense Amplifier calibration for PWM channel #1.
`111111` = Largest negative offset calibration
.
.
.
`100000` = Mid scale, no offset applied or
`000000` = Mid scale, no offset applied
.
.
.
`011111` = Largest positive offset calibration

bit 7-4 **IGMCAL<3:0>:** Current Loop EA transconductance for PWM channel #1.
`1111` = Largest negative offset calibration
.
.
.
`1000` = Mid scale, no offset applied or
`0000` = Mid scale, no offset applied
.
.
.
`0111` = Largest positive offset calibration

REGISTER 9-1: CONFIG: CALIBRATION WORD 1 (ADDRESS 2080H) (CONTINUED)

bit 3-0 **VGMCAL<3:0>:** Voltage Loop EA transconductance for PWM channel #1.

1111 = Largest negative offset calibration
.
.
.
1000 = Mid scale, no offset applied or
0000 = Mid scale, no offset applied
.
.
.
0111 = Largest positive offset calibration

9.2 Calibration Word 2

Calibration Word 2 at memory location 2081h contains the calibration bits for the current sense differential amplifier, voltage loop EA and current loop EA of PWM channel #2.

The DCSCAL<5:0> bits set the offset calibration for the current sense differential amplifier (10X) of the second PWM channel.

The IGMCAL<3:0> bits trim the transconductance of the current loop EA of PWM channel #2.

The VGMCAL<3:0> bits trim the transconductance of the voltage loop EA of PWM channel #2.

Firmware must load these values into the appropriate registers (DCSCAL2 and GMCAL2 registers).

REGISTER 9-2: CONFIG: CALIBRATION WORD 2 (ADDRESS 2081H)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DCSCAL5	DCSCAL4	DCSCAL3	DCSCAL2	DCSCAL1	DCSCAL0
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
IGMCAL3	IGMCAL2	IGMCAL1	IGMCAL0	VGMCAL3	VGMCAL2	VGMCAL1	VGMCAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-8 **DCSCAL<5:0>:** Differential Current Sense Amplifier offset calibration for PWM channel #2.

`111111` = Largest negative offset calibration
.
.
.
`100000` = Mid scale, no offset applied or
`000000` = Mid scale, no offset applied
.
.
.
`011111` = Largest positive offset calibration

bit 7-4 **IGMCAL<3:0>:** Current Loop EA offset for PWM channel #2.

`1111` = Largest negative offset calibration
.
.
.
`1000` = Mid scale, no offset applied or
`0000` = Mid scale, no offset applied
.
.
.
`0111` = Largest positive offset calibration

REGISTER 9-2: CONFIG: CALIBRATION WORD 2 (ADDRESS 2081H) (CONTINUED)

bit 3-0 **VGMCAL<3:0>:** Voltage Loop EA offset for PWM channel #2.

1111 = Largest negative offset calibration
.
.
.
1000 = Mid scale, no offset applied or
0000 = Mid scale, no offset applied
.
.
.
0111 = Largest positive offset calibration

9.3 Calibration Word 3

Calibration Word 3 is at memory location 2082h. It contains the calibration bits for the 4.096V reference generator, the pedestal voltage generator and for the Bandgap reference voltage generator.

The VR4VT<4:0> bits trim the 4.096V reference generator. Firmware must load these bits into VRCAL register.

The DACT<1:0> bits trim the DACs input current. Firmware must load these bits into DACBGRCAL register.

The BGRT<3:0> bits trim the bandgap voltage generator. Firmware must load these bits into DACBGRCAL register.

REGISTER 9-3: CONFIG: CALIBRATION WORD 3 (ADDRESS 2082H)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
VR4VT5	VR4VT4	VR4VT3	VR4VT2	VR4VT1	VR4VT0
bit 13					bit 8

U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	DACT1	DACT0	BGRT3	BGRT2	BGRT1	BGRT0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-8 **VR4VT<5:0>:** The 4.096V reference voltage calibration bits.

bit 7-6 **Unimplemented:** Read as ‘1’.

bit 5-4 **DACT<1:0>:** Calibration bits for the input current of the DACs.

bit 3-0 **BGRT<3:0>:** Band Gap Reference Voltage Generator calibration bits.

9.4 Calibration Word 4

Calibration Word 4 is at memory location 2083h. It contains the calibration bits for the pedestal voltage, offset voltage of the 2X differential amplifier and the trim bits for the over temperature set-point.

The PDST<5:0> bits trim the pedestal voltage. Firmware must load these bits into PDSCAL register.

The ADBOT<3:0> bits trim the offset of the 1/2X programmable gain differential amplifier. Firmware must load these bits into ADBT register.

The TTA<3:0> bits trim the over temperature set-point. Firmware must load these bits into TTCAL register.

REGISTER 9-4: CONFIG: CALIBRATION WORD 4 (ADDRESS 2083H)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
PDST5	PDST4	PDST3	PDST2	PDST1	PDST0
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
ADBOT3	ADBOT2	ADBOT1	ADBOT0	TTA3	TTA2	TTA1	TTA0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-8 **PDST<5:0>:** Pedestal voltage trim bits.

`111111` = Largest negative offset calibration
.
.
.
`100000` = Mid scale, no offset applied or
`000000` = Mid scale, no offset applied
.
.
.
`011111` = Largest positive offset calibration

bit 7-4 **ADBOT<3:0>:** 2X Differential Amplifier offset calibration bits.

bit 3-0 **TTA<3:0>:** Over temperature threshold calibration bits.

9.5 Calibration Word 5

Calibration Word 5 is at memory location 2084h. It contains the ADC reading of the TEMP_ANA input (ADC channel 0x0dh) when the diode temperature is at 30°C.

This is the actual 10-bit reading. Since the temperature coefficient is 16 mV/°C, the value stored at 2084h can be used to calibrate the ADC reading at any temperature.

REGISTER 9-5: CONFIG: CALIBRATION WORD 5 (ADDRESS 2084H)

U-0	U-0	U-0	U-0	R/P-1	R/P-1
—	—	—	—	TANA9	TANA8
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
TANA7	TANA6	TANA5	TANA4	TANA3	TANA2	TANA1	TANA0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-10 **Unimplemented:** Read as ‘1’.

bit 9-0 **TANA<9:0>:** ADC TEMP_ANA reading at 30°C calibration bits

9.6 Calibration Word 6

Calibration Word 6 is at memory location 2085h. The FCAL<6:0> bits set the internal oscillator calibration. Firmware must load these bits into OSCCAL register

REGISTER 9-6: CONFIG: CALIBRATION WORD 6 (ADDRESS 2085H)

U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—
bit 13					bit 8

U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13-7 **Unimplemented:** Read as '1'.

bit 6-0 **FCAL<6:0>:** Internal oscillator calibration bits.

0111111 = Maximum frequency
.
.
.
0000001
0000000 = Center frequency. Oscillator is running at the calibrated frequency
1111111
.
.
.
1000000 = Minimum frequency

9.7 Calibration Word 7

Calibration Word 7 is at memory location 2086h. The EACAL<7:4> bits trim the offset of the voltage loop error amplifier of PWM channel #2. The EACAL<3:0> bits trim the offset of the current loop error amplifier of PWM channel #2.

Firmware must load these bits into EACAL2 register.

REGISTER 9-7: CONFIG: CALIBRATION WORD 7 (ADDRESS 2086H)

U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
EACAL7	EACAL6	EACAL5	EACAL4	EACAL3	EACAL2	EACAL1	EACAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13-8 **Unimplemented:** Read as '1'.

bit 7-4 **EACAL<7:4>:** Trimming bits for the current loop error amplifier of PWM channel #2.

0111 = Maximum offset
.
.
.
0001
0000 = Center offset
1111
.
.
.
1000 = Minimum offset

bit 3-0 **EACAL<3:0>:** Trimming bits for the voltage loop error amplifier of PWM channel #2.

0111 = Maximum offset
.
.
.
0001
0000 = Center offset
1111
.
.
.
1000 = Minimum offset

9.8 Calibration Word 8

Calibration Word 8 is at memory location 2087h. This calibration register is reserved for future use.

REGISTER 9-8: CONFIG: CALIBRATION WORD 8 (ADDRESS 2087H)

U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-8 **Unimplemented:** Read as ‘1’.

bit 7-0 **CAL<7:0>:** Spare calibration bits.

9.9 Calibration Word 9

Calibration Word 9 is at memory location 2088h. The DACCAL<7:4> bits trim the DAC's input current of the current loop of PWM channel #2. The DACCAL<3:0> bits trim the DAC's input current of the voltage loop of PWM channel #2.

Firmware must load these bits into DACCAL2 register.

REGISTER 9-9: CONFIG: CALIBRATION WORD 9 (ADDRESS 2088H)

U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DACCAL7	DACCAL6	DACCAL5	DACCAL4	DACCAL3	DAC1CAL2	DAC1CAL1	DAC1CAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13-8 **Unimplemented:** Read as '1'.

bit 7-4 **DACCAL<7:4>:** Trimming bits for the current loop DAC of PWM channel #2.

0111 = Maximum current offset
.
.
.
0001
0000 = Center current offset
1111
.
.
.
1000 = Minimum current offset

bit 3-0 **DACCAL<3:0>:** Trimming bits for the voltage loop DAC of PWM channel #2.

0111 = Maximum current offset
.
.
.
0001
0000 = Center current offset
1111
.
.
.
1000 = Minimum current offset

9.10 Calibration Word 10

Calibration Word 10 is at memory location 2089h. The DACCAL<7:4> bits trim the DAC's input current of the current loop of PWM channel #1. The DACCAL<3:0> bits trim the DAC's input current of the voltage loop of PWM channel #1.

Firmware must load these bits into DACCAL1 register.

REGISTER 9-10: CONFIG: CALIBRATION WORD 10 (ADDRESS 2089H)

U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DACCAL7	DACCAL6	DACCAL5	DACCAL4	DACCAL3	DACCAL2	DACCAL1	DACCAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13-8 **Unimplemented:** Read as '1'.

bit 7-4 **DACCAL<7:4>:** Trimming bits for the current loop DAC of PWM channel #1

0111 = Maximum current offset
.
.
.
0001
0000 = Center current offset
1111
.
.
.
1000 = Minimum current offset

bit 3-0 **DACCAL<3:0>:** Trimming bits for the voltage loop DAC of PWM channel #1.

0111 = Maximum current offset
.
.
.
0001
0000 = Center current offset
1111
.
.
.
1000 = Minimum current offset

9.11 Calibration Word 11

Calibration Word 11 is at memory location 208Ah. The EACAL<7:4> bits trim the offset of the voltage loop error amplifier of PWM channel #1. The EACAL<3:0> bits trim the offset of the current loop error amplifier of PWM channel #1. Firmware must load these bits into EACAL1 register.

REGISTER 9-11: CONFIG: CALIBRATION WORD 11 (ADDRESS 208AH)

U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
EACAL7	EACAL6	EACAL5	EACAL4	EACAL3	EACAL2	EACAL1	EACAL0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-8 **Unimplemented:** Read as ‘1’.

bit 7-4 **EACAL<7:4>:** Trimming bits for the current loop error amplifier of PWM channel #1

0111 = Maximum offset
.
.
.
0001
0000 = Center offset
1111
.
.
.
1000 = Minimum offset

bit 3-0 **EACAL<3:0>:** Trimming bits for the voltage loop error amplifier of PWM channel #1

0111 = Maximum offset
.
.
.
0001
0000 = Center offset
1111
.
.
.
1000 = Minimum offset

9.12 Characterization Word 1

Characterization Word 1 is at memory location 208Bh. The GAIN<9:0> bits contain the measured voltage gain of the 10x differential amplifier of PWM channel #1. User can use this value to determine the real gain of the differential amplifier. The gain is calculated with Equation 9-1:

EQUATION 9-1:

$$GAIN = \frac{CONFIG}{1024}$$

REGISTER 9-12: CONFIG: CHARACTERIZATION WORD 1 (ADDRESS 208BH)

U-0	U-0	U-0	U-0	R/P-1	R/P-1
—	—	—	—	GAIN9	GAIN8
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
GAIN7	GAIN6	GAIN5	GAIN4	GAIN3	GAIN2	GAIN1	GAIN0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '1'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13-10 **Unimplemented:** Read as '1'.

bit 9-0 **GAIN<9:0>:** Measured voltage gain of the 10x differential amplifier of PWM channel #1

9.13 Characterization Word 2

Characterization Word 2 is at memory location 208Ch. The GAIN<9:0> bits contains the measured voltage gain of the 10x differential amplifier of PWM channel #2. User can use this value to determine the real gain of the differential amplifier. The gain is calculated with equation:

EQUATION 9-2:

$$GAIN = \frac{CONFIG}{1024}$$

REGISTER 9-13: CONFIG: CHARACTERIZATION WORD 2 (ADDRESS 208CH)

U-0	U-0	U-0	U-0	R/P-1	R/P-1
—	—	—	—	GAIN9	GAIN8
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
GAIN7	GAIN6	GAIN5	GAIN4	GAIN3	GAIN2	GAIN1	GAIN0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘1’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 13-10 **Unimplemented:** Read as ‘1’.

bit 9-0 **GAIN<9:0>:** Measured voltage gain of the 10X differential amplifier of PWM channel #2

NOTES:

10.0 MEMORY ORGANIZATION

There are two types of memory in the MCP19214/5:

- Program Memory
- Data Memory
  - Special Function Registers (SFRs)
  - General-Purpose RAM

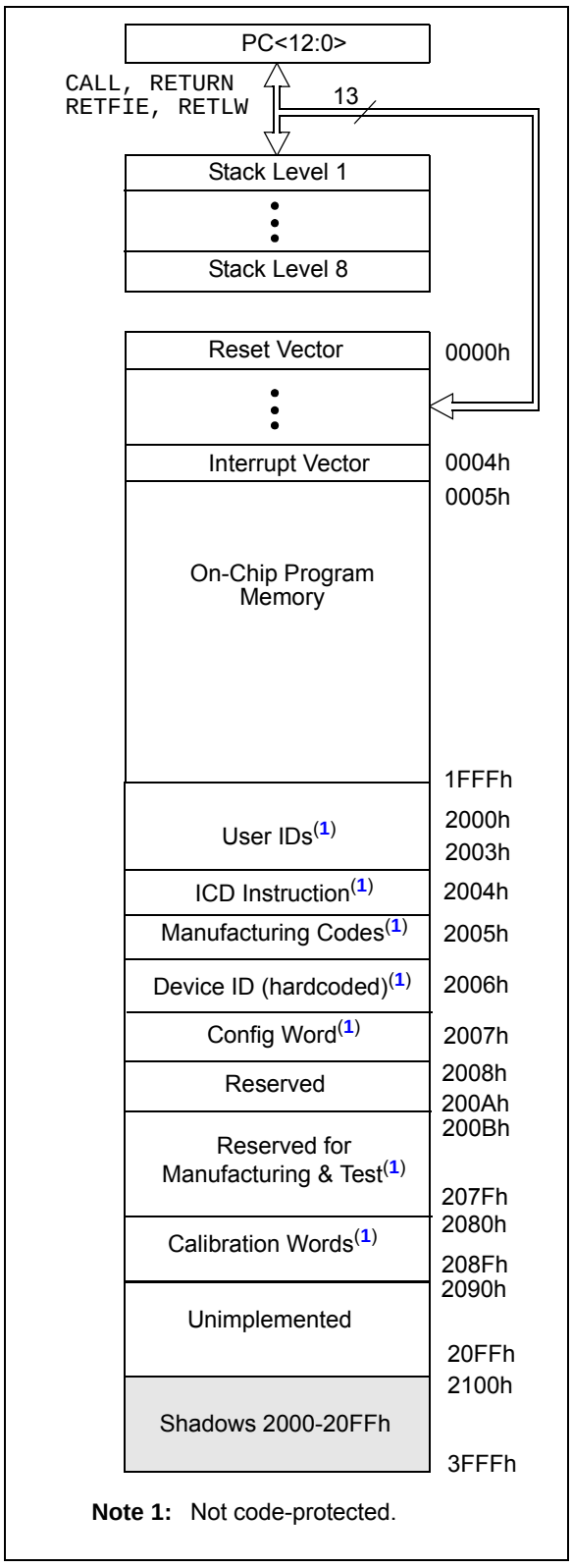
10.1 Program Memory Organization

The MCP19214/5 devices have a 13-bit program counter capable of addressing an 8192 x 14 program memory space.

The MCP19214/5 are 8K word devices and the address locations range is 0000h-1FFFh.

The Reset vector is at 0000h and the interrupt vector is at 0004h (refer to Figure 10-1). The width of the program memory bus (instruction word) is 14 bits. Since all instructions are a single word, the MCP19214/5 devices have space for 8192 instructions.

FIGURE 10-1: PROGRAM MEMORY MAP AND STACK FOR MCP19214/5 DEVICES



Note 1: Not code-protected.

10.1.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory:

- using tables of RETLW instructions
- setting a Files Select register (FSR) to point to the program memory.

10.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to the tables of constants. The recommended way to create such tables is shown in Example 10-1.

EXAMPLE 10-1: RETLW INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available, so the older table-read method must be used.

10.2 Data Memory Organization

The data memory (refer to Figure 10-1) is partitioned into four banks, which contain the General Purpose registers (GPR) and the Special Function registers (SFR). The Special Function registers are located in the first 32 locations of each bank. Register locations 20h-7Fh in Bank 0, A0h-EFh in Bank 1, and 120h-16Fh in Bank 2 are General Purpose registers, implemented as static RAM. All other RAM is unimplemented and returns '0' when read. The RP<1:0> bits in the STATUS register are the bank select bits.

EXAMPLE 10-2: BANK SELECT

RP1	RP0	
0	0	→ Bank 0 is selected
0	1	→ Bank 1 is selected
1	0	→ Bank 2 is selected
1	1	→ Bank 3 is selected

To move values from one register to another register, the value must pass through the W register. This means that for all register-to-register moves, two instruction cycles are required.

The entire data memory can be accessed either directly or indirectly. Direct addressing may require the use of the RP<1:0> bits. Indirect addressing requires the use of the FSR. Indirect addressing uses the Indirect Register Pointer (IRP) bit in the STATUS register for access to the Bank0/Bank1 or the Bank2/Bank3 areas of data memory.

10.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 64 x 8 in the MCP19214/5. Each register is accessed, either directly or indirectly, through the FSR (refer to **Section 10.5 "Indirect Addressing, INDF and FSR Registers"**).

10.2.2 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers can be addressed from any bank. These registers are listed below in Table 10-1. For detailed information, refer to Table 10-2.

TABLE 10-1: CORE REGISTERS

Addresses	BANKx
x00h, x80h, x100h, or x180h	INDF
x02h, x82h, x102h, or x182h	PCL
x03h, x83h, x103h, or x183h	STATUS
x04h, x84h, x104h, or x184h	FSR
x0Ah, x8Ah, x10Ah, or x18Ah	PCLATH
x0Bh, x8Bh, x10Bh, or x18Bh	INTCON

10.2.2.1 STATUS Register

The STATUS register contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (RAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, the `write` to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits.

Note 1: The C and DC bits operate as $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bits, respectively, in subtraction.

REGISTER 10-1: STATUS: STATUS REGISTER

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC(1)	C(1)
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	x = Bit is unknown	'0' = Bit is cleared
'1' = Bit is set		

bit 7 **IRP:** Register Bank Select bit (used for Indirect addressing)
`1` = Bank 2 & 3 (100h - 1FFh)
`0` = Bank 0 & 1 (00h - FFh)

bit 6-5 **RP<1:0>:** Register Bank Select bits (used for Direct addressing)
`00` = Bank 0 (00h - 7Fh)
`01` = Bank 1 (80h - FFh)
`10` = Bank 2 (100h - 17Fh)
`11` = Bank 3 (180h - 1FFh)

bit 4 **$\overline{TO}$:** Time-Out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time out occurred

Note 1: For $\overline{Borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high-order or low-order bit in the source register.

REGISTER 10-1: STATUS: STATUS REGISTER (CONTINUED)

bit 3 **$\overline{PD}$:** Power-Down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{\text{Digit Borrow}}$ bit(1) (ADDWF, ADDLW, SUBLW, SUBWF instructions)
1 = A carry-out from the 4th low-order bit of the result occurred
0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit(1) (ADDWF, ADDLW, SUBLW, SUBWF instructions)(1)
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note 1: For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high-order or low-order bit in the source register.

10.2.3 SPECIAL FUNCTION REGISTERS

The Special Function registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (refer to Figure 10-2). These registers are static RAM.

The special registers can be classified into two sets: core and peripheral. The Special Function registers associated with the microcontroller core are described in this section. Those related to the operation of the peripheral features are described in the associated section for that peripheral feature.

10.3 DATA MEMORY

TABLE 10-2: MCP19214/5 DATA MEMORY MAP

	File Address		File Address		File Address		File Address		File Address
Indirect addr.(1)	00h	Indirect addr. (1)	80h	Indirect addr.(1)	100h	Indirect addr. (1)	180h	Indirect addr. (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h	FSR	184h
PORTGPA	05h	TRISGPA	85h	WPUGPA	105h	IOCA	185h	IOCA	185h
PORTGPB	06h	TRISGPB	86h	WPUGPB	106h	IOCB	186h	IOCB	186h
PIR1	07h	PIE1	87h	PE1	107h	ANSELA	187h	ANSELA	187h
PIR2	08h	PIE2	88h		108h	ANSELB	188h	ANSELB	188h
PIR3(4)	09h	PIE3(4)	89h		109h		189h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON	18Bh
TMR1L	0Ch	VINUVLO	8Ch	ABECON1	10Ch	PORTICD(2)	18Ch	PORTICD(2)	18Ch
TMR1H	0Dh	VINOVLO	8Dh	ABECON2	10Dh	TRISICD(2)	18Dh	TRISICD(2)	18Dh
T1CON	0Eh	VINCON	8Eh		10Eh	ICKBUG(2)	18Eh	ICKBUG(2)	18Eh
TMR2	0Fh	CREFCON1	8Fh		10Fh	BIGBUG(2)	18Fh	BIGBUG(2)	18Fh
T2CON	10h	CREFCON2	90h	SSPADD	110h	PMCON1	190h	PMCON1	190h
PR2	11h	VREFCON1	91h	SSPBUF	111h	PMCON2	191h	PMCON2	191h
PCON	12h	VREFCON2	92h	SSPCON1	112h	PMADRL	192h	PMADRL	192h
PWM2PHL	13h	CC1RL	93h	SSPCON2	113h	PMADRH	193h	PMADRH	193h
PWM2PHH	14h	CC1RH	94h	SSPCON3	114h	PMDATL	194h	PMDATL	194h
PWM2RL	15h	CC2RL	95h	SSPMSK	115h	PMDATH	195h	PMDATH	195h
PWM2RH	16h	CC2RH	96h	SSPSTAT	116h	GMCAL1	196h	EACAL2(3)	196h
PWM1RL	17h	CCDCON	97h	SSPADD2	117h	GMCAL2	197h	SPARECAL(3)	197h
PWM1RH	18h	VDDCON	98h	SSPMSK2	118h	DCSCAL1	198h	DACCAL1(3)	198h
	19h	LOOPCON1	99h		119h	DCSCAL2	199h	DACCAL2(3)	199h
	1Ah	LOOPCON2	9Ah		11Ah	ADBT	19Ah	EACAL1(3)	19Ah
OSCTUNE	1Bh	TTCAL	9Bh	SPBRG(4)	11Bh	DACBGRCAL	19Bh	DACBGRCAL	19Bh
ADRESL	1Ch	SLPCRCON1	9Ch	RCREG(4)	11Ch	PDSCAL	19Ch	PDSCAL	19Ch
ADRESH	1Dh	SLPCRCON2	9Dh	TXREG(4)	11Dh	VRCAL	19Dh	VRCAL	19Dh
ADCON0	1Eh	ICOACON	9Eh	TXSTA(4)	11Eh	OSCCAL	19Eh	OSCCAL	19Eh
ADCON1	1Fh	ICLEBCON	9Fh	RCSTA(4)	11Fh		19Fh		19Fh
General Purpose Register 96 Bytes	20h . . . 7Fh	General Purpose Register 80 Bytes	A0h . . . EFh	General Purpose Register 80 bytes	120h . . . 16F	General Purpose Register 80 bytes	1A0h . . . 1EF	General Purpose Register 80 bytes	1A0h . . . 1EF
		Accesses Bank 0	F0h FFh	Accesses Bank 0	170h 17Fh	Accesses Bank 0	1F0h 1FFh	Accesses Bank 0	1F0h 1FFh
Bank 0		Bank 1		Bank2		Bank3		Bank4	

Legend: ▢ Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

2: Only accessible when DBGEN = 0 and ICKBUG<INBUG> = 1.

3: Only accessible when CALSEL = 1.

4: Only implemented in MCP19215; read as '0' in MCP19214.

TABLE 10-3: MCP19214/5 SPECIAL REGISTERS SUMMARY BANK 0

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other resets(1)
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter's (PC) Least Significant byte								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTGPA	GPA7	GPA6	GPA5	—	GPA3	GPA2	GPA1	GPA0	xxx- xxxx	uuu- uuuu
06h	PORTGPB	GPB7(3)	GPB6(3)	GPB5	GPB4	—	—	GPB1(3)	GPB0	xxxx --xx	uuuu --uu
07h	PIR1	OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
08h	PIR2	IVGD1IF	—	IVGD2IF	—	VDDUVIF	DRUVIF	OVLOIF	UVLOIF	0000 0000	0000 0000
09h	PIR3(3)							RCIF	TXIF	---- --00	---- --00
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF(2)	0000 000x	0000 000u
0Ch	TMR1L	Holding register for the Least Significant byte of the 16-bit TMR1								xxxx xxxx	uuuu uuuu
0Dh	TMR1H	Holding register for the Most Significant byte of the 16-bit TMR1								xxxx xxxx	uuuu uuuu
0Eh	T1CON	—	—	T1CKPS1	T1CKPS0	—	—	TMR1CS	TMR1ON	--00 --00	--uu --uu
0Fh	TMR2	Timer2 Module Register								0000 0000	uuuu uuuu
10h	T2CON	—	—	—	—	—	TMR2ON	T2CKPS1	T2CKPS0	---- -000	---- -000
11h	PR2	Timer2 Module Period Register								1111 1111	1111 1111
12h	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	---- --uu
13h	PWM2PHL	PWM2 (SLAVE) Phase Shift Register								xxxx xxxx	uuuu uuuu
14h	PWM2PHH	PWM2 (SLAVE) Phase Shift Register								xxxx xxxx	uuuu uuuu
15h	PWM2RL	PWM2 Register Low Byte								xxxx xxxx	uuuu uuuu
16h	PWM2RH	PWM2 Register High Byte								xxxx xxxx	uuuu uuuu
17h	PWM1RL	PWM1 Register Low Byte								xxxx xxxx	uuuu uuuu
18h	PWM1RH	PWM1 Register High Byte								xxxx xxxx	uuuu uuuu
19h	—	Unimplemented								_	_
1Ah	—	Unimplemented								_	_
1Bh	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---0 0000
1Ch	ADRESL	Least significant 8 bits of the A/D result								xxxx xxxx	uuuu uuuu
1Dh	ADRESH	Most significant 2 bits of the A/D right shifted								0000 00xx	0000 00uu
1Eh	ADCON0	CHS5	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	0000 0000	0000 0000
1Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition shaded = unimplemented

Note 1: Other (non power-up) resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: $\overline{MCLR}$ and WDT reset does not affect the previous value data latch. The IOCF bit will be cleared upon reset but will set again if the mismatch exists.

3: Available only in MCP19215.

TABLE 10-4: MCP19214/5 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Values on all other resets(1)
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
81h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC) Least Significant byte								0000 0000	0000 0000
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISGPA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	1110 1111	1110 1111
86h	TRISGPB	TRISB7(3)	TRISB6(3)	TRISB5	TRISB4	—	—	TRISB1(3)	TRISB0	1111 --11	1111 --11
87h	PIE1	OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
88h	PIE2	IVGD1IE	—	IVGD2IE	—	VDDUVIE	DRUVIE	OVLOIE	UVLOIE	0-0- 0000	0-00 -000
89h	PIE3(3)	—	—	—	—	—	—	RCIE	TXIE	---- --00	---- --00
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF(2)	0000 000x	0000 000u
8Ch	VINUVLO	—	—	UVLO5	UVLO4	UVLO3	UVLO2	UVLO1	UVLO0	--xx xxxx	--uu uuuu
8Dh	VINOVLO	—	—	OVLO5	OVLO4	OVLO3	OVLO2	OVLO1	OVLO0	--xx xxxx	--uu uuuu
8Eh	VINCON	UVLOEN	UVLOOUT	UVLOINTP	UVLOINTN	OVLOEN	OVLOOUT	OVLOINTP	OVLOINTN	0x00 0x00	0u00 0u00
8Fh	CREFCON1	CREF7	CREF6	CREF5	CREF4	CREF3	CREF2	CREF1	CREF0	0000 0000	0000 0000
90h	CREFCON2	CREF7	CREF6	CREF5	CREF4	CREF3	CREF2	CREF1	CREF0	0000 0000	0000 0000
91h	VREFCON1	VREF7	VREF6	VREF5	VREF4	VREF3	VREF2	VREF1	VREF0	0000 0000	0000 0000
92h	VREFCON1	VREF7	VREF6	VREF5	VREF4	VREF3	VREF2	VREF1	VREF0	0000 0000	0000 0000
93h	CC1RL	Capture1/Compare1 Register1 x Low Byte (LSB)								xxxx xxxx	uuuu uuuu
94h	CC1RH	Capture1/Compare1 Register2 x High Byte (MSB)								xxxx xxxx	uuuu uuuu
95h	CC2RL	Capture2/Compare2 Register1 x Low Byte (LSB)								xxxx xxxx	uuuu uuuu
96h	CC2RH	Capture2/Compare2 Register2 x High Byte (MSB)								xxxx xxxx	uuuu uuuu
97h	CCDCON	CC2M<3:0>				CC1M<3:0>				0000 0000	0000 0000
98h	VDDCON	VDDUVEN	VDDUVOUT	VDDUVINTP	VDDUVINTN	—	—	VDDUV1	VDDUV0	0x00 0000	0x00 0000
99h	LOOPCON1	IVLRES	—	IV_GOOD	IVGDINTP	IVGDINTN	IV_DOM	—	—	10x0 0x00	10x0 0x00
9Ah	LOOPCON2	IVLRES	—	IV_GOOD	IVGDINTP	IVGDINTN	IV_DOM	—	—	10x0 0x00	10x0 0x00
9Bh	TTCAL	TSTOT	—	—	—	TTA3	TTA2	TTA1	TTA0	0000 0000	0000 0000
9Ch	SLPCRCON1	—	SLPBY	SLPS5	SLPS4	SLPS3	SLPS2	SLPS1	SLPS0	-xxx xxxx	-uuu uuuu
9Dh	SLPCRCON2	—	SLPBY	SLPS5	SLPS4	SLPS3	SLPS2	SLPS1	SLPS0	-xxx xxxx	-uuu uuuu
9Eh	ICOACON	IC1OAC3	IC1OAC2	IC1OAC1	IC1OAC0	IC2OAC3	IC2OAC2	IC2OAC1	IC2OAC0	xxxx xxxx	uuuu uuuu
9Fh	ICLEBCON	—	—	—	—	IC1LEBC1	IC1LEBC0	IC2LEBC1	IC2LEBC0	---- xxxx	---- uuuu

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The IOCF bit will be cleared upon reset but will set again if the mismatch exists.

3: Available only in MCP19215.

TABLE 10-5: MCP19214/5 SPECIAL REGISTERS SUMMARY BANK 2

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other resets(1)
Bank 2											
100h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
101h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
102h	PCL	Program Counter's (PC) Least Significant byte								0000 0000	0000 0000
103h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
104h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
105h	WPUGPA	WPUA7	—	WPUA5	—	WPUA3	WPUA2	WPUA1	WPUA0	1-1- 1111	u-u- uuuu
106h	WPUGPB	WPUB7(3)	WPUB6(3)	WPUB5	WPUB4	—	—	WPUB1(3)	WPUB0	1111 ---1	uuuu --uu
107h	PE1	PDRV1EN	PDRV2EN	—	—	IS1PUEN	IS2PUEN	LDO_LV	LDO_LP	00-- 1100	00-- 1100
108h	—	—	—	—	—	—	—	—	—	—	—
109h	—	—	—	—	—	—	—	—	—	—	—
10Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
10Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF(2)	0000 000x	0000 000u
10Ch	ABECON1	DIGOEN	GCTRL	DCHSEL1	DCHSEL0	DRUVSEL	EA1DIS1	EA2DIS1	ANAOEN	0000 0000	0000 0000
10Dh	ABECON2	—	DSEL2	DSEL1	DSEL0	—	EA1DIS2	EA2DIS2	—	-000 -00-	-000 -00-
10Eh	—	Unimplemented								—	—
10Fh	—	Unimplemented								—	—
110h	SSPADD	ADD<7:0>								0000 0000	0000 0000
111h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
112h	SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000	0000 0000
113h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
114h	SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000
115h	SSPMSK	MSK<7:0>								1111 1111	1111 1111
116h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	—	—
117h	SSPADD2	ADD2<7:0>								0000 0000	0000 0000
118h	SSPMSK2	MSK2<7:0>								1111 1111	1111 1111
119h	—	Unimplemented								—	—
11Ah	—	Unimplemented								—	—
11Bh	SPBRG(3)	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
11Ch	RCREG(3)	USART Receive Data Register								0000 0000	0000 0000
11Dh	TXREG(3)	USART Transmit Data Register								0000 0000	0000 0000
11Eh	TXSTA(3)	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0000	0000 0000
11Fh	RCSTA(3)	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 0000	0000 0000

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition shaded = unimplemented

Note 1: Other (non power-up) resets include MCLR Reset and Watchdog Timer Reset during normal operation.

2: MCLR and WDT reset does not affect the previous value data latch. The IOCF bit will be cleared upon reset but will set again if the mismatch exists.

3: Available only in MCP19215.

TABLE 10-6: MCP19214/5 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3 (PMCON1.CALSEL = 0)

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Values on all other resets(1)
Bank 3											
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
181h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h	PCL	Program Counter's (PC) Least Significant byte								0000 0000	0000 0000
183h	STATUS	IRP(1)	RP1(2)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
184h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	IOCA	IOCA7	IOCA6	IOCA5	—	IOCA3	IOCA2	IOCA1	IOCA0	000- 0000	000- 0000
186h	IOCB	IOCB7(3)	IOCB6(3)	IOCB5	IOCB4	—	—	IOCB1(3)	IOCB0	0000 --00	0000 --00
187h	ANSELA	—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0	---- 1111	---- 1111
188h	ANSELB	—	ANSB6(3)	ANSB5	ANSB4	—	—	ANSB1(3)	—	-111 --1-	--11 --1-
189h	—	Unimplemented								—	—
18Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
18Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF(4)	0000 000x	0000 000u
18Ch	PORTICD(5)	In-Circuit Debug Port Register								xxxx --xx	uuuu --uu
18Dh	TRISICD(5)	In-Circuit Debug TRIS Register								1111 0011	1111 0011
18Eh	ICKBUG(5)	In-Circuit Debug Register								0000 0000	000u uuuu
18Fh	BIGBUG(5)	In-Circuit Debug Breakpoint Register								0000 0000	uuuu uuuu
190h	PMCON1	—	CALSEL	—	—	—	WREN	WR	RD	-0-- -000	-0-- -000
191h	PMCON2	Program Memory Control Register 2 (not a physical register)								---- ----	---- ----
192h	PMADRL	PMADRL7	PMADRL6	PMADRL5	PMADRL4	PMADRL3	PMADRL2	PMADRL1	PMADRL0	0000 0000	0000 0000
193h	PMADRH	—	—	—	—	PMADRH3	PMADRH2	PMADRH1	PMADRH0	---- 0000	---- 0000
194h	PMDATL	PMDATL7	PMDATL6	PMDATL5	PMDATL4	PMDATL3	PMDATL2	PMDATL1	PMDATL0	0000 0000	0000 0000
195h	PMDATH	—	—	PMDATH5	PMDATH4	PMDATH3	PMDATH2	PMDATH1	PMDATH0	--00 0000	--00 0000
196h	GMCAL1	IGMCAL3	IGMCAL2	IGMCAL1	IGMCAL0	VGMCAL3	VGMCAL2	VGMCAL1	VGMCAL0	xxxx xxxx	uuuu uuuu
197h	GMCAL2	IGMCAL3	IGMCAL2	IGMCAL1	IGMCAL0	VGMCAL3	VGMCAL2	VGMCAL1	VGMCAL0	xxxx xxxx	uuuu uuuu
198h	DCSCAL1	—	DCSCAL6	DCSCAL5	DCSCAL4	DCSCAL3	DCSCAL2	DCSCAL1	DCSCAL0	-xxx xxxx	-uuu uuuu
199h	DCSCAL2	—	DCSCAL6	DCSCAL5	DCSCAL4	DCSCAL3	DCSCAL2	DCSCAL1	DCSCAL0	-xxx xxxx	-uuu uuuu
19Ah	ADBT	ADBOT3	ADBOT2	ADBOT1	ADBOT0	—	—	—	—	xxxx ----	uuuu ----
19Bh	DACBGRCAL	SPARE	—	DACT1	DACT0	BGRT3	BGRT2	BGRT1	BGRT0	x-xx xxxx	u-uu uuuu
19Ch	PDSCAL	—	—	PDST5	PDST4	PDST3	PDST2	PDST1	PDST0	--xx xxxx	--uu uuuu
19Dh	VRCAL	—	—	VR4VT4	VR4VT4	VR4VT3	VR4VT2	VR4VT1	VR4VT0	--xx xxxx	--uu uuuu
19Eh	OSCCAL	—	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	-xxx xxxx	-uuu uuuu
19Fh											

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: IRP & RP1 bits are reserved, always maintain these bits clear.

3: Available only in MCP19215.

4: $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The IOCF bit will be cleared upon reset but will set again if the mismatch exists.

5: Only accessible when DBGEN = 0 and ICKBUG<INBUG> = 1.

TABLE 10-7: MCP19214/5 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3 (PMCON1.CALSEL = 1)

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Values on all other resets(1)
Bank 3											
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
181h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
182h	PCL	Program Counter's (PC) Least Significant byte								0000 0000	0000 0000
183h	STATUS	IRP(1)	RP1(2)	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
184h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
185h	IOCA	IOCA7	IOCA6	IOCA5	—	IOCA3	IOCA2	IOCA1	IOCA0	000- 0000	000- 0000
186h	IOCB	IOCB7(3)	IOCB6(3)	IOCB5	IOCB4	—	—	IOCB1(3)	IOCB0	0000 --00	0000 --00
187h	ANSELA	—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0	---- 1111	---- 1111
188h	ANSELB	—	ANSB6(3)	ANSB5	ANSB4	—	—	ANSB1(3)	—	-111 --1-	--11 --1-
189h	—	Unimplemented								—	—
18Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
18Bh	INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF(4)	0000 000x	0000 000u
18Ch	PORTICD(5)	In-Circuit Debug Port Register								xxxx --xx	uuuu --uu
18Dh	TRISICD(5)	In-Circuit Debug TRIS Register								1111 0011	1111 0011
18Eh	ICKBUG(5)	In-Circuit Debug Register								0000 0000	000u uuuu
18Fh	BIGBUG(5)	In-Circuit Debug Breakpoint Register								0000 0000	uuuu uuuu
190h	PMCON1	—	CALSEL	—	—	—	WREN	WR	RD	-0-- -000	-0-- -000
191h	PMCON2	Program Memory Control Register 2 (not a physical register)								---- ----	---- ----
192h	PMADRL	PMADRL7	PMADRL6	PMADRL5	PMADRL4	PMADRL3	PMADRL2	PMADRL1	PMADRL0	0000 0000	0000 0000
193h	PMADRH	—	—	—	—	PMADRH3	PMADRH2	PMADRH1	PMADRH0	---- 0000	---- 0000
194h	PMDATL	PMDATL7	PMDATL6	PMDATL5	PMDATL4	PMDATL3	PMDATL2	PMDATL1	PMDATL0	0000 0000	0000 0000
195h	PMDATH	—	—	PMDATH5	PMDATH4	PMDATH3	PMDATH2	PMDATH1	PMDATH0	--00 0000	--00 0000
196h	EACAL2	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	xxxx xxxx	uuuu uuuu
197h	SPARECAL2	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	xxxx xxxx	uuuu uuuu
198h	DACCAL1	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL1	xxxx xxxx	uuuu uuuu
199h	DACCAL2	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL1	xxxx xxxx	uuuu uuuu
19Ah	EACAL1	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	xxxx xxxx	uuuu uuuu
19Bh	DACBGRCAL	—	—	DACT1	DACT0	BGRT3	BGRT2	BGRT1	BGRT0	--xx xxxx	u-uu uuuu
19Ch	PDSCAL	—	—	IVROT5	IVROT4	IVROT3	IVROT2	IVROT1	IVROT0	--xx xxxx	--uu uuuu
19Dh	VRCAL	—	—	VR4VT5	VR4VT4	VR4VT3	VR4VT2	VR4VT1	VR4VT0	--xx xxxx	--uu uuuu
19Eh	OSCCAL	—	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	-xxx xxxx	-uuu uuuu
19Fh											

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

2: IRP & RP1 bits are reserved, always maintain these bits clear.

3: Available only in MCP19215.

4: $\overline{MCLR}$ and WDT Reset does not affect the previous value data latch. The IOCF bit will be cleared upon reset but will set again if the mismatch exists.

5: Only accessible when $\overline{DBGEN}$ = 0 and ICKBUG<INBUG> = 1.

10.3.1 OPTION_REG REGISTER

The OPTION_REG register is a readable and writable register, which contains various control bits to configure:

- Timer0/WDT prescaler
- External GPA2/INT interrupt
- Timer0
- Weak pull-ups on PORTGPA and PORTGPB

Note: To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT by setting PSA bit to '1' in the OPTION_REG register. Refer to **Section 21.1.3 "Software Programmable Prescaler"**.

REGISTER 10-2: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	x = Bit is unknown	'0' = Bit is cleared
'1' = Bit is set		

bit 7 **$\overline{\text{RAPU}}$:** Port GPx Pull-Up Enable bit(1)
- 1 = Port GPx pull-ups are disabled
- 0 = Port GPx pull-ups are enabled

bit 6 **INTEDG:** Interrupt Edge Select bit
- 0 = Interrupt on rising edge of INT pin
- 1 = Interrupt on falling edge of INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
- 1 = Transition on T0CKI pin
- 0 = Internal instruction cycle clock

bit 4 **T0SE:** TMR0 Source Edge Select bit
- 1 = Increment on high-to-low transition on T0CKI pin
- 0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
- 1 = Prescaler is assigned to WDT
- 0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1: 2	1: 1
001	1: 4	1: 2
010	1: 8	1: 4
011	1: 16	1: 8
100	1: 32	1: 16
101	1: 64	1: 32
110	1: 128	1: 64
111	1: 256	1: 128

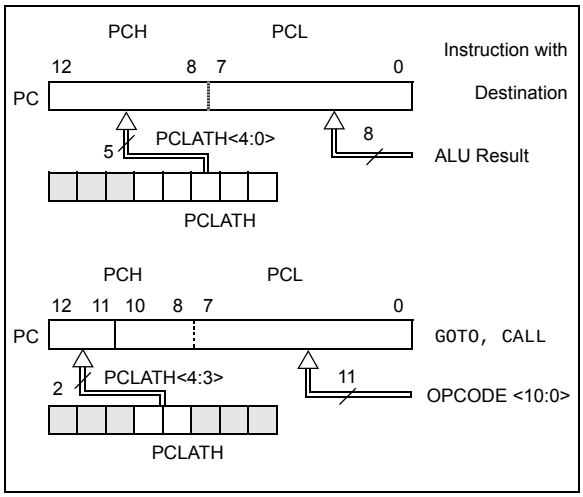
Note 1: Individual WPUx bit must also be enabled.

10.4 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any reset, the PC is cleared. Figure 10-2 shows the two situations for loading the PC:

- the upper example shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH)
- the lower example in Figure 10-2 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 10-2: PROGRAM COUNTER (PC) LOADING IN DIFFERENT SITUATIONS



10.4.1 MODIFYING PCL REGISTER

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 5 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

10.4.2 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). Care should be exercised when jumping into a look-up table or program branch table (computed GOTO) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower 8 bits of the memory address roll over from 0xFFh to 0X00h in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the table location within the table.

For more information, refer to Application Note AN556, *"Implementing a Table Read"* (DS00556E).

10.4.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<4:3> is loaded with PCLATH<4:3>.

10.4.4 STACK

The MCP19214/5 devices have an 8-level x 13-bit wide hardware stack (refer to Figure 10-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the 9th push overwrites the value that was stored from the first push. The 10th push overwrites the 2nd push (and so on).

Note 1: There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.

2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

10.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

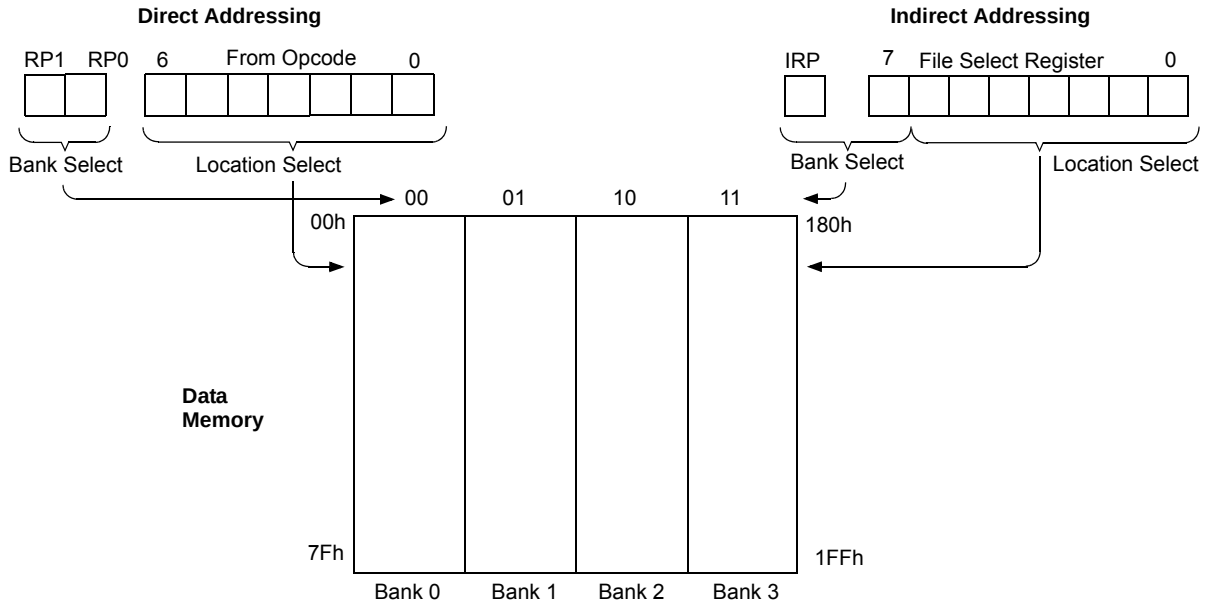
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register directly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit in the STATUS register, as shown in Figure 10-3.

A simple program to clear RAM location 40h-7Fh using indirect addressing is shown in Example 10-3.

EXAMPLE 10-3: INDIRECT ADDRESSING

```
        MOVLW   0x40    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR     ;inc pointer
        BTFSS   FSR,7   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```

FIGURE 10-3: DIRECT/INDIRECT ADDRESSING



For memory map detail, refer to Figure 10-2.

NOTES:

11.0 DEVICE CONFIGURATION

Device Configuration consists of Configuration Word, Code Protection and Device ID.

Note:	The $\overline{\text{DBGEN}}$ bit in Configuration Word is managed automatically by device development tools, including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'. Debug is available only on the MCP19215.

11.1 Configuration Word

There are several Configuration Word bits that allow different timers to be enabled and memory protection options. These are implemented as Configuration Word at 2007h.

REGISTER 11-1: CONFIG: CONFIGURATION WORD

R/P-1	U-1	R/P-1	R/P-1	U-1	R/P-1
$\overline{\text{DBGEN}}$	—	WRT1	WRT0	—	BOREN
bit 13					bit 8

U-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1	U-1	U-1
—	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 13 **$\overline{\text{DBGEN}}$:** ICD Debug bit
1 = ICD debug mode disabled
0 = ICD debug mode enabled

bit 12 **Unimplemented:** Read as '0'

bit 11-10 **WRT<1:0>:** Flash Program Memory Self-Write Enable bit
11 =Write protection off
10 =000h to 3FFh write protected, 400h to FFFh may be modified by PMCON1 control
01 =000h to 7FFh write protected, 800h to FFFh may be modified by PMCON1 control
00 =000h to FFFh write protected, entire program memory is write protected.

bit 9 **Unimplemented:** Read as '0'

bit 8 **BOREN:** Brown-Out Reset Enable bit
1 = BOR disabled during Sleep and Enabled during operation
0 = BOR disabled

bit 7 **Unimplemented:** Read as '0'

bit 6 **$\overline{\text{CP}}$:** Code Protection
1 = Program memory is not code protected
0 = Program memory is external read and write protected

bit 5 **MCLRE:** $\overline{\text{MCLR}}$ Pin Function Select
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$ function and weak internal pull-up is enabled
0 = $\overline{\text{MCLR}}$ pin is alternate function, $\overline{\text{MCLR}}$ function is internally disabled

bit 4 **$\overline{\text{PWRTE}}$:** Power-Up Timer Enable bit(1)
1 = PWRT disabled
0 = PWRT enabled

bit 3 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 2-0 **Unimplemented:** Read as '0'

Note 1: Bit is reserved and not controlled by user.

11.2 Code Protection

Code protection allows the device to be protected from unauthorized access. Internal access to the program memory is unaffected by any code protection setting.

11.2.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in the Configuration Word. When $\overline{CP}$ = 0, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. Refer to **Section 11.3 "Write Protection"** for more information.

11.3 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in the Configuration Word define the size of the program memory block that is protected.

11.4 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are not accessible during normal execution, but are readable and writable during Program/Verify mode. Only the Least Significant 7 bits of the ID locations are reported when using MPLAB Integrated Development Environment (IDE).

12.0 RESETS

The reset logic is used to place the MCP19214/5 into a known state. The source of the reset can be determined by using the device status bits.

There are multiple ways to reset these devices:

- Power-on Reset (POR)
- Overtemperature Reset (OT)
- $\overline{\text{MCLR}}$ Reset
- WDT Reset
- Brown-out Reset (BOR)

To allow V_{DD} to stabilize, an optional power-up timer can be enabled to extend the Reset time after a POR event.

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a Reset state on:

- Power-on Reset
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset

WDT (Watchdog Timer) wake-up does not cause register resets in the same manner as a WDT Reset, since wake-up is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 12-1. The software can use these bits to determine the nature of the Reset. Refer to Table 12-2 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 12-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. Refer to **Section 5.0 "Digital Electrical Characteristics"** for pulse-width specifications.

FIGURE 12-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

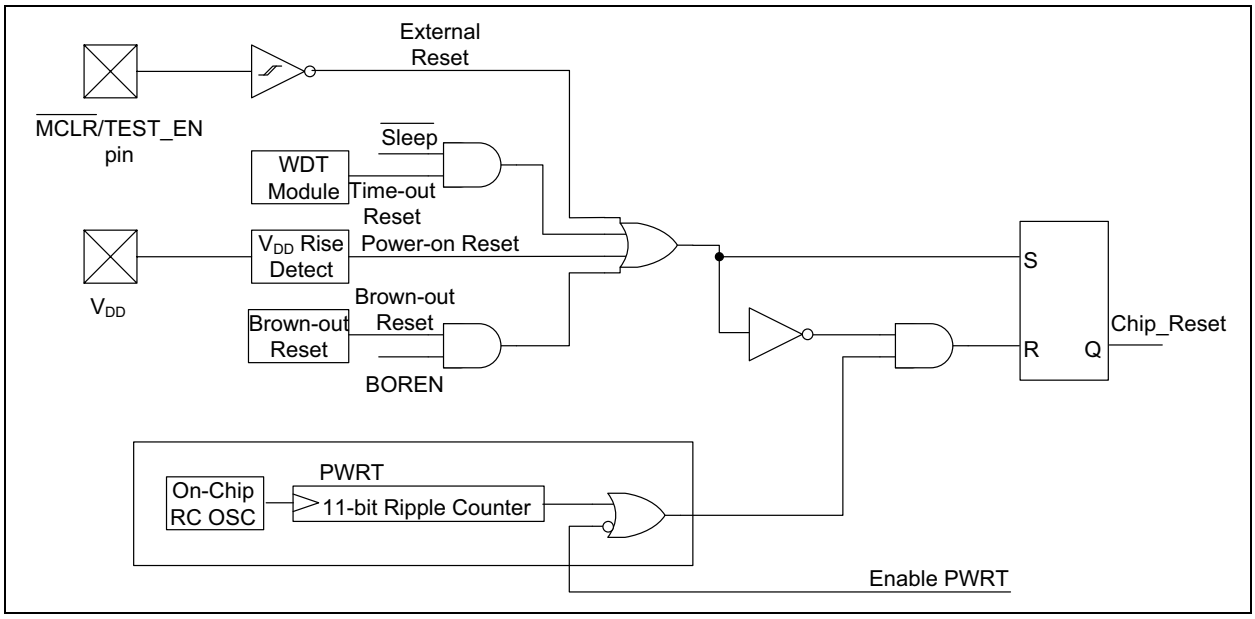


TABLE 12-1: TIME-OUT IN VARIOUS SITUATIONS

Power-Up		Wake-Up from Sleep
$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1	
T_{PWRT}	—	—

TABLE 12-2: STATUS/PCON BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	Condition
0	x	1	1	Power-on Reset
u	0	1	1	Brown-out Reset
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-Up
u	u	u	u	$\overline{MCLR}$ Reset during normal operation
u	u	1	0	$\overline{MCLR}$ Reset during Sleep

Legend: u = unchanged, x = unknown

12.1 Power-on Reset (POR)

The on-chip POR circuit holds the chip in Reset until V_{DD} has reached a high enough level for proper operation. To take advantage of the POR, simply connect the $\overline{MCLR}$ pin through a resistor to V_{DD}. This will eliminate external RC components usually needed to create Power-on Reset.

When the device starts normal operation (exits the Reset condition), device operating parameters (i.e., voltage, frequency, temperature, etc.) must be met to ensure proper operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

12.2 $\overline{MCLR}$

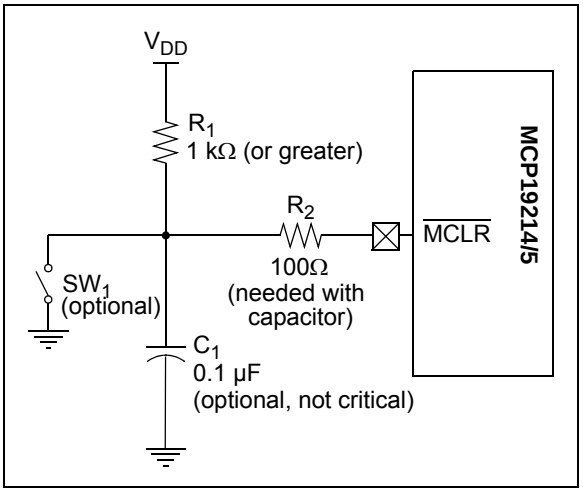
MCP19214/5 devices have a noise filter in the $\overline{MCLR}$ Reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive the $\overline{MCLR}$ pin low.

Voltages applied to the $\overline{MCLR}$ pin that exceed its specification can result in both $\overline{MCLR}$ Resets and excessive current beyond the device specification during the ESD event. For this reason, Microchip recommends that the $\overline{MCLR}$ pin no longer be tied directly to V_{DD}. The use of a Resistor-Capacitor (RC) network, as shown in Figure 12-2, is recommended.

An internal $\overline{MCLR}$ option is enabled by clearing the MCLRE bit in the CONFIG register. When MCLRE = 0, the Reset signal to the chip is generated internally. When MCLRE = 1, the $\overline{MCLR}$ pin becomes an external Reset input. In this mode, the $\overline{MCLR}$ pin has a weak pull-up to V_{DD}.

FIGURE 12-2: RECOMMENDED $\overline{MCLR}$ CIRCUIT

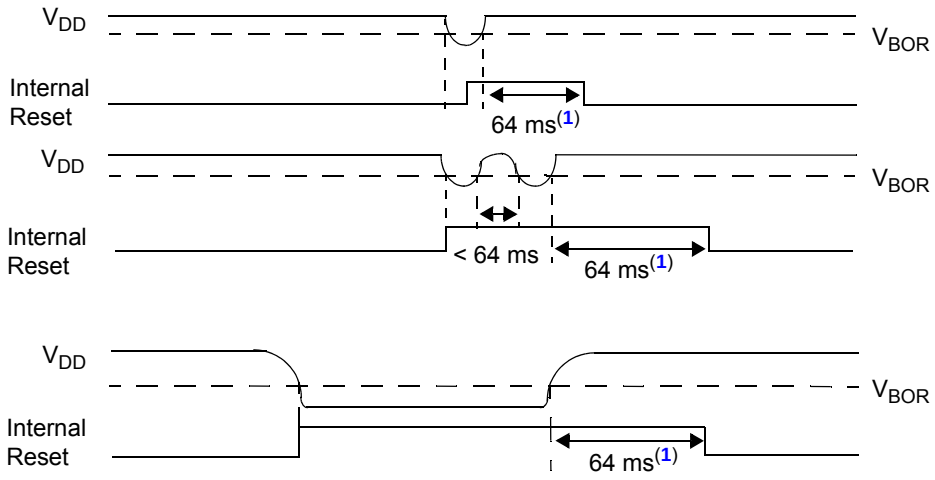


12.3 Brown-out Reset (BOR)

The BOREN bit <8> in the CONFIG register enables or disables the BOR mode, as defined in the CONFIG register. A brown-out occurs when V_{DD} falls below V_{BOR} for greater than 100 µs minimum. On any Reset (Power-on, Brown-out, Watchdog Timer, etc.), the chip will remain in Reset until V_{DD} rises above V_{BOR} (refer to Figure 12-3). If enabled, the Power-Up Timer will be invoked by the Reset and will keep the chip in Reset for an additional 64 ms. During power-up, it is recommended that the BOR configuration bit is enabled, holding the MCU in Reset (OSC turned off and no code execution) until V_{DD} exceeds the V_{BOR} threshold. Users have the option of adding an additional 64 ms delay by clearing the PWRTE bit. At this time, the V_{DD} voltage level is high enough to operate the MCU functions only; all other device functionality is not operational. This is independent of the value of V_{IN}, which is typically $V_{DD} + V_{DROPOUT}$. During power-down with BOR enabled, the MCU operation will be held in Reset when V_{DD} falls below the V_{BOR} threshold. With BOR disabled or while operating in Sleep mode, the POR will hold the part in Reset when V_{DD} falls below the V_{POR} threshold. Since this device runs at F_{OSC} = 8 MHz and the processor is fully operational at V_{DD} = 2V, it is recommended that BOR always be enabled during power-up and power-down.

Note: The Power-Up Timer is enabled by the $\overline{PWRTE}$ bit in the CONFIG register. If V_{DD} drops below V_{BOR} while the Power-Up Timer is running, the chip will go back into a Brown-out Reset and the Power-Up Timer will be re-initialized. Once the V_{DD} rises above V_{BOR}, the Power-Up Timer will execute a 64 ms reset.

FIGURE 12-3: BROWN-OUT SITUATIONS



Note 1: 64 ms delay only if PWRTE bit is programmed to '0'.

12.4 Power-Up Timer (PWRT)

The Power-Up Timer provides a fixed 64 ms (nominal) time-out on power-up only, from POR Reset. The Power-Up Timer operates from an internal RC oscillator. The chip is kept in Reset as long as PWRT is active. The PWRT delay allows the V_{DD} to rise to an acceptable level. A bit ($\overline{PWRTE}$) in the CONFIG register can disable (if set) or enable (if cleared or programmed) the Power-Up Timer.

The Power-Up Timer delay will vary from chip to chip due to:

- V_{DD} variation
- Temperature variation
- Process variation

Note: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a low level to the $\overline{MCLR}$ pin, rather than pulling this pin directly to V_{SS}.

The Power-Up Timer optionally delays device execution after a POR event. This timer is typically used to allow V_{DD} to stabilize before allowing the device to start running.

The Power-Up Timer is controlled by the $\overline{PWRTE}$ bit in the CONFIG register.

12.5 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the WDT Reset. Refer to **Section 14.0 "Watchdog Timer (WDT)"** for more information.

12.6 Start-Up Sequence

Upon the release of a POR, the following must occur before the device begins executing:

- Power-Up Timer runs to completion (if enabled)
- Oscillator start-up timer runs to completion
- $\overline{MCLR}$ must be released (if enabled)

The total time out will vary based on $\overline{PWRTE}$ bit status. For example, with PWRTE bit erased (PWRT disabled), there will be no time out at all. Figures 12-4, 12-5 and 12-6 represent time-out sequences.

Since the time outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then, bringing $\overline{MCLR}$ high will begin execution immediately (refer to Figure 12-5). This is useful for testing purposes or to synchronize more than one MCP19214/5 device operating in parallel.

12.6.1 POWER CONTROL (PCON) REGISTER

The Power Control (PCON) register (address 8Eh) has two Status bits to indicate what type of Reset occurred last.

FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 1

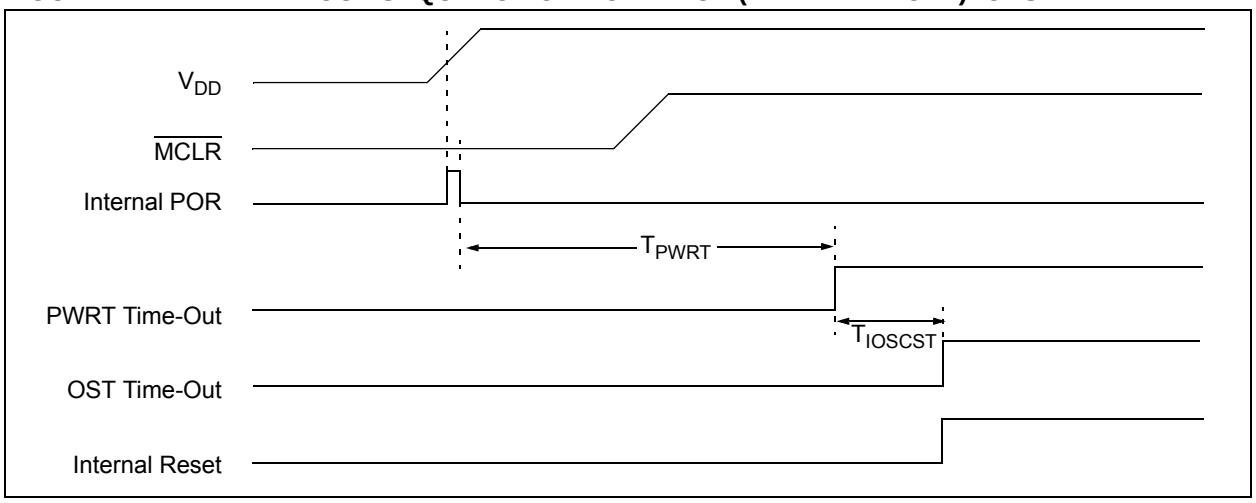


FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{MCLR}$): CASE 2

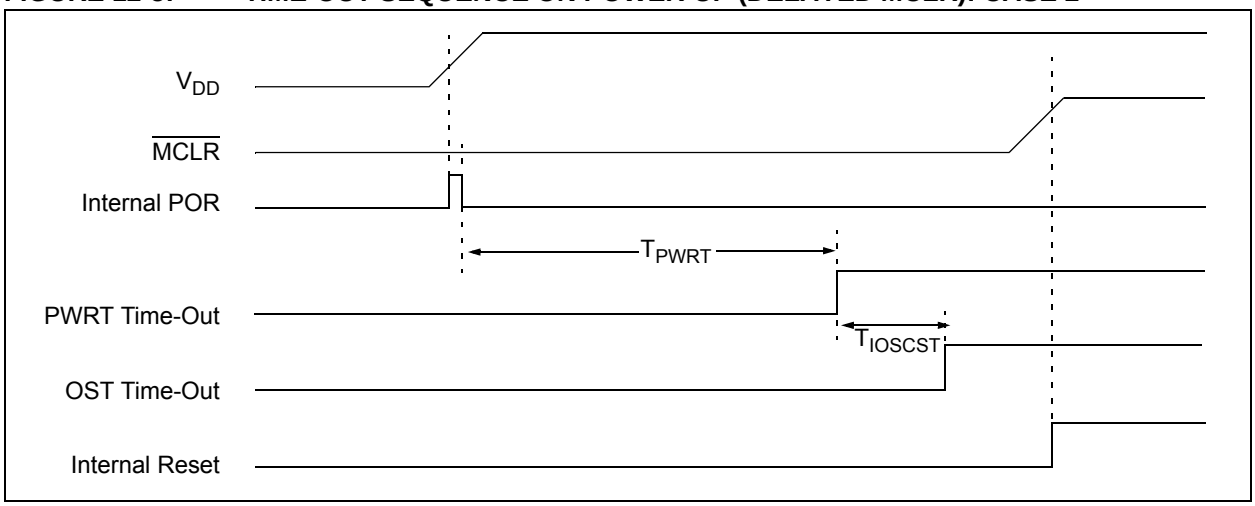
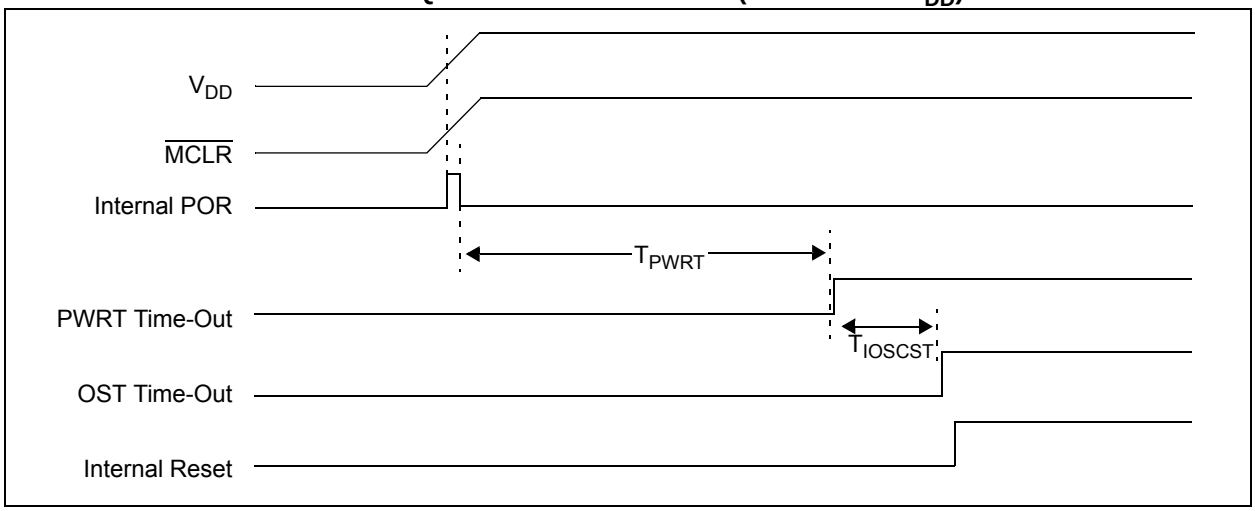


FIGURE 12-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ WITH V_{DD})



12.7 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset.Table 12-3 and 12-4 show the Reset conditions of these registers.

TABLE 12-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	Condition
0	x	1	1	Power-On Reset
u	0	1	1	Brown-Out Reset
u	u	0	u	WDT Reset
u	u	0	0	WDT Wake-Up from Sleep
u	u	1	0	Interrupt Wake-Up from Sleep
u	u	u	u	$\overline{MCLR}$ Reset during normal operation
u	u	1	0	$\overline{MCLR}$ Reset during Sleep
0	u	0	x	Not allowed. $\overline{TO}$ is set on POR.
0	u	x	0	Not allowed. $\overline{PD}$ is set on POR.

TABLE 12-4: RESET CONDITION FOR SPECIAL REGISTERS (Note 1)

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	0001 1xxx	---- --0u
Brown-out Reset	0000	0001 1xxx	---- --u0
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	000u uuuu	---- --uu
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	0001 0uuu	---- --uu
WDT Reset	0000h	0000 uuuu	---- --uu
WDT Wake-Up from Sleep	PC + 1	uuu0 0uuu	---- --uu
Interrupt Wake-Up from Sleep	PC + 1(2)	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: If a Status bit is not implemented, that bit will be read as '0'.

2: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

12.8 Power Control (PCON) Register

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)

The PCON register bits are shown in Register 12-1.

REGISTER 12-1: PCON: POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

TABLE 12-5: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	97
STATUS	IPR	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	77

Legend: — = unimplemented bit, read as '0'. Shaded cells are not used by Resets.

Note 1: Other (non-power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

NOTES:

13.0 INTERRUPTS

The MCP19214/5 devices have multiple sources of interrupt:

- External Interrupt (INT pin)
- Interrupt-On-Change (IOC)
- Timer0 Overflow Interrupt
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- ADC Interrupt
- System Input Undervoltage Error
- System Input Overvoltage Error
- Master Synchronous Serial Port (MSSP)
- Bus Collision on MSSP (BCL)
- I/V Good Comparators
- Gate Drive UVLO Circuit
- Internal Voltage Regulator (V_{DD}) UVLO Circuit
- Capture/Compare 1
- Capture/Compare 2
- USART TX interrupt (**MCP19215** only)
- USART RX complete interrupt (**MCP19215** only)
- Overtemperature

The Interrupt Control (INTCON) register and the Peripheral Interrupt Request (PIRx) registers record individual interrupt requests in flag bits. The INTCON register also has individual and global interrupt enable bits.

The Global Interrupt Enable bit (GIE) in the INTCON register enables (if set) all unmasked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in the INTCON register and PIEx registers. GIE is cleared on Reset.

When an interrupt is serviced, the following actions occur automatically:

- the GIE is cleared to disable any further interrupt
- the return address is pushed onto the stack
- the PC is loaded with 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

Note 1: Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt operation, refer to its peripheral chapter.

13.1 Interrupt Latency

For external interrupt events, such as the INT pin or PORTGPx change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends upon when the interrupt event occurs (refer to Figure 13-2). The latency is the same for one- or two-cycle instructions.

13.2 GPA2/INT Interrupt

The external interrupt on the GPA2/INT pin is edge-triggered, either on the rising edge if the INTEDG bit in the OPTION_REG register is set, or the falling edge if the INTEDG bit is clear. When a valid edge appears on the GPA2/INT pin, the INTF bit in the INTCON register is set. This interrupt can be disabled by clearing the INTE control bit in the INTCON register. The INTF bit must be cleared by software in the Interrupt Service Routine before re-enabling this interrupt. The GPA2/INT interrupt can wake up the processor from Sleep, if the INTE bit was set prior to going into Sleep. Refer to **Section 19.0 “Power-Down Mode (Sleep)”** for details on Sleep and **Section 19.1 “Wake-Up from Sleep”** for timing of wake-up from Sleep through GPA2/INT interrupt.

Note: The ANSEL register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’ and cannot generate an interrupt.

FIGURE 13-1: INTERRUPT LOGIC

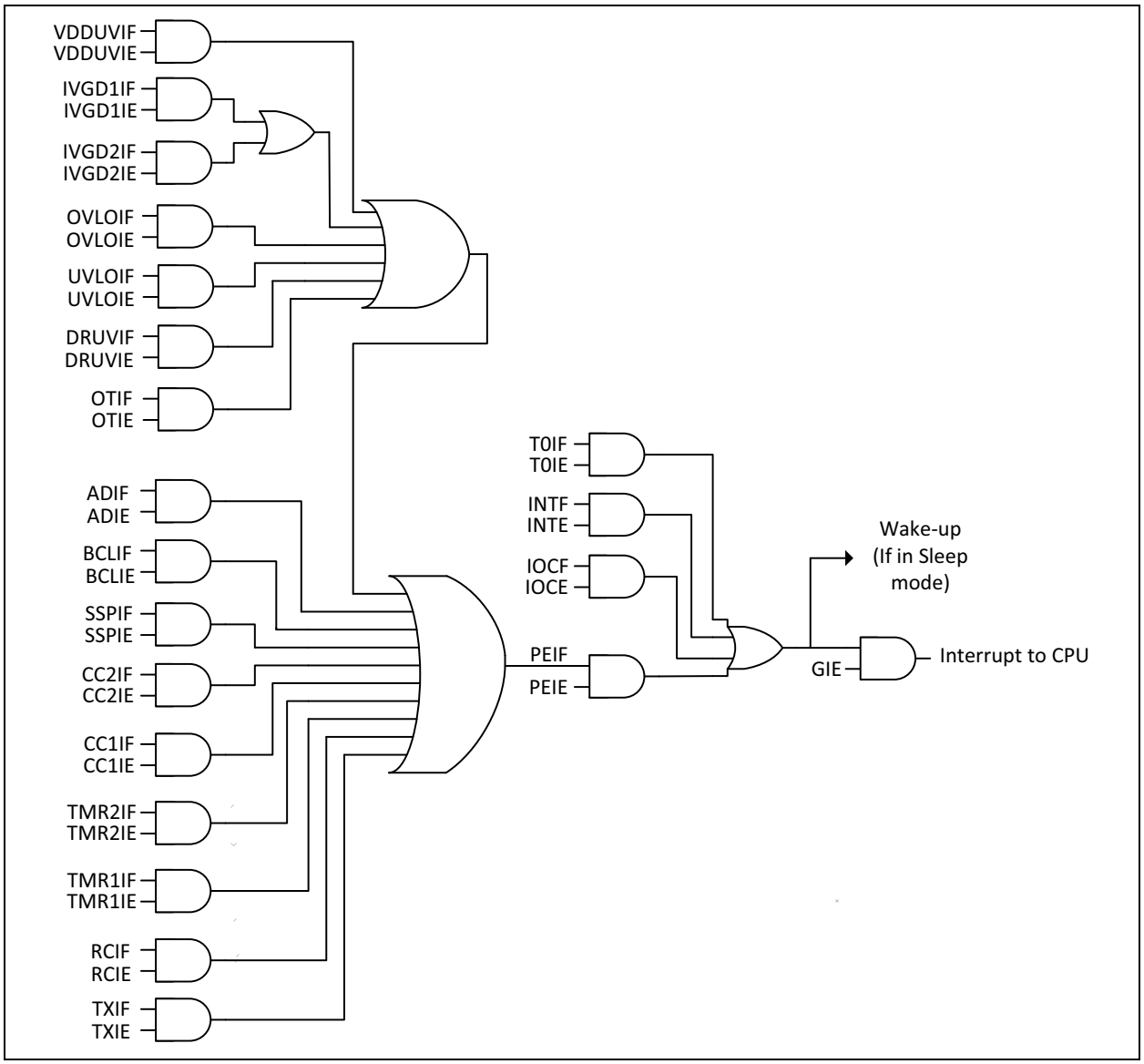
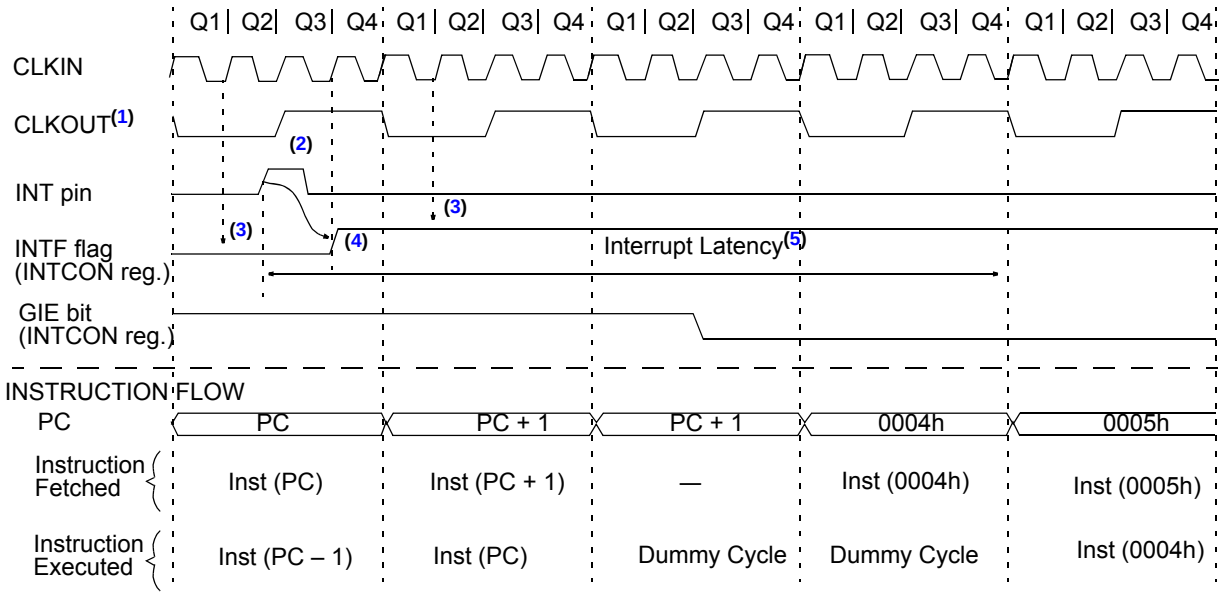


FIGURE 13-2: INT PIN INTERRUPT TIMING



Note 1: CLKOUT is available only in INTOSC and RC Oscillator modes.

2: For minimum width of INT pulse, refer to AC specifications in **Section 5.0 "Digital Electrical Characteristics"**.

3: INTF flag is sampled here (every Q1).

4: INTF is enabled to be set any time during the Q4-Q1 cycles.

5: Asynchronous interrupt latency = 3-4 T_{CY}. Synchronous latency = 3 T_{CY}, where T_{CY} = instruction cycle time. Latency is the same whether Inst (PC) is a single-cycle or a two-cycle instruction.

13.3 Interrupt Control Registers

13.3.1 INTCON REGISTER

The INTCON register is a readable and writable register that contains the various enable and flag bits for the TMR0 register overflow, interrupt-on-change and external INT pin interrupts.

Note:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit (GIE) in the INTCON register. The user's software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 13-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **GIE:** Global Interrupt Enable bit
- 1 = Enables all unmasked interrupts
- 0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
- 1 = Enables all unmasked peripheral interrupts
- 0 = Disables all peripheral interrupts

bit 5 **T0IE:** TMR0 Overflow Interrupt Enable bit
- 1 = Enables the TMR0 interrupt
- 0 = Disables the TMR0 interrupt

bit 4 **INTE:** INT External Interrupt Enable bit
- 1 = Enables the INT external interrupt
- 0 = Disables the INT external interrupt

bit 3 **IOCE:** Interrupt-on-Change Enable bit(1)
- 1 = Enables the interrupt-on-change
- 0 = Disables the interrupt-on-change

bit 2 **T0IF:** TMR0 Overflow Interrupt Flag bit(2)
- 1 = TMR0 register has overflowed (must be cleared in software)
- 0 = TMR0 register did not overflow

bit 1 **INTF:** External Interrupt Flag bit
- 1 = The external interrupt occurred (must be cleared in software)
- 0 = The external interrupt did not occur

bit 0 **IOCF:** Interrupt-on-Change Interrupt Flag bit
- 1 = When at least one of the interrupt-on-change pins changed state
- 0 = None of the interrupt-on-change pins have changed state

Note 1: IOCx registers must also be enabled.

2: T0IF bit is set when TMR0 rolls over. TMR0 is unchanged on Reset and should be initialized before clearing T0IF bit.

13.3.1.1 PIE1 Register

The PIE1 register contains the Peripheral Interrupt Enable bits, as shown in Register 13-2.

Note 1: Bit PEIE in the INTCON register must be set to enable any peripheral interrupt.

REGISTER 13-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1 (ADDRESS: 87h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **OTIE:** Over Temperature Interrupt Enable bit
1 = Enables the Over temperature interrupt
0 = Disables the Over temperature interrupt

bit 6 **ADIE:** ADC Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **BCLIE:** MSSP Bus Collision Interrupt Enable bit
1 = Enables the MSSP Bus Collision Interrupt
0 = Disables the MSSP Bus Collision Interrupt

bit 4 **SSPIE:** Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 3 **CC2IE:** Capture2/Compare2 Interrupt enable bit
1 = Enables the Capture2/Compare2 interrupt
0 = Disables the Capture2/Compare2 interrupt

bit 2 **CC1IE:** Capture1/Compare1 Interrupt enable bit
1 = Enables the Capture1/Compare1 interrupt
0 = Disables the Capture1/Compare1 interrupt

bit 1 **TMR2IE:** Timer2 Interrupt Enable
1 = Enables the Timer2 interrupt
0 = Disables the Timer2 interrupt

bit 0 **TMR1IE:** Timer1 Interrupt Enable
1 = Enables the Timer1 interrupt
0 = Disables the Timer1 interrupt

13.3.1.2 PIE2 Register

The PIE2 register contains the Peripheral Interrupt Enable bits, as shown in Register 13-3.

Note 1: Bit PEIE in the INTCON register must be set to enable any peripheral interrupt.

REGISTER 13-3: PIE2: PERIPHERAL INTERRUPT FLAG REGISTER 2 (ADDRESS: 08h)

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
IVGD1IF	—	IVGD2IF	—	VDDUVIF	DRUVIF	OVLOIF	UVLOIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IVGD1IF:** IV_GOOD comparator interrupt flag bit (PWM #1)
1 = IV_GOOD event has occurred
0 = IV_GOOD event has not occurred

bit 6 **Unimplemented:** Read as '0'

bit 5 **IVGD2IF:** IV_GOOD comparator interrupt flag bit (PWM #2)
1 = IV_GOOD event has occurred
0 = IV_GOOD event has not occurred

bit 4 **Unimplemented:** Read as '0'

bit 3 **VDDUVIF:** Internal LDO UVLO interrupt flag bit
1 = V_{DD} UVLO event has occurred
0 = V_{DD} UVLO event has not occurred

bit 2 **DRUVIF:** Gate Drive Undervoltage Lockout Interrupt flag bit
1 = Gate Drive Undervoltage Lockout has occurred
0 = Gate Drive Undervoltage Lockout has not occurred

bit 1 **OVLOIF:** V_{IN} Overvoltage Lockout interrupt flag bit
With OVLOINTP bit set
1 = A V_{IN} Not Overvoltage to V_{IN} Overvoltage edge has been detected
0 = A V_{IN} Not Overvoltage to V_{IN} Overvoltage edge has Not been detected
With OVLOINTN bit set
1 = A V_{IN} Overvoltage to V_{IN} Not Overvoltage edge has been detected
0 = A V_{IN} Overvoltage to V_{IN} Not Overvoltage edge has Not been detected

bit 0 **UVLOIF:** V_{IN} Undervoltage Lockout interrupt flag bit
With UVLOINTP bit set
1 = A V_{IN} Not Undervoltage to V_{IN} Undervoltage edge has been detected
0 = A V_{IN} Not Undervoltage to V_{IN} Undervoltage edge has Not been detected
With UVLOINTN bit set
1 = A V_{IN} Undervoltage to V_{IN} Not Undervoltage edge has been detected
0 = A V_{IN} Undervoltage to V_{IN} Not Undervoltage edge has Not been detected

13.3.1.3 PIE3 Register

The PIE3 register contains the Peripheral Interrupt Enable bits, as shown in Register 13-4. This register is present only in MCP19215 device.

Note 1: Bit PEIE in the INTCON register must be set to enable any peripheral interrupt.

REGISTER 13-4: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3 (ADDRESS: 89h)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RCIE	TXIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **RCIE:** USART Receive Interrupt Flag bit (**MCP19215** only)
- 1 = Enables the USART receive interrupt
- 0 = Disables the USART receive interrupt

bit 0 **TXIE:** USART Transmit Interrupt Flag bit (**MCP19215** only)
- 1 = Enables the USART transmit interrupt
- 0 = Disables the USART transmit interrupt

13.3.1.4 PIR1 Register

The PIR1 register contains the Peripheral Interrupt Flag bits, as shown in Register 13-5.

Note 1:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit (GIE) in the INTCON register. The user's software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 13-5: PIR1: PERIPHERAL INTERRUPT FLAG REGISTER 1 (ADDRESS: 07h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **OTIF:** Over Temperature interrupt flag bit
1 = Over temperature event has occurred
0 = Over temperature event has not occurred

bit 6 **ADIF:** ADC Interrupt Flag bit
1 = ADC conversion complete
0 = ADC conversion has not completed or has not been started

bit 5 **BCLIF:** MSSP Bus Collision Interrupt Flag bit
1 = Bus collision has occurred
0 = Bus collision has not occurred

bit 4 **SSPIF:** Synchronous Serial Port (MSSP) Interrupt Flag bit
1 = Transmission or Reception has occurred
0 = Transmission or Reception has not occurred

bit 3 **CC2IF:** Capture2/Compare2 interrupt flag bit
1 = Capture or Compare has occurred
0 = Capture or Compare has not occurred

bit 2 **CC1IF:** Capture1/Compare1 interrupt flag bit
1 = Capture or Compare has occurred
0 = Capture or Compare has not occurred

bit 1 **TMR2IF:** Timer2 to PR2 Match Interrupt Flag
1 = Timer2 to PR2 match occurred (must be cleared in software)
0 = Timer2 to PR2 match did not occur

bit 0 **TMR1IF:** Timer1 Interrupt Flag
1 = Timer1 rolled over (must be cleared in software)
0 = Timer1 has not rolled over

13.3.1.5 PIR2 Register

The PIR2 register contains the Peripheral Interrupt Flag bits, as shown in Register 13-6.

Note 1:	Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit (GIE) in the INTCON register. The user's software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 13-6: PIR2: PERIPHERAL INTERRUPT FLAG REGISTER 2 (ADDRESS: 08h)

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
IVGD1IF	—	IVGD2IF	—	VDDUVIF	DRUVIF	OVLOIF	UVLOIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IVGD1IF:** IV_GOOD comparator interrupt flag bit (PWM #1)
1 = IV_GOOD event has occurred
0 = IV_GOOD event has not occurred

bit 6 **Unimplemented:** Read as '0'

bit 5 **IVGD2IF:** IV_GOOD comparator interrupt flag bit (PWM #2)
1 = IV_GOOD event has occurred
0 = IV_GOOD event has not occurred

bit 4 **Unimplemented:** Read as '0'

bit 3 **VDDUVIF:** Internal LDO UVLO interrupt flag bit
1 = V_{DD} UVLO event has occurred
0 = V_{DD} UVLO event has not occurred

bit 2 **DRUVIF:** Gate Drive Undervoltage Lockout Interrupt flag bit
1 = Gate Drive Undervoltage Lockout has occurred
0 = Gate Drive Undervoltage Lockout has not occurred

bit 1 **OVLOIF:** V_{IN} Overvoltage Lockout interrupt flag bit
With OVLOINTP bit set
1 = V_{IN} Not Overvoltage to V_{IN} Overvoltage edge has been detected
0 = A V_{IN} Not Overvoltage to V_{IN} Overvoltage edge has Not been detected
With OVLOINTN bit set
1 = A V_{IN} Overvoltage to V_{IN} Not Overvoltage edge has been detected
0 = A V_{IN} Overvoltage to V_{IN} Not Overvoltage edge has Not been detected

bit 0 **UVLOIF:** V_{IN} Undervoltage Lockout interrupt flag bit
With UVLOINTP bit set
1 = A V_{IN} Not Undervoltage to V_{IN} Undervoltage edge has been detected
0 = A V_{IN} Not Undervoltage to V_{IN} Undervoltage edge has Not been detected
With UVLOINTN bit set
1 = A V_{IN} Undervoltage to V_{IN} Not Undervoltage edge has been detected
0 = A V_{IN} Undervoltage to V_{IN} Not Undervoltage edge has Not been detected

13.3.1.6 PIR3 Register

The PIR3 register contains the Peripheral Interrupt Flag bits, as shown in Register 13-7. This register is present only in MCP19215 device.

Note 1: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit (GIE) in the INTCON register. The user's software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 13-7: PIR3: PERIPHERAL INTERRUPT FLAG REGISTER 2 (ADDRESS: 09h)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RCIF	TXIF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **RCIF:** USART Receive Interrupt Flag bit (**MCP19215** only)
- 1 = The USART receive buffer is full (cleared by reading RCREG)
- 0 = The USART receive buffer is not full

bit 0 **TXIF:** USART Transmit Interrupt Flag bit (**MCP19215** only)
- 1 = The USART transmit buffer is empty (cleared by writing to TXREG)
- 0 = The USART transmit buffer is full

TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	85
PIE1	OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	103
PIE2	IVGD1IE	—	IVGD2IE	OTIE	VDDUVIE	DRUVIE	OVLOIE	UVLOIE	104
PIE3	—	—	—	—	—	—	RCIE	TXIE	105
PIR1	OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	106
PIR2	IVGD1IF	—	IVGD2IF	—	VDDUVIF	DRUVIF	OVLOIF	UVLOIF	107
PIR3	—	—	—	—	—	—	RCIF	TXIF	108

Legend: — = unimplemented locations, read as '0'. Shaded cells are not used by Interrupts.

13.4 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may want to save key registers during an interrupt (e.g., W and STATUS registers). This must be implemented in software.

Temporary holding registers W_TEMP and STATUS_TEMP should be placed in the last 16 bytes of GPR (refer to Figure 10-2). These 16 locations are common to all banks and do not require banking. This makes context save and restore operations simpler. The code shown in Example 13-1 can be used to:

- Store the W register
- Store the STATUS register
- Execute the ISR code
- Restore the Status (and Bank Select Bit) register
- Restore the W register

Note: The MCP19214/5 devices do not require saving the PCLATH. However, if computed GOTOs are used in both the ISR and the main code, the PCLATH must be saved and restored in the ISR.

EXAMPLE 13-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP              ;Copy W to TEMP register
SWAPF   STATUS,W            ;Swap status to be saved into W
                            ;Swaps are used because they do not affect the status bits
MOVWF   STATUS_TEMP         ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                      ;Insert user code here
:
SWAPF   STATUS_TEMP,W       ;Swap STATUS_TEMP register into W
                            ;(sets bank to original state)
MOVWF   STATUS              ;Move W into STATUS register
SWAPF   W_TEMP,F            ;Swap W_TEMP
SWAPF   W_TEMP,W            ;Swap W_TEMP into W
```

NOTES:

14.0 WATCHDOG TIMER (WDT)

The Watchdog Timer is a free-running timer. The WDT is enabled by setting the WDTE bit in the CONFIG register (default setting).

14.1 Watchdog Timer (WDT) Operation

During normal operation, a WDT time-out generates a device reset. If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation; this is known as a WDT wake-up. The WDT can be permanently disabled by clearing the WDTE bit in the CONFIG register. Refer to **Section 11.1 "Configuration Word"** for more information.

The postscaler assignment is fully under software control and can be changed during program execution.

14.2 WDT Period

The WDT has a nominal time-out period of 18 ms (with no prescaler). The time-out periods vary with temperature, V_{DD} and process variations from part to part (refer to Table 5-2). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION_REG register. Thus, time-out periods up to 2.3 seconds can be realized.

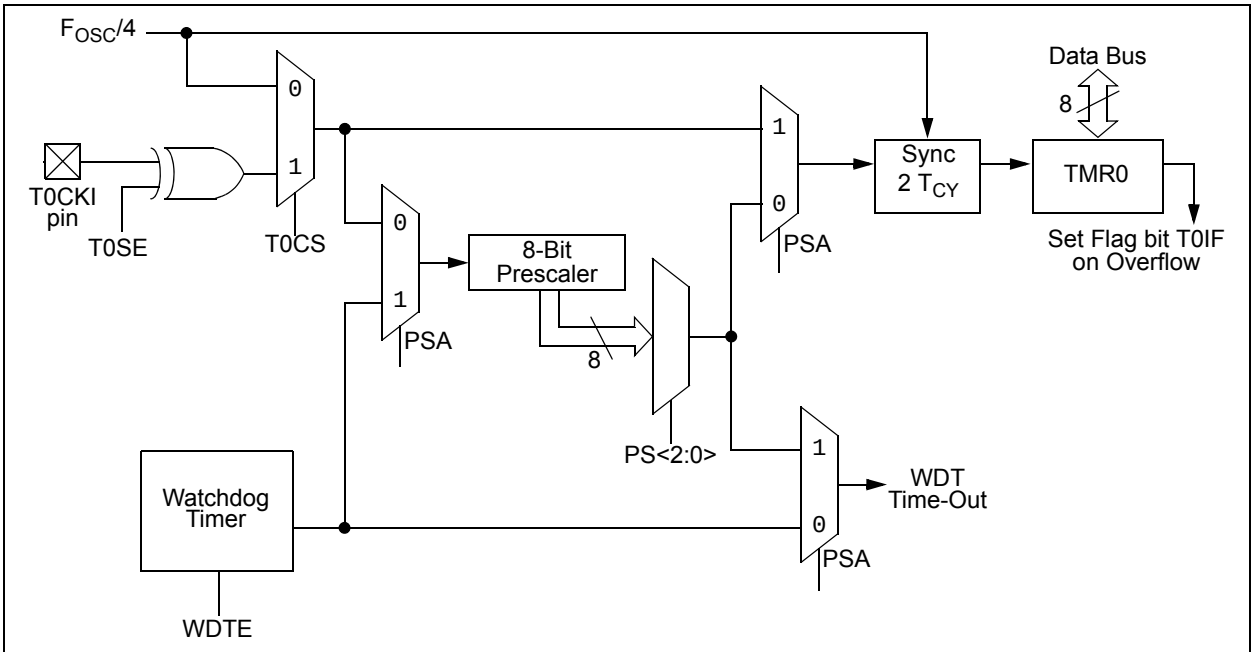
The `CLRWDT` and `SLEEP` instructions clear the WDT and the prescaler, if assigned to the WDT, and prevent it from timing out and generating a device reset.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

14.3 WDT Programming Considerations

Under worst-case conditions (i.e., V_{DD} = Minimum, Temperature = Maximum, Maximum WDT prescaler), it may take several seconds before a WDT time-out occurs.

FIGURE 14-1: WATCHDOG TIMER WITH SHARED PRESCALE BLOCK DIAGRAM



Note 1: T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION_REG register.

2: WDTE bit is in the CONFIG register.

TABLE 14-1: WDT STATUS

Conditions	WDT
WDTE = 0	Cleared
CLRWDT Command	
Exit Sleep	

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	85

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: Refer to Register 11-1 for operation of all the bits in the CONFIG register.

TABLE 14-3: SUMMARY OF CONFIGURATION WORD ASSOCIATED WITH WATCHDOG TIMER

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG	13:8	—	—	$\overline{\text{DBGEN}}$	—	WRT1	WRT0	—	BOREN	89
	7:0	—	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	—	—	—	

Legend: — = unimplemented location, read as '1'. Shaded cells are not used by Watchdog Timer.

15.0 INTERRUPT-ON-CHANGE

Each PORTGPA and PORTGPB pin is individually configurable as an interrupt-on-change pin. Control bits IOCA and IOCB enable or disable the interrupt function for each pin. Refer to Registers 15-1 and 15-2. The interrupt-on-change is disabled on a Power-on Reset.

The interrupt-on-change on GPA5 is disabled when configured as MCLR pin in the CONFIG register.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTGPA or PORTGPB. The mismatched outputs of the last read of all the PORTGPA and PORTGPB pins are ORed together to set the Interrupt-on-Change Interrupt Flag (IOCF) bit in the INTCON register.

15.1 Enabling the Module

To allow individual port pins to generate an interrupt, the IOCE bit in the INTCON register must be set. If the IOCE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

15.2 Individual Pin Configuration

To enable a pin to detect an interrupt-on-change, the associated IOCAx or IOCBx bit in the IOCA or IOCB registers is set.

15.3 Clearing Interrupt Flags

The user, in the Interrupt Service Routine, clears the interrupt by:

a) Any read of PORTGPA or PORTGPB AND Clear flag bit IOCF. This will end the mismatch condition.

OR

b) Any write of PORTGPA or PORTGPB AND Clear flag bit IOCF. This will end the mismatch condition.

A mismatch condition will continue to set flag bit IOCF. Reading PORTGPA or PORTGPB will end the mismatch condition and allow flag bit IOCF to be cleared. The latch holding the last read value is not affected by a MCLR Reset. After this Reset, the IOCF flag will continue to be set if a mismatch is present.

Note:	If a change on the I/O pin occurs when any PORTGPA or PORTGPB operation is being executed, the IOCF interrupt flag may not get set.

15.4 Operation in Sleep

The interrupt-on-change interrupt sequence wakes the device from Sleep mode, if the IOCE bit is set.

15.5 Interrupt-On-Change Registers

REGISTER 15-1: IOCA: INTERRUPT-ON-CHANGE PORTGPA REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCA7	IOCA6	IOCA5	—	IOCA3	IOCA2	IOCA1	IOCA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **IOCA<7:6>**: Interrupt-on-Change PORTGPA register bits
1 = Interrupt-on-change enabled on the pin.
0 = Interrupt-on-change disabled on the pin.

bit 5 **IOCA<5>**: Interrupt-on-Change PORTGPA register bits(1)
1 = Interrupt-on-change enabled on the pin.
0 = Interrupt-on-change disabled on the pin.

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **IOCA<3:0>**: Interrupt-on-Change PORTGPA register bits
1 = Interrupt-on-change enabled on the pin.
0 = Interrupt-on-change disabled on the pin.

Note 1: The Interrupt-on-Change on GPA5 is disabled if GPA5 is configured as $\overline{\text{MCLR}}$.

REGISTER 15-2: IOCB: INTERRUPT-ON-CHANGE PORTGPB REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
IOCB7(1)	IOCB6(1)	IOCB5	IOCB4	—	—	IOCB1(1)	IOCB0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-4 **IOCB<7:4>**: Interrupt-on-Change PORTGPB register bits
1 = Interrupt-on-change enabled on the pin.
0 = Interrupt-on-change disabled on the pin.

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1-0 **IOCB<1:0>**: Interrupt-on-Change PORTGPB register bits
1 = Interrupt-on-change enabled on the pin.
0 = Interrupt-on-change disabled on the pin.

Note 1: MCP19215 only.

TABLE 15-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0	128
ANSELB	—	ANSB6(1)	ANSB5(1)	ANSB4	—	—	ANSB1(1)	—	132
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
IOCA	IOCA7	IOCA6	IOCA5	—	IOCA3	IOCA2	IOCA1	IOCA0	114
IOCB	IOCB7(1)	IOCB6(1)	IOCB5	IOCB4	—	—	IOCB1(1)	IOCB0	114
TRISGPA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	127
TRISGPB	TRISB7(1)	TRISB6(1)	TRISB5	TRISB4	—	—	TRISB1(1)	TRISB0	131

Legend: — = unimplemented locations, read as '0'. Shaded cells are not used by interrupt-on-change.

Note 1: MCP19215 only.

NOTES:

16.0 OSCILLATOR MODES

The MCP19214/5 devices have one oscillator configuration which is an 8 MHz internal oscillator.

16.1 Internal Oscillator (INTOSC)

The Internal Oscillator module provides a system clock source of 8 MHz. The frequency of the internal oscillator can be trimmed with a calibration value in the OSCTUNE register.

16.2 Oscillator Calibration

The 8 MHz internal oscillator is factory-calibrated. The factory calibration values reside in the read-only CALWD6 register. These values must be read from the CALWD6 register and stored in the OSCCAL register. Refer to **Section 17.0 "Flash Program Memory Control"** for the procedure on reading the program memory.

Note:	The FCAL<6:0> bits in the CALWD6 register must be written into the OSCCAL register to calibrate the internal oscillator.

16.3 Frequency Tuning in User Mode

In addition to the factory calibration, the base frequency can be tuned in the user's application. This frequency tuning capability allows the user to deviate from the factory-calibrated frequency. The user can tune the frequency by writing to the OSCTUNE register (refer to Register 16-1).

REGISTER 16-1: OSCTUNE: OSCILLATOR TUNING REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **TUN<4:0>:** Frequency Tuning bits

01111 = Maximum frequency
01110
•
•
•
00001
00000 = Center frequency. Oscillator Module is running at the calibrated frequency.
11111
•
•
•
10000 = Minimum frequency

16.3.1 OSCILLATOR DELAY UPON POWER-UP, WAKE-UP AND BASE FREQUENCY CHANGE

In applications where the OSCTUNE register is used to shift the frequency of the internal oscillator, the application should not expect the frequency of the internal oscillator to stabilize immediately. In this case, the frequency may shift gradually toward the new value. The time for this frequency shift is less than eight cycles of the base frequency.

On power-up, the device is held in reset by the power-up time if the power-up timer is enabled.

Following a wake-up from Sleep mode or POR, an internal delay of ~10 µs is invoked to allow the memory bias to stabilize before program execution can begin.

TABLE 16-1: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	117

Legend: — = unimplemented locations, read as '0'. Shaded cells are not used by clock sources.

TABLE 16-2: SUMMARY OF CONFIGURATION WORD ASSOCIATED WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG6	13:8	—	—	—	—	—	—	—	—	66
	7:0	—	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	

Legend: — = unimplemented locations, read as '0'. Shaded cells are not used by clock sources.

17.0 FLASH PROGRAM MEMORY CONTROL

The Flash program memory is readable and writable during normal operation (full V_{IN} range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function registers (refer to Registers 17-1 to 17-5). There are six SFRs used to read and write this memory:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When interfacing the program memory block, the PMDATL and PMDATH registers form a two-byte word that holds the 14-bit data for read/write, and the PMADRL and PMADRH registers form a two-byte word that holds the 13-bit address of the FLASH location being accessed. These devices have 8k words of program Flash with an address range from 0000h-1FFFh.

The program memory allows single-word read and a four-word write. A four-word write automatically erases the row of the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the Flash program memory.

Depending on the settings of the Flash Program Memory Enable (WRT<1:0>) bits, the device may or may not be able to write certain blocks of the program memory; however, reads of the program memory are allowed.

When the Flash Program Memory Code Protection ($\overline{CP}$) bit is enabled, the program memory is code-protected and the device programmer (ICSP) cannot access data or program memory.

17.1 PMADRH and PMADRL Registers

The PMADRH and PMADRL registers can address up to a maximum of 8,192 words of program memory.

When selecting a program address value, the Most Significant Byte (MSB) of the address is written to the PMADRH register and the Least Significant Byte (LSB) is written to the PMADRL register.

17.2 PMCON1 and PMCON2 Registers

The PMCON1 register is the control register for the data program memory accesses.

Control bits RD and WR initiate read and write, respectively. In software, these bits can only be set, not cleared. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear.

The CALSEL bit allows the user to read locations in test memory in case there are calibration bits stored in the calibration word locations that need to be transferred to SFR trim registers. The CALSEL bit is only for reads. If a write operation is attempted with CALSEL = 1, no write will occur.

PMCON2 is not a physical register. Reading PMCON2 will read all '0's. The PMCON2 register is used exclusively in the flash memory write sequence.

17.3 Flash Program Memory Control Registers

REGISTER 17-1: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMDATL<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **PMDATL<7:0>**: 8 Least Significant Data bits to Write or Read from Program Memory

REGISTER 17-2: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMADRL<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **PMADRL<7:0>**: 8 Least Significant Address bits for Program Memory Read/Write Operation

REGISTER 17-3: PMDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMDATH<5:0>					
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **PMDATH<5:0>**: 6 Most Significant Data bits from Program Memory

REGISTER 17-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PMADRH<3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **PMADRH<3:0>**: 4 Most Significant Address bits or High bits for Program Memory Reads

REGISTER 17-5: PMCON1: PROGRAM MEMORY CONTROL REGISTER 1

U-1	R/W-0	U-0	U-0	U-0	R/W-0	R/S-0	R/S-0
—	CALSEL	—	—	—	WREN	WR	RD
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown
S = Bit can only be set			

bit 7 **Unimplemented:** Read as '1'

bit 6 **CALSEL:** Program Memory calibration space select bit
- 1 = Select test memory area for reads only (for loading calibration trim registers)
- 0 = Select user area for reads

bit 5-3 **Unimplemented:** Read as '0'

bit 2 **WREN:** Program Memory Write Enable bit
- 1 = Allows write cycles
- 0 = Inhibits write to the EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a write cycle to program memory. (The bit is cleared by hardware when write is complete. The WR bit can only be set (not cleared) in software.)
- 0 = Write cycle to the Flash memory is complete

bit 0 **RD:** Read Control bit
- 1 = Initiates a program memory read. (The read takes one cycle. The RD is cleared in hardware; the RD bit can only be set (not cleared) in software.)
- 0 = Does not initiate a Flash memory read

17.3.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must write two bytes of the address to the PMADRL and PMADRH registers, and then set control bit RD (bit 0 in the PMCON1 register). Once the read control bit is set, the Program Memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the BSF PMCON1,RD instruction to be ignored. The data is available, in the very next cycle, in the PMDATL and PMDATH registers; it can be read as two bytes in the following instructions. PMDATL and PMDATH registers will hold this value until another read or until it is written to by the user (during a write operation).

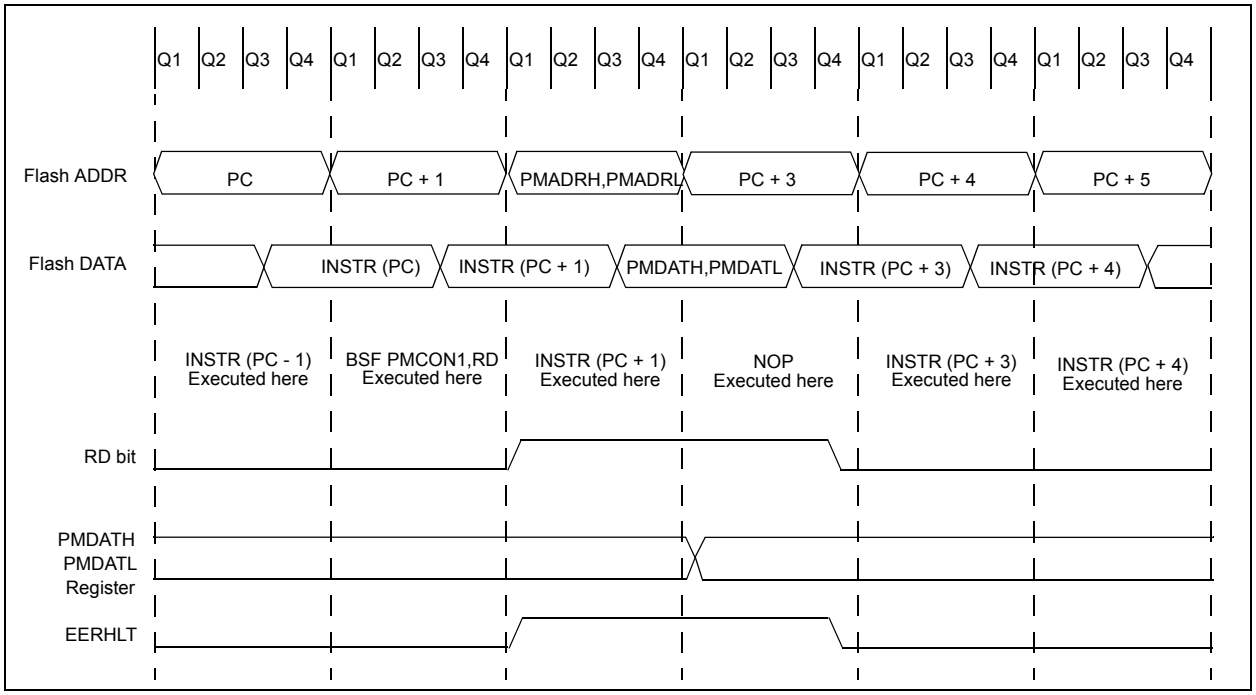
EXAMPLE 17-1: FLASH PROGRAM READ

```
    BANKSELPM_ADR; Change STATUS bits RP1:0 to select bank with PMADR
    MOVLWMS_PROG_PM_ADDR;
    MOVWFPMADRH; MS Byte of Program Address to read
    MOVLWLS_PROG_PM_ADDR;
    MOVWFPMADRL; LS Byte of Program Address to read
    BANKSELPMCON1; Bank to containing PMCON1
    BSF PMCON1, RD; EE Read

    NOP     ; First instruction after BSF PMCON1,RD executes normally

    NOP     ; Any instructions here are ignored as program
            ; memory is read in second cycle after BSF PMCON1,RD
            ;
    BANKSELPMDATL; Bank to containing PMADRL
    MOVFPMDATL, W; W = LS Byte of Program PMDATL
    MOVFPMDATH, W; W = MS Byte of Program PMDATL
```

FIGURE 17-1: FLASH PROGRAM MEMORY READ CYCLE EXECUTION – NORMAL MODE



17.3.2 WRITING TO THE FLASH PROGRAM MEMORY

A word of the Flash program memory may only be written to if the word is in an unprotected segment of the memory, as defined in **Section 11.1 "Configuration Word"** (bits <WRT1:0>).

> **Note:** The write protect bits are used to protect the user's program from modification by the user's code. They have no effect when programming is performed by ICSP. The code-protect bits, when programmed for code protection, will prevent the program memory from being written via the ICSP interface.

Flash program memory must be written in eight-word blocks. Refer to Figures 17-2 and 17-3 for more details. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where PMADRL<1:0> = 00. All block writes to program memory are done as 16-word erase by four-word write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, the WREN bit must first be loaded into the buffer registers (refer to Figure 17-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATL and PMDATH. After the address and data have been set, the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set the WR control bit in the PMCON1 register.

All eight buffer register locations should be written to with correct data. If less than eight words are being written to in the block of eight words, a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then the sequence of events to transfer data to the buffer registers must be executed.

To transfer data from the buffer registers to the program memory, the PMADRL and PMADRH must point to the last location in the eight-word block (PMADRL<2:0> = 111). Then the following sequence of events must be executed:

1. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
2. Set control bit WR in the PMCON1 register to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (000, 001, 010, 011). When the write is performed on the last word (PMADRL<2:0> = 111), a block of 16 words is automatically erased and the content of the eight-word buffer registers are written into the program memory.

After the BSF PMCON1,WR instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms only during the cycle in which the erase takes place (i.e., the last word of the 16-word block erase). This is not Sleep mode, as the clocks and peripherals will continue to run. After the eight-word write cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction. The above sequence must be repeated for the higher 12 words.

> **Note:** An erase is only initiated for the write of four words just after a row boundary; or PMCON1<WR> set with PMADRL<3:0> = xxxx0011.

Refer to Figure 17-2 for a block diagram of the buffer registers and the control signals for test mode.

17.3.3 PROTECTION AGAINST SPURIOUS WRITE

There are conditions when the device should not write to the program memory. To protect against spurious writes, various mechanisms have been built-in. On power-up, WREN is cleared. Also, the Power-Up Timer (72 ms duration) prevents program memory writes.

The write initiate sequence and the WREN bit help prevent an accidental write during a power glitch or software malfunction.

17.3.4 OPERATION DURING CODE PROTECT

When the device is code-protected, the CPU is able to read and write unscrambled data to the program memory. The test mode access is disabled.

17.3.5 OPERATION DURING WRITE PROTECT

When the program memory is write-protected, the CPU can read and execute from the program memory. The portions of program memory that are write-protected cannot be modified by the CPU using the PMCON registers. The write protection has no effect in ICSP mode.

FIGURE 17-2: BLOCK WRITES TO 8K FLASH PROGRAM MEMORY

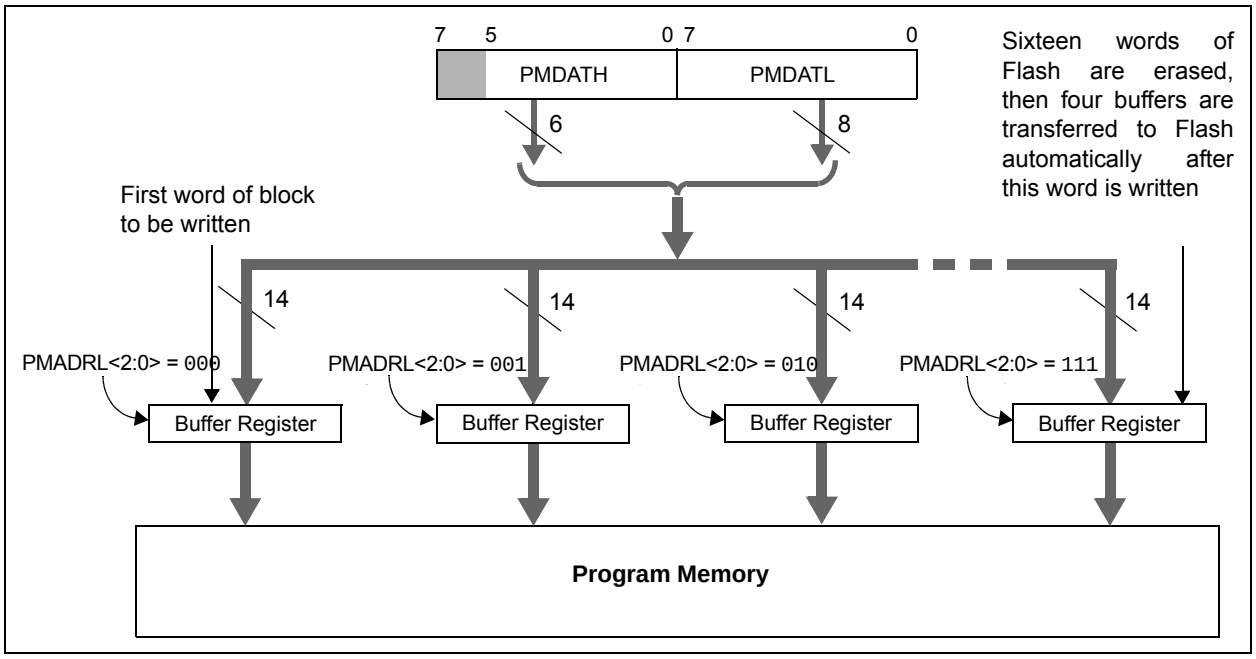
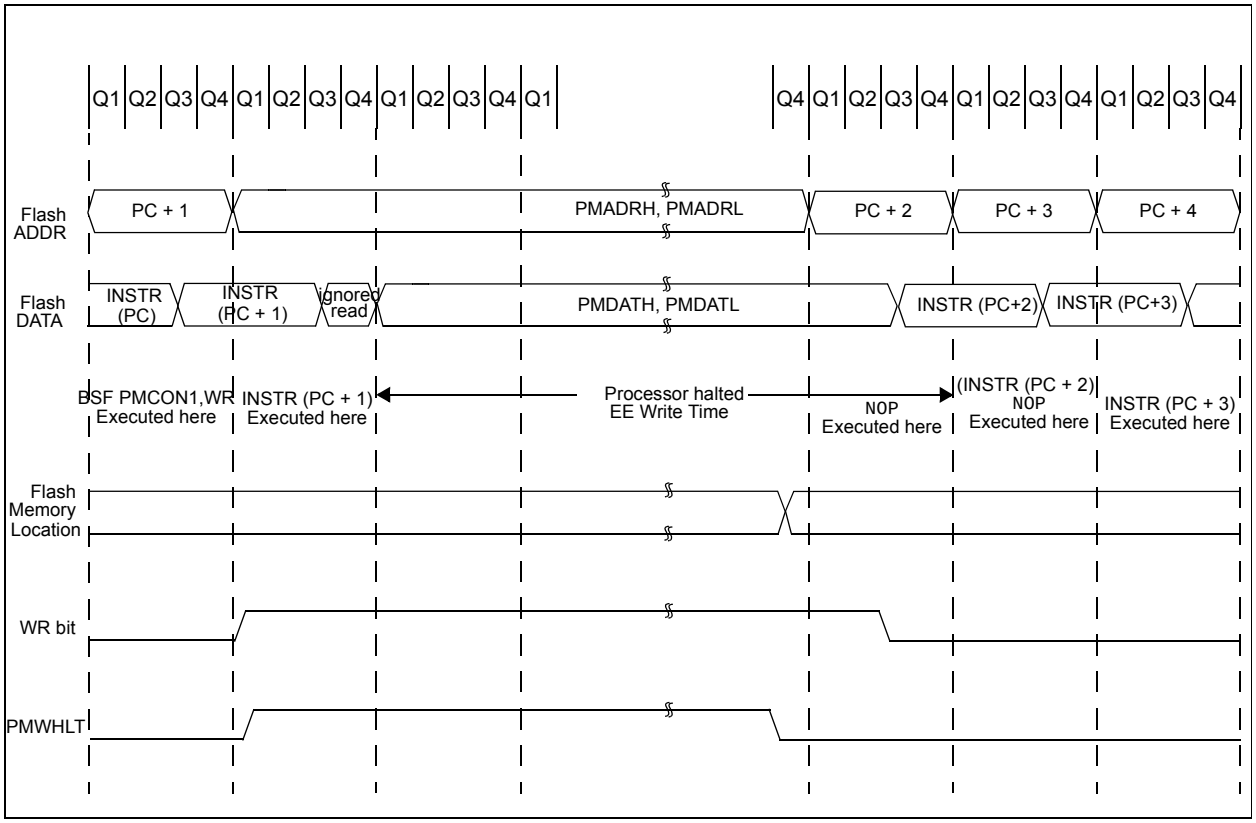


FIGURE 17-3: FLASH PROGRAM MEMORY LONG WRITE CYCLE EXECUTION



18.0 I/O PORTS

In general, when a peripheral is enabled, that pin may not be used as a general-purpose I/O pin.

Each port has the registers for its operation. These registers are:

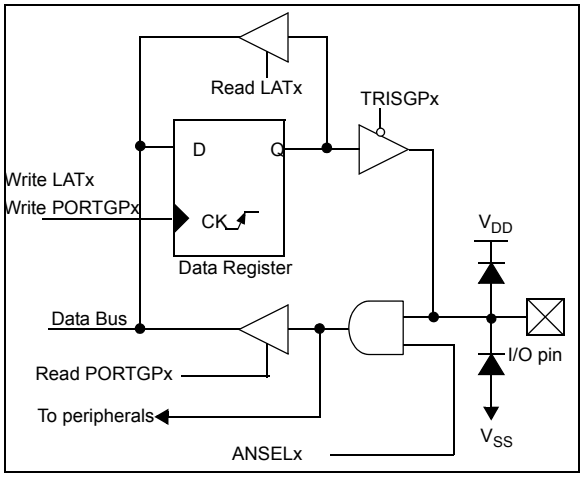
- TRISGPx registers (data direction register)
- PORTGPx registers (read the levels on the pins of the device)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUGPx (weak pull-up)

Ports with analog functions also have an ANSELx register, which can disable the digital input and save power. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 18-1.

FIGURE 18-1: GENERIC I/O PORTGPX OPERATION



EXAMPLE 18-1: INITIALIZING PORTGPA

```
; This code example illustrates
; initializing the PORTGPA register. The
; other ports are initialized in the same
; manner.

BANKSEL PORTGPA;
CLRF    PORTGPA;Init PORTA
BANKSEL ANSELA;
CLRF    ANSELA;digital I/O
BANKSEL TRISGPA;
MOVLW   B'00011111';Set GPA<3:0> as
            ;inputs
MOVWF   TRISGPA;and set GPA<7:5> as
            ;outputs
```

18.1 PORTGPA and TRISGPA Registers

PORTGPA is an 8-bit wide, bidirectional port consisting of six CMOS I/Os and one input-only pin (GPA5). The GPA4 I/O is not available. The corresponding data direction register is TRISGPA. Setting a TRISGPA bit to 1 will make the corresponding PORTGPA pin an input (i.e., disable the output driver). Clearing a TRISGPA bit set to 0 will make the corresponding PORTGPA pin an output (i.e., enables output driver). The exception is GPA5, which is input-only and its TRISGPA bit will always read as '1'. Example 18-1 shows how to initialize an I/O port.

Reading the PORTGPA register reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations.

The TRISGPA register controls the PORTGPA pin output drivers, even when they are being used as analog inputs. The user must ensure the bits in the TRISGPA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'. If the pin is configured for a digital output (either port or alternate function), the TRISGPA bit must be cleared in order for the pin to drive the signal, and a read will reflect the state of the pin.

18.1.1 INTERRUPT-ON-CHANGE

Each PORTGPA pin is individually configurable as an interrupt-on-change pin. Control bits IOCA<7:5> and IOCA<3:0> enable or disable the interrupt function for each pin. The interrupt-on-change feature is disabled on a Power-on Reset. Reference **Section 15.0 "Interrupt-On-Change"** for more information.

18.1.2 WEAK PULL-UPS

PORTGPA <7:5> and PORTGPA <3:0> have an internal weak pull-up. Individual control bits can enable or disable the internal weak pull-ups (refer to Register 18-3). The weak pull-up is automatically turned off when the port pin is configured as an output or as an alternative function or on a Power-on Reset setting the RAPU bit in the OPTION_REG register. The weak pull-up on GPA5 is enabled when configured as MCLR pin by setting bit 5 in the CONFIG register, and disabled when GPA5 is an I/O. There is no software control of the MCLR pull-up.

18.1.3 ANSELA REGISTER

The ANSELA register is used to configure the input mode of an I/O pin to analog. Setting the appropriate ANSELA bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELA bits has no effect on digital output functions. A pin with TRISGPA cleared and ANSELx set will still operate as a digital output, but the input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELA bits default to the Analog mode after Reset. To use any pins as digital general-purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by the user's software.

18.1.4 PORTGPA FUNCTIONS AND OUTPUT PRIORITIES

Each PORTGPA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 18-1. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as ADC, are not shown in the priority lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELA register. Digital output functions may control the pin when it is in Analog mode with the priority shown in Table 18-1.

TABLE 18-1: PORTGPA OUTPUT PRIORITY

Pin Name	Function Priority(1)
GPA0	GPA0
	TS_OUT1
GPA1	GPA1
GPA2	GPA2
	T0CKI
	INT
GPA3	GPA3
	TS_OUT2
GPA5	GPA5 (input only)
	$\overline{\text{MCLR}}$
	TEST_EN
GPA6	GPA6
	CCD1
	ICSPDAT
GPA7	GPA7
	SCL

Note 1: Output function priority listed from lowest to highest.

REGISTER 18-1: PORTGPA: PORTGPA REGISTER

R/W-x	R/W-x	R-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
GPA7	GPA6	GPA5	—	GPA3	GPA2	GPA1	GPA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **GPA<7:6>:** General-Purpose I/O pin
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

bit 5 **GPA5/MCLR/TEST_EN:** General-Purpose input pin

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **GPA<3:0>:** General-Purpose I/O pin
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

REGISTER 18-2: TRISGPA: PORTGPA TRI-STATE REGISTER

R/W-1	R/W-1	R-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1
TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **TRISA<7:6>:** PORTGPA Tri-State Control bit
1 = PORTGPA pin configured as an input (tri-stated)
0 = PORTGPA pin configured as an output

bit 5 **TRISA5:** GPA5 Port Tri-State Control bit
This bit is always ‘1’ as GPA5 is an input only

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **TRISA<3:0>:** PORTGPA Tri-State Control bit
1 = PORTGPA pin configured as an input (tri-stated)
0 = PORTGPA pin configured as an output

REGISTER 18-3: WPUGPA: WEAK PULL-UP PORTGPA REGISTER (Note 1)

R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1
WPUA7	WPUA6	WPUA5	—	WPUA3	WPUA2	WPUA1	WPUA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **WPUA<7:5>:** Weak Pull-Up Register bit(2)
1 = Pull-up enabled
0 = Pull-up disabled

bit 4 **Unimplemented:** Read as '0'

bit 3-0 **WPUA<3:0>:** Weak Pull-Up Register bit
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: The weak pull-up device is enabled only when the global $\overline{\text{RAPU}}$ bit is enabled, the pin is in input mode (TRISGPA = 1) and the individual WPUA bit is enabled (WPUA = 1), and the pin is not configured as an analog input.

2: GPA5 weak pull-up is also enabled when the pin is configured as $\overline{\text{MCLR}}$ in the CONFIG register.

REGISTER 18-4: ANSELA: ANALOG SELECT GPA REGISTER

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **ANSA<3:0>:** Analog Select GPA Register bit
1 = Analog input. Pin is assigned as analog input.(1)
0 = Digital I/O. Pin is assigned to port or special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change if available. The corresponding TRISA bit must be set to Input mode in order to allow external control of the voltage on the pin.

TABLE 18-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTGPA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0	128
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	85
PORTGPA	GPA7	GPA6	GPA5	—	GPA3	GPA2	GPA1	GPA0	127
TRISGPA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	127
WPUGPA	WPUA7	WPUA6	WPUA5	—	WPUA3	WPUA2	WPUA1	WPUA0	128

Legend: — = unimplemented locations read as '0'. Shaded cells are not used by PORTGPA.

18.2 PORTGPB and TRISGPB Registers

Due to special function pin requirements, a limited number of the PORTGPB I/Os are utilized. On the 28-pin QFN MCP19214, GPB0, GPB4 and GPB5 are implemented. The 32-pin QFN MCP19215 has three additional general-purpose PORTGPB I/O pins, GPB1, GPB6 and GPB7. The corresponding data direction register is TRISGPB. Setting a TRISGPB bit to 1 will make the corresponding PORTGPB pin an input (i.e., disable the output driver). Clearing a TRISGPB bit to 0 will make the corresponding PORTGPB pin an output (i.e., enable the output driver). Example 18-1 shows how to initialize an I/O port.

Some pins for PORTGPB are multiplexed with an alternate function for the peripheral or a clock function. In general, when a peripheral or clock function is enabled, that pin may not be used as a general-purpose I/O pin.

Reading the PORTGPB register reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations.

The TRISGPB register controls the PORTGPB pin output drivers, even when they are being used as analog inputs. It is recommended that the user ensures the bits in the TRISGPB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'. If the pin is configured for a digital output (either port or alternate function), the TRISGPB bit must be cleared in order for the pin to drive the signal and a read will reflect the state of the pin.

18.2.1 INTERRUPT-ON-CHANGE

Each PORTGPB pin is individually configurable as an interrupt-on-change pin. Control bits IOCB<7:4> and IOCB<1:0> enable or disable the interrupt function for each pin. The interrupt-on-change feature is disabled on a Power-on Reset. Reference **Section 15.0 "Interrupt-On-Change"** for more information.

18.2.2 WEAK PULL-UPS

Each of the PORTGPB pins has an individually configurable internal weak pull-up. Control bits WPUB<7:4> and WPUB<1:0> enable or disable each pull-up (refer to Register 18-7). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-On Reset by the $\overline{\text{RAPU}}$ bit in the OPTION_REG register.

18.2.3 ANSELB REGISTER

The ANSELB register is used to configure the input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no effect on the digital output functions. A pin with TRISGPB clear and ANSELB set will still operate as a digital output, but the input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

The TRISGPB register controls the PORTGPB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISGPB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

Note:	The ANSELB bits default to the Analog mode after Reset. To use any pins as digital general-purpose or peripheral inputs, the corresponding ANSELB bits must be initialized to '0' by the user's software.

18.2.4 PORTGPB FUNCTIONS AND OUTPUT PRIORITIES

Each PORTGPB pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 18-3. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as ADC, and some digital input functions are not included in the list below. These inputs are active when the I/O pin is set for Analog mode using the ANSELB register. Digital output functions may control the pin when it is in Analog mode, with the priority shown in Table 18-3.

TABLE 18-3: PORTGPB OUTPUT PRIORITY

Pin Name	Function Priority(1)
GPB0	GPB0
	SDA
GPB1	GPB1 (**MCP19215** only)
	RX
GPB4	GPB4
	ICSPDAT
GPB5	GPB5
	ICSPCLK
GPB6	GPB6 (**MCP19215** only)
	TX/CK
GPB7	GPB7 (**MCP19215** only)
	CCD2

Note 1: Output function priority listed from lowest to highest.

REGISTER 18-5: PORTGPB: PORTGPB REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0	R/W-x	R/W-x
GPB7(1)	GPB6(1)	GPB5	GPB4	—	—	GPB1(1)	GPB0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-4 **GPB<7:4>:** General-Purpose I/O Pin bit
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1-0 **GPB<1:0>:** General-Purpose I/O Pin bit
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

Note 1: MCP19215 only.

REGISTER 18-6: TRISGPB: PORTGPB TRI-STATE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	R/W-1	R/W-1
TRISB7(1)	TRISB6(1)	TRISB5	TRISB4	—	—	TRISB1(1)	TRISB0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **TRISB<7:4>**: PORTGPB Tri-State Control bit
1 = PORTGPB pin configured as an input (tri-stated)
0 = PORTGPB pin configured as an output

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **TRISB<1:0>**: PORTGPB Tri-State Control bit
1 = PORTGPB pin configured as an input (tri-stated)
0 = PORTGPB pin configured as an output

Note 1: MCP19215 only.

REGISTER 18-7: WPUGPB: WEAK PULL-UP PORTGPB REGISTER (Note 1)

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	R/W-1	R/W-1
WPUB7(2)	WPUB6(2)	WPUB5	WPUB4	—	—	WPUB1(2)	WPUB0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **WPUB<7:4>**: Weak Pull-up Register bit
1 = Pull-up enabled
0 = Pull-up disabled

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **WPUB<1:0>**: Weak Pull-up Register bit
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: The weak pull-up device is enabled only when the global $\overline{\text{RAPU}}$ bit is enabled, the pin is in input mode (TRISGPB = 1) and the individual WPUB bit is enabled (WPUB = 1), and the pin is not configured as an analog input.
2: MCP19215 only.

REGISTER 18-8: ANSELB: ANALOG SELECT GPB REGISTER

U-0	R/W-1	R/W-1	R/W-1	U-0	U-0	R/W-1	U-0
—	ANSB6(1)	ANSB5	ANSB4	—	—	ANSB1(1)	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **ANSB<6:4>**: Analog Select GPB Register bit
1 = Analog input. Pin is assigned as analog input(2).
0 = Digital I/O. Pin is assigned to port or special function.

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **ANSB1**: Analog Select GPB Register bit
1 = Analog input. Pin is assigned as analog input(2).
0 = Digital I/O. Pin is assigned to port or special function.

bit 0 **Unimplemented:** Read as '0'

Note 1: MCP19215 only.

2: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

TABLE 18-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTGPB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	ANSB6(1)	ANSB5	ANSB4	—	—	ANSB1(1)	—	132
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	85
PORTGPB	GPB7(1)	GPB6(1)	GPB5	GPB4	—	—	GPB1(1)	GPB0	130
TRISGPB	TRISB7(1)	TRISB6(1)	TRISB5	TRISB4	—	—	TRISB1(1)	TRISB0	131
WPUGPB	WPUB7(1)	WPUB6(1)	WPUB5	WPUB4	—	—	WPUB1(1)	WPUB0	131

Legend: — = unimplemented locations, read as '0'. Shaded cells are not used by the PORTGPB register.

Note 1: MCP19215 only.

19.0 POWER-DOWN MODE (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running, if enabled for operation during Sleep.
2. $\overline{PD}$ bit in the STATUS register is cleared.
3. $\overline{TO}$ bit in the STATUS register is set.
4. CPU clock is disabled.
5. The ADC is inoperable.
6. I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).
7. Resets other than WDT are not affected by Sleep mode.
8. Power to the internal analog circuitry is removed during Sleep mode.
9. POR level changes to 1.2V typical.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating.
- External circuitry sinking current from I/O pins.
- Internal circuitry sourcing current from I/O pins.
- Current draw from pins with internal weak pull-ups.
- Modules using Timer1 oscillator.

I/O pins that are high-impedance inputs should be pulled to V_{DD} or GND externally to avoid switching currents caused by floating inputs.

The SLEEP instruction removes power from the analog circuitry. Internal AV_{DD} voltage is shut down to minimize current draw in Sleep mode and to maintain a shutdown current of 30 µA typical. The 5V LDO (V_{DD}) voltage drops to 3V (typical) in Sleep mode if bit LDO_LV from register PE1 is cleared. If this bit is set, the voltage during sleep is maintained at 5V (typical). External current draw from the 5V LDO (V_{DD}) should be limited to less than 200 µA. Loads drawing more than 200 µA externally during Sleep mode risk loading down the V_{DD} voltage and tripping POR. A POR event during Sleep mode will wake the device from Sleep. The enable state of the analog circuitry does not change with the execution of the SLEEP instruction.

19.1 Wake-Up from Sleep

The device can wake up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin, if enabled
2. POR Reset
3. Watchdog Timer, if enabled
4. Any external interrupt
5. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information)

The first two events will cause a device reset. The last three events are considered a continuation of program execution. To determine whether a device reset or wake-up event occurred, refer to **Section 12.7 "Determining the Cause of a Reset"**.

The following peripheral interrupts can wake the device from Sleep:

1. Interrupt-on-change
2. External Interrupt from INT pin

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is enabled, the device executes the instruction after the SLEEP instruction and will then call the Interrupt Service Routine. In cases where the execution of the instruction following SLEEP is not desirable, the user should have an NOP after the SLEEP instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

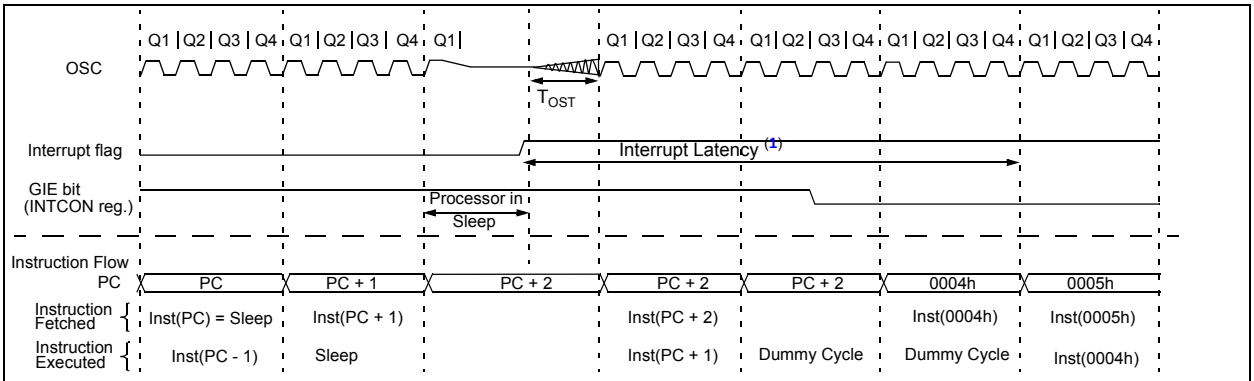
19.1.1 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction
  - SLEEP instruction will execute as an NOP
  - WDT and WDT prescaler will not be cleared
  - $\overline{TO}$ bit in the STATUS register will not be set
  - $\overline{PD}$ bit in the STATUS register will not be cleared
- If the interrupt occurs **during or after** the execution of a SLEEP instruction
  - SLEEP instruction will be completely executed
  - Device will immediately wake up from Sleep
  - WDT and WDT prescaler will be cleared
  - $\overline{TO}$ bit in the STATUS register will be set
  - $\overline{PD}$ bit in the STATUS register will be cleared

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as an NOP.

FIGURE 19-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
IOCA	IOCA7	IOCA6	IOCA5	—	IOCA3	IOCA2	IOCA1	IOCA0	114
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	IOCB1	IOCB0	114
PIE1	OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	103
PIE2	IVGD1IE	—	IVGD2IE	—	VDDUVIE	DRUVIE	OVLOIE	UVLOIE	104
PIR1	OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	106
PIR2	IVGD1IF	—	IVGD2IF	—	VDDUVIF	DRUVIF	OVLOIF	UVLOIF	107
STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	77

Legend: — = unimplemented, read as '0'. Shaded cells are not used in Power-Down mode.

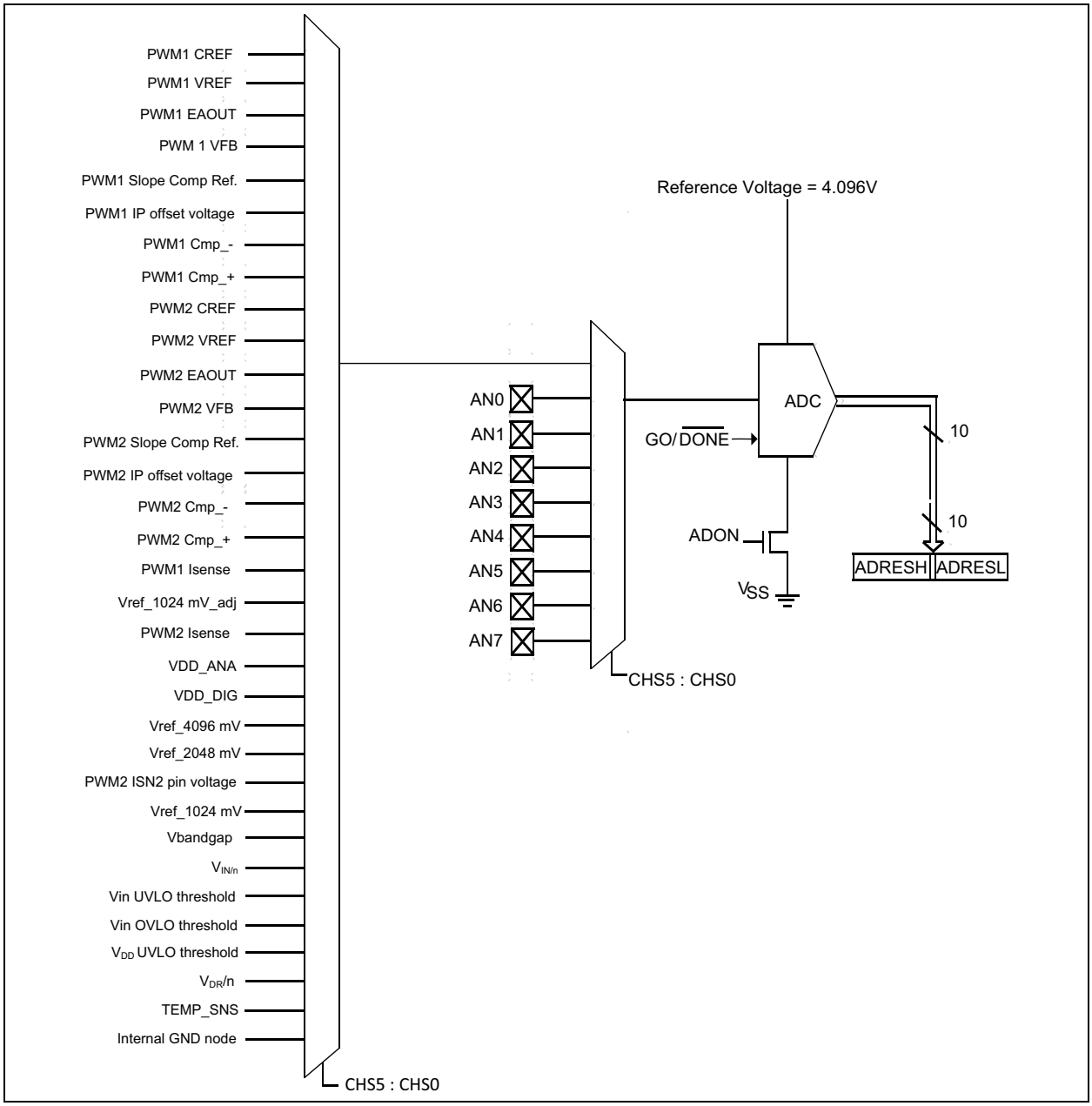
20.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs which are multiplexed into a single sample-and-hold circuit. The output of the sample-and-hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the right justified conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 20-1 shows the block diagram of the ADC.

The internal band gap supplies the voltage reference to the ADC.

Note:	Once V_{IN} is greater than $V_{DD} + V_{DROPOUT}$, V_{DD} is in regulation, allowing A/D readings to be accurate.

FIGURE 20-1: ADC BLOCK DIAGRAM



20.1 ADC Configuration

When configuring and using the ADC, the following functions must be considered:

- Port configuration
- Channel selection
- ADC conversion clock source
- Interrupt control
- Result formatting

20.1.1 PORT CONFIGURATION

The ADC is used to convert analog signals into a corresponding digital representation. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 18.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

20.1.2 CHANNEL SELECTION

There are up to 29 channel selections available for the MCP19214 and 31 channels for the MCP19215:

- AN<4:0> pins
- AN<7:5> pins (**MCP19215** Only)
- V_{IN}: 1/16 of the input voltage (V_{IN})
- Reference voltages for current and voltage regulating loops
- Internal references: 4096 mV, 2048 mV and 1024 mV
- V_{BGR}: band gap reference
- VFB pins voltages
- Outputs of the 10X current sense amplifiers
- The inputs of the PWM comparators
- Pedestal voltage
- IP_ADJ: I_P after pedestal and offset adjust
- IP_OFF_REF: I_P offset reference
- V_{DR}: V_{DR} * 0.229V/V
- TEMP_SNS: analog voltage representing internal temperature (refer to Equation 25-1)
- SLPCMP_REF: slope compensation reference

The CHS<4:0> bits in the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 20.2 "ADC Operation"** for more information.

20.1.3 ADC CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits in the ADCON1 register. There are five possible clock options:

- $F_{OSC}/8$
- $F_{OSC}/16$
- $F_{OSC}/32$
- $F_{OSC}/64$
- F_{RC} (clock derived from internal oscillator with a divisor of 16)

The time to complete one-bit conversion is defined as T_{AD}. One full 10-bit conversion requires 11 T_{AD} periods, as shown in Figure 20-2.

For a correct conversion, the appropriate T_{AD} specification must be met. Refer to the A/D conversion requirements in **Section 4.0 "Electrical Characteristics"** for more information. Table 20-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the F_{RC}, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

TABLE 20-1: ADC CLOCK PERIOD (T_{AD}) VS. DEVICE OPERATING FREQUENCIES

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})
ADC Clock Source	**ADCS<2:0>**	**8 MHz**
$F_{OSC}/8$	001	1.0 µs(1)
$F_{OSC}/16$	101	2.0 µs
$F_{OSC}/32$	010	4.0 µs
$F_{OSC}/64$	110	8.0 µs(2)
F_{RC}	x11	2.0-6.0 µs(3, 4)

Legend: Shaded cells are outside of recommended range.

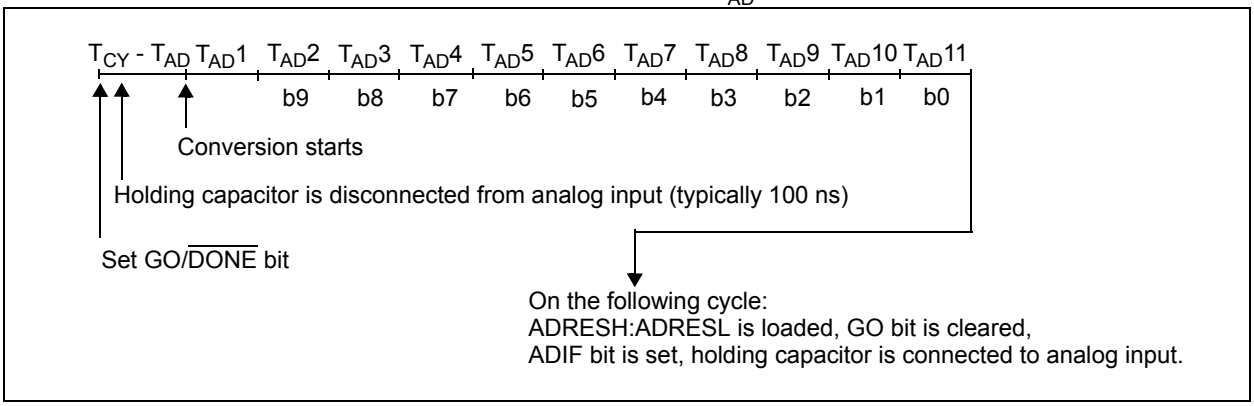
Note 1: These values violate the minimum required T_{AD} time.

2: For faster conversion times, the selection of another clock source is recommended.

3: The F_{RC} source has a typical T_{AD} time of 4 µs for V_{DD} > 3.0V.

4: The F_{RC} clock source is only recommended if the conversion will be performed during Sleep.

FIGURE 20-2: ANALOG-TO-DIGITAL CONVERSION T_{AD} CYCLES



20.1.4 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an analog-to-digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

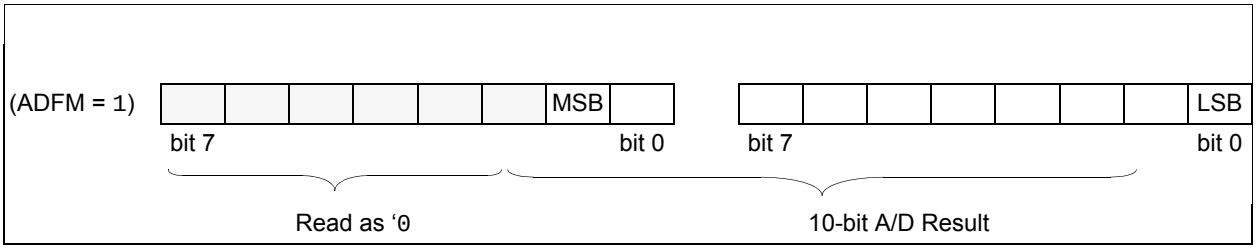
Note 1:	The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
2:	The ADC operates during Sleep only when the F_{RC} oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake up from Sleep and resume in-line code execution, the GIE and PEIE bits in the INTCON register must be disabled. If the GIE and PEIE bits in the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

20.1.5 RESULT FORMATTING

The 10-bit A/D conversion result is supplied in right justified format only.

FIGURE 20-3: 10-BIT A/D RESULT FORMAT



20.2 ADC Operation

20.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit in the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit in the ADCON0 register to a '1' will start the analog-to-digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 20.2.5 "A/D Conversion Procedure"**.

20.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH:ADRESL registers with new conversion result

20.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH:ADRESL registers will not be updated with the partially complete analog-to-digital conversion sample. Instead, the ADRESH:ADRESL register pair will retain the value of the previous conversion. Additionally, two ADC clock cycles are required before another acquisition can be initiated. Following the delay, an input acquisition is automatically started on the selected channel.

Note:	A device reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

20.2.4 ADC OPERATION DURING SLEEP

The ADC is not operational during Sleep mode. The AV_{DD} 4V reference has been removed to minimize Sleep current.

20.2.5 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an analog-to-digital conversion:

1. Configure Port:
   - Disable pin output driver (refer to the TRISGPx registers)
   - Configure pin as analog (refer to the ANSELx registers)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1:	The global interrupt can be disabled if the user is attempting to wake up from Sleep and resume in-line code execution.
2:	Refer to **Section 20.4 "A/D Acquisition Requirements"**.

EXAMPLE 20-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Frc clock and AN0 input.
;
;Conversion start & polling for completion ;
are included.
;
BANKSELADCON1;
MOVLWB'01110000';Frc clock
MOVWFADCON1;
BANKSELTRISGPA;
BSF TRISGPA,0;Set GPA0 to input
BANKSELANSELA;
BSF ANSELA,0;Set GPA0 to analog
BANKSELADCON0;
MOVLWB'01100001';Select channel AN0
MOVWFADCON0;Turn ADC On
CALLSampleTime;Acquisiton delay
BSF ADCON0,1;Start conversion
BTFSCADCON0,1;Is conversion done?
GOTO$-1 ;No, test again
BANKSELADRESH;
MOVFADRESH,W;Read upper 2 bits
MOVWFRESULTHI;store in GPR space
BANKSELADRESL;
MOVFADRESL,W;Read lower 8 bits
MOVWFRESULTLO;Store in GPR space
```

20.3 ADC Register Definitions

The following registers are used to control the operation of the ADC:

REGISTER 20-1: ADCON0: ANALOG-TO-DIGITAL CONTROL REGISTER 0 (ADDRESS: 1Eh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHS5	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-2 **CHS<5:0>:** Analog Channel Select bits

000000 = PWM #1 EA1 (Current loop) reference voltage
000001 = PWM #1 EA2 (Voltage loop) reference voltage
000010 = PWM #1 Error Amplifier output voltage
000011 = PWM #1 VFB1 pin voltage
000100 = PWM #1 Slope Compensation reference voltage
000101 = PWM #1 IP1 signal offset reference voltage
000110 = PWM #1 PWM Comparator negative input
000111 = PWM #1 PWM Comparator positive input
001000 = PWM #2 EA1 (Current loop) reference voltage
001001 = PWM #2 EA2 (Voltage loop) reference voltage
001010 = PWM #2 Error Amplifier output voltage
001011 = PWM #2 VFB2 pin voltage
001100 = PWM #2 Slope Compensation reference voltage
001101 = PWM #2 IP2 signal offset reference voltage
001110 = PWM #2 PWM Comparator negative input
001111 = PWM #2 PWM Comparator positive input
010000 = PWM #1 A1 Current Sense Amplifier output
010001 = 1024 mV reference adjust
010010 = PWM #2 A1 Current Sense Amplifier output
010011 = Internal V_{DD} for analog circuitry
010100 = Internal V_{DD} for digital circuitry
010101 = 4096 mV Reference Voltage
010110 = 2048 mV Reference Voltage
010111 = PWM #2 ISN2 pin voltage
011000 = 1024 mV Reference Voltage
011001 = Bandgap Reference Voltage
011010 = V_{IN}/n voltage
011011 = V_{IN} UVLO Threshold
011100 = V_{IN} OVLO Threshold
011101 = V_{DD} UVLO voltage
011110 = V_{DR}/n (MOSFET drivers supply voltage)
011111 = TEMP_SNS temperature sensor voltage measurement
100000 = Internal GND node
111000 = AN0 analog input
111001 = AN1 analog input
111010 = AN2 analog input
111011 = AN3 analog input
111100 = AN4 analog input
111101 = AN5 analog input
111110 = AN6 analog input
111111 = AN7 analog input

REGISTER 20-1: ADCON0: ANALOG-TO-DIGITAL CONTROL REGISTER 0 (ADDRESS: 1Eh) (CONTINUED)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Conversion Status bit
1 = A/D converter module is operating
0 = A/D converter is shut off and consumes no operating current

REGISTER 20-2: ADCON1: A/D CONTROL REGISTER 1

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCS2	ADCS1	ADCS0	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = Reserved
001 = $F_{OSC}/8$
010 = $F_{OSC}/32$
x11 = F_{RC} (clock derived from internal oscillator with a divisor of 16)
100 = Reserved
101 = $F_{OSC}/16$
110 = $F_{OSC}/64$

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 20-3: ADRESH: ADC RESULT REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	R-x	R-x
—	—	—	—	—	—	ADRES9	ADRES8
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1-0 **ADRES<9:8>**: Most Significant A/D Results

REGISTER 20-4: ADRESL: ADC RESULT REGISTER LOW

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **ADRES<7:0>**: Least Significant A/D results

20.4 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 20-4. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}) (refer to Figure 20-4). **The maximum recommended impedance for analog sources is 10 kΩ.**

As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 20-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 20-1: ACQUISITION TIME EXAMPLE

Assumptions: *Temperature = +50°C and external impedance of 10 kΩ 5.0V* V_{DD}

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\,\mu s + T_C + [(\text{Temperature} - 25°C)(0.05\,\mu s/°C)]$$

The value for T_C *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD} \quad ;[1]\ V_{CHOLD} \text{ charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \quad ;\text{combining [1] and [2]}$$

Note: *Where n = number of bits of the ADC.*

Solving for T_C*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/2047)$$
$$= -10\ pF(1\ k\Omega + 7\ k\Omega + 10\ k\Omega)\ ln(0.0004885)$$
$$= 1.37\mu s$$

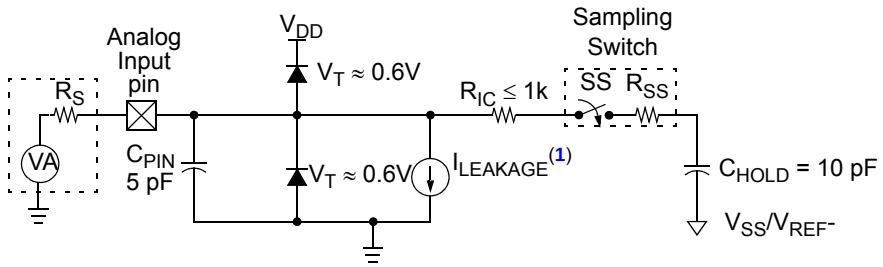
Therefore:

$$T_{ACQ} = 2\,\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.67\,\mu s$$

Note 1: The charge holding capacitor (C_{HOLD}) is not discharged after each conversion.

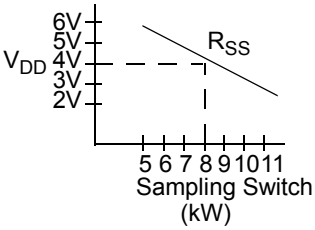
2: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

FIGURE 20-4: ANALOG INPUT MODEL



Legend:

C_{HOLD}	=	Sample/Hold Capacitance
C_{PIN}	=	Input Capacitance
$I_{LEAKAGE}$	=	Leakage current at the pin due to various junctions
R_{IC}	=	Interconnect Resistance
R_{SS}	=	Resistance of Sampling Switch
SS	=	Sampling Switch
V_T	=	Threshold Voltage



Note 1: Refer to **Section 4.0 "Electrical Characteristics"**.

FIGURE 20-5: ADC TRANSFER FUNCTION

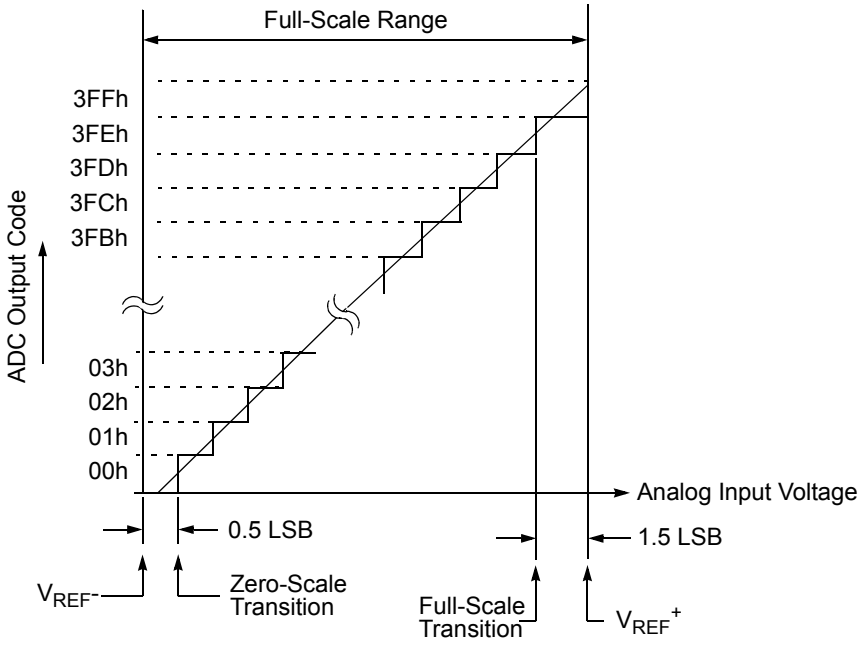


TABLE 20-2: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	CHS5	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	139
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	140
ADRESH	—	—	—	—	—	—	ADRES9	ADRES8	140
ADRESL	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0	141
ANSELA	—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0	128
ANSELB	—	ANSB6	ANSB5	ANSB4	—	—	ANSB1	—	132
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
PIE1	OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	103
PIR1	OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	106
TRISGPA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	127
TRISGPB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	TRISB1	TRISB0	131

Legend: — = unimplemented, read as '0'. Shaded cells are not used for ADC module.

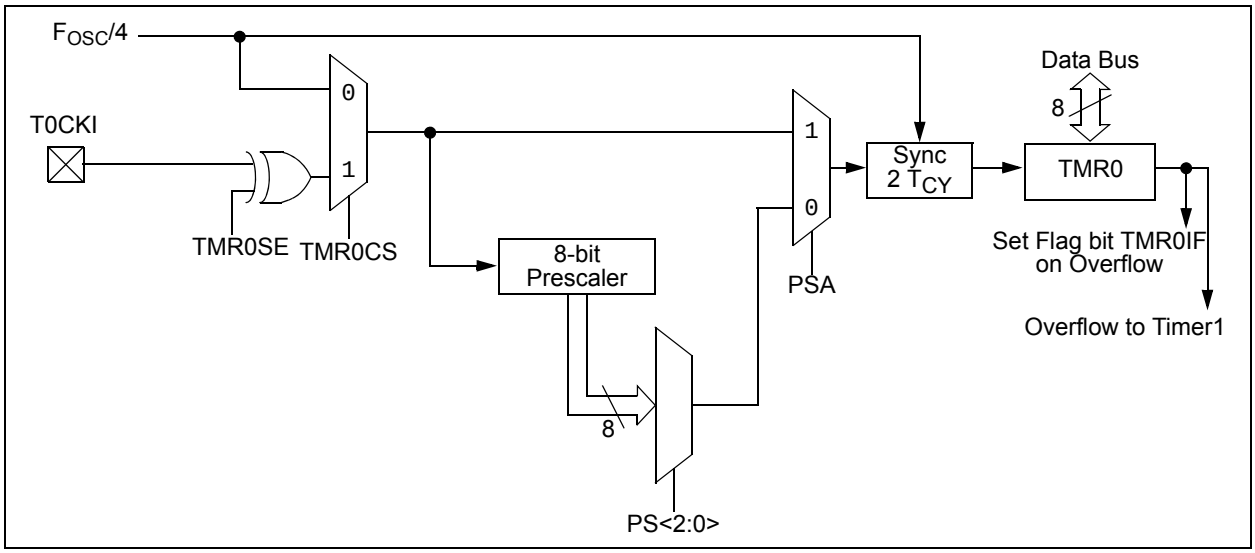
21.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow

Figure 21-1 is a block diagram of the Timer0 module.

FIGURE 21-1: TIMER0 BLOCK DIAGRAM



21.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

21.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-Bit Timer mode is selected by clearing the T0CS bit in the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note:	The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

21.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin. The incrementing edge is determined by the T0SE bit in the OPTION_REG register.

8-Bit Counter mode using the T0CKI pin is selected by setting the T0CS bit in the OPTION_REG register to ‘1’.

21.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with either Timer0 or the Watchdog Timer (WDT), but not both simultaneously. The prescaler assignment is controlled by the PSA bit in the OPTION_REG register. To assign the prescaler to Timer0, the PSA bit must be cleared to ‘0’.

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits in the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit in the OPTION_REG register.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

21.1.4 SWITCHING PRESCALER BETWEEN TIMER0 AND WDT MODULES

As a result of having the prescaler assigned to either Timer0 or the WDT, it is possible to generate an unintended device reset when switching prescaler values. When changing the prescaler assignment from Timer0 to the WDT module, the instruction sequence shown in Example 21-1 must be executed.

EXAMPLE 21-1: CHANGING PRESCALER (TIMER0 → WDT)

```
BANKSELTMR0;
CLRWDT  ;Clear WDT
CLRFTMR0;Clear TMR0 and
        ;prescaler
BANKSELOPTION_REG;
BSF OPTION_REG,PSA;Select WDT
CLRWDT  ;
        ;
MOVLWb'11111000';Mask prescaler
ANDWFOPTION_REG,W;bits
IORLWb'00000101';Set WDT prescaler
MOVWFOPTION_REG;to 1:32
```

When changing the prescaler assignment from the WDT to the Timer0 module, the following instruction sequence must be executed (refer to Example 21-2).

EXAMPLE 21-2: CHANGING PRESCALER (WDT → TIMER0)

```
CLRWDT  ;Clear WDT and
        ;prescaler
BANKSELOPTION_REG;
MOVLWb'11110000';Mask TMR0 select and
ANDWFOPTION_REG,W;prescaler bits
IORLWb'00000011';Set prescale to 1:16
MOVWFOPTION_REG;
```

21.1.5 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The T0IF interrupt flag bit in the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The T0IF bit can only be cleared in software. The Timer0 interrupt enable is the T0IE bit in the INTCON register.

Note:	The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

21.1.6 USING TIMER0 WITH AN EXTERNAL CLOCK

When Timer0 is in Counter mode, the synchronization of the T0CKI input and the Timer0 register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in **Section 4.0 "Electrical Characteristics"**.

21.1.7 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	IOCIE	T0IF	INTF	IOCIF	102
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	85
TMR0	Timer0 Module Register								145*
TRISGPA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	127

Legend: — = Unimplemented locations, read as '0'. Shaded cells are not used by the Timer0 module.

* Page provides register information.

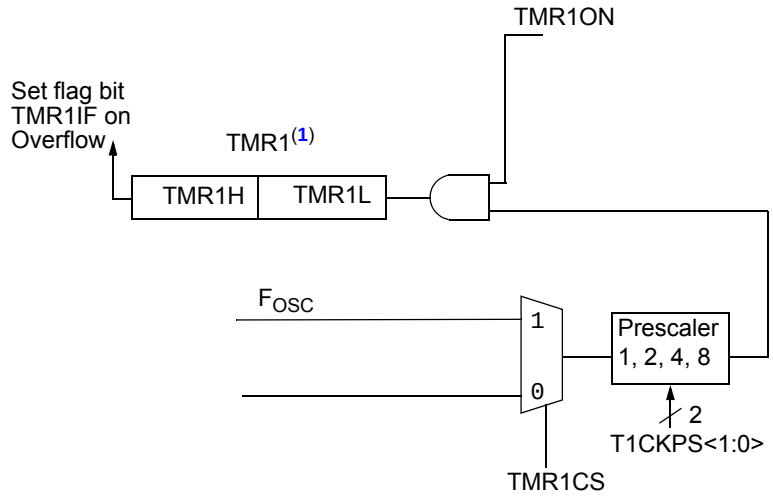
22.0 TIMER1 MODULE

The Timer1 module is a 16-bit timer with the following features:

- 16-bit timer register pair (TMR1H:TMR1L)
- Readable and Writable (both registers)
- Selectable internal clock source
- 2-bit prescaler
- Interrupt on overflow

Figure 22-1 is a block diagram of the Timer1 module.

FIGURE 22-1: TIMER1 BLOCK DIAGRAM



Note 1: TMR1 register increments on rising edge.

22.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing timer which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter. The timer is incremented on every instruction cycle.

Timer1 is enabled by configuring the TMR1ON bit in the T1CON register. Table 22-1 displays the Timer1 enable selections.

22.2 Clock Source Selection

The TMR1CS bit in the T1CON register is used to select the clock source for Timer1. Table 22-1 displays the clock source selections.

22.2.1 INTERNAL CLOCK SOURCE

The TMR1H:TMR1L register pair will increment on multiples of F_{OSC} or $F_{OSC}/4$ as determined by the Timer1 prescaler.

As an example, when the F_{OSC} internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle.

TABLE 22-1: CLOCK SOURCE SELECTIONS

TMR1CS	Clock Source
1	8 MHz system clock (F_{OSC})
0	2 MHz instruction clock ($F_{OSC}/4$)

22.3 Timer1 Prescaler

Timer1 has four prescaler options, allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits in the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

22.4 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit in the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit in the T1CON register
- TMR1IE bit in the PIE1 register
- PEIE bit in the INTCON register
- GIE bit in the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note:	The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

22.5 Timer1 in Sleep

Unlike other standard mid-range Timer1 modules, the MCP19214/5 Timer1 module only clocks from an internal system clock, and thus cannot run during Sleep mode, nor can it be used to wake the device from this mode.

22.6 Timer1 Control Register

The Timer1 Control (T1CON) register, shown in Register 22-1, is used to control Timer1 and select the various features of the Timer1 module.

REGISTER 22-1: T1CON: TIMER1 CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	—	—	TMR1CS	TMR1ON
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
- 11 = 1:8 Prescale value
- 10 =1:4 Prescale value
- 01 = 1:2 Prescale value
- 00 = 1:1 Prescale value

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **TMR1CS:** Timer1 Clock Source Control bit
- 1 = 8 MHz system clock (F_{OSC})
- 0 = 2 MHz instruction clock ($F_{OSC/4}$)

bit 0 **TMR1ON:** Timer1 On bit
- 1 = Enables Timer1
- 0 = Stops Timer1, Clears Timer1 gate flip-flop

TABLE 22-2: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
PIE1	OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	103
PIR1	OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	106
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								147*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								147*
T1CON	—	—	T1CKPS1	T1CKPS0	—	—	TMR1CS	TMR1ON	148

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

* Page provides register information.

NOTES:

23.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)

Refer to Figure 23-1 for a block diagram of Timer2.

23.1 Timer2 Operation

The clock input to the Timer2 module is the system clock (F_{OSC}). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, TMR2 is reset to 00h on the next increment cycle.

The match output of the Timer2/PR2 comparator is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

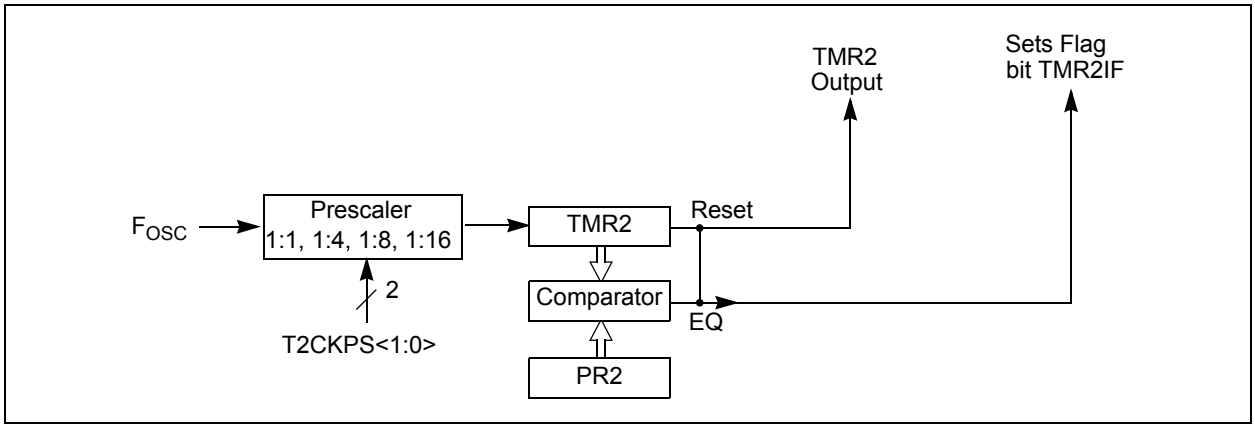
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to a '1'. Timer2 is turned off by clearing the TMR2ON bit to a '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The prescaler counter are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device reset occurs (Power-On Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-Out Reset).

Note:	TMR2 is not cleared when T2CON is written.

FIGURE 23-1: TIMER2 BLOCK DIAGRAM



23.2 Timer2 Control Register

REGISTER 23-1: T2CON: TIMER2 CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
10 = Prescaler is 8
11 = Prescaler is 16

TABLE 23-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
PIE1	OTIE	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	103
PIR1	OTIF	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	106
PR2	Timer2 Module Period Register								151*
T2CON	—	—	—	—	—	TMR2ON	T2CKPS1	T2CKPS0	152
TMR2	Holding Register for the 8-bit TMR2 Time Base								151*

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used for Timer2 module.
* Page provides register information.

24.0 ENHANCED PWM MODULE

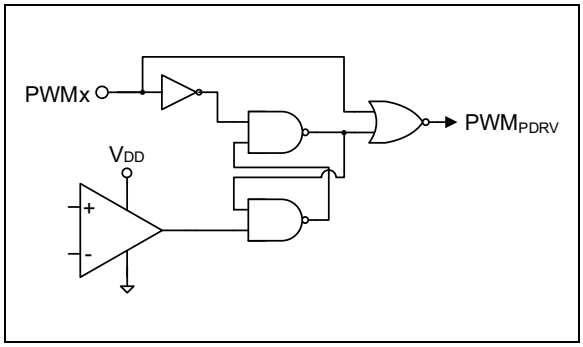
The PWM module implemented on the MCP19214/5 is a scaled-down version of the Capture/Compare/PWM (CCP) module found in standard mid-range microcontrollers. The module only features the PWM module, which is slightly modified from standard mid-range microcontrollers. In the MCP19214/5, the PWM module is used to generate the system clock or system oscillator. This system clock can control the MCP19214/5 switching frequency, as well as set the maximum allowable duty cycle. The PWM module does not continuously adjust the duty cycle to control the output voltage. This is accomplished by the analog control loop and associated circuitry.

24.1 Standard Pulse-Width Modulation Mode

The CCP will only function in PWM mode. The PWM signal is used to set the operating frequency, the maximum allowable duty cycle and the phase shift for both channels of the MCP19214/5. Figure 24-1 is a snippet of the MCP19214/5 block diagram showing the PWM signal from the CCP module.

Figure 24-2 shows a simplified block diagram of the CCP module in PWM mode. The first channel (PWM #1) is considered to be the MASTER channel while the second channel (PWM #2) is considered to be the SLAVE channel. The switching frequency is identical for both channels, but the phase shift and the maximum duty cycle can be independently adjusted.

FIGURE 24-1: MCP19214/5 SNIPPET SHOWING SYSTEM CLOCK FROM CCP MODULE



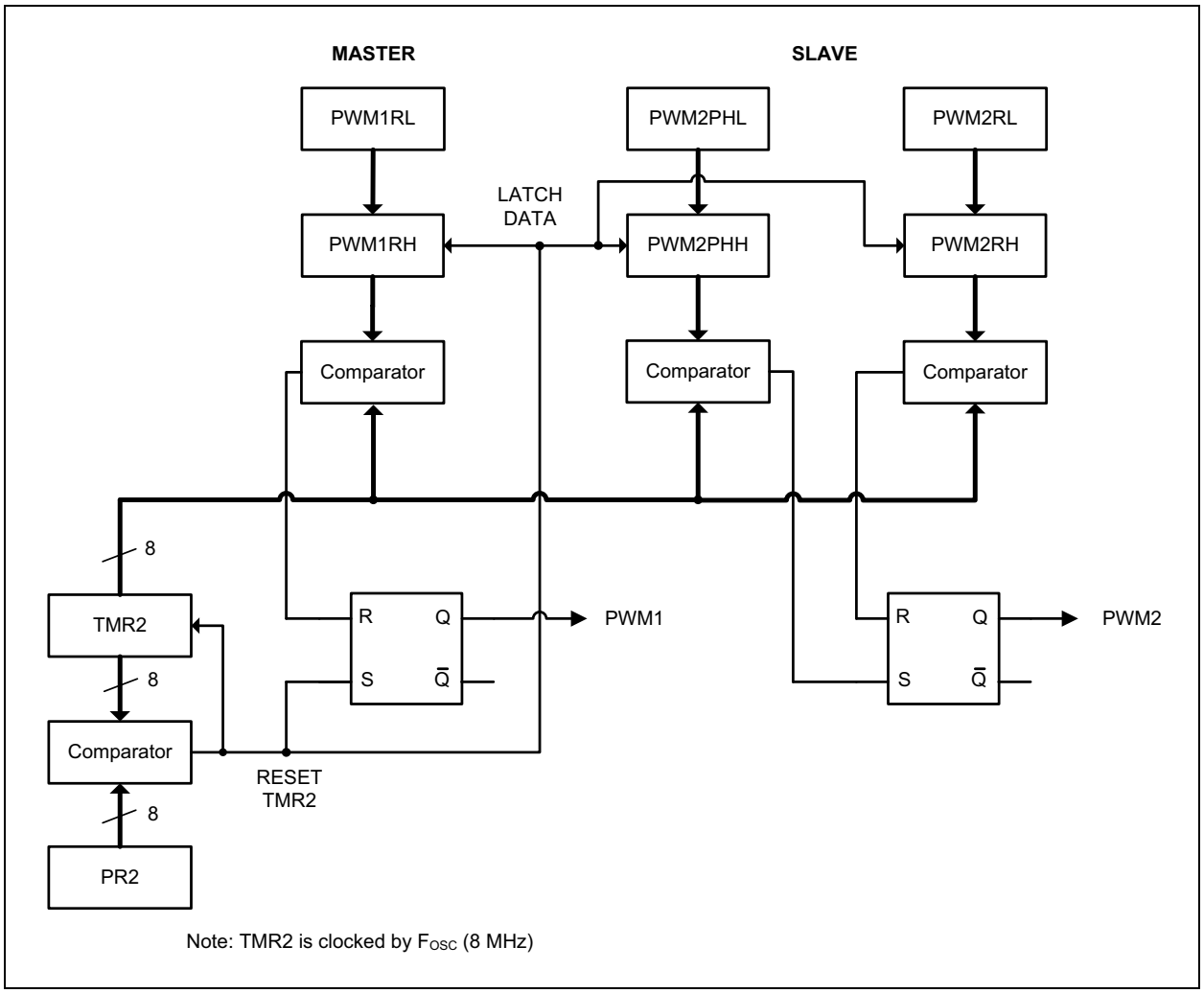
The PWM1 and PWM2 signals act as the clock signals for the analog PWM controllers of MCP19214/5. These signals will set the switching frequency of the converter, the maximum allowable duty cycle and the phase shift between channels.

The programmed maximum duty cycle is not adjusted on a cycle-by-cycle basis to control the MCP19214/5 system output. The required duty cycle ($D_{PDRVxON}$) to control the output is adjusted by the MCP19214/5 analog control loop and associated circuitry. D_{PWMx} does however set the maximum allowable $D_{PDRVxON}$.

EQUATION 24-1:

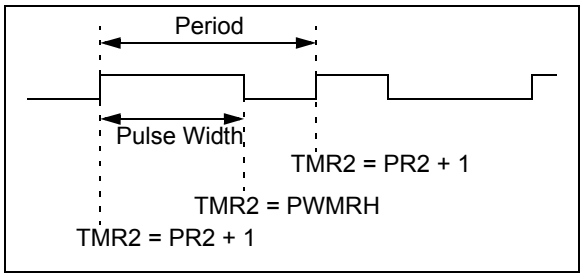
$$D < D_{PWMx}$$

FIGURE 24-2: SIMPLIFIED PWM BLOCK DIAGRAM



A PWMx output (Figure 24-2) has a time base (period) and a time when the output stays high (pulse width). The frequency of the PWMx is the inverse of the period (1/period).

FIGURE 24-3: PWM OUTPUT



24.1.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period (in seconds) can be calculated using the following equation:

EQUATION 24-2:

$$PWM_{PERIOD} = [(PR2) + 1] \times T_{OSC} \times (T2_{PRESCALE\ VALUE})$$

When TMR2 is equal to PR2, the following events occur on the next increment cycle:

- TMR2 is cleared.
- The PWM duty cycle is latched from PWM1RL into PWM1RH. This will set the maximum duty cycle of the PWM channel 1.
- The PWM duty cycle is latched from PWM2RL into PWM2RH. This will set the maximum duty cycle of the PWM channel 2.
- The phase shift is latched from PWM2PHL into PWM2PHH. This will set the phase shift between PWM channels.

24.1.2 MAXIMUM PWM DUTY CYCLE OF THE MASTER CHANNEL (PWM #1)

The PWM duty cycle of the first channel is specified by writing to the PWM1RL register. Up to 8-bit resolution is available. The following equation is used to calculate the PWM pulse width of PWM1 signal.

EQUATION 24-3:

$$PWM1_{PULSE\ WIDTH} = PWM1RL \times T_{OSC} \times (T2_{PRESCALE\ VALUE})$$

The PWM1RL bits can be written to at any time, but the duty cycle value is not latched into PWM1RH until after a match between PR2 and TMR2 occurs.

The following equation is used to calculate the maximum duty cycle of PWM channel 1 (MASTER channel).

EQUATION 24-4:

$$PWM1_{DUTY\ CYCLE} = \frac{PWM1RL}{PR2 + 1} \cdot 100$$

The calculated duty cycle is expressed in percent.

24.1.3 PHASE SHIFT FOR SLAVE CHANNEL (PWM #2)

In order to avoid excessive current ripple into the input filter capacitor of the converter, a phase shift between channels can be introduced.

The amount of phase shift between PWM channels can be adjusted by the value of the PWM2PHL register. The PWM #2 channel phase shift (in seconds) can be calculated by using the following equation.

EQUATION 24-5:

$$PWM2_{PHASE\ SHIFT} = PWM2PHL \times T_{OSC} \times (T2_{PRESCALE\ VALUE})$$

The phase shift expressed in electrical degrees can be calculated with the equation:

EQUATION 24-1:

$$PWM2_{PHASE\ SHIFT} = \frac{PWM2PHL}{PR2 + 1} \cdot 360$$

24.1.4 MAXIMUM PWM DUTY CYCLE OF THE SLAVE CHANNEL (PWM #2)

The maximum duty cycle of the SLAVE PWM channel can be calculated with the following equation:

EQUATION 24-6:

$$PWM2_{DUTYCYCLE} = \frac{PWM2RL - PWM2PHL}{PR2 + 1} \cdot 100$$

If the result of the above equation is negative, add 100 in order to find the correct value. The calculated duty cycle is expressed in percent.

To calculate the value of the PWM2RL register for a desired maximum duty cycle, use the following equation:

EQUATION 24-7:

$$PWM2RL = \left[round\left(\frac{D}{100} \cdot (PR2 + 1) + PWM2PHL\right)\right]_{modulo(PR2 + 1)}$$

The duty cycle D must be expressed in percent.

24.2 Operation During Sleep

When the device is placed in Sleep mode, the allocated timer will not increment and the state of the module will not change. When the device wakes up, it will continue from this state.

TABLE 24-1: SUMMARY OF REGISTERS ASSOCIATED WITH PWM MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PWM2RL	PWM2 (SLAVE) Register Low Byte								158
PWM2PHL	PWM2 (SLAVE) Phase Shift Register								158
T2CON	—	—	—	—	—	TMR2ON	T2CKPS1	T2CKPS0	152
PR2	Timer2 Module Period Register								151
PWM1RL	PWM1 (MASTER) Register Low Byte								158*

Legend: — = Unimplemented locations, read as '0'. Shaded cells are not used by PWM mode.

* Page provides register information.

NOTES:

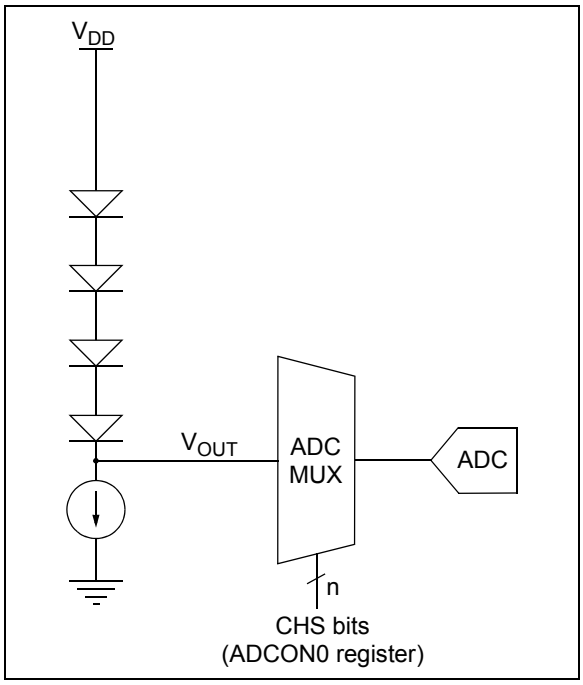
25.0 INTERNAL TEMPERATURE INDICATOR MODULE

The MCP19214/5 devices are equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between –40°C and +125°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

25.1 Circuit Operation

This internal temperature measurement circuit is always enabled.

FIGURE 25-1: TEMPERATURE CIRCUIT DIAGRAM



25.2 Temperature Output

The output of the circuit is measured using the internal analog-to-digital converter. Channel 13 is reserved for the temperature circuit output. Refer to **Section 20.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

The temperature of the silicon die can be calculated by the ADC measurement, using Equation 25-1. A factory-stored 10-bit ADC value for +30°C is located at address 2084h. The temperature coefficient for this circuit is 15.7 mV/°C (±0.8 mV/°C). Other temperature readings can be calculated from the 30°C mark.

EQUATION 25-1: SILICON DIE TEMPERATURE

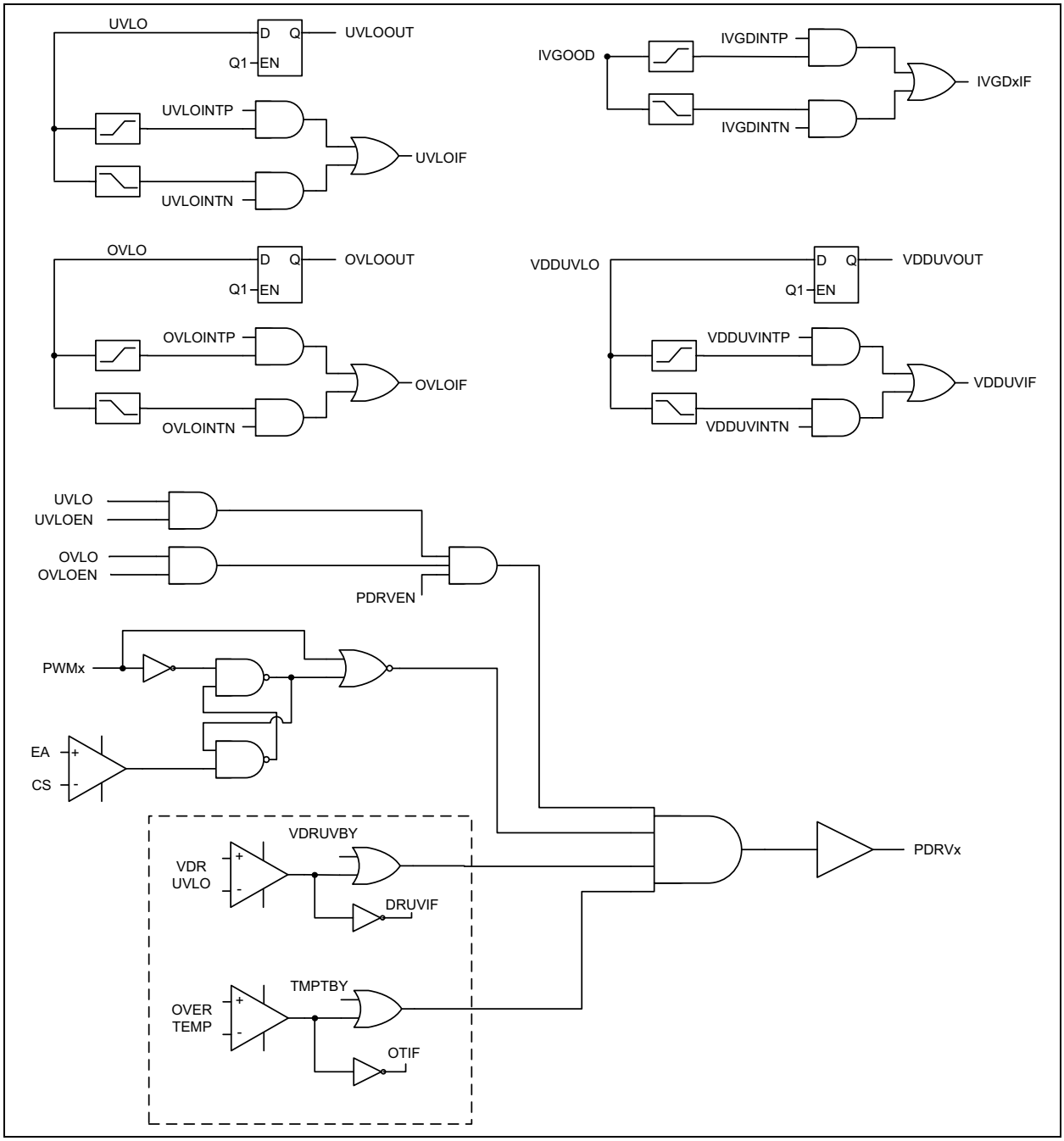
$$TEMP\_DIE(\,^{\circ}C) = \frac{(ADC\_READING\ (counts) - ADC\_30\ ^{\circ}C\_READING\ (counts)}{4.0\ (counts/\,^{\circ}C)} + 30\ ^{\circ}C$$

NOTES:

26.0 PWM CONTROL LOGIC

The PWM Control Logic implements standard comparator modules to identify events such as input undervoltage, input overvoltage and V_{DD} UVLO. The control logic takes action in hardware to appropriately enable/disable the output drive (PDRVx), as well as to set corresponding interrupt flags to be read by software. This control logic also defines normal PWM operation. For definition of individual bits within the control logic, refer to the Special Function register (SFR) sections.

FIGURE 26-1: PWM CONTROL LOGIC



NOTES:

27.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

27.1 MSSP Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module in the MCP19214/5 only operates in Inter-Integrated Circuit (I^2C) mode.

The I^2C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited Multi-Master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Dual Address masking
- Address Hold and Data Hold modes
- Selectable SDA hold times

Figure 27-1 is a block diagram of the I^2C interface module in Master mode. Figure 27-2 is a diagram of the I^2C interface module in Slave mode.

FIGURE 27-1: MSSP BLOCK DIAGRAM (I^2C MASTER MODE)

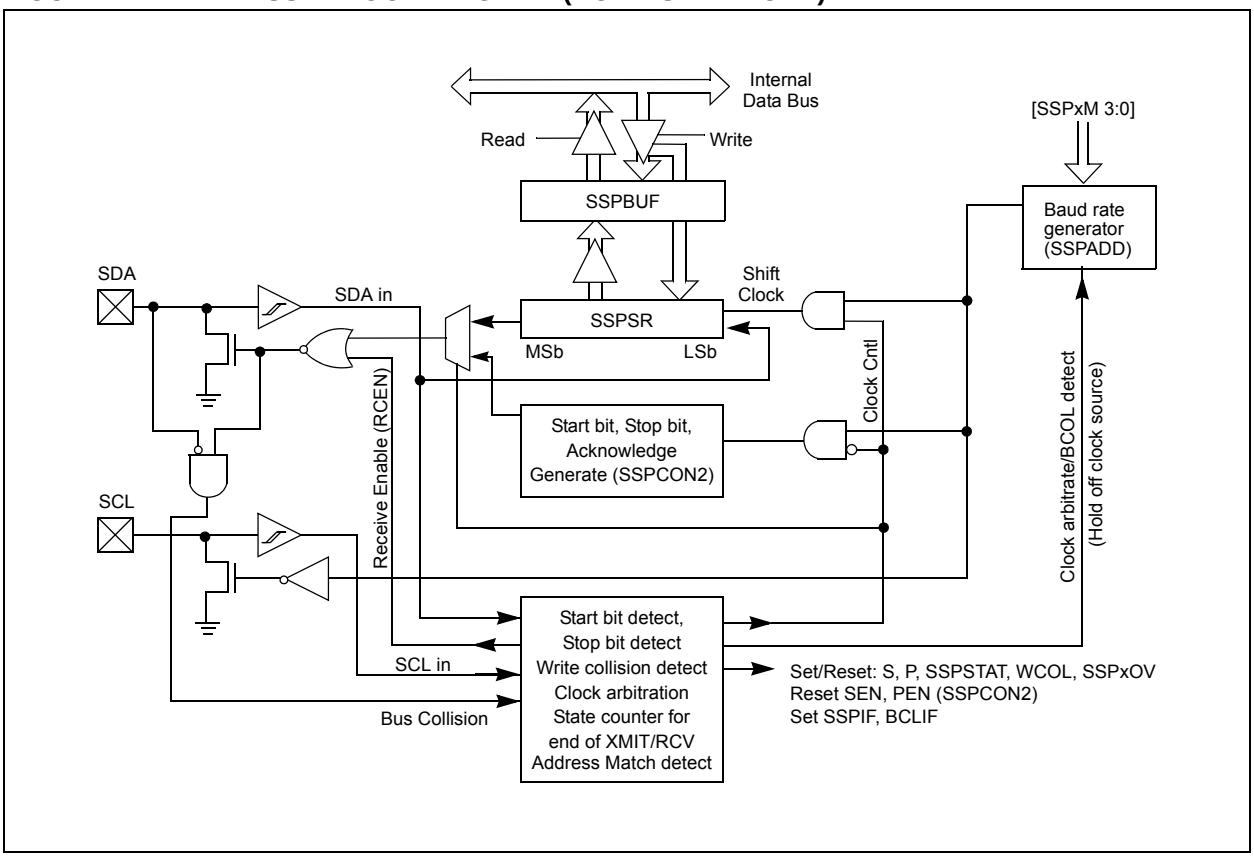
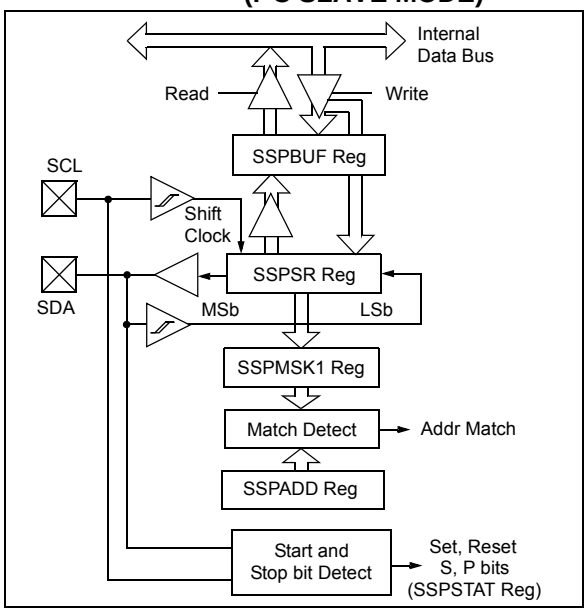


FIGURE 27-2: MSSP BLOCK DIAGRAM (I^2C SLAVE MODE)



27.2 I^2C MODE OVERVIEW

The Inter-Integrated Circuit Bus (I^2C) is a multi-master serial data communication bus. Devices communicate in a master/slave environment, where the master devices initiate the communication. A slave device is controlled through addressing.

The I^2C bus specifies two signal connections:

- Serial Clock (SCL)
- Serial Data (SDA)

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero; letting the line float is considered a logical one.

Figure 27-3 shows a typical connection between two devices configured as master and slave.

The I^2C bus can operate with one or more master devices and one or more slave devices.

There are four potential modes of operation for a given device:

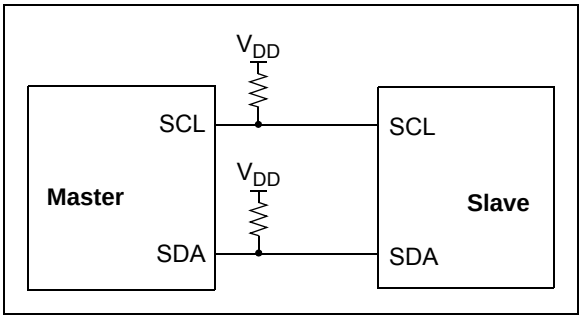
- Master Transmit mode
  (master is transmitting data to a slave)
- Master Receive mode
  (master is receiving data from a slave)
- Slave Transmit mode
  (slave is transmitting data to a master)
- Slave Receive mode
  (slave is receiving data from a master)

To begin communication, a master device starts out in Master Transmit mode. The master device sends out a Start bit followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an $\overline{ACK}$. The master then continues in either Transmit mode or Receive mode and the slave continues in the complement, either in Receive mode or Transmit mode, respectively.

A Start bit is indicated by a high-to-low transition of the SDA line while the SCL line is held high. Address and data bytes are sent out Most Significant bit (MSb) first. The Read/Write bit is sent out as a logical one when the master intends to read data from the slave, and is sent out as a logical zero when it intends to write data to the slave.

FIGURE 27-3: I^2C MASTER/ SLAVE CONNECTION



The Acknowledge bit ($\overline{ACK}$) is an active-low signal that holds the SDA line low to indicate to the transmitter that the slave device has received the transmitted data and is ready to receive more.

The transition of a data bit is always performed while the SCL line is held low. Transitions that occur while the SCL line is held high are used to indicate Start and Stop bits.

If the master intends to write to the slave, it repeatedly sends out a byte of data, with the slave responding after each byte with an $\overline{ACK}$ bit. In this example, the master device is in Master Transmit mode and the slave is in Slave Receive mode.

If the master intends to read from the slave, it repeatedly receives a byte of data from the slave and responds after each byte with an $\overline{ACK}$ bit. In this example, the master device is in Master Receive mode and the slave is in Slave Transmit mode.

On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends the Stop bit in place of the last ACK bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send another Start bit in place of the Stop bit or last ACK bit when it is in Receive mode.

The I^2C bus specifies three message protocols:

- Single message where a master writes data to a slave
- Single message where a master reads data from a slave
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves

When one device is transmitting a logical one, or letting the line float, and a second device is transmitting a logical zero, or holding the line low, the first device can detect that the line is not a logical one. This detection, when used on the SCL line, is called clock stretching. Clock stretching gives slave devices a mechanism to control the flow of data. When this detection is used on the SDA line, it is called arbitration. Arbitration ensures that there is only one master device communicating at any single time.

27.2.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

27.2.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an idle state.

However, two master devices may try to initiate a transmission at or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels don't match loses arbitration and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it must also stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission. It can do so without any complications, because so far the transmission appears exactly as expected, with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

27.3 I²C MODE OPERATION

All MSSP I²C communication is byte-oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC microcontroller and with the user's software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

27.3.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice versa, followed by an Acknowledge bit sent back. After the 8th falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained in the following sections.

27.3.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. Such word usage is defined in Table 27-1 and may be used in the rest of this document without explanation. The information in this table was adapted from the Philips I²C specification.

27.3.3 SDA AND SCL PINS

Selecting any I²C mode with the SSPEN bit set forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

Note: Data is tied to output zero when an I²C mode is enabled.

27.3.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit in the SSPCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 27-1: I²C BUS TERMS

Term	Description
Transmitter	The device that shifts data out onto the bus
Receiver	The device that shifts data in from the bus
Master	The device that initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master
Multi-Master	A bus with more than one device that can initiate data transfers
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus
Idle	No master is controlling the bus and both SDA and SCL lines are high
Active	Any time one or more master devices are controlling the bus
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSPADDx
Write Request	Slave receives a matching address with R/$\overline{W}$ bit clear and is ready to clock in data
Read Request	Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus holds SCL low to stall communication
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state

27.3.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDA from a high state to a low state, while SCL line is high. A Start condition is always generated by the master, and signifies the transition of the bus from an Idle to an Active state. Figure 27-4 shows the wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I^2C Specification that states no bus collision can occur on a Start.

27.3.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

Note: At least one SCL low time must appear before a Stop is valid. Therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

27.3.7 RESTART CONDITION

A Restart is valid any time that a Stop is valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave.

In 10-bit Addressing Slave mode, a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained. Until a Stop condition, a high address with R/$\overline{W}$ clear or a high address match fails.

27.3.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits in the SSPCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. These bits will have no effect on slave modes where interrupt on Start and Stop detect are already enabled.

FIGURE 27-4: I^2C START AND STOP CONDITIONS

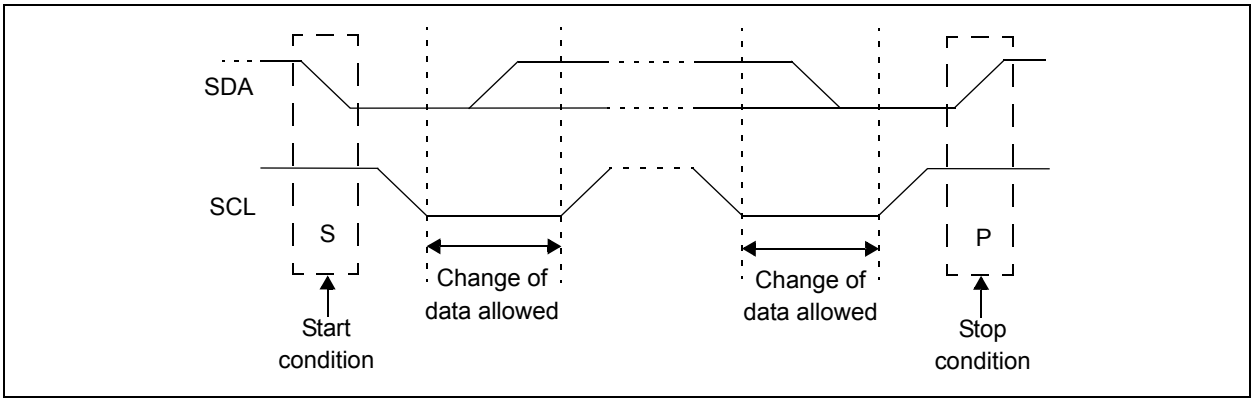
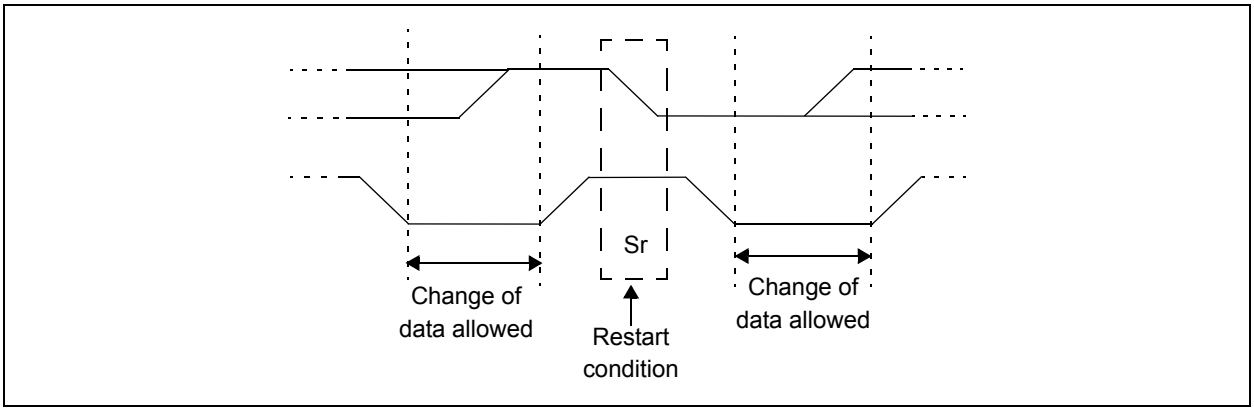


FIGURE 27-5: I^2C RESTART CONDITION



27.3.9 ACKNOWLEDGE SEQUENCE

The 9th SCL pulse for any transferred byte in I2C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{ACK}$) is an active-low signal, pulling the SDA line low, indicating to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{ACK}$ is placed in the ACKSTAT bit in the SSPCON2 register.

Slave software, when the AHEN and DHEN bits are set, allows the user to set the $\overline{ACK}$ value sent back to the transmitter. The ACKDT bit in the SSPCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{ACK}$ response if the AHEN and DHEN bits in the SSPCON3 register are clear.

There are certain conditions where an $\overline{ACK}$ will not be sent by the slave. If the BF bit in the SSPSTAT register or the SSPOV bit in the SSPCON1 register are set when a byte is received, the $\overline{ACK}$ will not be sent.

When the module is addressed, after the 8th falling edge of SCL on the bus, the ACKTIM bit in the SSPCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM status bit is only active when the AHEN or DHEN bits are enabled.

27.4 I2C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of the four modes selected in the SSPM bits in SSPCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing mode operates the same as 7-bit, with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes, with SSPIF additionally getting set upon detection of a Start, Restart or Stop condition.

27.4.1 SLAVE MODE ADDRESSES

The SSPADD register contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSPMSK1 register affects the address matching process. Refer to **Section 27.4.10 "SSPMSK1 Register"** for more information.

27.4.2 SECOND SLAVE MODE ADDRESS

The SSPADD2 register contains a second 7-bit Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSPMSK2 register affects the address matching process. Refer to **Section 27.4.10 "SSPMSK1 Register"** for more information.

27.4.2.1 I2C Slave 7-Bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

27.4.2.2 I2C Slave 10-Bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb of the 10-bit address and are stored in bits 2 and 1 in the SSPADD register.

After the high byte has been acknowledged, the UA bit is set and SCL is held low until the user updates SSPADD with the low address. The low address byte is clocked in and all 8 bits are compared to the low address value in SSPADD. Even if there is no address match, SSPIF and UA are set and SCL is held low until SSPADD is updated to receive a high byte again. When SSPADD is updated, the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address-byte match.

27.4.3 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit in the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and acknowledged.

When an overflow condition exists for a received address, then Not Acknowledge is given. An overflow condition is defined as either bit BF in the SSPSTAT register is set, or bit SSPOV in the SSPCON1 register is set. The BOEN bit in the SSPCON3 register modifies this operation. For more information, refer to Register 27-4.

An MSSP interrupt is generated for each transferred data byte. The flag bit SSPIF must be cleared by software.

When the SEN bit in the SSPCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit in the SSPCON1 register, except sometimes in 10-bit mode.

27.4.3.1 7-Bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I^2C Slave in 7-bit Addressing mode, including all decisions made by hardware or software and their effects on reception. Figures 27-5 and 27-6 are used as a visual reference for this description.

This is a step-by-step process of what typically must be done to accomplish I^2C communication:

1. Start bit detected.
2. S bit in the SSPSTAT register is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low, sending an $\overline{ACK}$ to the master, and sets SSPIF bit.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPBUF, clearing the BF flag.
7. If SEN = 1, Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSPIF bit.
10. Software clears SSPIF.
11. Software reads the received byte from SSPBUF clearing BF.
12. Steps 8 to 11 are repeated for all received bytes from the Master.
13. Master sends Stop condition, setting P bit in the SSPSTAT register, and the bus goes idle.

27.4.3.2 7-Bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operates the same as it does without these options, with extra interrupts and clock stretching added after the 8th falling edge of SCL. These additional interrupts allow the slave software to decide if it wants the $\overline{ACK}$ to receive address or data byte, rather than the hardware.

Figure 27-7 displays a module using both address and data holding. Figure 27-8 includes the operation with the SEN bit in the SSPCON2 register set. This list describes the steps that need to be taken by slave software to use these options for I^2C communication:

1. S bit in the SSPSTAT register is set; SSPIF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSPIF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSPIF.
4. Slave can look at the ACKTIM bit in the SSPCON3 register to determine if the SSPIF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPxIF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSPIF.

Note: SSPIF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to Master is SSPIF not set.

11. SSPIF set and CKP cleared after 8th falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit in the SSPCON3 register to determine the source of the interrupt.
13. Slave reads the received data from SSPBUF clearing BF.
14. Steps 7 to 14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1 or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit in the SSPSTAT register.

FIGURE 27-6: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)

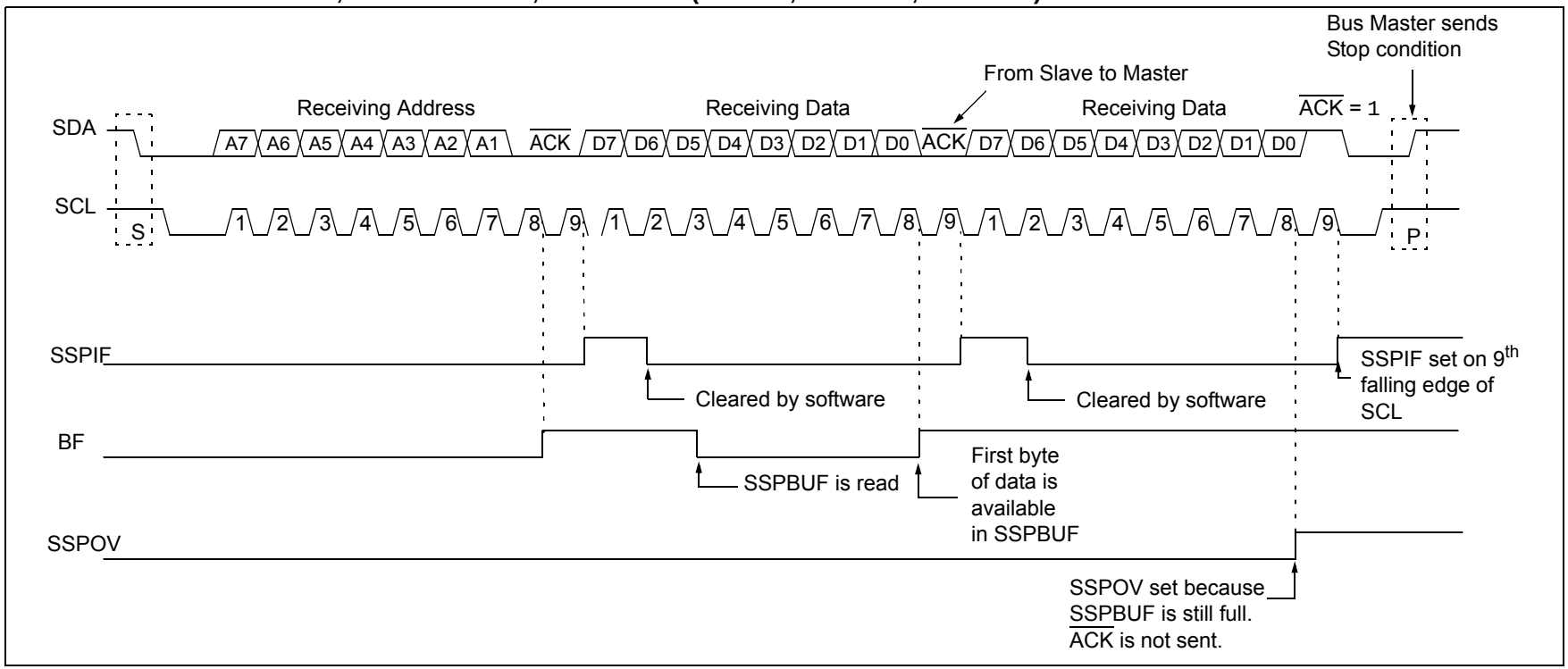


FIGURE 27-7: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)

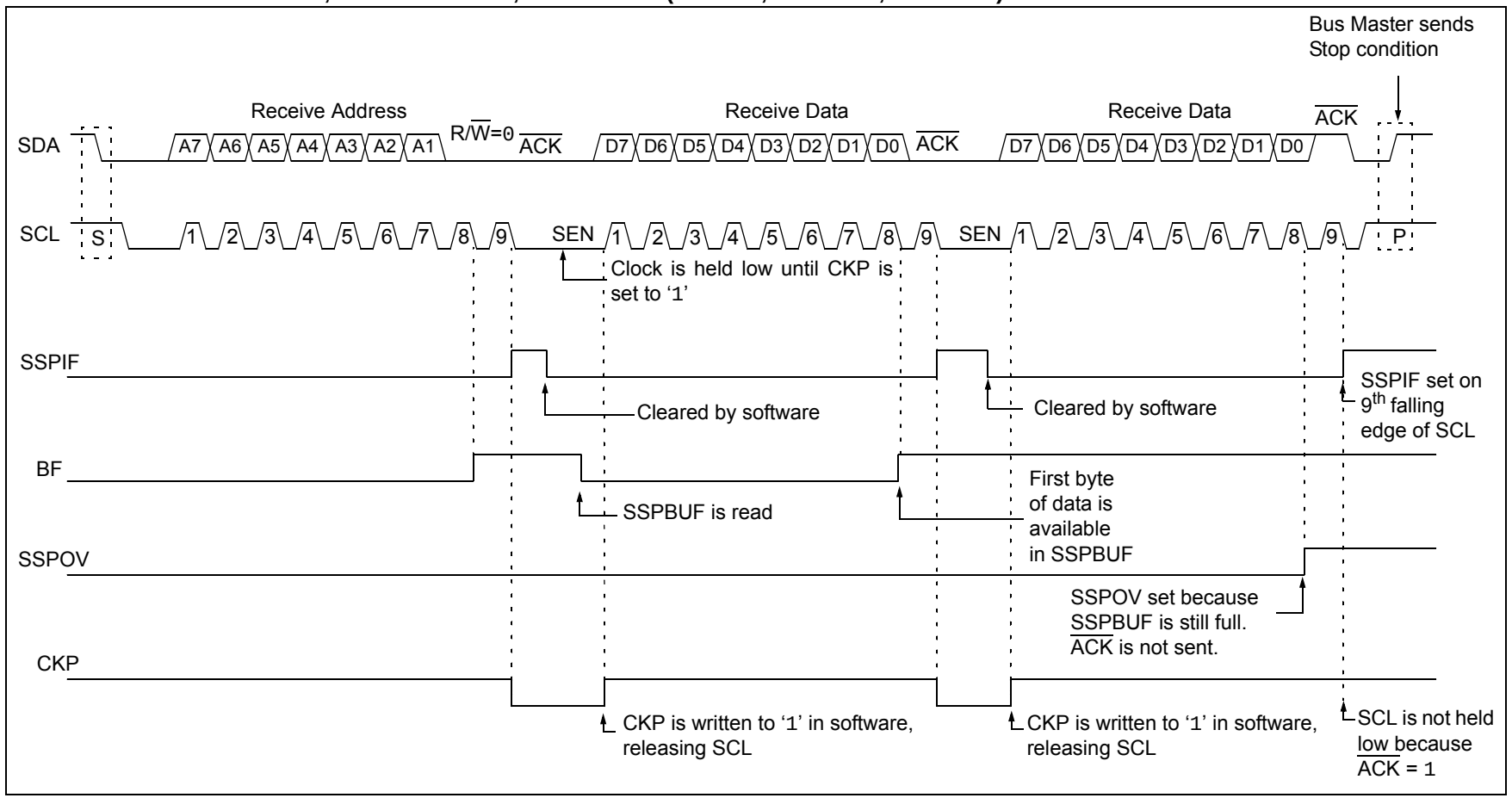


FIGURE 27-8: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)

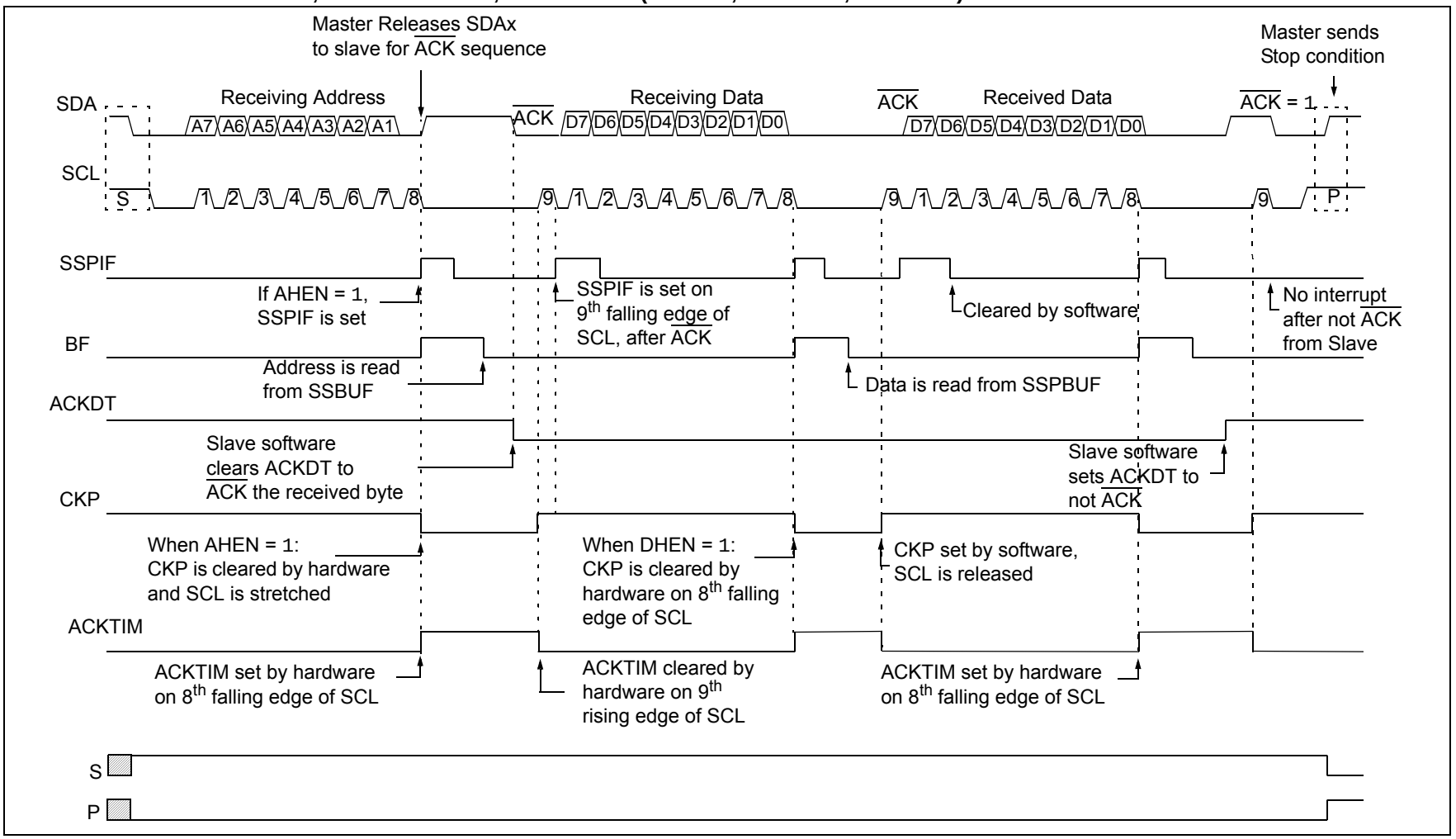
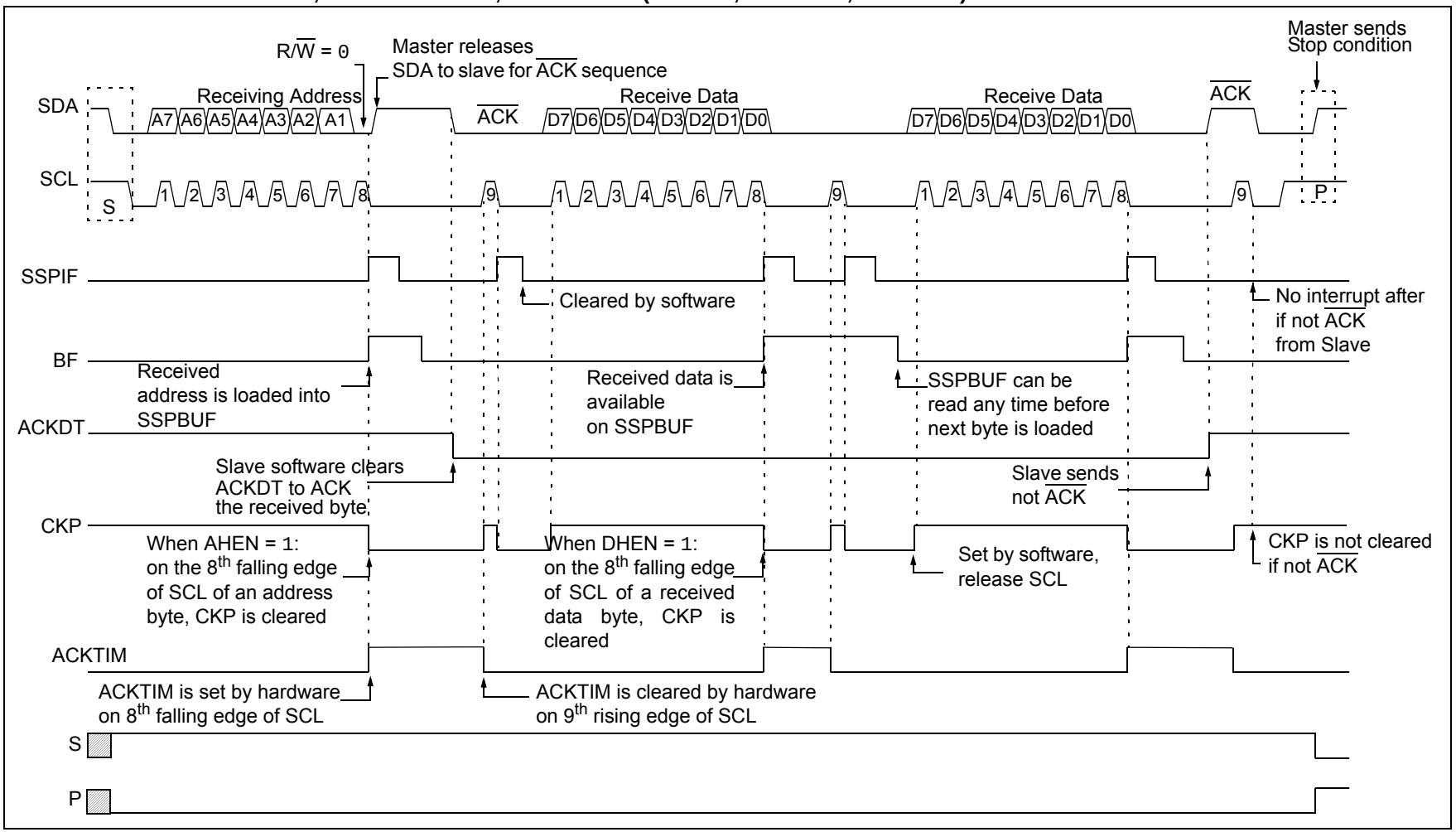


FIGURE 27-9: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)



27.4.4 SLAVE TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit in the SSPSTAT register is set. The received address is loaded into the SSPBUF register and an $\overline{ACK}$ pulse is sent by the slave on the 9th bit.

Following the $\overline{ACK}$, slave hardware clears the CKP bit and the SCL pin is held low. Refer to **Section 27.4.7 "Clock Stretching"** for more details. By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be released by setting the CKP bit in the SSPCON1 register. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time.

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the 9th SCL input pulse. This $\overline{ACK}$ value is copied to the ACKSTAT bit in the SSPCON2 register. If ACKSTAT is set (not $\overline{ACK}$), the data transfer is complete. In this case, when the not $\overline{ACK}$ is latched by the slave, the slave goes idle and waits for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPBUF register. Again, the SCL pin must be released by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared by software, and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the 9th clock pulse.

27.4.4.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDA line. If a bus collision is detected and the SBCDE bit in the SSPCON3 register is set, the BCLIF bit in the PIR register is set. Once a bus collision is detected, the slave goes idle and waits to be addressed again. The user's software can use the BCLIF bit to handle a slave bus collision.

27.4.4.2 7-Bit Transmission

A master device can transmit a read request to a slave, and then it clocks data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 27-10 can be used as a reference to this list:

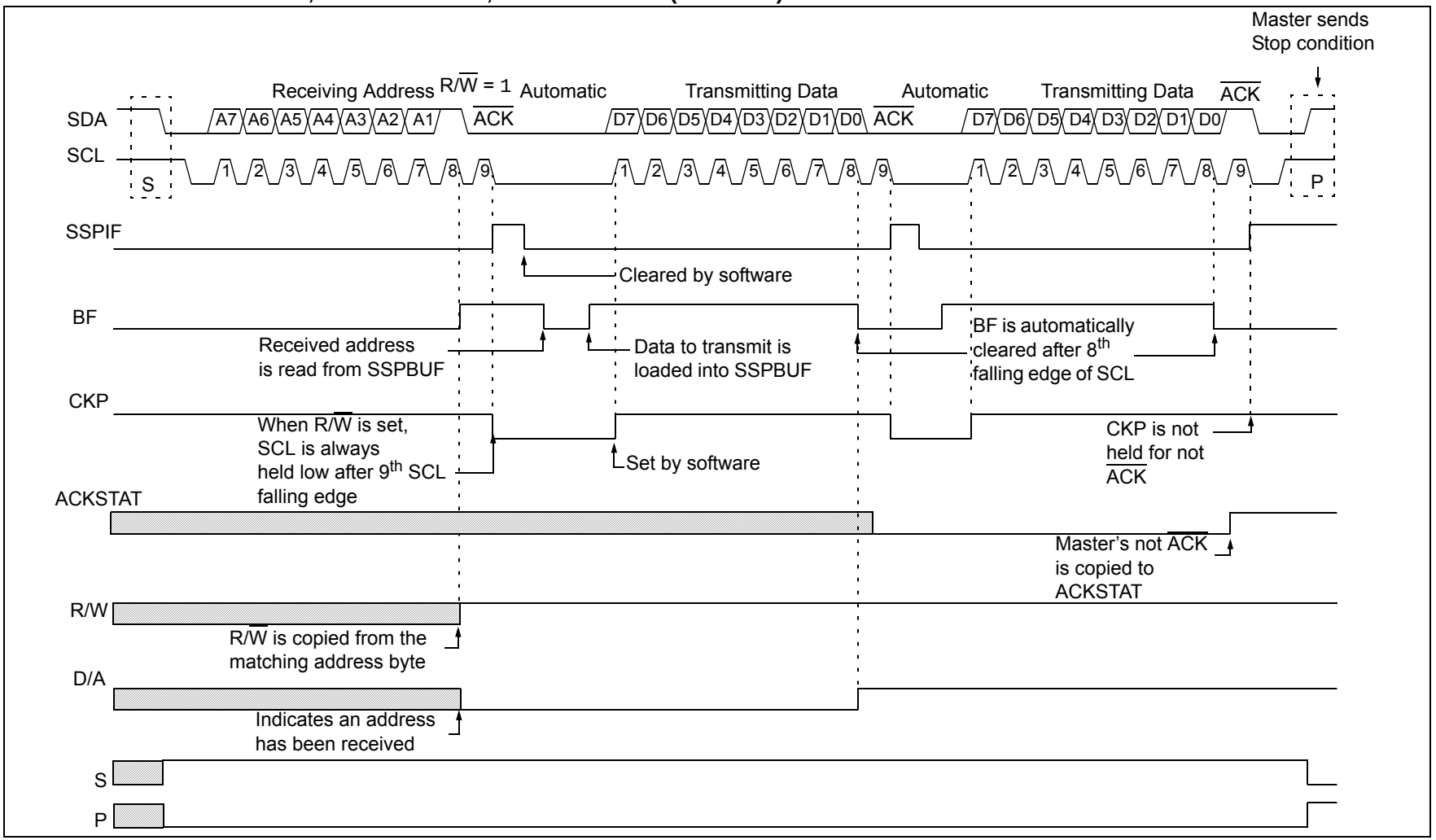
1. Master sends a Start condition on SDA and SCL.
2. S bit in the SSPSTAT register is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit set is received by the Slave setting SSPIF bit.
4. Slave hardware generates an $\overline{ACK}$ and sets SSPIF.
5. SSPIF bit is cleared by user.
6. Software reads the received address from SSPBUF, clearing BF.
7. R/$\overline{W}$ is set so CKP was automatically cleared after the $\overline{ACK}$.
8. The slave software loads the transmit data into SSPBUF.
9. CKP bit is set releasing SCL, allowing the master to clock the data out of the slave.
10. SSPIF is set after the $\overline{ACK}$ response from the master is loaded into the ACKSTAT register.
11. SSPIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master $\overline{ACK}$s, the clock will be stretched.

2: ACKSTAT is the only bit updated on the rising edge of SCL (9th) rather than on the falling edge.

13. Steps 9 to 13 are repeated for each transmitted byte.
14. If the master sends a not $\overline{ACK}$, the clock is not held, but SSPIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

FIGURE 27-10: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)



27.4.4.3 7-Bit Transmission with Address Hold Enabled

Setting the AHEN bit in the SSPCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPIF interrupt is set.

Figure 27-11 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

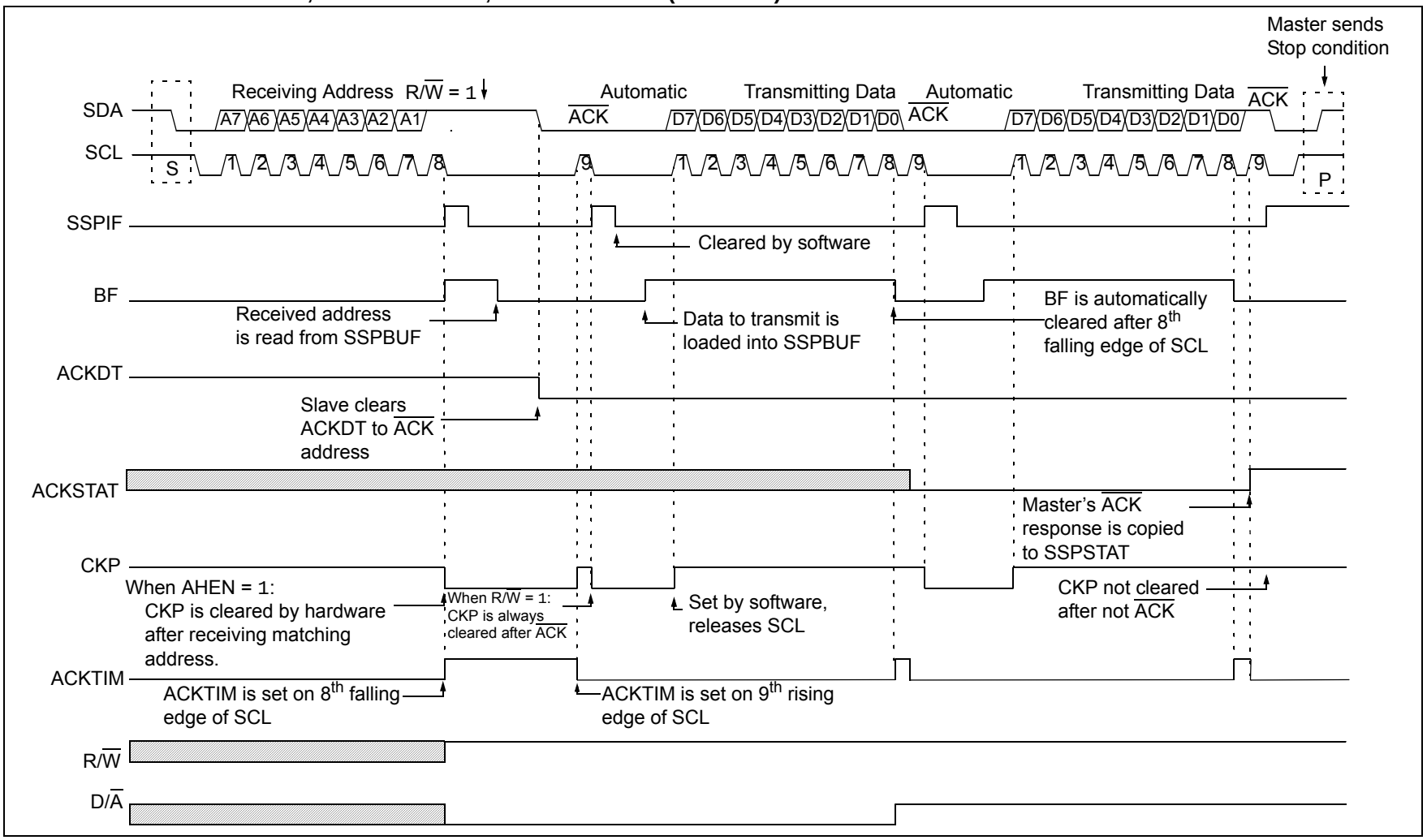
1. Bus starts idle.
2. Master sends Start condition; the S bit in the SSPSTAT register is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the 8th falling edge of the SCL line, the CKP bit is cleared and SSPIF interrupt is generated.
4. Slave software clears SSPIF.
5. Slave software reads ACKTIM bit in the SSPCON3 register and R/$\overline{W}$ and D/$\overline{A}$ bits in the SSPSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPBUF register, clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$, and sets ACKDT bit in the SSPCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPIF.
12. Slave loads value to transmit to the master into SSPBUF setting the BF bit.

Note:	SSPBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the 9th SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit in the SSPCON2 register.
16. Steps 10 to 15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$, the slave releases the bus, allowing the master to send a Stop and end the communication.

Note:	Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.

FIGURE 27-11: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 1)



27.4.5 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSP module configured as an I^2C Slave in 10-bit Addressing mode.

Figure 27-12 is used as a visual reference for this description.

This is a step-by-step process of what must be done by slave software to accomplish I^2C communication:

1. Bus starts idle.
2. Master sends Start condition; S bit in the SSPSTAT register is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching high address with R/$\overline{W}$ bit clear; UA bit in the SSPSTAT register is set.
4. Slave sends $\overline{ACK}$ and SSPIF is set.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPBUF, clearing the BF flag.
7. Slave loads low address into SSPADD, releasing SCL.
8. Master sends matching low-address byte to the Slave; UA bit is set.

Note: Updates to the SSPADD register are not allowed until after the $\overline{ACK}$ sequence.

9. Slave sends $\overline{ACK}$ and SSPIF is set.

Note: If the low address does not match, SSPIF and UA are still set so that the slave software can set SSPADD back to the high address. BF is not set because there is no match. CKP is unaffected.

10. Slave clears SSPIF.
11. Slave reads the received matching address from SSPBUF, clearing BF.
12. Slave loads high address into SSPADD.
13. Master clocks a data byte to the slave and clocks out the slave's $\overline{ACK}$ on the 9th SCL pulse; SSPIF is set.
14. If SEN bit in the SSPCON2 register is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPIF.
16. Slave reads the received byte from SSPBUF, clearing BF.
17. If SEN is set, the slave sets CKP to release the SCL.
18. Steps 13 to 17 are repeated for each received byte.
19. Master sends Stop to end the transmission.

27.4.6 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and the SCL line is held low, is the same. Figure 27-13 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 27-14 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

FIGURE 27-12: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)

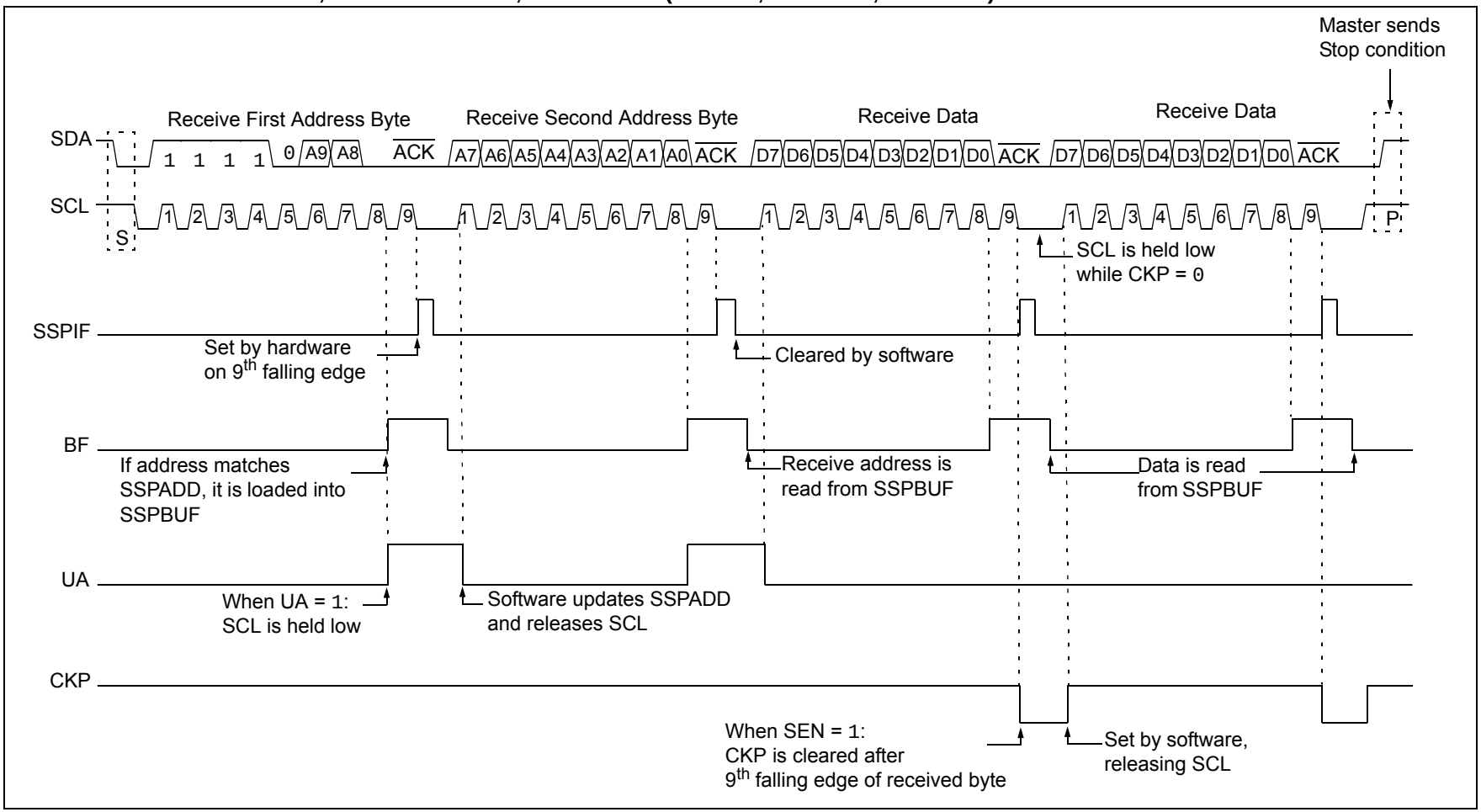


FIGURE 27-13: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)

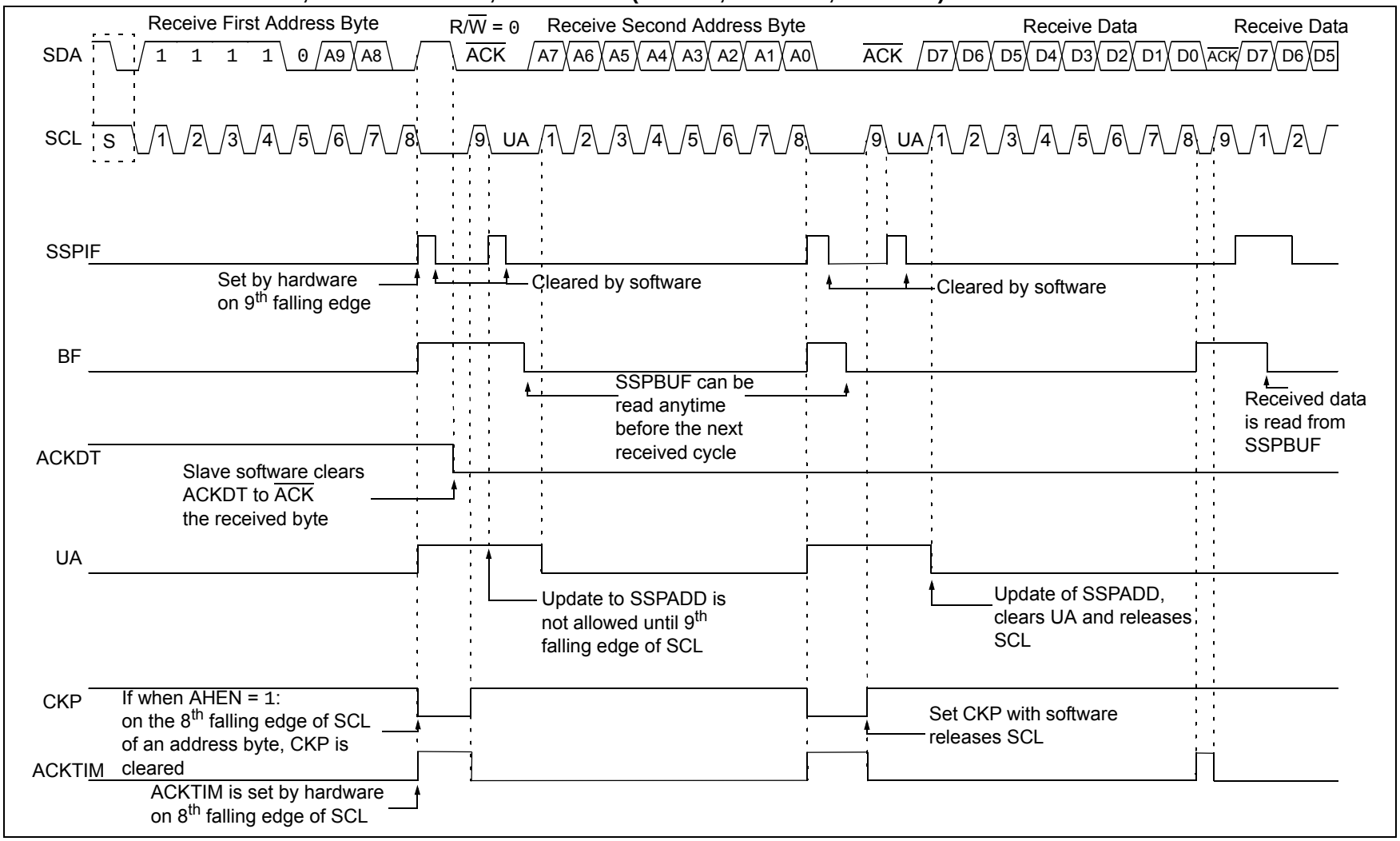
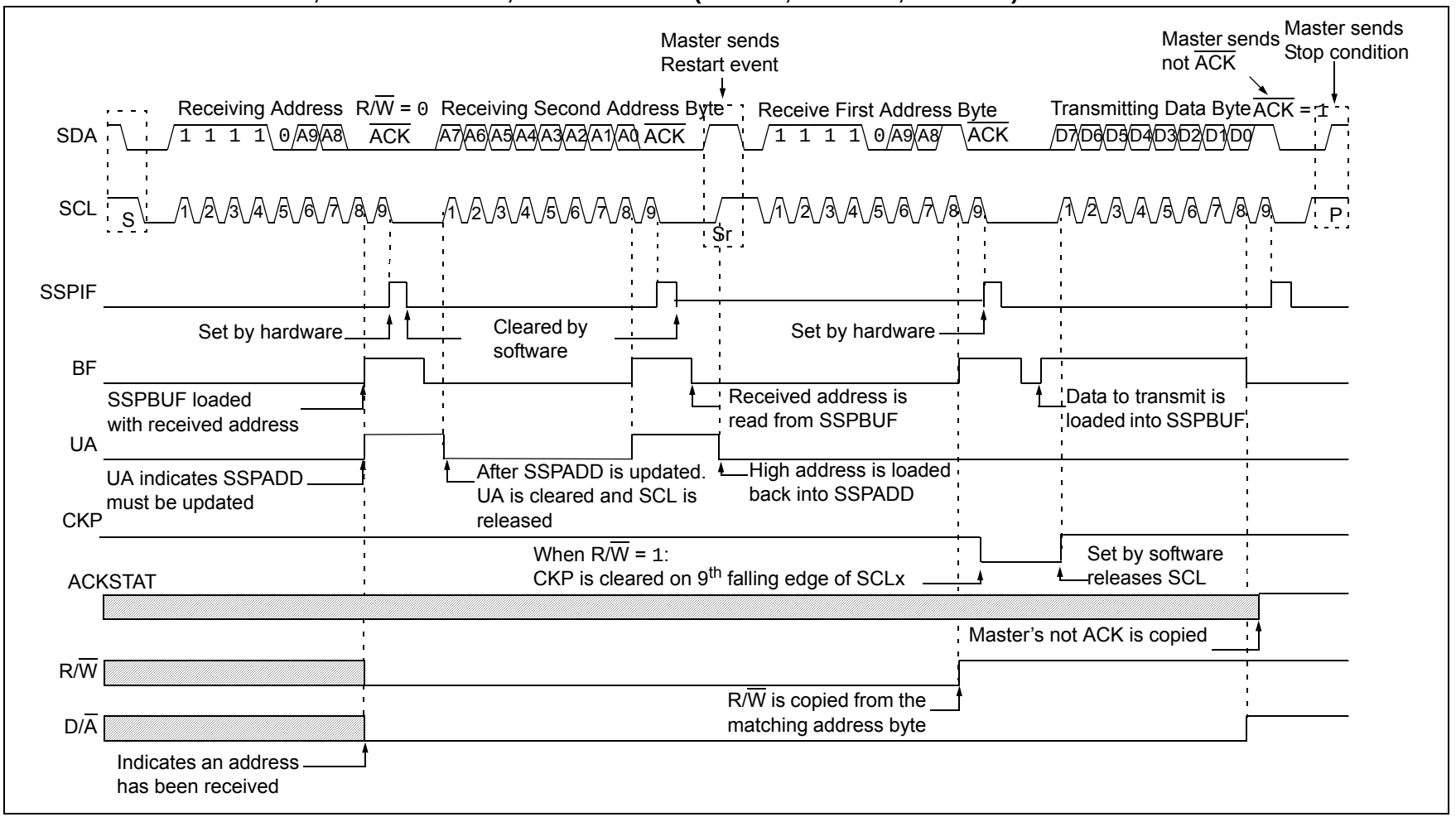


FIGURE 27-14: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)



27.4.7 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching, as it is stretching anytime it is active on the bus and not transferring data. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit in the SSPCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

27.4.7.1 Normal Clock Stretching

Following an $\overline{ACK}$, if the R/$\overline{W}$ bit in the SSPSTAT register is set, causing a read request, the slave hardware will clear CKP. This allows the slave time to update SSPBUF with data to transfer to the master. If the SEN bit in the SSPCON2 register is set, the slave hardware will always stretch the clock after the $\overline{ACK}$ sequence. Once the slave is ready, CKP is set by software, and communication resumes.

Note 1:	The BF bit has no effect on whether the clock will be stretched or not. This is different from previous versions of the module that would not stretch the clock or clear CKP if SSPBUF was read before the 9th falling edge of SCL.
2:	Previous versions of the module did not stretch the clock for a transmission if SSPBUF was loaded before the 9th falling edge of SCL. It is now always cleared for read requests.

27.4.7.2 10-Bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set, the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPADD.

Note:	Previous versions of the module did not stretch the clock if the second address byte did not match.

27.4.7.3 Byte NACKing

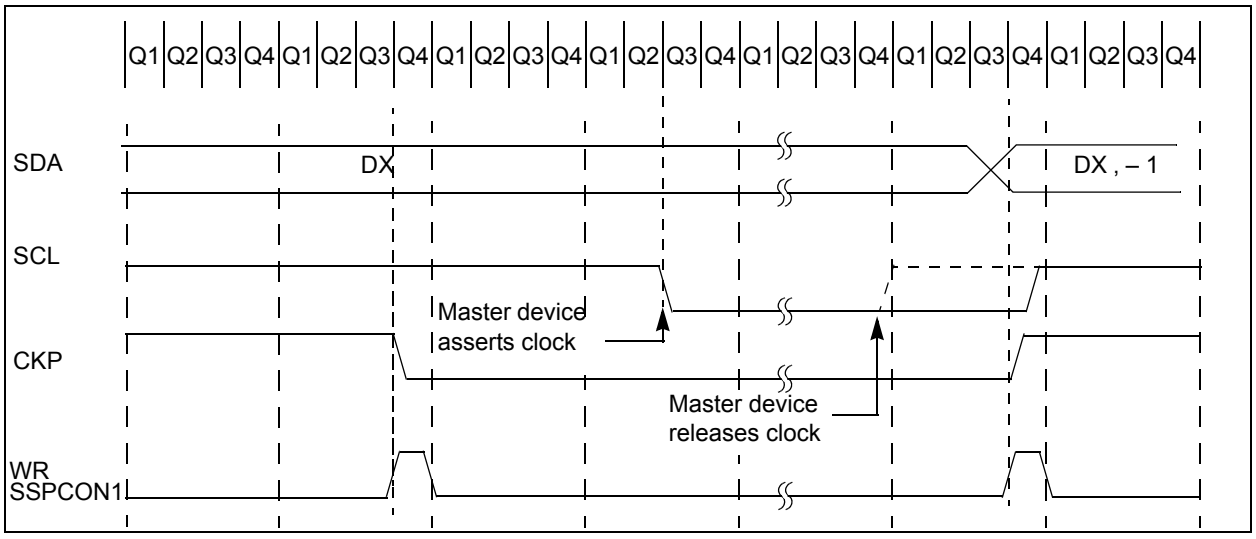
When AHEN bit in the SSPCON3 register is set, CKP is cleared by hardware after the 8th falling edge of SCL for a received matching address byte. When DHEN bit in the SSPCON3 register is set, CKP is cleared after the 8th falling edge of SCL for received data.

Stretching after the 8th falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

27.4.8 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (refer to Figure 27-16).

FIGURE 27-15: CLOCK SYNCHRONIZATION TIMING



27.4.9 GENERAL CALL ADDRESS SUPPORT

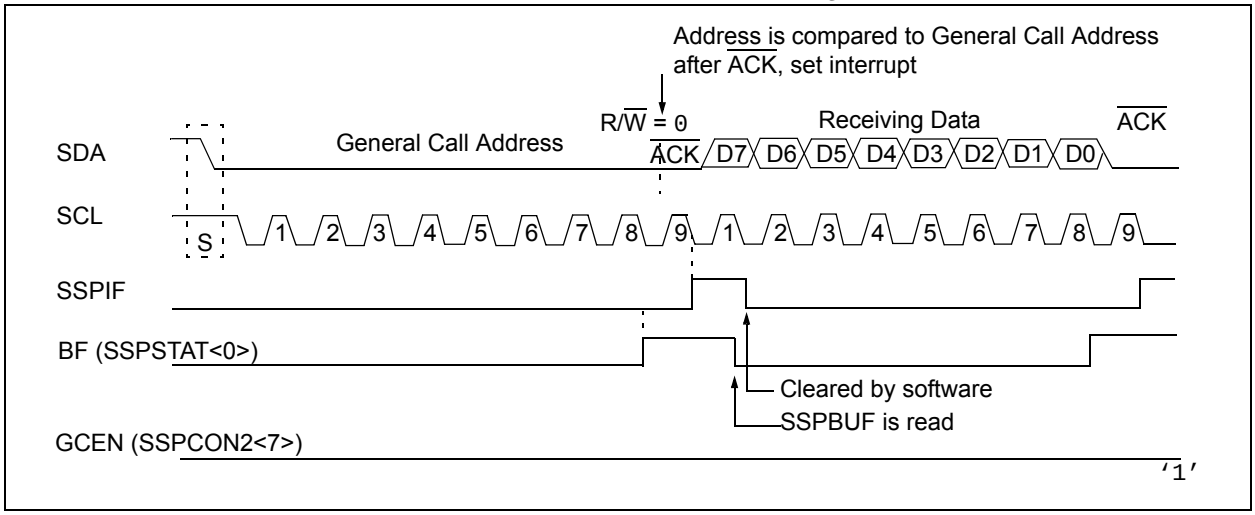
In the addressing procedure for the I^2C bus, the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address, which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I^2C protocol, defined as address 0x00. When the GCEN bit in the SSPCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address, regardless of the value stored in SSPADD. After the slave clocks in an address of all zeros with the R/$\overline{W}$ bit clear, an interrupt is generated and slave software can read SSPBUF and respond. Figure 27-7 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit in the SSPCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the 8th falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 27-16: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



27.4.10 SSPMSK1 REGISTER

An SSP Mask (SSPMSK1) register is available in I^2C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPMSK1 register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSPMSK1 register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

27.5 I²C MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary, to drive the pins low.

The Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set or the bus is idle.

In Firmware-Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user's software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit (SSPIF) to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1:	The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.
2:	When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

27.5.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer ends with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmit mode, serial data is output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

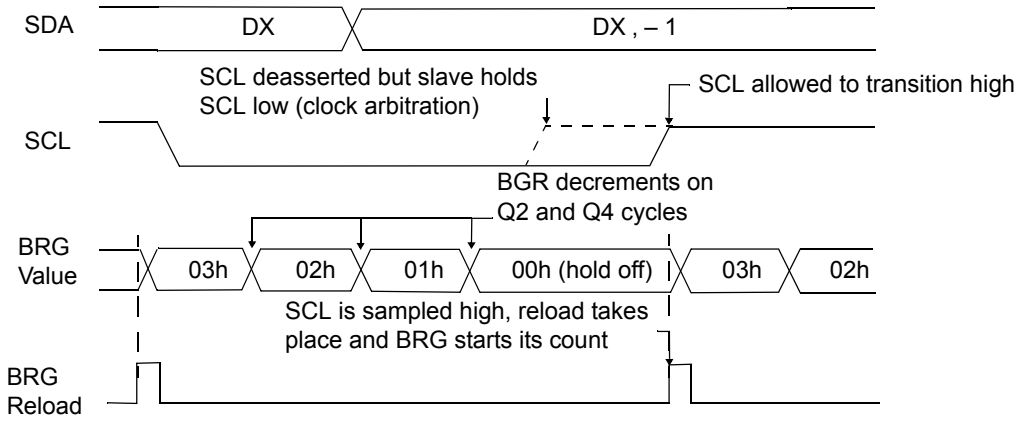
In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. Refer to **Section 27.6 "Baud Rate Generator"** for more details.

27.5.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any Receive, Transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 27-17).

FIGURE 27-17: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



27.5.3 WCOL STATUS FLAG

If the user writes the SSPBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set, it indicates that an action on SSPBUF was attempted while the module was not idle.

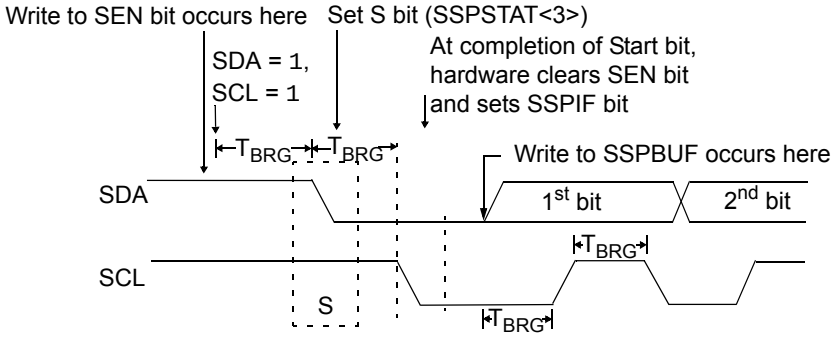
Note:	Because queuing of events is not allowed, writing to the lower 5 bits in the SSPCON2 register is disabled until the Start condition is complete.

27.5.4 I^2C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit (SEN) in the SSPCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (T_{BRG}), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit in the SSPSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<7:0> and resumes its count. When the Baud Rate Generator times out (T_{BRG}), the SEN bit in the SSPCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

Note 1:	If, at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if, during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag (BCLIF), is set, the Start condition is aborted and the I^2C module is reset into its idle state.
2:	The Philips I^2C Specification states that a bus collision cannot occur on a Start.

FIGURE 27-18: FIRST START BIT TIMING



27.5.5 I²C MASTER MODE REPEATED START CONDITION TIMING

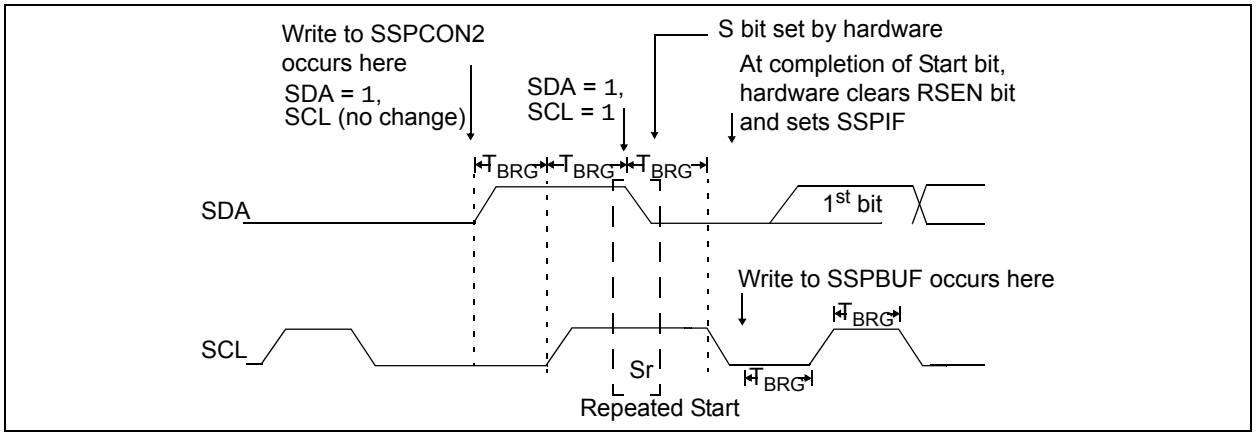
A Repeated Start condition occurs when the RSEN bit in the SSPCON2 register is programmed high and the Master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (T_{BRG}). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one T_{BRG}. This action is then followed by assertion of the SDA pin (SDA = 0) for one T_{BRG} while SCL is high. SCL is asserted low. Following this, the RSEN bit in the SSPCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit in the SSPSTAT register will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.

2: A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low to high
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

FIGURE 27-19: REPEAT START CONDITION WAVEFORM



27.5.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full (BF) flag bit and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (T_{BRG}). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for T_{BRG}. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the 8th bit is shifted out (the falling edge of the 8th clock), the BF flag is cleared and the master releases the SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the 9th bit time if an address match occurred or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the 9th clock. If the master receives an Acknowledge, the Acknowledge Status bit (ACKSTAT) is cleared. If not, the bit is set. After the 9th clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 27-20).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the 8th clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the 9th clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT status bit in the SSPCON2 register. Following the falling edge of the 9th clock transmission of the address, the SSPIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

27.5.6.1 BF Status Flag

In Transmit mode, the BF bit in the SSPSTAT register is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

27.5.6.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

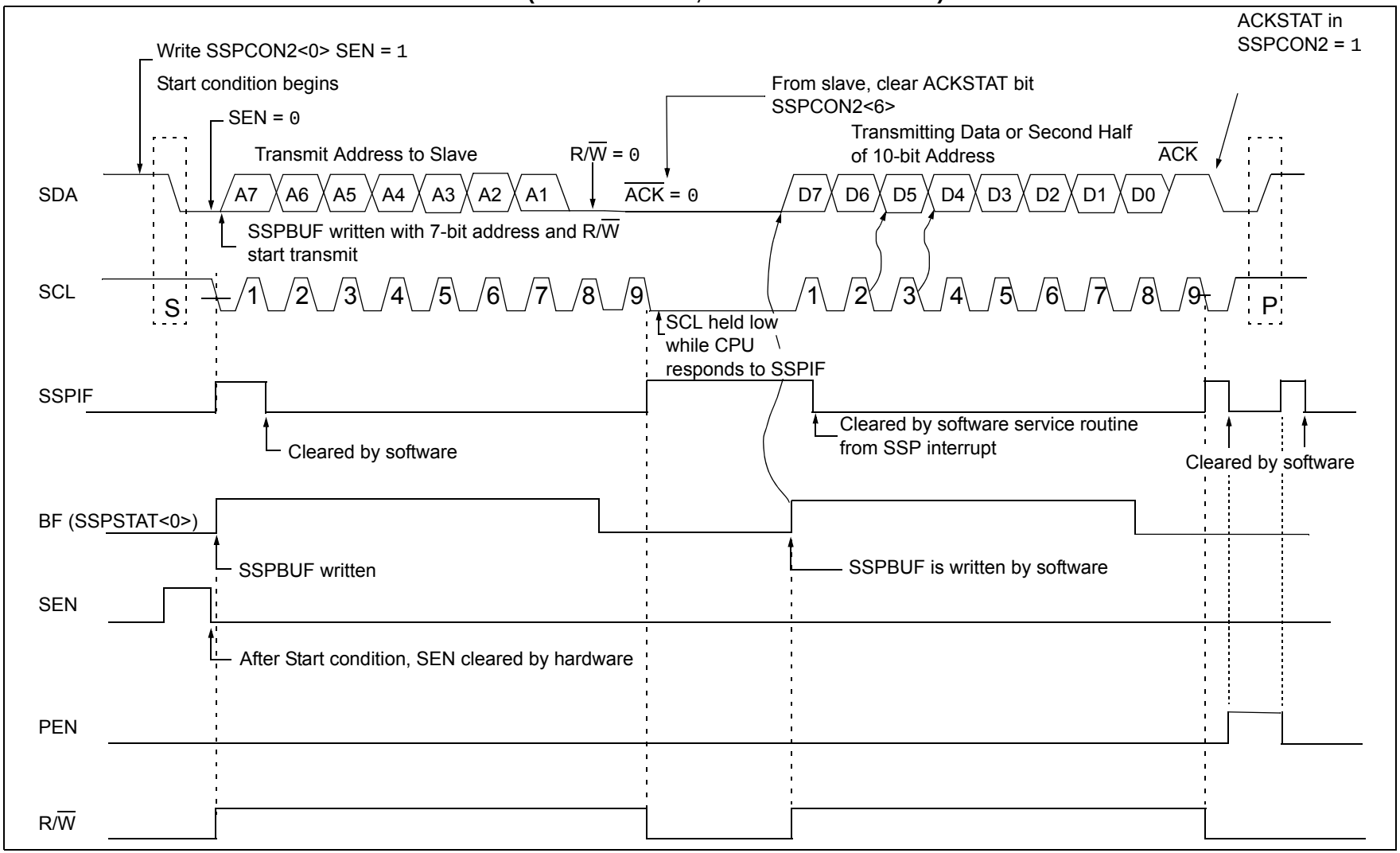
27.5.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit in the SSPCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call) or when the slave has properly received its data.

27.5.6.4 Typical Transmit Sequence:

1. The user generates a Start condition by setting the SEN bit in the SSPCON2 register.
2. SSPIF is set by hardware on completion of the Start.
3. SSPIF is cleared by software.
4. The MSSP module will wait for the required start time before any other operation takes place.
5. The user loads the SSPBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all 8 bits are transmitted. Transmission begins as soon as SSPBUF is written to.
7. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit in the SSPCON2 register.
8. The MSSP module generates an interrupt at the end of the 9th clock cycle by setting the SSPIF bit.
9. The user loads the SSPBUF with 8 bits of data.
10. Data is shifted out the SDA pin until all 8 bits are transmitted.
11. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit in the SSPCON2 register.
12. Steps 8 to 11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits in the SSPCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

FIGURE 27-20: I^2C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



27.5.7 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable (RCEN) bit in the SSPCON2 register.

Note:	The MSSP module must be in an idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and, upon each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the 8th clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable (ACKEN) bit in the SSPCON2 register.

27.5.7.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

27.5.7.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

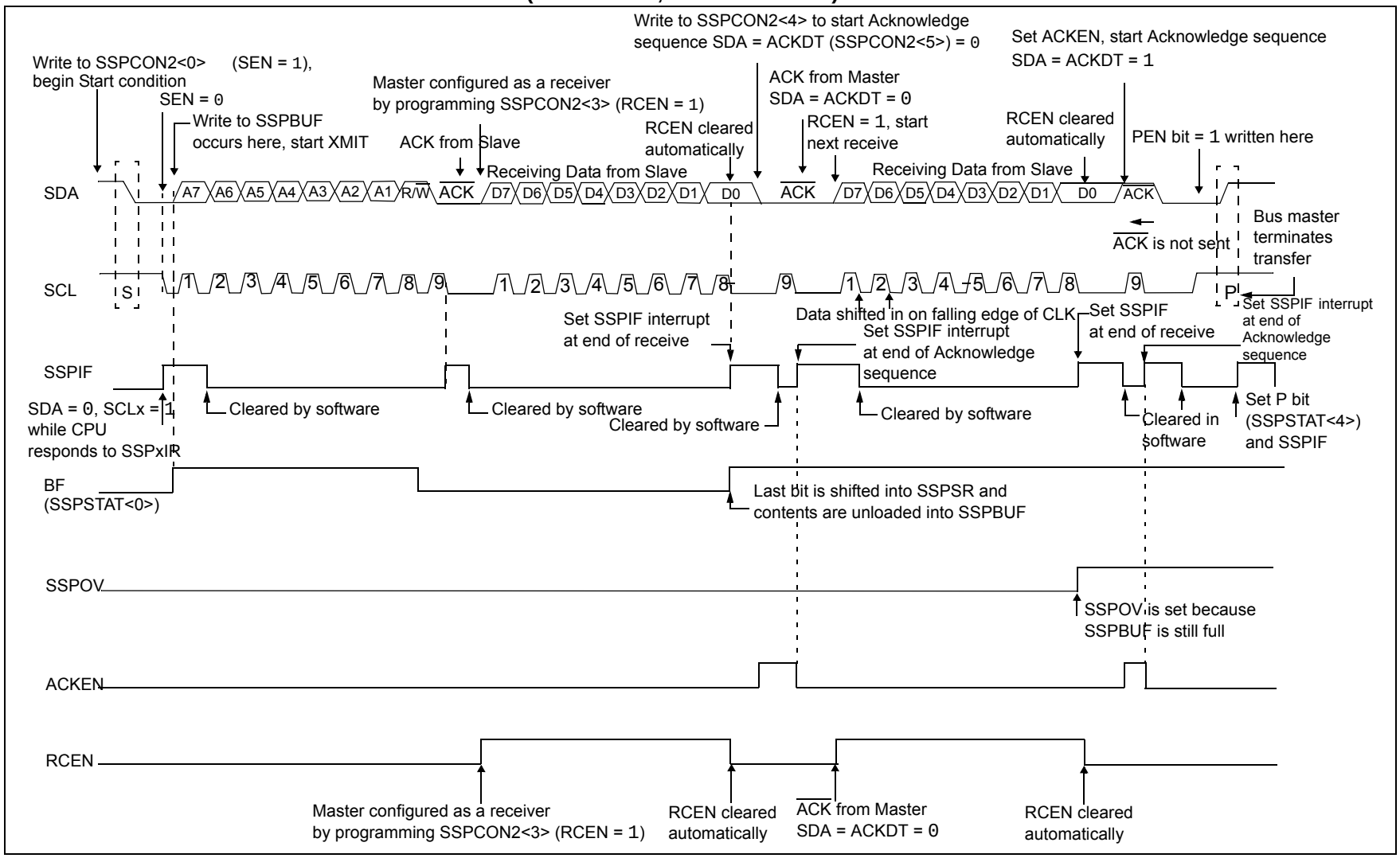
27.5.7.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

27.5.7.4 Typical Receive Sequence

1. The user generates a Start condition by setting the SEN bit in the SSPCON2 register.
2. SSPIF is set by hardware on completion of the Start.
3. SSPIF is cleared by software.
4. User writes SSPBUF with the slave address to transmit and the R/$\overline{W}$ bit set.
5. Address is shifted out the SDA pin until all 8 bits are transmitted. Transmission begins as soon as SSPBUF is written to.
6. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit in the SSPCON2 register.
7. The MSSP module generates an interrupt at the end of the 9th clock cycle by setting the SSPIF bit.
8. User sets the RCEN bit in the SSPCON2 register and the Master clocks in a byte from the slave.
9. After the 8th falling edge of SCL, SSPIF and BF are set.
10. Master clears SSPIF and reads the received byte from SSPUF, then clears BF.
11. Master sets $\overline{ACK}$ value sent to slave in ACKDT bit in the SSPCON2 register and initiates the $\overline{ACK}$ by setting the ACKEN bit.
12. Masters $\overline{ACK}$ is clocked out to the Slave and SSPIF is set.
13. The user clears SSPIF.
14. Steps 8 to 13 are repeated for each received byte from the slave.
15. Master sends a not $\overline{ACK}$ or Stop to end communication.

FIGURE 27-21: I^2C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



27.5.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable (ACKEN) bit in the SSPCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (T_{BRG}) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for T_{BRG}. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 27-22).

27.5.8.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, WCOL is set and the contents of the buffer are unchanged (the write does not occur).

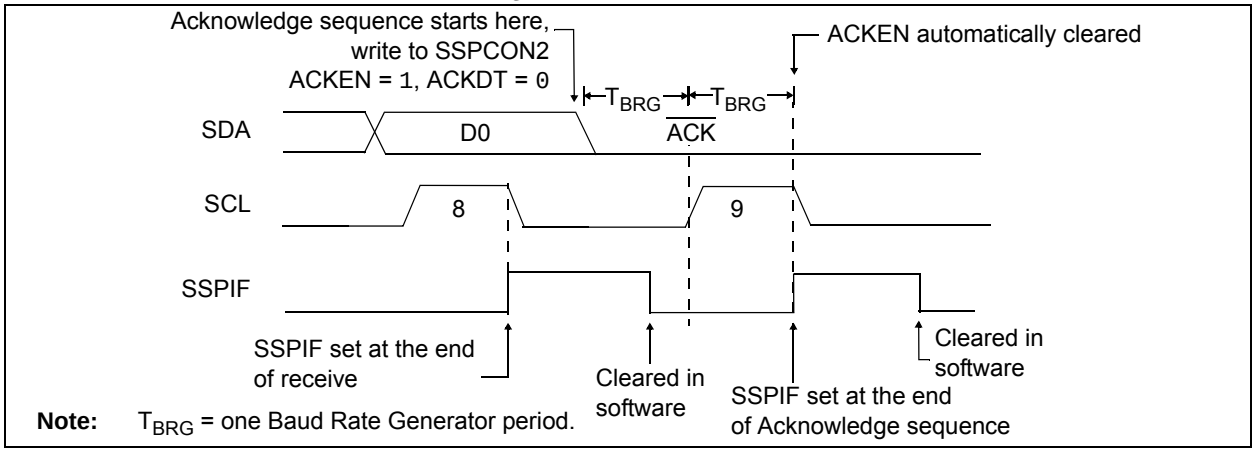
27.5.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit (PEN) in the SSPCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the 9th clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and then, one T_{BRG} (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit in the SSPSTAT register, is set. A T_{BRG} later, the PEN bit is cleared and the SSPIF bit is set (Figure 27-23).

27.5.9.1 WCOL Status Flag

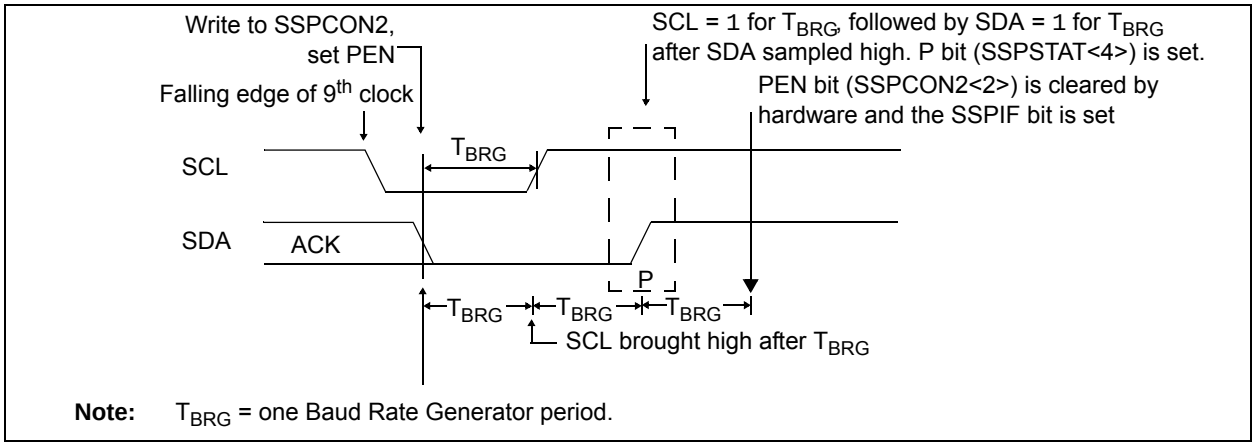
If the user writes the SSPBUF when a Stop sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 27-22: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: T_{BRG} = one Baud Rate Generator period.

FIGURE 27-23: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: T_{BRG} = one Baud Rate Generator period.

27.5.10 SLEEP OPERATION

While in Sleep mode, the I^2C slave module can receive addresses or data and, when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

27.5.11 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

27.5.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I^2C bus may be taken when the P bit in the SSPSTAT register is set or the bus is idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

27.5.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high, and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', a bus collision has taken place. The master sets the Bus Collision Interrupt Flag (BCLIF) and resets the I^2C port to its Idle state (Figure 27-24).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

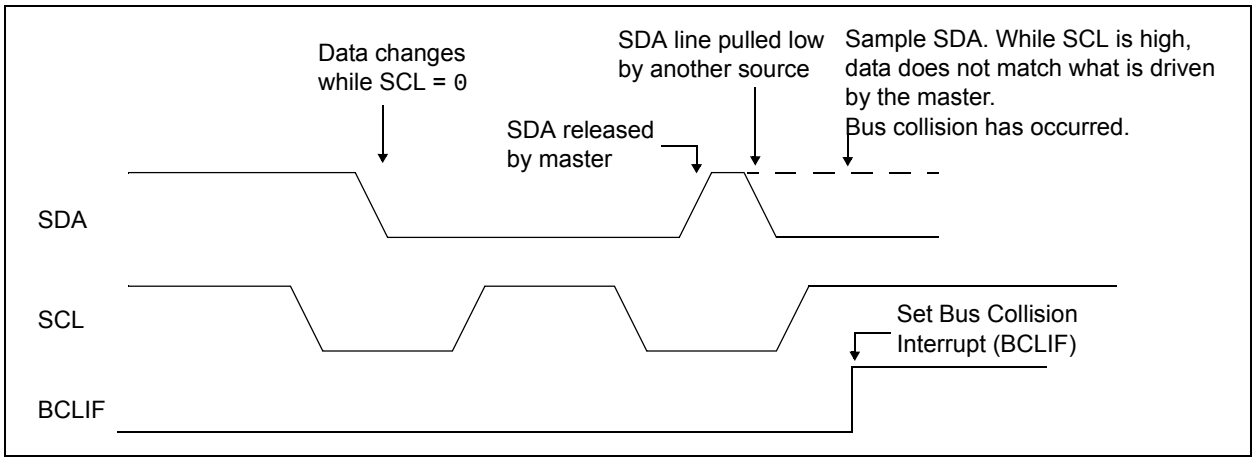
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit is set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is idle and the S and P bits are cleared.

FIGURE 27-24: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



27.5.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 27-25)

b) SCL is sampled low before SDA is asserted low (Figure 27-26)

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low or the SCL pin is already low, all of the following occur:

- the Start condition is aborted
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 27-25)

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 27-27). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note:	The reason why bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 27-25: BUS COLLISION DURING A START CONDITION (SDA ONLY)

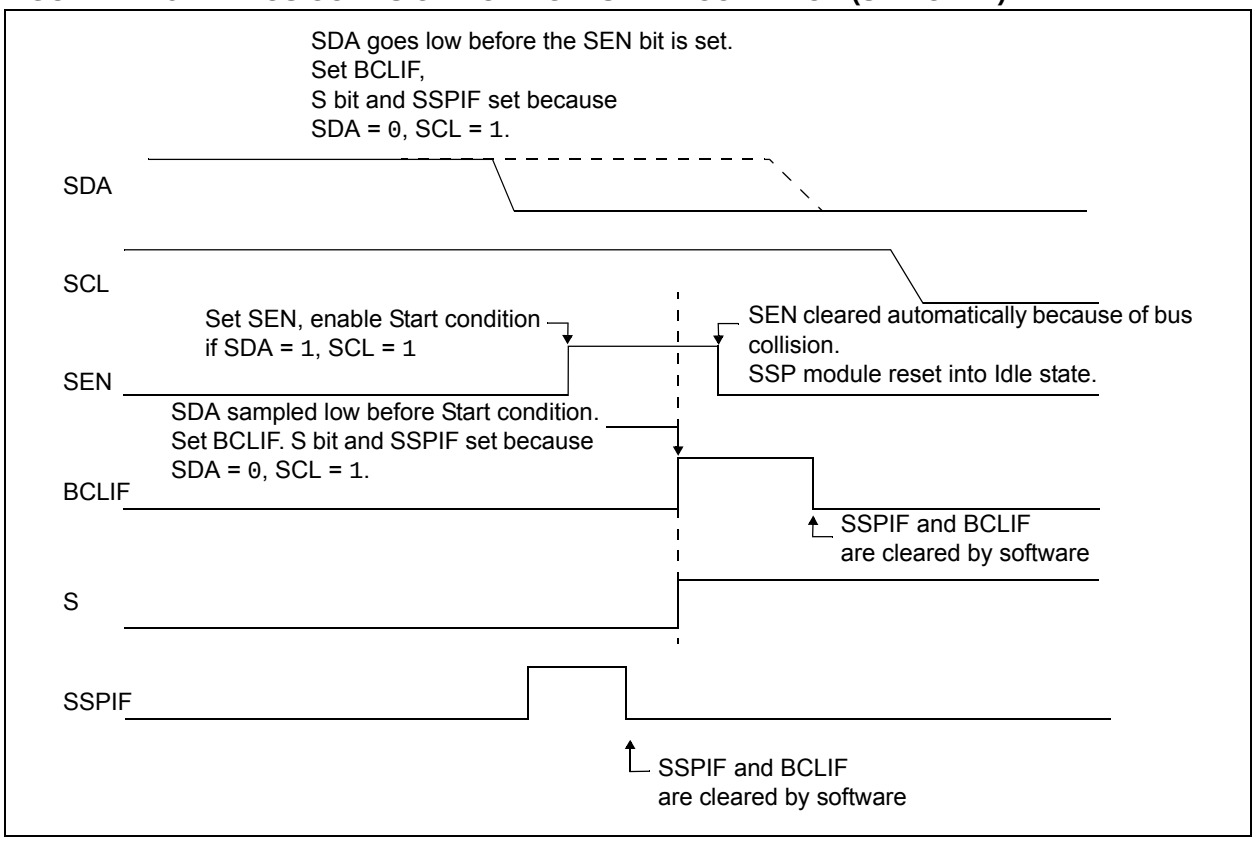


FIGURE 27-26: BUS COLLISION DURING A START CONDITION (SCL = 0)

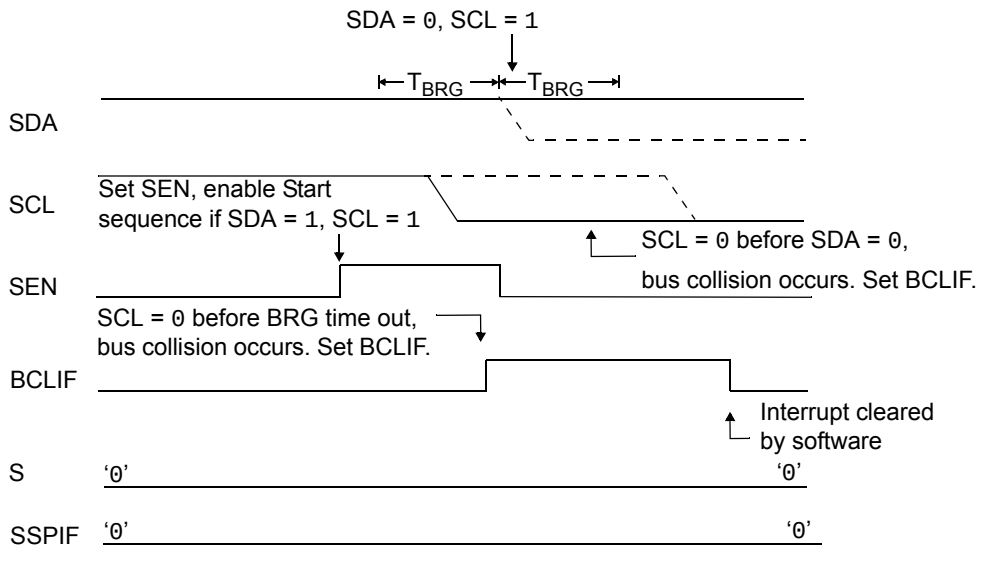
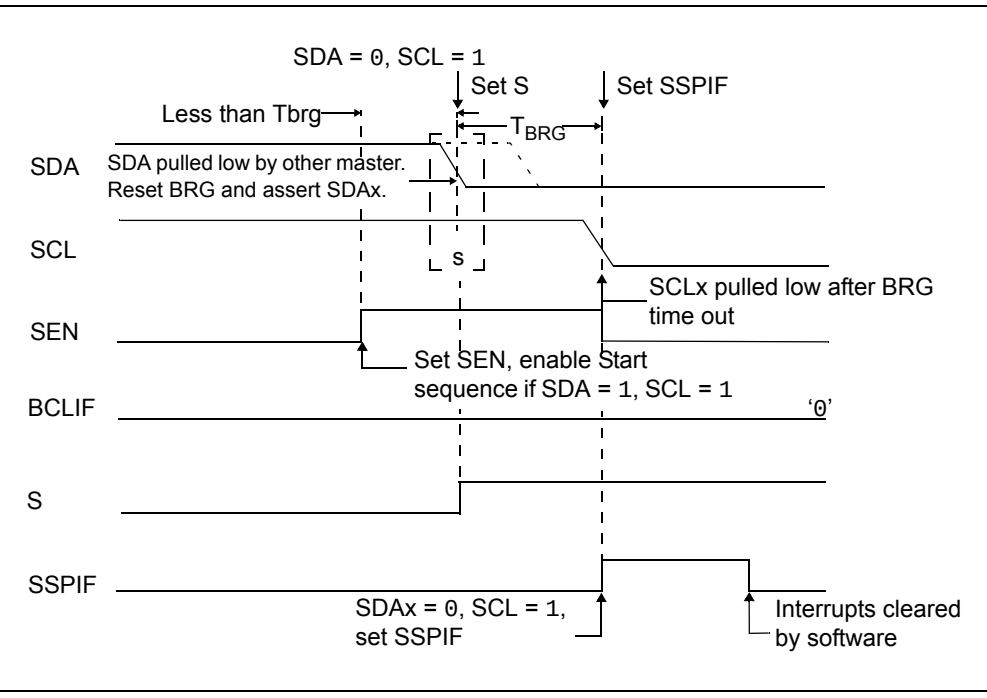


FIGURE 27-27: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



27.5.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) a low level is sampled on SDA when SCL goes from low level to high level

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPADD and counts down to zero. The SCL pin is then deasserted and, when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 27-28). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (refer to Figure 27-29).

If, at the end of the BRG time out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 27-28: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

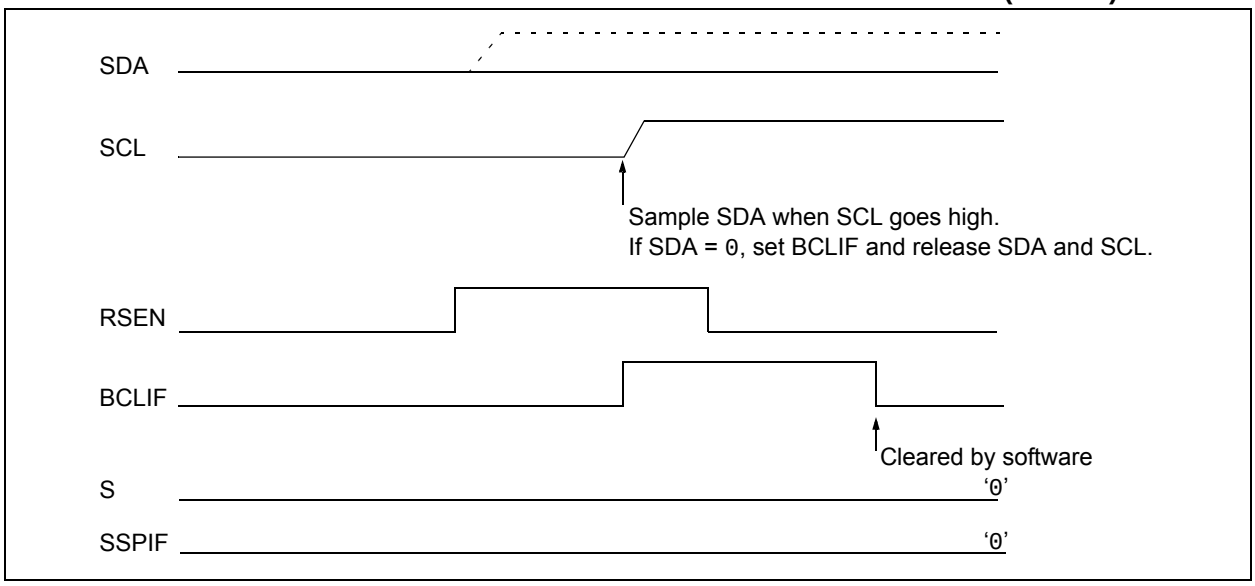
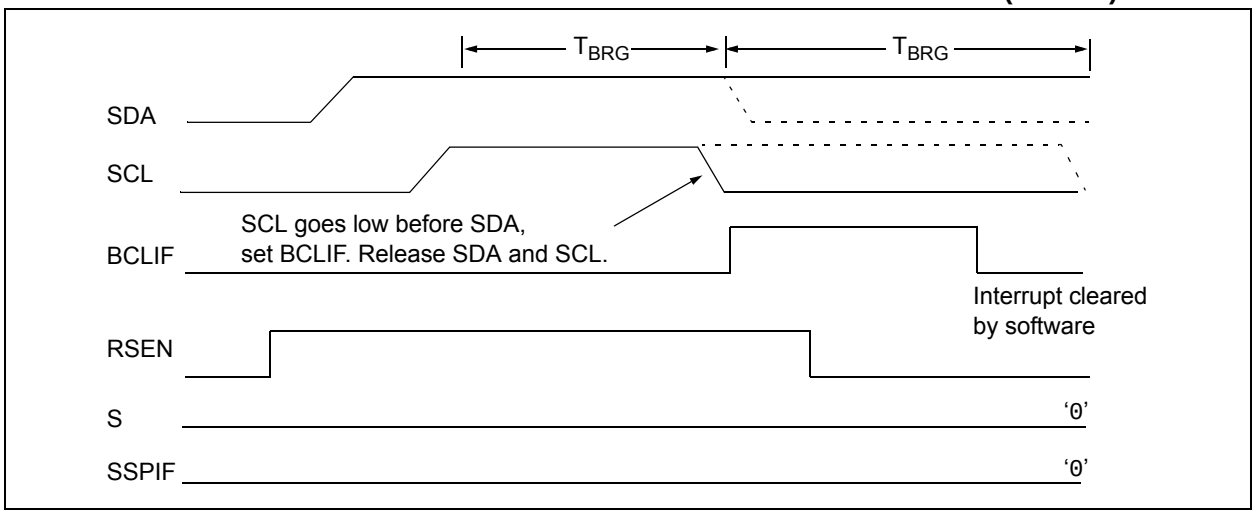


FIGURE 27-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 2)



27.5.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) after the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.

b) after the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 27-30). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 27-31).

FIGURE 27-30: BUS COLLISION DURING A STOP CONDITION (CASE 1)

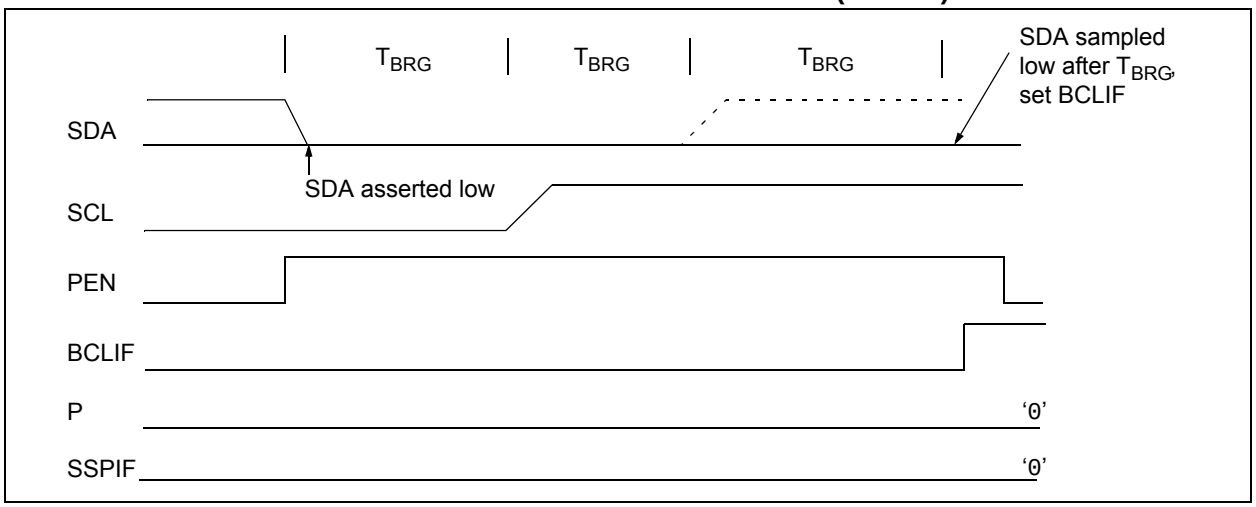


FIGURE 27-31: BUS COLLISION DURING A STOP CONDITION (CASE 2)

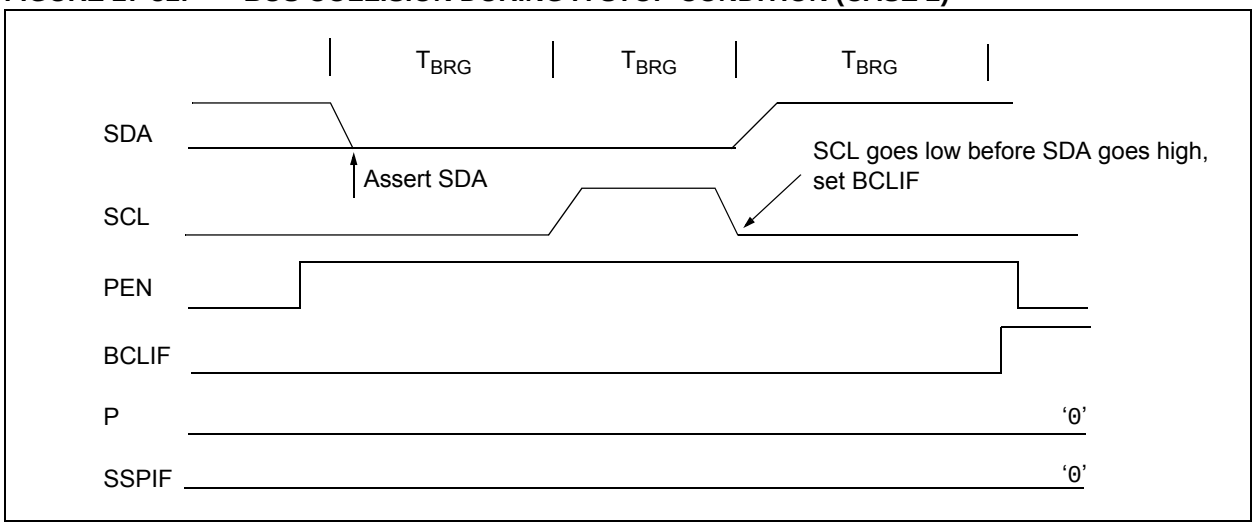


TABLE 27-2: SUMMARY OF REGISTERS ASSOCIATED WITH I²C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE	PEIE	T0IE	INTE	IOCE	T0IF	INTF	IOCF	102
PIE1	—	ADIE	BCLIE	SSPIE	CC2IE	CC1IE	TMR2IE	TMR1IE	103
PIR1	—	ADIF	BCLIF	SSPIF	CC2IF	CC1IF	TMR2IF	TMR1IF	106
TRISGPA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	127
TRISGPB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	TRISB1	TRISB0	131
SSPADD	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0	202
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								161*
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	198
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	200
SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	201
SSPMSK1	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	202
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	197
SSPMSK2	MSK27	MSK26	MSK25	MSK24	MSK23	MSK22	MSK21	MSK20	203
SSPADD2	ADD27	ADD26	ADD25	ADD24	ADD23	ADD22	ADD21	ADD20	203

Legend: – = unimplemented, read as '0'. Shaded cells are not used by the MSSP module in I²C mode.

* Page provides register information.

27.6 Baud Rate Generator

The MSSP module has a Baud Rate Generator available for clock generation in the I^2C Master mode. The Baud Rate Generator (BRG) reload value is placed in the SSPADD register. When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

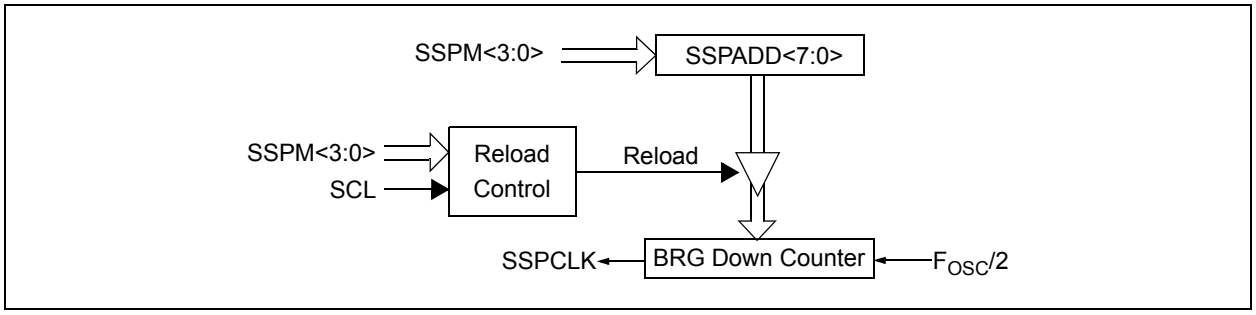
An internal signal “Reload” in Figure 27-32 triggers the value from SSPADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 27-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

EQUATION 27-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPADD + 1)(4)}$$

FIGURE 27-32: BAUD RATE GENERATOR BLOCK DIAGRAM



Note: Values of 0x00, 0x01 and 0x02 are not valid for SSPADD when used as a Baud Rate Generator for I^2C. This is an implementation limitation.

TABLE 27-3: MSSP CLOCK RATE W/BRG

F_{OSC}	F_{CY}	BRG Value	F_{CLOCK} (2 rollovers of BRG)
8 MHz	2 MHz	04h	400 kHz(1)
8 MHz	2 MHz	0Bh	166 kHz
8 MHz	2 MHz	13h	100 kHz

Note 1: The I^2C interface does not conform to the 400 kHz I^2C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

REGISTER 27-1: SSPSTAT: SSP STATUS REGISTER

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **SMP:** Data Input Sample bit
1 = Slew rate control disabled for standard-speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high-speed mode (400 kHz)

bit 6 **CKE:** Clock Edge Select bit
1 = Enable input logic so that thresholds are compliant with SM bus specification
0 = Disable SM bus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit
(This bit is cleared when the MSSP module is disabled; SSPEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit
(This bit is cleared when the MSSP module is disabled; SSPEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ bit information
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{\text{ACK}}$ bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full status bit
Receive:
1 = Receive complete; SSPBUF is full
0 = Receive not complete; SSPBUF is empty
Transmit:
1 = Data transmit in progress (does not include the ACK and Stop bits), SSPBUF is full
0 = Data transmit complete (does not include the ACK and Stop bits), SSPBUF is empty

REGISTER 27-2: SSPCON1: SSP CONTROL REGISTER 1

R/C/HS-0	R/C/HS-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>			
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	HS = Bit is set by hardware C = User cleared

bit 7 **WCOL:** Write Collision Detect bit

Master mode:

1 = A write to the SSPBUF register was attempted while the I^2C conditions were not valid for a transmission to be started.

0 = No collision

Slave mode:

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software).

0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit(1)

1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).

0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

In both modes, when enabled, these pins must be properly configured as input or output.

1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins(2)

0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit

In I^2C Slave mode:

SCL release control

1 = Enable clock

0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

In I^2C Master mode:

Unused in this mode

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

2: When enabled, the SDA and SCL pins must be configured as inputs.

3: SSPADD values of 0, 1 or 2 are not supported for I^2C mode.

REGISTER 27-2: SSPCON1: SSP CONTROL REGISTER 1 (CONTINUED)

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits

0000 = Reserved
0001 = Reserved
0010 = Reserved
0011 = Reserved
0100 = Reserved
0101 = Reserved
0110 = I^2C Slave mode, 7-bit address
0111 = I^2C Slave mode, 10-bit address
1000 = I^2C Master mode, clock = $F_{OSC}/(4 * (SSPADD+1))$ (3)
1001 = Reserved
1010 = Reserved
1011 = I^2C Firmware-Controlled Master mode (Slave idle)
1100 = Reserved
1101 = Reserved
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

2: When enabled, the SDA and SCL pins must be configured as inputs.

3: SSPADD values of 0, 1 or 2 are not supported for I^2C mode.

REGISTER 27-3: SSPCON2: SSP CONTROL REGISTER 2

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN(1)	RCEN(1)	PEN(1)	RSEN(1)	SEN(1)
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	H = Bit is set by hardware S = User set

bit 7 **GCEN:** General Call Enable bit (in I^2C Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPSR register
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in I^2C mode only)
1 = Acknowledge was not received
0 = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in I^2C mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in I^2C Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in I^2C Master mode only)
1 = Enables Receive mode for I^2C
0 = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in I^2C Master mode only)
SCK Release Control:
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Stop condition idle

bit 1 **RSEN:** Repeated Start Condition Enabled bit (in I^2C Master mode only)
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition idle

bit 0 **SEN:** Start Condition Enabled bit (in I^2C Master mode only)
In Master mode:
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition idle
In Slave mode:
1 = Clock stretching is enabled for both Slave Transmit and Slave Receive (stretch enabled)
0 = Clock stretching is disabled

Note 1: If the I^2C module is not in the Idle mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

REGISTER 27-4: SSPCON3: SSP CONTROL REGISTER 3

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **ACKTIM:** Acknowledge Time status bit (I^2C mode only)(1)
- 1 = Indicates the I^2C bus is in an Acknowledge sequence, set on 8th falling edge of SCL clock
- 0 = Not an Acknowledge sequence, cleared on 9th rising edge of SCL clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit (I^2C mode only)
- 1 = Enable interrupt on detection of Stop condition
- 0 = Stop detection interrupts are disabled(2)

bit 5 **SCIE**: Start Condition Interrupt Enable bit (I^2C mode only)
- 1 = Enable interrupt on detection of Start or Restart conditions
- 0 = Start detection interrupts are disabled(2)

bit 4 **BOEN:** Buffer Overwrite Enable bit

In I^2C Master mode:
This bit is ignored.

In I^2C Slave mode:
- 1 = SSPBUF is updated and $\overline{ACK}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
- 0 = SSPBUF is only updated when SSPOV is clear.

bit 3 **SDAHT:** SDA Hold Time Selection bit
- 1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
- 0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)

If, on the rising edge of SCL, SDA is sampled low when the module outputs a high state, the BCLIF bit in the PIR1 register is set and bus goes idle.
- 1 = Enable slave bus collision interrupts
- 0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit (I^2C Slave mode only)
- 1 = Following the 8th falling edge of SCL for a matching received address byte; CKP bit in the SSPCON1 register will be cleared and the SCL will be held low.
- 0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (I^2C Slave mode only)
- 1 = Following the 8th falling edge of SCL for a received data byte; slave hardware clears the CKP bit in the SSPCON1 register and SCL is held low.
- 0 = Data holding is disabled

Note 1: The ACKTIM status bit is only active when the AHEN bit or DHEN bit is set.

2: This bit has no effect in Slave modes where Start and Stop condition detection is explicitly listed as enabled.

REGISTER 27-5: SSPMSK1: SSP MASK REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-1 **MSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD<n> to detect I^2C address match
0 = The received address bit n is not used to detect I^2C address match

bit 0 **MSK<0>:** Mask bit for I^2C Slave mode, 10-bit Address
I^2C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPADD<0> to detect I^2C address match
0 = The received address bit 0 is not used to detect I^2C address match I^2C Slave mode, 7-bit address, the bit is ignored

REGISTER 27-6: SSPADD: MSSP ADDRESS AND BAUD RATE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

Master mode:

bit 7-0 **ADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) * 4)/F_{OSC}

10-Bit Slave mode — Most Significant Address byte:

bit 7-3 **Not used:** Unused for Most Significant Address byte. Bit state of this register is a "don't care". Bit pattern sent by master is fixed by I^2C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **ADD<2:1>:** Two Most Significant bits of 10-bit address.

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care".

10-Bit Slave mode — Least Significant Address byte:

bit 7-0 **ADD<7:0>:** Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

bit 7-1 **ADD<7:1>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care".

REGISTER 27-7: SSPMSK2: SSP MASK REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK2<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-1 **MSK2<7:1>:** Mask bits
1 = The received address bit n is compared to SSPADD2<n> to detect I^2C address match
0 = The received address bit n is not used to detect I^2C address match

bit 0 **MSK2<0>:** Mask bit for I^2C Slave mode, 10-bit Address
I^2C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPADD2<0> to detect I^2C address match
0 = The received address bit 0 is not used to detect I^2C address match I^2C Slave mode, 7-bit address, the bit is ignored

REGISTER 27-8: SSPADD2: MSSP ADDRESS 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD2<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
‘1’ = Bit is set	‘0’ = Bit is cleared	

Master mode:

bit 7-0 **ADD2<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) * 4)/F_{OSC}

10-Bit Slave mode — Most Significant Address byte:

bit 7-3 **Not used:** Unused for Most Significant Address byte. Bit state of this register is a “don’t care”. Bit pattern sent by master is fixed by I^2C specification and must be equal to ‘11110’. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **ADD2<2:1>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care”.

10-Bit Slave mode — Least Significant Address byte:

bit 7-0 **ADD2<7:0>:** Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

bit 7-1 **ADD2<7:1>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a “don’t care”.

NOTES:

28.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (AUSART)

28.1 AUSART Module Overview

The Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The AUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The features of that module include:

- Asynchronous and Synchronous modes
  - -Asynchronous (full duplex)
  - -Synchronous - Master (half duplex)
  - -Synchronous - Slave (half duplex)
- 8- and 9-bit data operations
- Single and Continuous Receive modes
- Address detect
- Two byte FIFOs for Transmit and Receive operations
- Majority bit detection in Asynchronous mode
- 8-bit Baud Rate generator with speed selection
  - -Fosc/16 or Fosc/64 for Asynchronous mode
  - -Fosc/4 for Synchronous mode
- Status bits for
  - -Framing Error
  - -Overrun Error
  - -Transmit Shift Register Status

28.2 Module Reset

When the SPEN (RCSTA<7>) is cleared, all USART state machines are held in Reset. This allows for software re-initialization of the module by toggling the SPEN bit. This also causes all status bits to be reset. All other R/W bits are available to the user, which allows them to preconfigure the module prior to setting the SPEN bit.

28.3 PIN PLACEMENT AND PORT INTERACTION

The bi-directional TX/CK pin is located on GPB6/AN7/TX/CK. If TRISB<6> is configured as input (‘1’), the USART control will automatically reconfigure the pin from input to output as needed.

28.4 USART ASYNCHRONOUS MODE

In this mode, the USART uses standard non-return-to-zero (NRZ) format (one START bit, eight or nine data bits and one STOP bit). The BRG is used to derive the baud rate frequencies from the system clock. The USART transmits and receives the LSB first. The transmitter and receiver are functionally independent, but use the same data format and baud rate. The BRG produces a clock, either x4, x16 or x64 of the bit shift rate, depending on its configuration (see **Section 28.4.2 “Asynchronous Receive Mode”**). Parity is not supported by the hardware, but can be implemented in software using the ninth data bit option. Asynchronous mode is stopped during Sleep. Asynchronous mode is selected by clearing the SYNC (TXSTA<4>) bit.

28.4.1 ASYNCHRONOUS TRANSMIT MODE

The USART transmitter block diagram is shown in Figure 28-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer (TXREG). The TXREG register is loaded with data via software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, TSR is loaded with new data from the TXREG register (if available). The transmit register (TXREG) is double buffered. As the user writes to TXREG, the data is transferred from the buffer to the transmit shift register (TSR), thus freeing up the buffer register. The interrupt flag TXIF is set as long as TXEN (TXSTA<5>) bit is set and TXREG is empty, indicating that the transmit buffer register (TXREG) is enabled and free to accept another word. Flag bit TXIF (Transmit Buffer Empty) is read-only and will be set, regardless of the state of the TXIE bit, and cannot be cleared in software. It will be reset only when new data is loaded into TXREG.

Transmission is enabled by setting enable bit TXEN. The actual transmission will not occur until the TXREG register has been loaded with data and the BRG has produced a shift clock (Figure 28-2). The transmission can also be started by first loading TXREG and then setting enable bit TXEN. Normally, when transmission is first started, TSR is empty. At that point, transfer to TXREG will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 28-3). Clearing the enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the GPB6/AN7/TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, transmit bit TX9 (TXSTA<6>) is set and the ninth bit written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to TXREG because the data write to TXREG results in an immediate transfer of the data to the TSR (if empty).

While TXIF indicates the status of the transmit buffer register, the TRMT (TXSTA<1>) bit indicates the status of the transmit operation. The TRMT bit is cleared automatically upon a byte transfer from TXREG to the shift register and is set at the end of a stop bit. A '1' value in the TRMT bit signifies that the transmit state machine is idle. The TRMT bit is read-only and is valid for both Asynchronous and Synchronous transmission. No interrupt is associated with the TRMT bit. See Figures 28-2 and 28-3 for timing details of the TRMT bit.

When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, set GIE (INTCON<7>) and PEIE (INTCON<6>) bits.

FIGURE 28-1: USART TRANSMIT BLOCK DIAGRAM

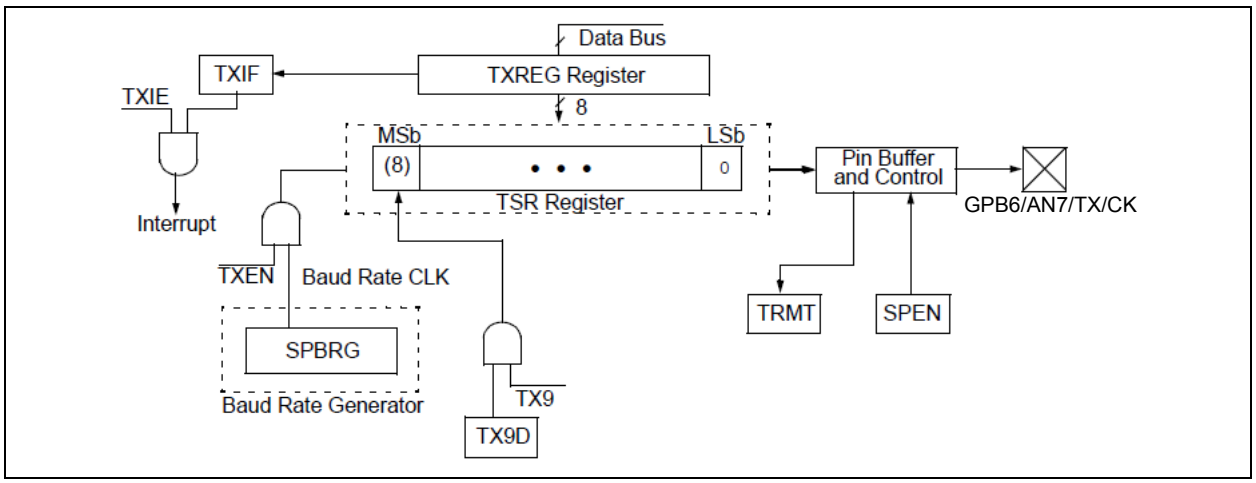


FIGURE 28-2: ASYNCHRONOUS MASTER TRANSMISSION

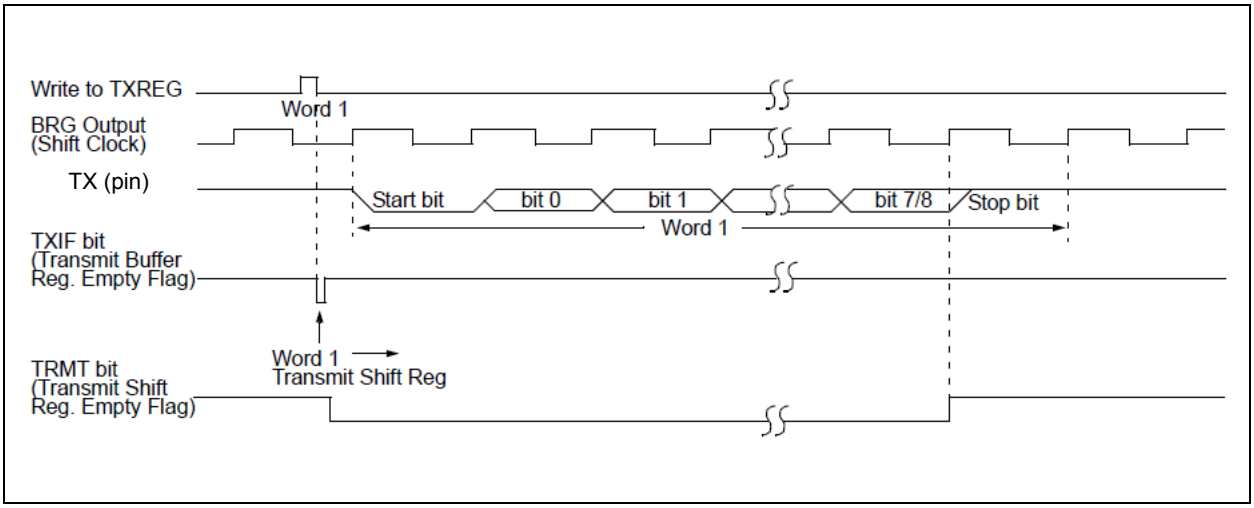
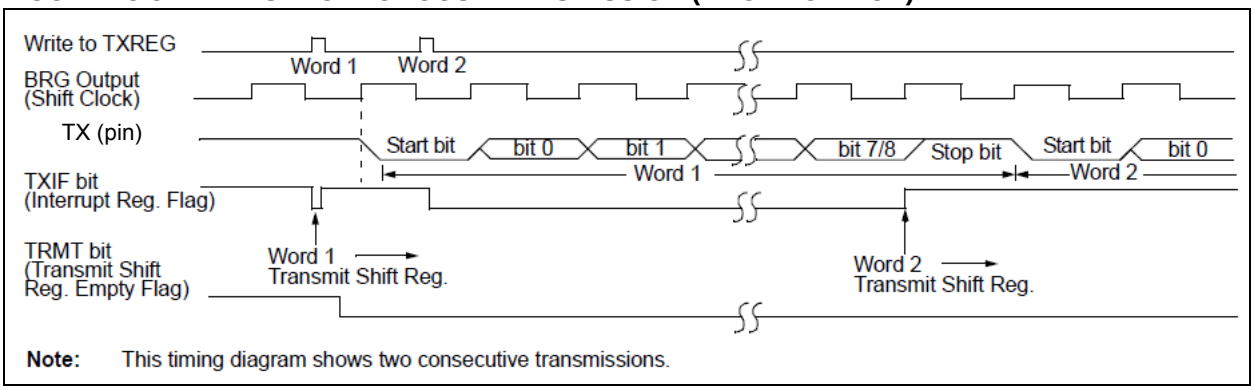


FIGURE 28-3: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

28.4.2 ASYNCHRONOUS RECEIVE MODE

The receiver block diagram is shown in Figure 28-4. The data is received on the GPB1/AN4/RX pin and drives the data recovery block. The data recovery block is a shifter operating at x64, x16 or x4 times the baud rate. The main receive serial shifter operates at the bit rate or at Fosc. Once asynchronous mode is selected, reception is enabled by setting the CREN (RCSTA<4>) bit.

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to RCREG (if empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The interrupt can be enabled by setting the RCIE (PIE1<5>) bit. Flag bit RCIF is read-only and cleared by hardware. It is cleared when RCREG has been read and is empty. RCREG is double buffered (two deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR. On detection of the STOP bit of the third byte, if RCREG is full, the overrun error bit OERR (RCSTA<1>) will be set. The word in RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to RCREG are inhibited and no further data will be received. The OERR bit can then be cleared in software. Framing error bit FERR (RCSTA<2>) is set if a STOP bit is detected as clear. The FERR bit and the ninth receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values. The user will need to read the RCSTA register before reading RCREG in order to not lose the old FERR and RX9D data. The USART module has a special provision for multi-processor communication. When the RX9 bit is set in the RCSTA register, 9-bits are received and the ninth bit is placed in the RX9D status bit of the RSTA register. The port can be programmed such that when the stop bit is received, the serial port interrupt will only activated if the RX9D bit is set. This feature is enabled by setting the ADDEN bit in the RCSTA register and can be used in a multi-processor system in the following manner.

To transmit a block of data in a multi-processor system, the master processor must first send an address byte that identifies the target slave. An address byte is identified by the RX9D bit being a '1' (instead of a '0' for a data byte). If the ADDEN bit is set in the slave's

RCSTA register, all data bytes will be ignored. However, if the ninth received bit is equal to a '1', indicating that the received byte is an address, the slave will be interrupted and the contents of the Receive Shift Register (RSR) will be transferred into the receive buffer. This allows the slave to be interrupted only by addresses, so that the slave can examine the received byte to see if it is addressed. The addressed slave will then clear its ADDEN bit and prepare to receive data bytes from the master. When the ADDEN bit is set, all data bytes are ignored. Following the STOP bit, the data will not be loaded into the receive buffer and no interrupt will occur. If another byte is shifted into the RSR, the previous data byte will be lost.

The ADDEN bit will only take affect when the receiver is configured in 9-bit mode. To indicate that a reception is in progress, the RCIDL bit (BAUDCTL<6>) reflects the current state of the receive operation. This bit is cleared ('0') on the leading edge of a start bit and set ('1') upon the end of a stop bit. A '1' value in the RCIDL bit signifies that the receive state machine is idle. The RCIDL bit is read-only and is valid for both Asynchronous and Synchronous receptions. No interrupt is associated with the RCIDL bit. See Figures 28-5, 28-6 and 28-7 for timing details of the RCIDL signal.

When setting up an Asynchronous Reception, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Set ADDEN if address detect is needed.
6. Enable the reception by setting bit CREN.
7. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing enable bit CREN.
11. If using interrupts, set GIE (INTCON<7>) and PEIE (INTCON<6>) bits.

FIGURE 28-4: USART RECEIVE BLOCK DIAGRAM

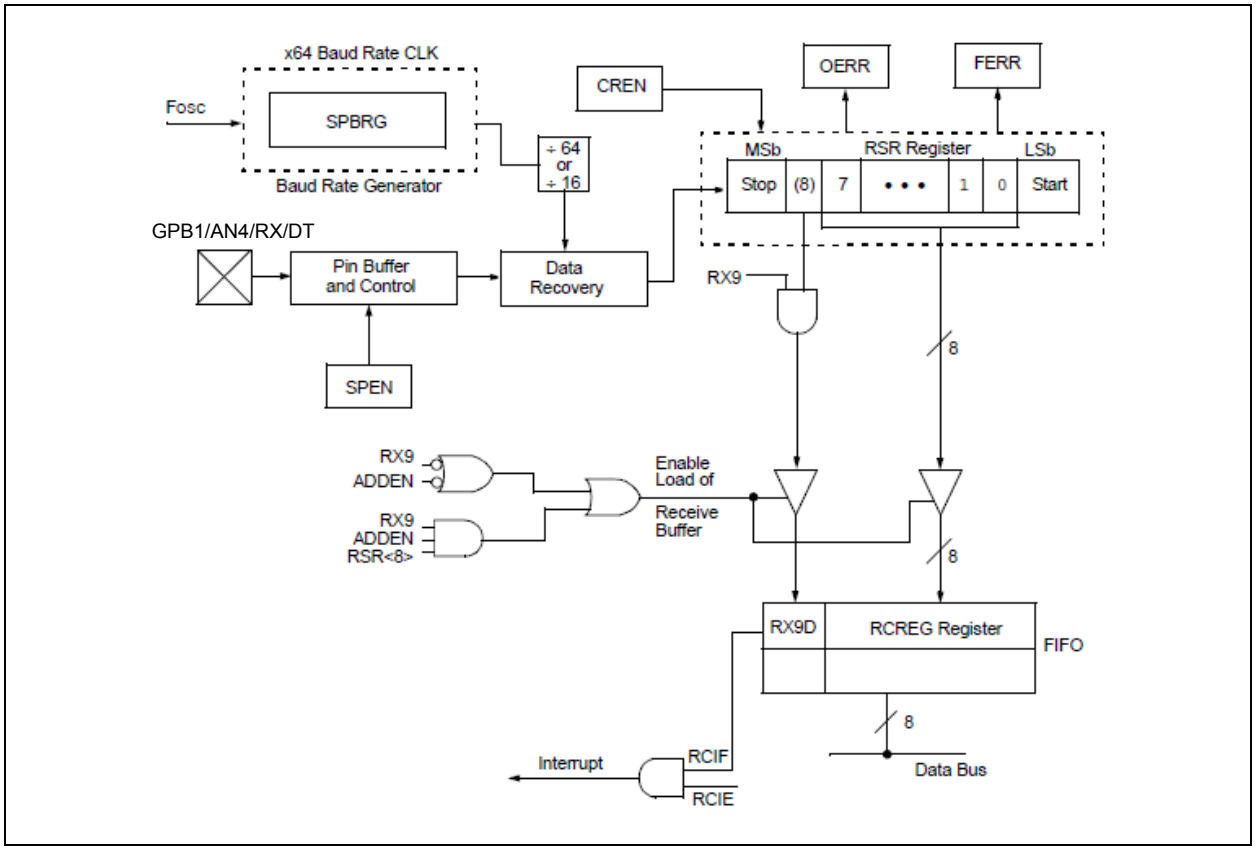
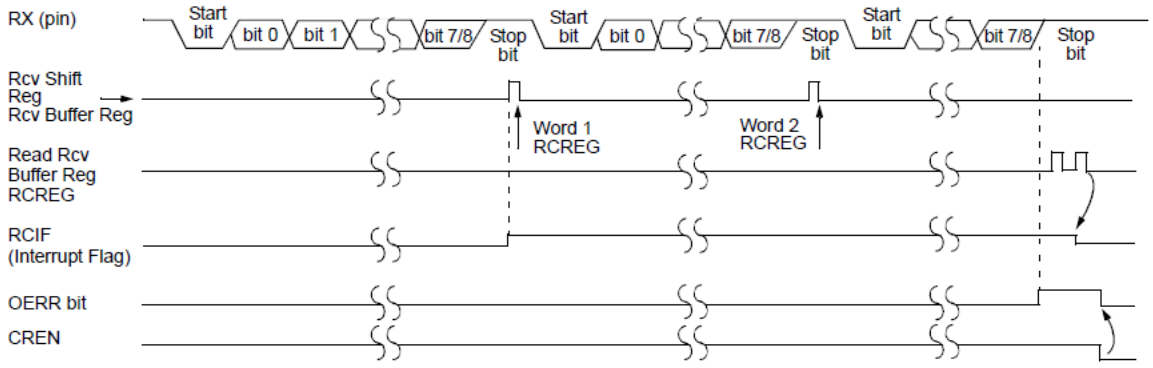
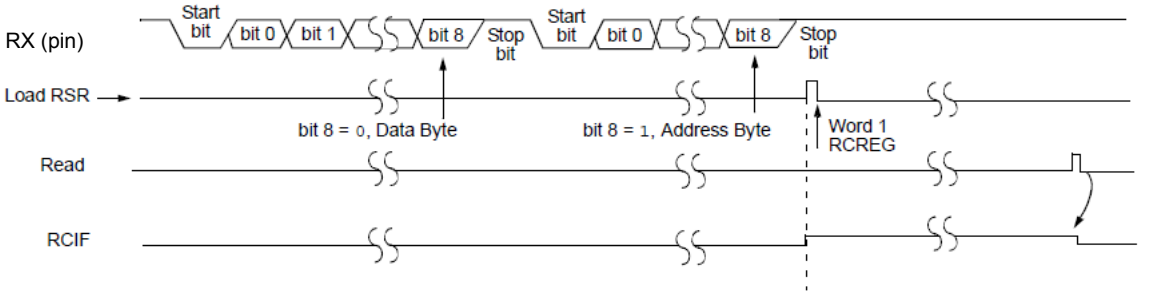


FIGURE 28-5: ASYNCHRONOUS RECEPTION



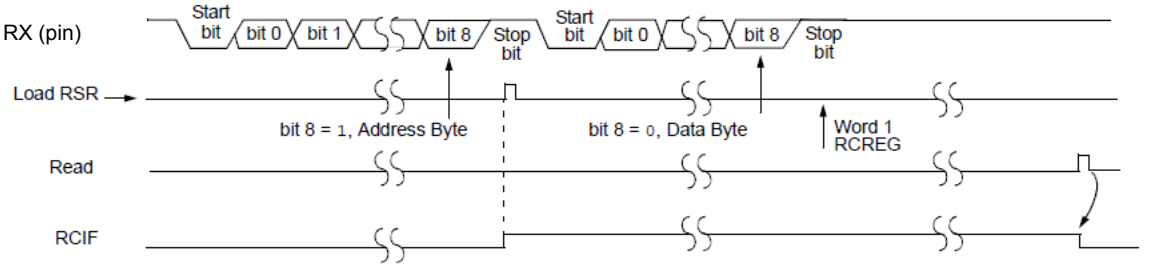
Note: This timing diagram shows three words appearing on the RX input. The RCREG (Receive Buffer) is read after the third word, causing the OERR (Overrun Error) bit to be set.

FIGURE 28-6: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the RCREG (Receive Buffer) because ADDEN = 1.

FIGURE 28-7: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the RCREG (Receive Buffer) because ADDEN was not updated and still = 0.

28.5 USART SYNCHRONOUS MASTER MODE

In Synchronous Master mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition, enable bit SPEN (RCSTA<7>) is set in order to configure the GPB6/AN7//TX/CK and GPB1/AN4/RX/DT I/O pins to CK (clock) and DT (data) lines, respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit CSRC (TXSTA<7>).

28.5.1 SYNCHRONOUS MASTER TRANSMIT MODE

Synchronous Master transmit mode works similarly to Asynchronous Transmit mode, **Section 28.4.1 "Asynchronous Transmit Mode"**. In Synchronous Transmit mode, the first data byte will be shifted out on the next available rising edge of the CK line. Data out is stable relative to the falling edge of the synchronous clock.

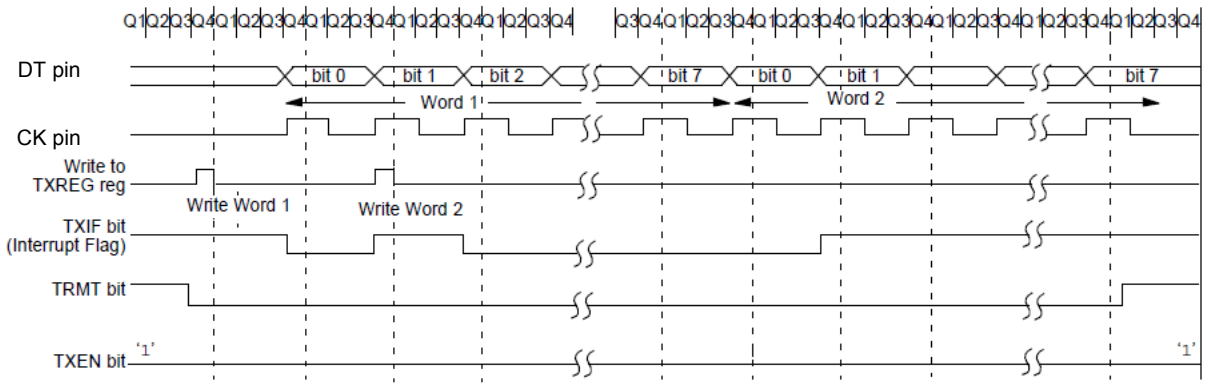
Clearing enable bit TXEN (TXSTA<5>) during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to high-impedance. If either bit CREN (RCSTA<4>) or bit SREN (RCSTA<5>) is set during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception).

The CK pin will remain an output if bit CSRC (TXSTA<7>) is set (internal clock). The transmitter logic, however, is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting, since bit TXEN is still set. The DT line will immediately switch from high-impedance Receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

Steps to follow when setting up a Synchronous Master Transmission:

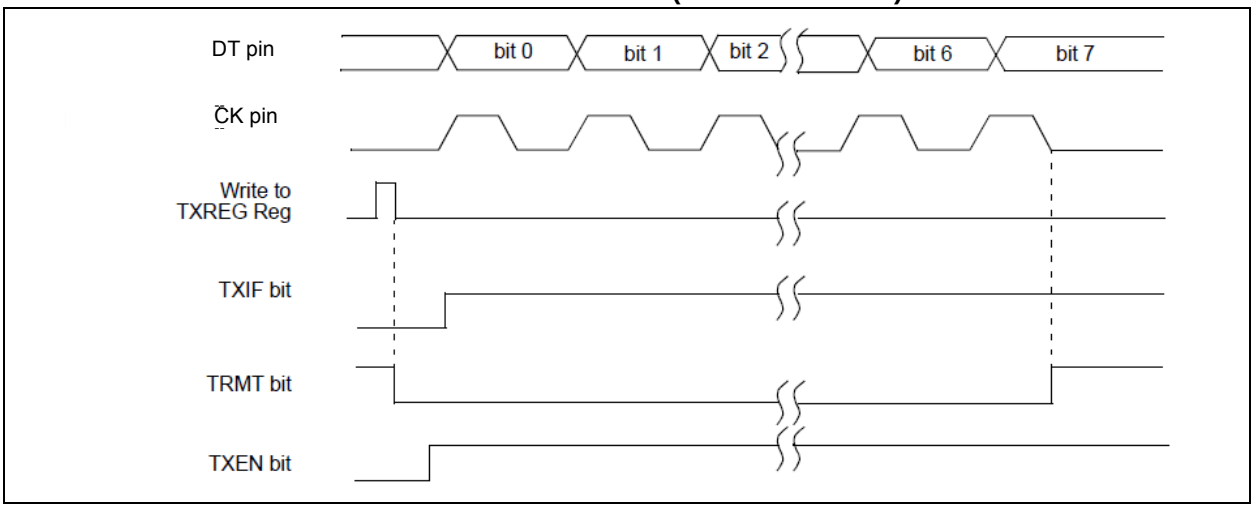
1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 28-8: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.

FIGURE 28-9: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



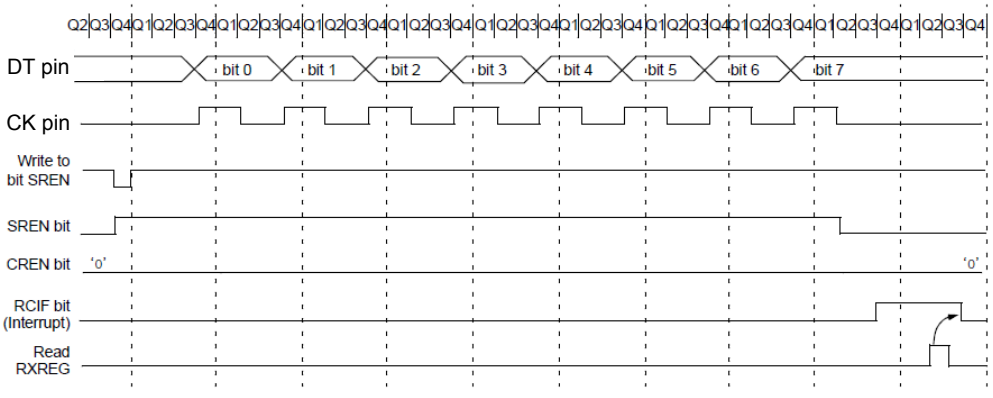
28.5.2 SYNCHRONOUS MASTER RECEIVE MODE

Synchronous Master Receive mode works similarly to Asynchronous Receive mode, **Section 28.4.2 "Asynchronous Receive Mode"**. In Synchronous Receive mode, reception is enabled by setting either enable bit SREN (RCSTA<5>), or enable bit CREN (RCSTA<4>). Data is sampled on the GPB1/AN4/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence.

When setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. CREN and SREN bits are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 28-10: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Sync Master mode with bit SREN = 1 and bit BRG = 0.

28.5.3 USART SYNCHRONOUS SLAVE MODE

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the GPB6/AN7//TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

28.5.3.1 Synchronous Slave Transmit Mode

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

1. The first word will immediately transfer to the TSR register and transmit.
2. The second word will remain in TXREG register.
3. Flag bit TXIF will not be set.
4. When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
5. If enable bit TXIE is set, the interrupt will wake the chip from Sleep and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Transmission, follow these steps:

1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. Clear bits CREN and SREN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set bit TX9.
5. Enable the transmission by setting enable bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

28.5.3.2 Synchronous Slave Receive Mode

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode. Bit SREN is a “don't care” in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during Sleep. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Reception, follow these steps:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete and an interrupt will be generated, if enable bit RCIE was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

REGISTER 28-1: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS: 11Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **SPEN[(1)]:** Serial Port Enable bit
1 = Serial port enabled - configures GPB6/AN7//TX/CK and GPB1/AN4/RX/DT pins as serial port pins
0 = Serial port disabled - module and its state machines held in Reset

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Unused in this mode - value ignored
Synchronous mode - master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave:
Unused in this mode - value ignored

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared
0 = Disables continuous receive
Synchronous mode - master:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
RX9 = 1:
1 = Enables address detection - enable interrupt and load of the receive buffer when the ninth bit in the receive buffer is set
0 = Disables address detection - all bytes are received, and ninth bit can be used as parity bit
RX9 = 0:
Unused in this mode

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of received data (can be parity bit)

Note 1: The USART module automatically changes the pin from tri-state to drive as needed. Configure TRISB<1> = 1 and TRISB<6> = 1.

REGISTER 28-2: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS: 11EH)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Unused in this mode - value ignored
Synchronous mode:
1 = Master mode - Clock generated internally from BRG
0 = Slave mode - Clock from external source

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN(1):** Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode - value ignored

bit 1 **TRMT:** Transmit Operation Idle Status bit
1 = Transmit Operation Idle
0 = Transmit Operation Active

bit 0 **TX9D:** 9th bit of transmit data; can be used as parity bit

Note 1: SREN/CREN overrides TXEN in Synchronous mode.

REGISTER 28-3: SPBRG: BAUD RATE GENERATOR REGISTER (ADDRESS: 11BH)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **BRG<7:0>:** Lower 8-bits of the Baud Rate Generator

29.0 APPLICATION HINTS

This chapter presents the typical steps that must be performed by the user in order to start-up the application based on the MCP19214/5 controllers.

29.1 Part Calibration

The calibration step must be performed in order to achieve a good accuracy of the controlled parameters. The calibration coefficients are determined during production and they are stored into the non-volatile memory of MCP19214/5. User must read these coefficients from the internal memory and store them into the specific registers. Refer to **Section 9.0 "Device Calibration"** for details about each calibration coefficient. The user can run the code snippet from Example 29-1 to perform the calibration of the part.

29.2 Configure The Analog Controllers

The configuration of the analog controllers and the additional protection circuitry must be performed before starting the converter.

For convenience, Figure 29-1 depicts the simplified diagram of the internal PWM controllers. Near each functional block there is a short list with the associated registers.

The reference voltages of the voltage loops are controlled by the VREFCON1 and VREFCON2 registers. These are 8-bit registers therefore, the output voltage of the converter can be controlled in 256 steps. The typical adjustment step is 8 mV. If D_R is the output voltage feedback divider's ratio, the regulating point can be calculated with the following equation:

EQUATION 29-1:

$$VOUTx = VREFCONx \cdot 0.008 \cdot \frac{1}{D_R}$$

The reference voltages for the current loops are controlled by the CREFCON1 and CREFCON2 registers. The output current of the converter can be adjusted in 256 steps. If G_S is the gain of the output current sense sensor (in A/V), the regulating point can calculated with the following equation:

EQUATION 29-2:

$$IOUTx = CREFCON_x \cdot 0.004 \cdot G_S$$

If the output current sensor is a shunt resistor, the gain of the sensor is:

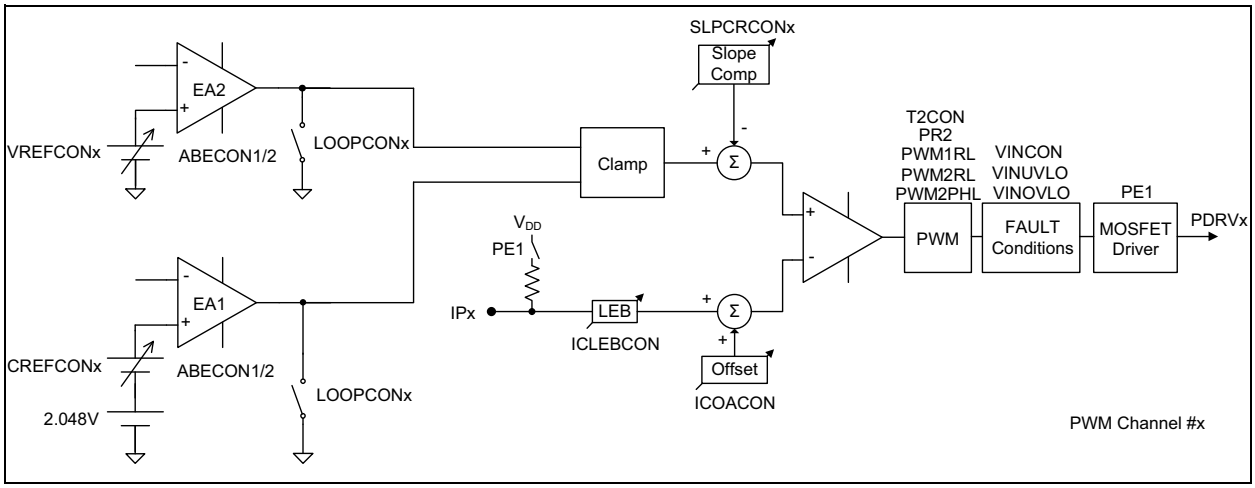
EQUATION 29-3:

$$G_S = 10 \cdot R_S$$

In the above equation, 10 is the typical value of the gain of current sense differential amplifier. The user can calculate the real value of the differential amplifiers gain measured during production phase, using Equation 9-1 and Equation 9-2.

Both parameters, the output voltage and the output current, are affected by the accuracy of the external components like the resistors of the voltage divider and the shunt resistor. The final accuracy of the converter can be further improved by using specific calibration functions in firmware.

FIGURE 29-1: ANALOG PWM CONTROLLER AND THE ASSOCIATED CONTROL REGISTERS



EXAMPLE 29-1: CODE FOR PERFORMING CALIBRATION

```
#define CAL_ADR_HI          0x20    //the start address of the calibration coefficients area
#define CAL_ADR_LO          0x80
#define CAL_ADR_NUM         11      //number of the calibration coefficients

unsigned char temp;

    for(temp = 0; temp < CAL_ADR_NUM; temp++)  //read the entire calibration coefficients area
    {
      PMCON1bits.CALSEL = 1;       //set the CALSEL bit in order to access the registers from Bank 4
      PMADRH = CAL_ADR_HI;
      PMADRL = CAL_ADR_LO + temp;
      PMCON1bits.RD = 1;           //initiates a program memory read
      NOP();
      NOP();
      if(PMADRL == 0x80)           //the values from address 0x2080 must be stored in DCSCAL1 and GMCAL1
        {
            PMCON1bits.CALSEL = 0;
            DCSCAL1     = PMDATH;
            GMCAL1      = PMDATL;
        } else
        if(PMADRL == 0x81)         //the values from address 0x2081 must be stored in DCSCAL2 and GMCAL2
        {
            PMCON1bits.CALSEL = 0;
            DCSCAL2     = PMDATH;
            GMCAL2      = PMDATL;
        } else
        if(PMADRL == 0x82)         //the values from address 0x2082 must be stored in VRCAL and DACBGRCAL
        {
            PMCON1bits.CALSEL = 0;
            VRCAL     = PMDATH;
            DACBGRCAL = PMDATL;
        } else
        if(PMADRL == 0x83)         //the values from address 0x2083 must be stored in PDSCAL, ADBT and TTCAL
        {
            PMCON1bits.CALSEL = 0;
            PDSCAL      = PMDATH;
            ADBT        = PMDATL >> 4;
            TTCAL       = PMDATL & 0x07;
        } else
       if(PMADRL == 0x84)        //the value from this address can be used to calibrate the temperature sensor
        {
            PMCON1bits.CALSEL = 0;
            // use 10 bit value as 30 C temperature measurement
        } else
        if(PMADRL == 0x85)       //the value from address 0x2085 must be stored in OSCCAL register
        {
            PMCON1bits.CALSEL = 0;
            OSCCAL = PMDATL;
        } else
        if(PMADRL == 0x86)       //the value from address 0x2086 must be stored in EACAL2 register
        {
            EACAL2 = PMDATL;
        } else
        if(PMADRL == 0x88)       //the value from address 0x2088 must be stored in DACCAL2 register
        {
            DACCAL2 = PMDATL;
        } else
        if(PMADRL == 0x89)
        {
            DACCAL1 = PMDATL;    //the value from address 0x2089 must be stored in DACCAL1 register
        } else
        if(PMADRL == 0x8A)
        {
            EACAL1 = PMDATL;     //the value from address 0x208A must be stored in EACAL1 register
        }
    }
```

In order to avoid severe overshoot of the controlled parameter (voltage or current) during start-up sequence or when the converter restarts at the output of each error amplifier, there is a switch that resets the compensation network. These switches are under the control of the user and the control bits are located in registers LOOPCON1 and LOOPCON2.

In order to improve the current consumption, the error amplifiers can be disabled if they are not used in the actual configuration/state of the hardware. The control bits are located in registers ABECON1 and ABECON2.

The ISx inputs are used to sense the inductor's current necessary for the control of the regulating loop. Each ISx input has a week pull-up resistor to the internal AV_{DD} node. These resistors provide a protection against an open circuit condition on the current sense circuitry (e.g., the shunt resistor defective). These pull-up resistors can be disabled using the associated control bits from register PE1.

The Leading Edge Blanking Circuit associated with ISx inputs can be controlled by the bits from register ICLEBCON. During the blanking period, any signal applied to ISx inputs will be ignored and will not produce the reset of the current PWM cycle.

An adjustable DC offset voltage can be added to the ISx input signal. This offset allows a more flexible control of the point where the controller starts to limit the peak of the inductor current. The value of this offset is controlled by the bits of register ICOACON.

The Peak Current mode control requires a programmable ramp for the so-called slope compensation. In the case of MCP19214/5 controllers, this ramp is internally generated and is subtracted from the signal produced by the error amplifiers. The amplitude of this ramp is controlled by the bits of registers SLPCRCON1 and SLPCRCON2. The internal ramp generator can be disabled by setting bit SLPBY in registers SLPCRCONx.

The MCP19214/5 controllers integrate the Undervoltage Lockout (UVLO)/Overvoltage Lockout (OVLO) circuit. This circuit monitors the input voltage and, if the value of this parameter is outside the programmed limits, it disables the output of the MOSFETs drivers. The behavior of UVLO/OVLO circuit is controlled by the bits of the VINCON register. The value of the thresholds is controlled by the bits of VINUVLO and VINOVLO registers. The UVLO/OVLO circuit can generate a specific interrupt that informs the core if an event related to input voltage occurs.

The MOSFETs drivers can be disabled using the associated bits from register PE1. The MOSFETs drivers circuits are equipped with an UVLO circuit. If V_{DR} voltage is below a certain level, the outputs of the drivers are disabled. There are two thresholds for this UVLO circuit: 2.7V and 5.4V (typical). Selection between these thresholds are done using bit DRUVSEL from the ABECON1 register.

NOTES:

30.0 INSTRUCTION SET SUMMARY

The MCP19214/5 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type, and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 30-1, while the various opcode fields are summarized in Table 30-1.

Table 30-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a normal instruction execution time of 1 µs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

30.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (RMW) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTA instruction will read PORTGPA, clear all the data bits, then write the result back to PORTGPA. This example would have the unintended consequence of clearing the condition that sets the IOCIF flag.

TABLE 30-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
$\overline{TO}$	Time-Out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

FIGURE 30-1: GENERAL FORMAT FOR INSTRUCTIONS

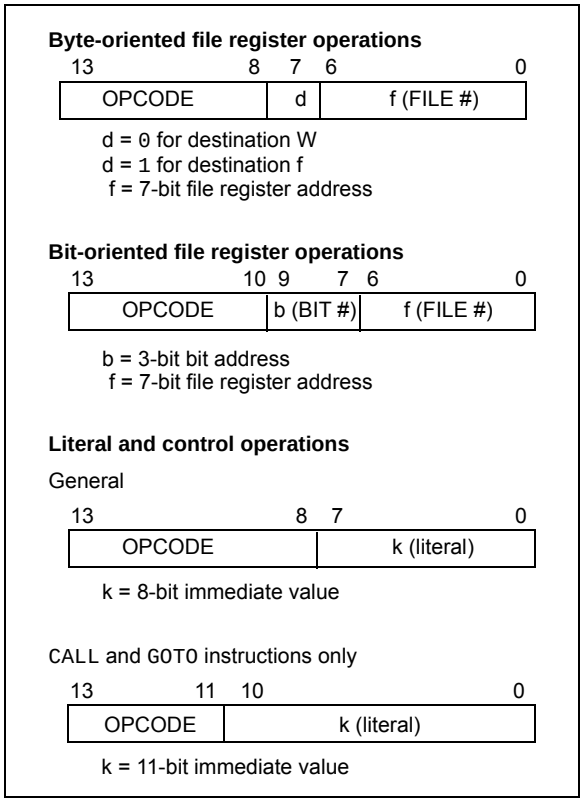


TABLE 30-2: MCP19214/5 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C, DC, Z	1, 2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1, 2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	–	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1, 2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1**(2)**	00 1011 dfff ffff		1, 2, 3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1, 2
INCFSZ	f, d	Increment f, Skip if 0	1**(2)**	00 1111 dfff ffff		1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1, 2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1, 2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	–	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1, 2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1, 2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C, DC, Z	1, 2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1, 2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1, 2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1, 2
BTFSC	f, b	Bit Test f, Skip if Clear	1 **(2)**	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 **(2)**	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call Subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	–	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	–	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	–	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	–	Go into Standby mode	1	00 0000 0110 0011	$\overline{TO}$, $\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTA, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as an NOP.

30.2 Instruction Descriptions

ADDLW	Add literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ANDLW	AND literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are ANDed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded, and an NOP is executed instead, making this a two-cycle instruction.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', the next instruction is discarded and an NOP is executed instead, making this a two-cycle instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero (Z) bit is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Bits $\overline{TO}$ and $\overline{PD}$ in the STATUS register are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', an NOP is executed instead, making it a two-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', an NOP is executed instead, making it a two-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are ORed with the 8-bit literal 'k'. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

MOVF **Move f**

Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If d = 0, the destination is W register. If d = 1, the destination is file register 'f' itself. d = 1 is useful to test a file register since STATUS flag Z is affected.
Words:	1
Cycles:	1
Example:	MOVF FSR, 0 After Instruction W = value in FSR register Z = 1

MOVLW **Move literal to W**

Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Description:	The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	MOVLW 0x5A After Instruction W = 0x5A

MOVWF **Move W to f**

Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	MOVW OPTION F Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F W = 0x4F

NOP **No Operation**

Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	NOP

RETFIE **Return from Interrupt**

Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.
Words:	1
Cycles:	2
Example:	`RETFIE` After Interrupt PC = TOS GIE = 1

RETLW **Return with literal in W**

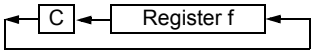
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the 8-bit literal ‘k’. The program counter is loaded from the Top-of-Stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2

Example:	`CALL TABLE;W contains`
	`;table offset`
	`;value`
	`GOTO DONE`
	`TABLE`
	•
	•
	`ADDWF PC ;W = offset`
	`RETLW k1 ;Begin table`
	`RETLW k2 ;`
	•
	•
	•
	`RETLW kn ;End of table`
	`DONE`
	Before Instruction
	W = 0x07
	After Instruction
	W = value of k8

RETURN **Return from Subroutine**

Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the Top-of-Stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

RLF **Rotate Left f through Carry**

Syntax:	[*label* RLF f,d]
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register ‘f’ are rotated one bit to the left through the Carry flag. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’. 
Words:	1
Cycles:	1
Example:	`RLF REG1,0` Before Instruction REG1 = 1110 0110 C = 0 After Instruction REG1 = 1110 0110 W = 1100 1100 C = 1

RRF **Rotate Right f through Carry**

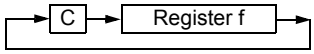
Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP **Enter Sleep mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down STATUS bit, $\overline{PD}$, is cleared. Time-Out STATUS bit, $\overline{TO}$, is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (two's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

Result	Condition
C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (two's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

XORLW **Exclusive OR literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Description: The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

NOTES:

31.0 DUAL CAPTURE/COMPARE (CCD) MODULE

The CCD module is implemented on the MCP19214/5. This is a new module based on the standard CCP module. It has two capture and compare only register sets with no PWM function.

31.1 Capture Mode

In Capture mode, the CCxRH:CCxRL register set captures the 16-bit value of the TMR1 register when an event occurs on the DIMI pin. An event is defined as one of the following:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

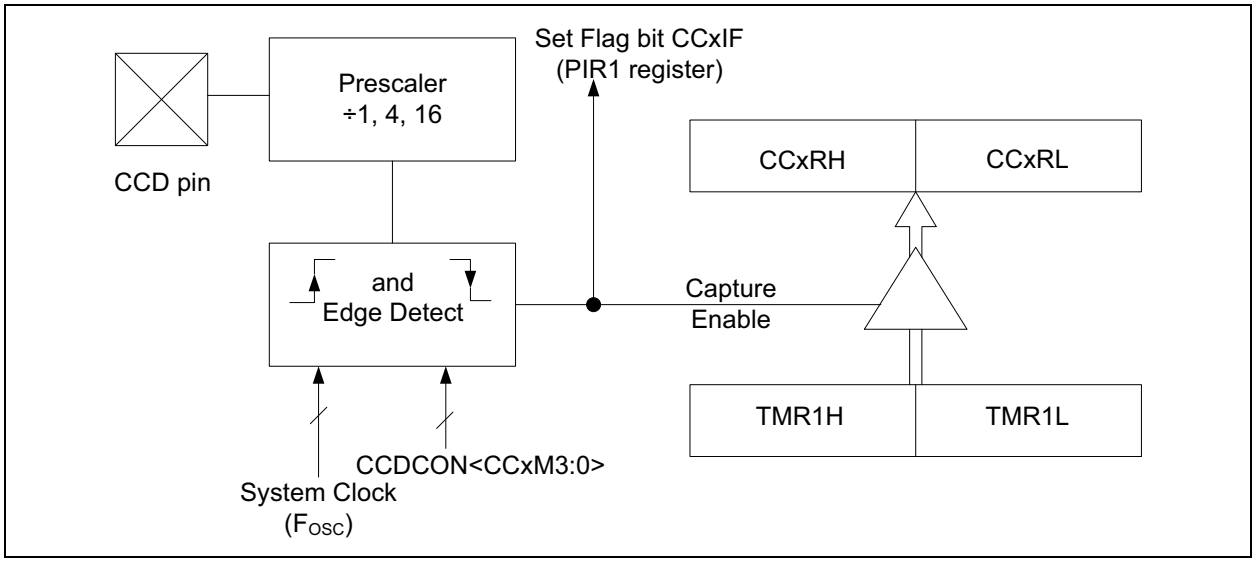
The type of event is configured by control bits CCxM3:CCxM0 (CCDCON<3:0> for register set 1 or CCDCON<7:4> for register set 2). When a capture is made, the interrupt request flag bit, CCxIF (PIR1<2> for register set 1 or PIR1<3> for register set 2), is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the register set is read, the old captured value is overwritten by the new value.

31.1.1 CCX PIN CONFIGURATION

In Capture mode, the CCx pin should be configured as an input by setting the TRIS bit for that pin.

Note: If the CCx pin is configured as an output, a write to the port can cause a capture condition.

FIGURE 31-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



31.1.2 TIMER1 MODE SELECTION

Timer1 must be running off of the instruction clock for the CCD module to use the capture feature. If Timer1 is running off of the 8 MHz clock, the capture feature may not function correctly.

31.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the enable for the capture interrupt clear in order to avoid false interrupts and should clear the flag bit, CCxIF, following any such change in the operating mode.

31.1.4 CCD PRESCALER

There are four prescaler settings, specified by bits CCxM3:CCxM0. Whenever the CCD register set is disabled or not set to Capture mode, the prescaler counter is cleared. Any reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore, the first capture may be from a nonzero prescaler. It is recommended to disable the register set (CCxM3:0 = 00xx) prior to changing the prescaler value.

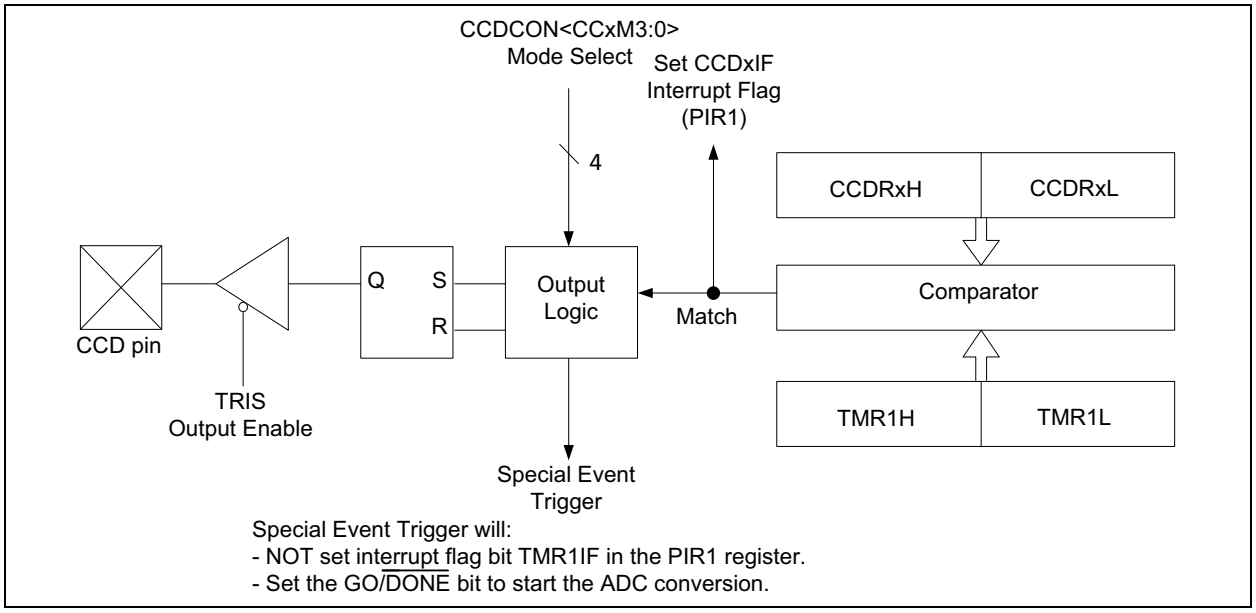
31.2 Compare Mode

In Compare mode, the 16-bit CCDRx register value is constantly compared against the TMR1 register pair value. When a match occurs, the CMPx pin:

- is driven high
- is driven low
- toggles
- remains unchanged

The action on the pin is based on the value of the control bits, CCxM3:CCxM0. At the same time, interrupt flag bit CCP1IF is set.

FIGURE 31-2: COMPARE MODE OPERATION BLOCK DIAGRAM



31.2.1 CMPX PIN CONFIGURATION

The user must configure the CMPx pin as an output by clearing the TRIS bit for that pin.

Note:	Clearing the CCxM<3:0> bits will set the CMPx compare output latch to the default state. This is not the GPIO pin data latch. The default state for set-on-match or toggle on match is 0, but the default state for clear on match is 1.

31.2.2 TIMER1 MODE SELECTION

Timer1 must be running off of the instruction clock for the CCD module to use the compare feature. If Timer1 is running off of the 8 MHz clock, the compare feature may not function correctly.

31.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the CCP1 pin is not affected. The CCP1IF bit is set, causing a CCx interrupt (if enabled).

31.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated, which may be used to initiate an action. The Special Event Trigger output of CCD does not reset the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled).

Note:	The Special Event Trigger from the CCD module will not set the interrupt flag bit TMR1IF (bit 0 in the PIR1 register).

31.3 Dual Capture/Compare Register

The Dual Capture/Compare Module is a new module based on the standard CCP. It has no PWM function.

REGISTER 31-1: CCDCON: DUAL CAPTURE/COMPARE CONTROL MODULE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CC2M3	CC2M2	CC2M1	CC2M0	CC1M3	CC1M2	CC1M1	CC1M0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n = Value at POR
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **CC2M<3:0>:** CC Register Set 2 Mode Select bits

- 00xx = Capture/Compare off (resets the module)
- 0100 = Capture mode, every falling edge
- 0101 = Capture mode, every rising edge
- 0110 = Capture mode, every 4th rising edge
- 0111 = Capture mode, every 16th rising edge
- 1000 = Compare mode, set output on match (CC2IF bit is set)
- 1001 = Compare mode, clear output on match (CC2IF bit is set)
- 1010 = Compare mode, toggle output on match (CC2IF bit is set)
- 1011 = Reserved
- 11xx = Compare mode, generate software interrupt on match (CC2IF bit is set, CMP2 pin is unaffected and configured as an I/O)
- 1111 = Compare mode, trigger special event (CC2IF bit is set; **CC2 does not reset TMR1(1) and starts an A/D conversion, if the A/D module is enabled**. CMP2 pin is unaffected and configured as an I/O port).

bit 3-0 **CC1M<3:0>:** CC Register Set 1 Mode Select bits

- 00xx = Capture/Compare off (resets the module)
- 0100 = Capture mode, every falling edge
- 0101 = Capture mode, every rising edge
- 0110 = Capture mode, every 4th rising edge
- 0111 = Capture mode, every 16th rising edge
- 1000 = Compare mode, set output on match (CC1IF bit is set)
- 1001 = Compare mode, clear output on match (CC1IF bit is set)
- 1010 = Compare mode, toggle output on match (CC1IF bit is set)
- 1011 = Reserved
- 11xx = Compare mode, generate software interrupt on match (CC1IF bit is set, CMP1 pin is unaffected and configured as an I/O)
- 1111 = Compare mode, trigger special event (CC1IF bit is set; **CC1 resets TMR1 and starts an A/D conversion, if the A/D module is enabled**. CMP1 pin is unaffected and configured as an I/O port).

Note 1: When the Compare interrupt is set, a PIC will typically reset TMR1. This module does NOT reset TMR1.

NOTES:

32.0 IN-CIRCUIT SERIAL PROGRAMMING™ (ICSP™)

ICSP allows customers to manufacture circuit boards with unprogrammed devices. Programming can be done after the assembly process, allowing the device to be programmed with the most recent firmware or a custom firmware. Five pins are needed for ICSP programming:

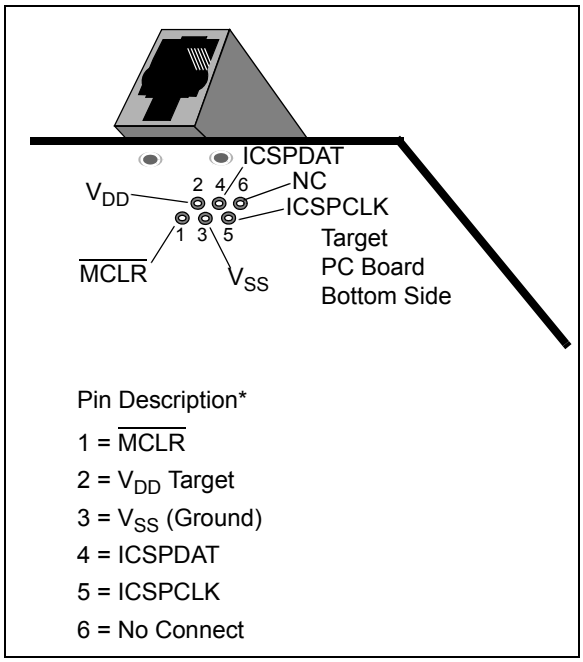
- ICSPCLK
- ICSPDAT
- $\overline{MCLR}$
- V_{DD}
- V_{SS} (A_{GND})

In Program/Verify mode, the Program Memory, User IDs and the Configuration Words are programmed through serial communications. The ICSPDAT pin is a bidirectional I/O used for transferring the serial data and the ICSPCLK pin is the clock input. The device is placed into a Program/Verify mode by holding the $\overline{MCLR}$ pin low, while raising the $\overline{MCLR}$ pin from V_{IL} to V_{IHH}.

32.1 Common Programming Interfaces

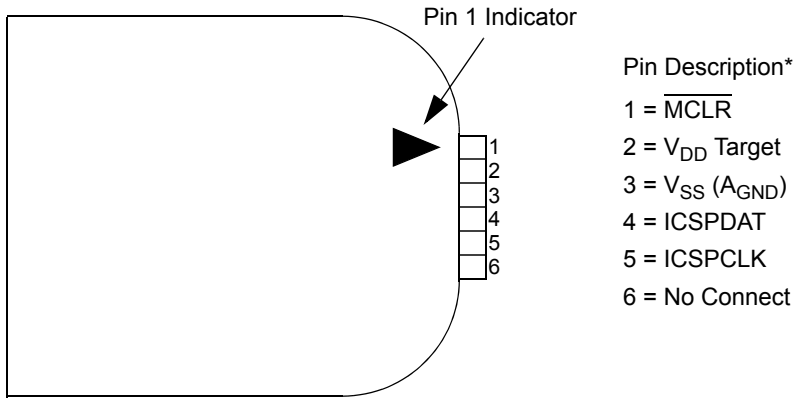
Connection to a target device is typically done through an ICSP header. A commonly found connector on development tools is the RJ-11 in the 6P6C (6-pin, 6-connector) configuration. Refer to Figure 32-1.

FIGURE 32-1: ICD RJ-11 STYLE CONNECTOR INTERFACE



Another connector often found in use with the PICkit™ programmers is a standard 6-pin header with 0.1 inch spacing. Refer to Figure 32-2.

FIGURE 32-2: PICkit™ STYLE CONNECTOR INTERFACE

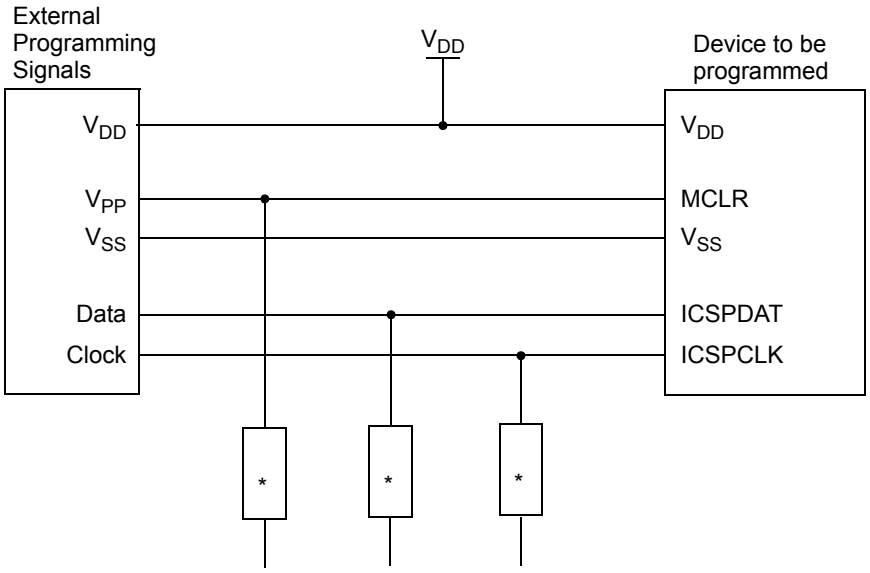


* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes or even jumpers. Refer to Figure 32-3 for more information.

FIGURE 32-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



* Isolation devices (as required).

Note: If powering via back feeding V_{DD}, V_{DD} must be tied to V_{IN}.

33.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

33.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

33.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

33.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

33.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

33.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

33.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

33.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8-, 16- and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to 3 meters) interconnection cables.

33.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

33.9 PICkit 3 In-Circuit Debugger/Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

33.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

33.11 Demonstration/Development Boards, Evaluation Kits and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web site (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

33.12 Third-Party Development Tools

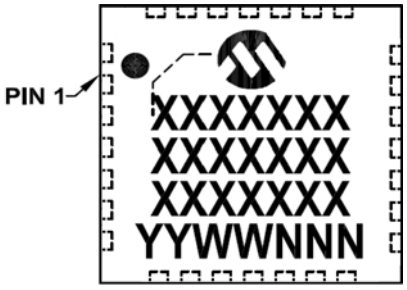
Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

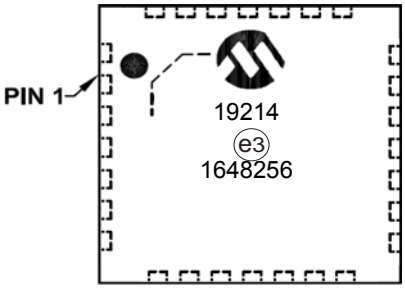
34.0 PACKAGING INFORMATION

34.1 Package Marking Information

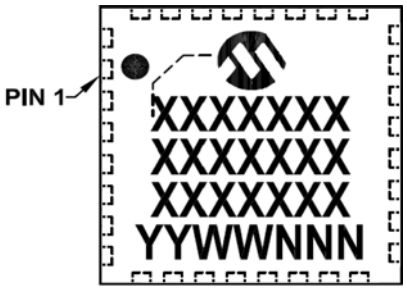
28-Lead QFN (5x5 mm) **(MCP19214 only)**



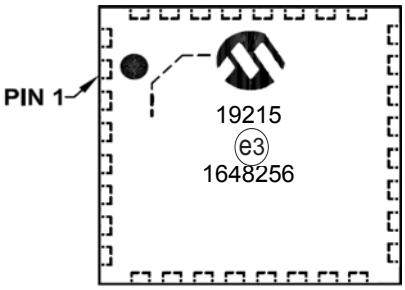
Example



32-Lead QFN (5x5 mm) **(MCP19215 only)**



Example

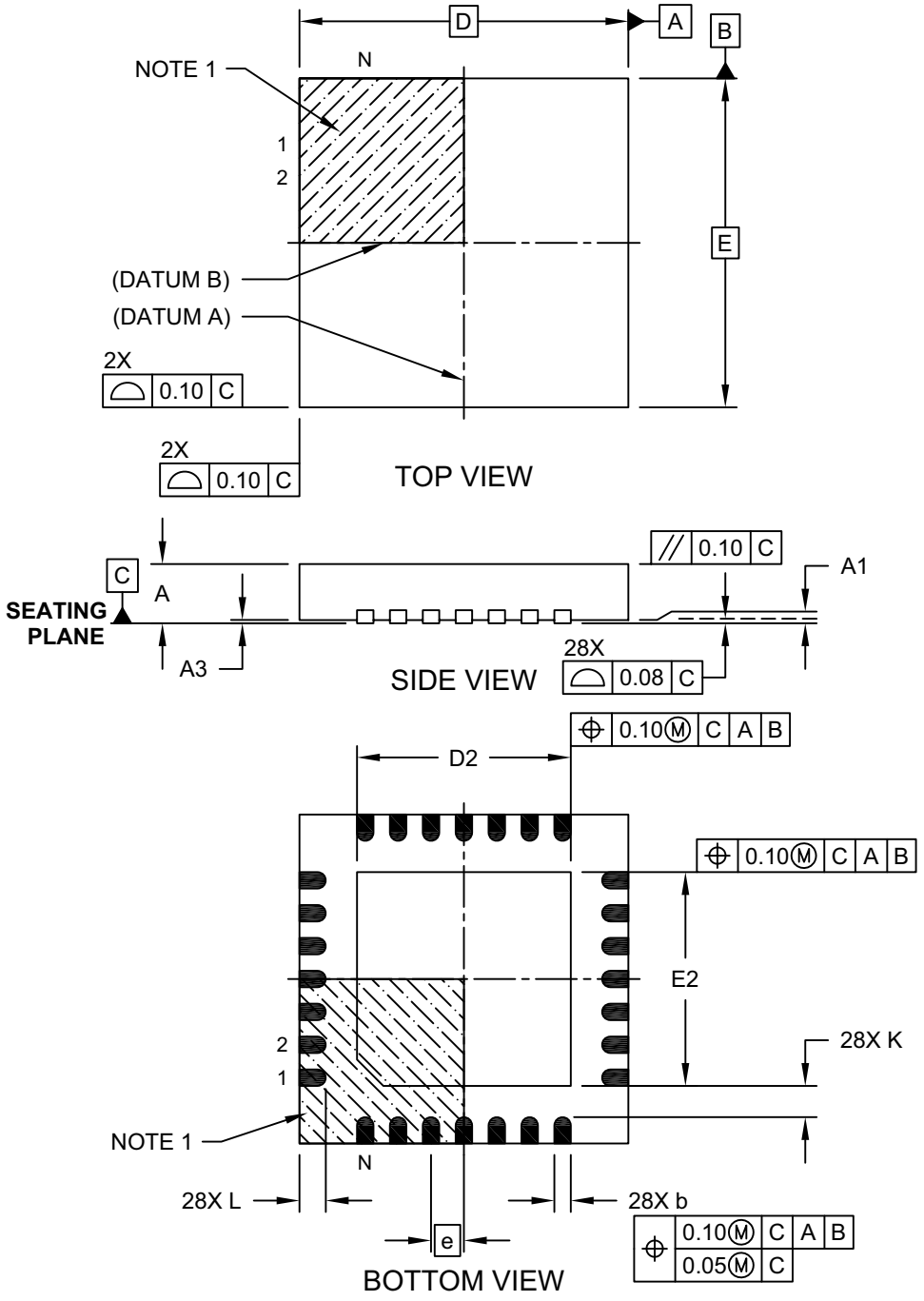


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN or VQFN]

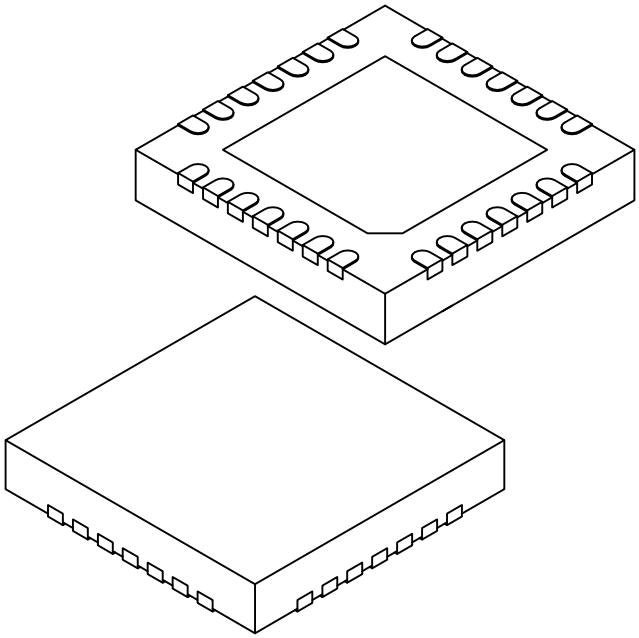
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-140C Sheet 1 of 2

28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN or VQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		5.00 BSC	
Exposed Pad Width	E2	3.15	3.25	3.35
Overall Length	D		5.00 BSC	
Exposed Pad Length	D2	3.15	3.25	3.35
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.35	0.40	0.45
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

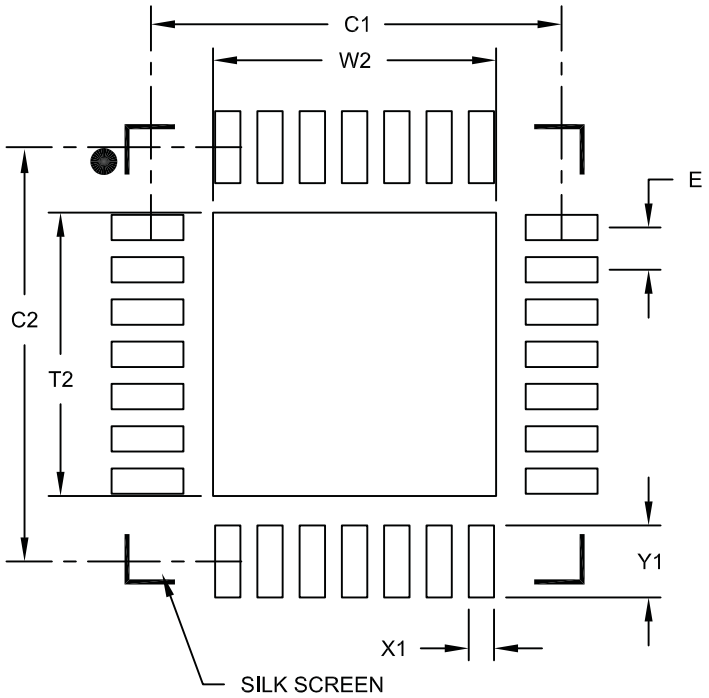
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-140C Sheet 2 of 2

28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5 mm Body [QFN] Land Pattern With 0.55 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			3.35
Optional Center Pad Length	T2			3.35
Contact Pad Spacing	C1		4.90	
Contact Pad Spacing	C2		4.90	
Contact Pad Width (X28)	X1			0.30
Contact Pad Length (X28)	Y1			0.85

Notes:

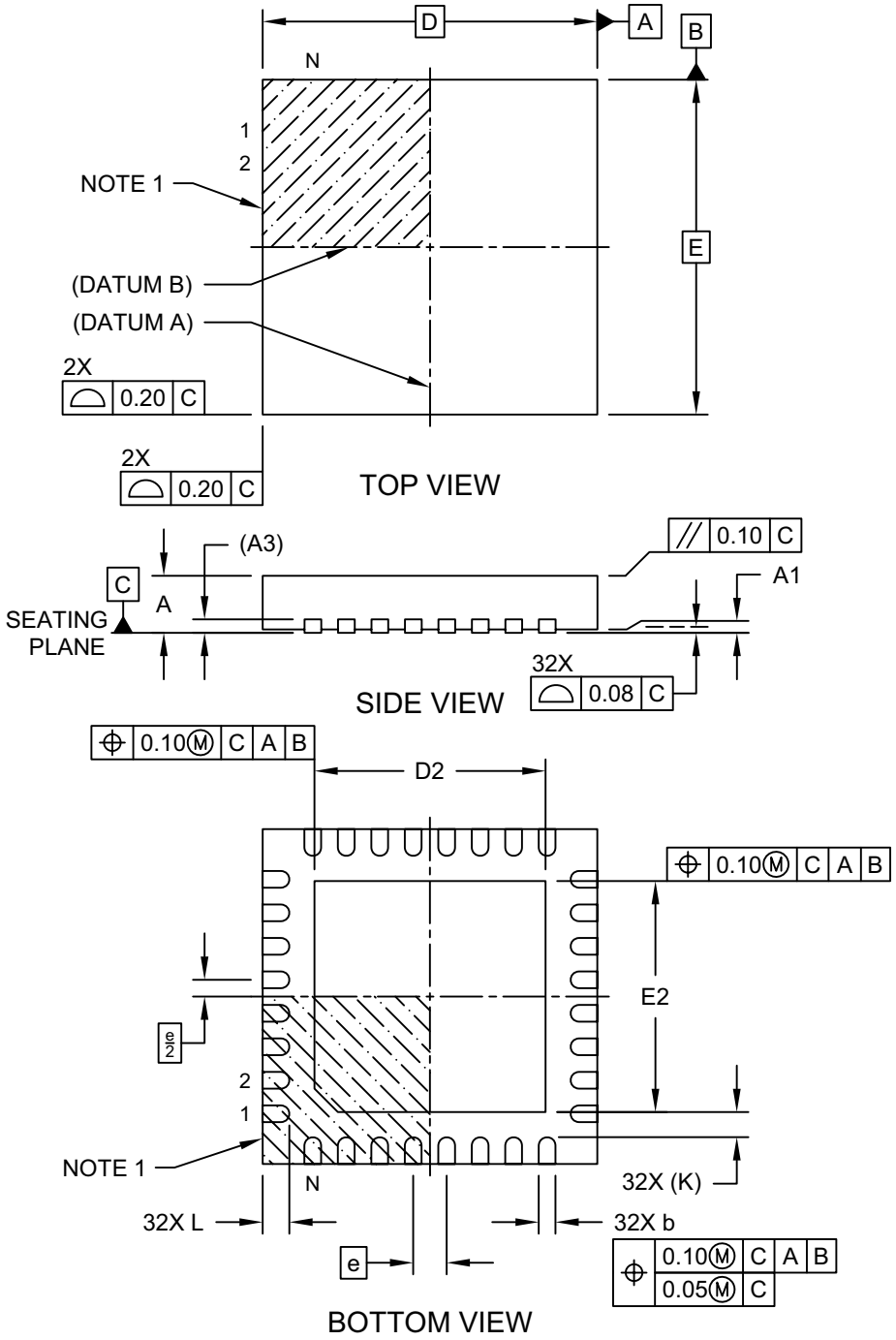
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2140A

32-Lead Very Thin Plastic Quad Flat, No Lead Package (S8) - 5x5x0.9 mm Body [VQFN] With 3.7x3.7 mm Exposed Pad

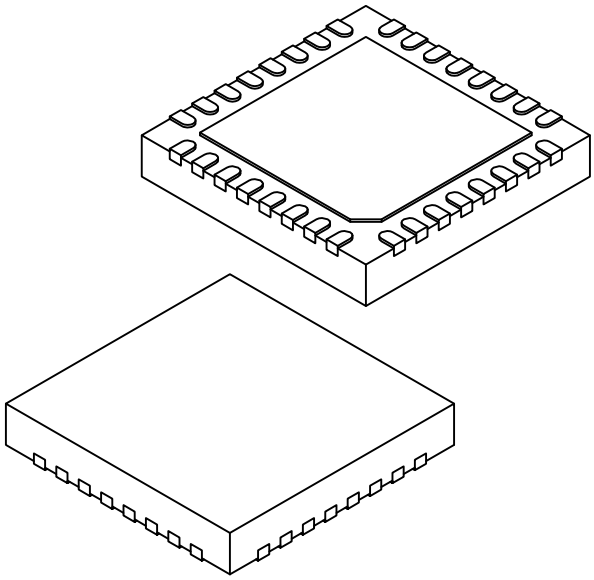
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-160B S8 Sheet 1 of 2

32-Lead Very Thin Plastic Quad Flat, No Lead Package (S8) - 5x5x0.9 mm Body [VQFN] With 3.7x3.7 mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		32	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		5.00 BSC	
Exposed Pad Width	E2	3.60	3.70	3.80
Overall Length	D		5.00 BSC	
Exposed Pad Length	D2	3.60	3.70	3.80
Terminal Width	b	0.18	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		0.25 REF	

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

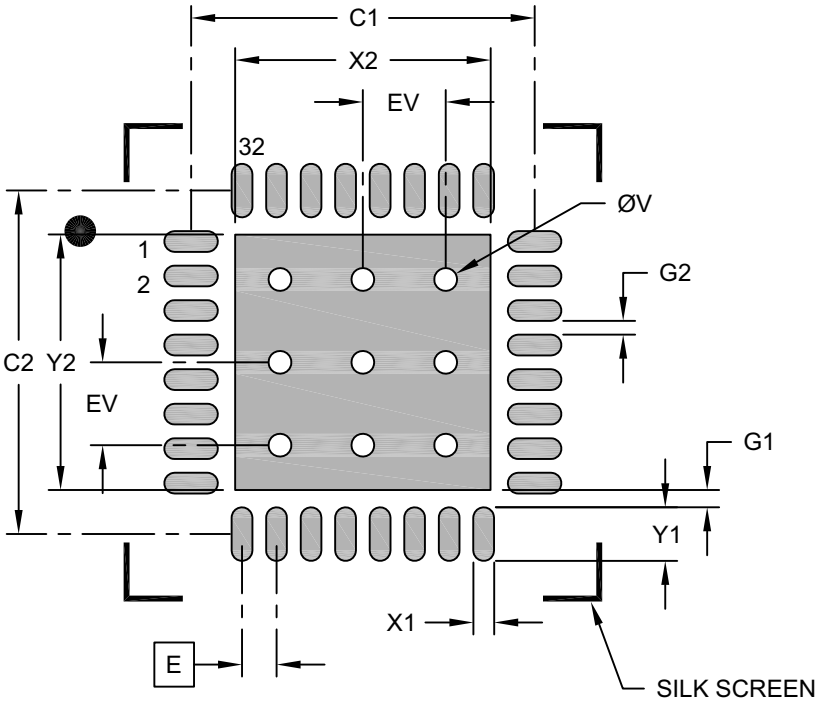
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-160B S8 Sheet 2 of 2

32-Lead Very Thin Plastic Quad Flat, No Lead Package (S8) - 5x5 mm Body [VQFN] With 3.7x3.7 mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	X2			3.70
Optional Center Pad Length	Y2			3.70
Contact Pad Spacing	C1		4.98	
Contact Pad Spacing	C2		4.98	
Contact Pad Width (X32)	X1			0.30
Contact Pad Length (X32)	Y1			0.78
Contact Pad to Center Pad (X32)	G1	0.25		
Contact Pad to Contactr Pad (X28)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2160C-S8

NOTES:

APPENDIX A: REVISION HISTORY

Revision A (March 2017)

- Original Release of this Document.

NOTES:

INDEX

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. The web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: www.microchip.com/support

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](1)	- X	/XX
Device	Tape and Reel Option	Temperature Range	Package

Device:	MCP19214: Digitally Enhanced Power Analog PWM Controllers MCP19215: Digitally Enhanced Power Analog PWM Controllers
Tape and Reel Option:	Blank = Standard packaging (tube) T = Tape and Reel
Temperature Range:	E = -40°C to +125°C (Extended)
Package:	MQ = 28-Lead Plastic Quad Flat, No Lead Package - 5x5 mm Body (QFN) S8 = 32-Lead Plastic Quad Flat, No Lead Package - 5x5 mm Body (QFN)

Examples:

a) MCP19214-E/MQ: Extended Temperature, 28 LD QFN 5x5 package

b) MCP19214T-E/MQ: Tape and Reel, Extended Temperature, 28 LD QFN 5x5 package

a) MCP19215-E/S8: Extended Temperature, 32 LD QFN 5x5 package

b) MCP19215T-E/S8: Tape and Reel, Extended Temperature, 32 LD QFN 5x5 package

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-1323-3

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820