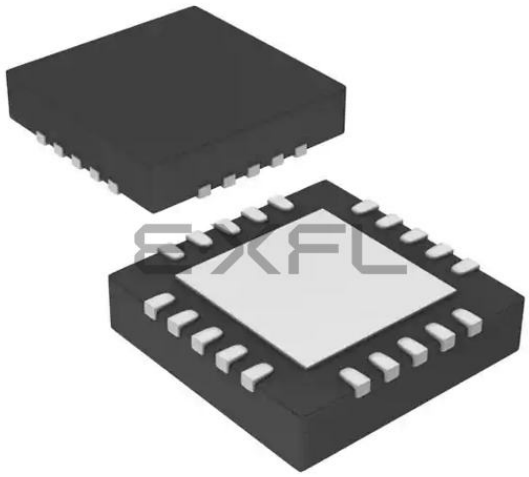


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1508-e-ml

3.3.6 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-8 can be addressed from any Bank.

TABLE 3-8: CORE FUNCTION REGISTERS SUMMARY

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 0-31											
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
x02h or x82h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
x03h or x83h	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
x04h or x84h	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
x05h or x85h	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
x06h or x86h	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
x07h or x87h	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
x08h or x88h	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000
x09h or x89h	WREG	Working Register								0000 0000	uuuu uuuu
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 0											
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx
00Dh	PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	xxxx ----
00Eh	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	xxxx xxxx
010h	—	Unimplemented								—	—
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	0000 0-00	0000 0-00
012h	PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	000- -00-	000- -00-
013h	PIR3	—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF	---- 0000	---- 0000
014h	—	Unimplemented								—	—
015h	TMR0	Holding Register for the 8-bit Timer0 Count								xxxx xxxx	uuuu uuuu
016h	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count								xxxx xxxx	uuuu uuuu
017h	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								xxxx xxxx	uuuu uuuu
018h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	0000 00-0	uuuu uu-u
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu
01Ah	TMR2	Timer2 Module Register								0000 0000	0000 0000
01Bh	PR2	Timer2 Period Register								1111 1111	1111 1111
01Ch	T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
01Dh to 01Fh	—	Unimplemented								—	—
Bank 1											
08Ch	TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
08Dh	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
08Eh	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
08Fh	—	Unimplemented								—	—
090h	—	Unimplemented								—	—
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	0000 0-00	0000 0-00
092h	PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	000- 00--	000- 00--
093h	PIE3	—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE	---- 0000	---- 0000
094h	—	Unimplemented								—	—
095h	OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	00-1 11qq	qq-q qquu
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
098h	—	Unimplemented								—	—
099h	OSCCON	—	IRCF<3:0>				—	SCS<1:0>		-011 1-00	-011 1-00
09Ah	OSCSTAT	SOSCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS	0-q0 --00	q-qq --qq
09Bh	ADRESL	ADC Result Register Low								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	ADC Result Register High								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		0000 --00	0000 --00
09Fh	ADCON2	TRIGSEL<3:0>				—	—	—	—	0000 ----	0000 ----

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: PIC16F1508/9 only.

2: Unimplemented, read as '1'.

SCS<1:0> = 00:

Change the SCS bits in the OSCCON register to select the FOSC Configuration Word clock selection. Since the OST is not applicable with external clocks, the clock module will immediately switch to the external clock, and the fail-safe condition will be cleared.

If the Fail-Safe condition still exists, the OSFIF bit will again be set by hardware.

5.5.3.3 Secondary Oscillator with SCS<1:0> = 01

When a Fail-Safe condition occurs with the clock switch selected to run from the Secondary Oscillator selection (SCS<1:0> = 01), regardless of the FOSC selection, the condition is cleared by performing the following procedure.

SCS<1:0> = 01 (Secondary Oscillator)

SCS<1:0> = 1x:

Change the SCS bits in the OSCCON register to select the internal oscillator block.

OSFIF = 0:

Clear the OSFIF bit in the PIR2 register.

Read SOSCR:

The OST is not used with the secondary oscillator, therefore, the user must determine if the secondary oscillator is ready by monitoring the SOSCR bit in the OSCSTAT register. When the SOSCR bit is set, the secondary oscillator is ready.

SCS<1:0> = 01:

Change the SCS bits in the OSCCON register to select the secondary oscillator. The clock module will immediately switch to the secondary oscillator and the fail-safe condition will be cleared.

If the Fail-Safe condition still exists, the OSFIF bit will again be set by hardware.

5.5.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect external oscillator or external clock failures.

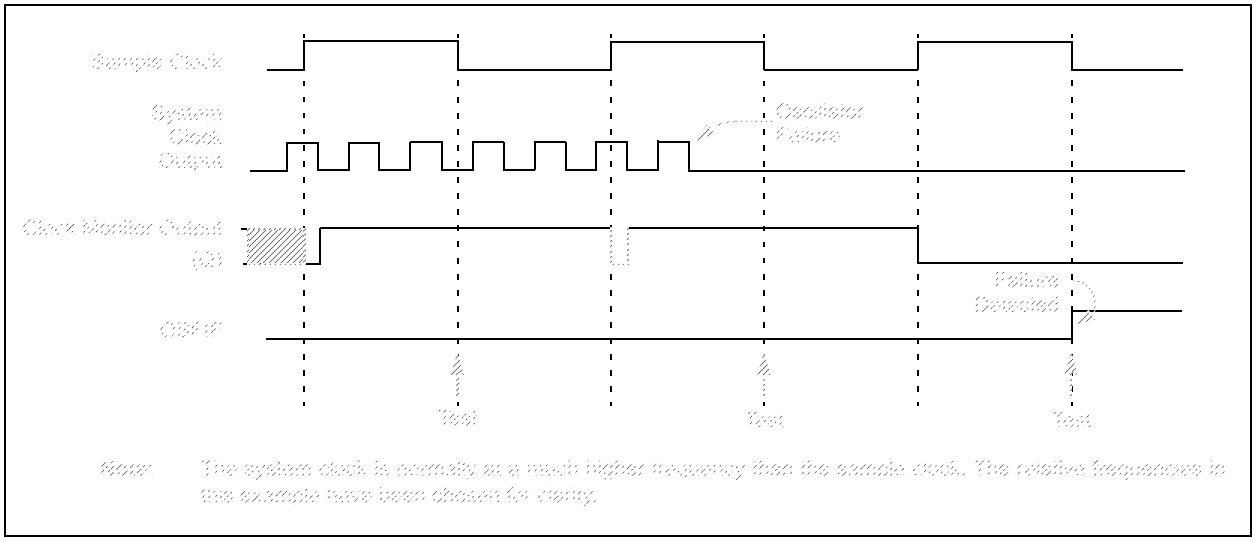
When FSCM is used with an external oscillator, the Oscillator Start-up Timer (OST) count must expire before the FSCM becomes active. The OST is used after waking up from Sleep and after any type of Reset.

When the FSCM is used with external clocks, the OST is not used and the FSCM will be active as soon as the Reset or wake-up has completed.

When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

Note:	Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep).

FIGURE 5-10: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

11.7 PORTC Registers

11.7.1 DATA REGISTER

PORTC is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 11-13). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 11-12) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATC).

11.7.2 DIRECTION CONTROL

The TRISC register (Register 11-13) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.7.3 ANALOG CONTROL

The ANSELC register (Register 11-15) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC 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 ANSELC bits has no effect on digital output functions. A pin with TRIS clear and ANSELC 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 ANSELC 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 user software.

11.7.4 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-8.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input and some digital input functions are not included in the output priority list. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the output priority list.

TABLE 11-8: PORTC OUTPUT PRIORITY

Pin Name	Function Priority(1)
RC0	CLC2 RC0
RC1	NCO1(2) PWM4 RC1
RC2	RC2
RC3	PWM2 RC3
RC4	CWG1B CLC4 C2OUT RC4
RC5	CWG1A CLC1(3) PWM1 RC5
RC6	NCO1(3) RC6
RC7	SDO RC7

Note 1: Priority listed from highest to lowest.
2: Default pin (see APFCON register).
3: Alternate pin (see APFCON register).

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR>1:0>		ADFVR<1:0>		125

Legend: Shaded cells are unused by the temperature indicator module.

REGISTER 15-3: ADCON2: ADC CONTROL REGISTER 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
TRIGSEL<3:0>(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	

bit 7-4 **TRIGSEL<3:0>:** Auto-Conversion Trigger Selection bits(1)

0000 = No auto-conversion trigger selected
0001 = Reserved
0010 = Reserved
0011 = Timer0 – T0_overflow(2)
0100 = Timer1 – T1_overflow(2)
0101 = Timer2 – T2_match
0110 = Comparator C1 – C1OUT_sync
0111 = Comparator C2 – C2OUT_sync
1000 = CLC1 – LC1_out
1001 = CLC2 – LC2_out
1010 = CLC3 – LC3_out
1011 = CLC4 – LC4_out
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Reserved

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

Note 1: This is a rising edge sensitive input for all sources.
2: Signal also sets its corresponding interrupt flag.

16.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The positive input source (VSOURCE+) of the DAC can be connected to:

- External VREF+ pin
- VDD supply voltage

The negative input source (VSOURCE-) of the DAC can be connected to:

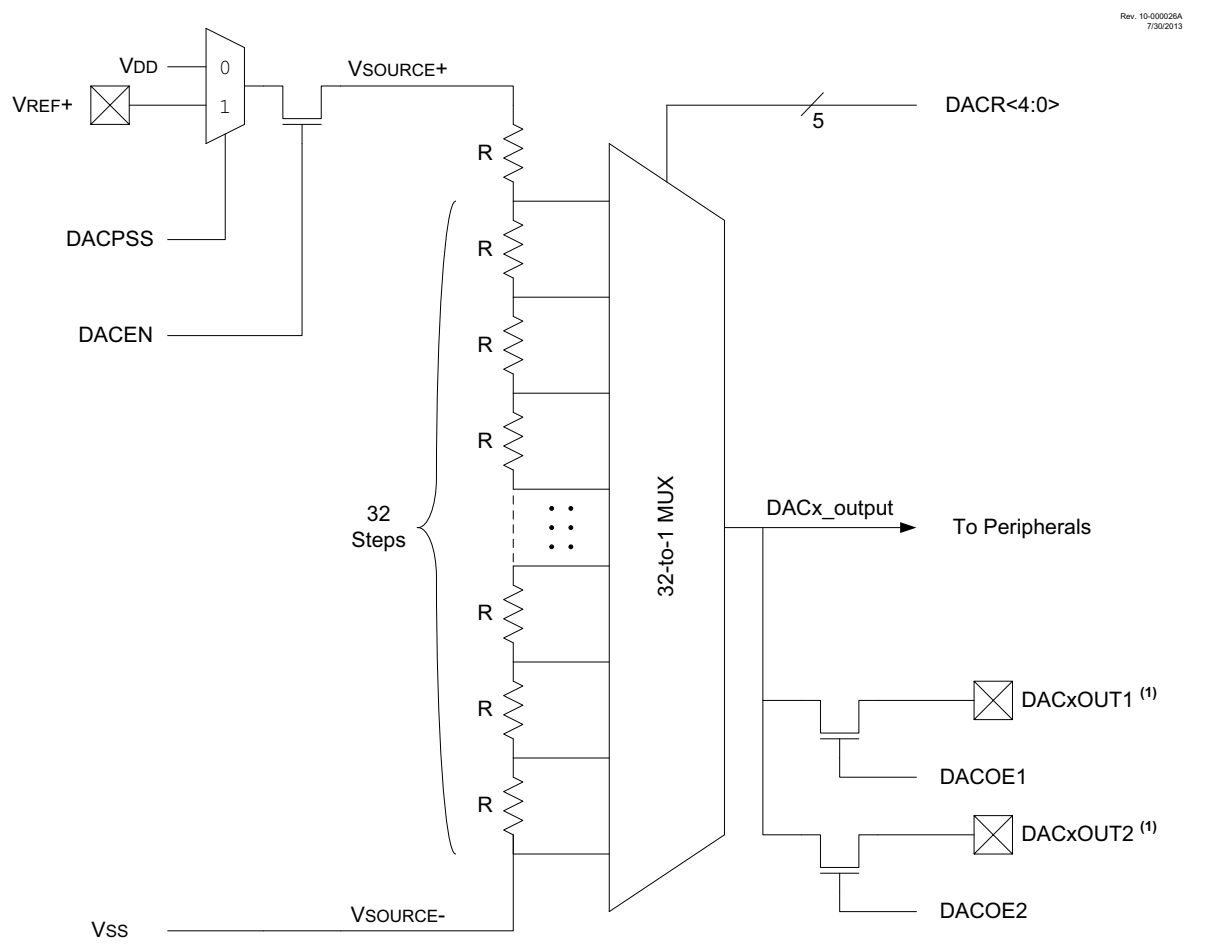
- Vss

The output of the DAC (DACx_output) can be selected as a reference voltage to the following:

- Comparator positive input
- ADC input channel
- DACxOUT1 pin
- DACxOUT2 pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the DACEN bit of the DACxCON0 register.

FIGURE 16-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



Note 1: The unbuffered DACx_output is provided on the DACxOUT pin(s).

FIGURE 19-5: TIMER1 GATE SINGLE-PULSE MODE

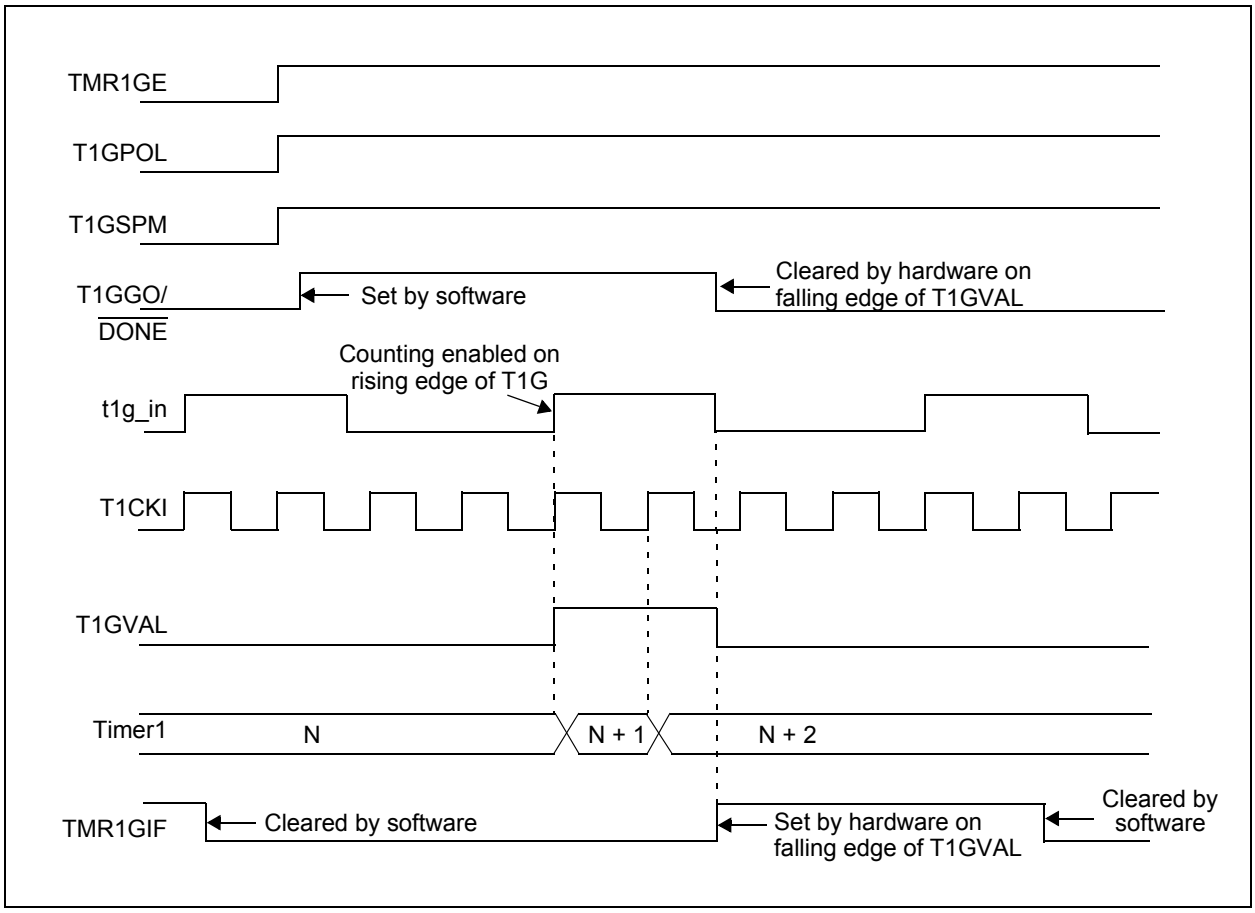
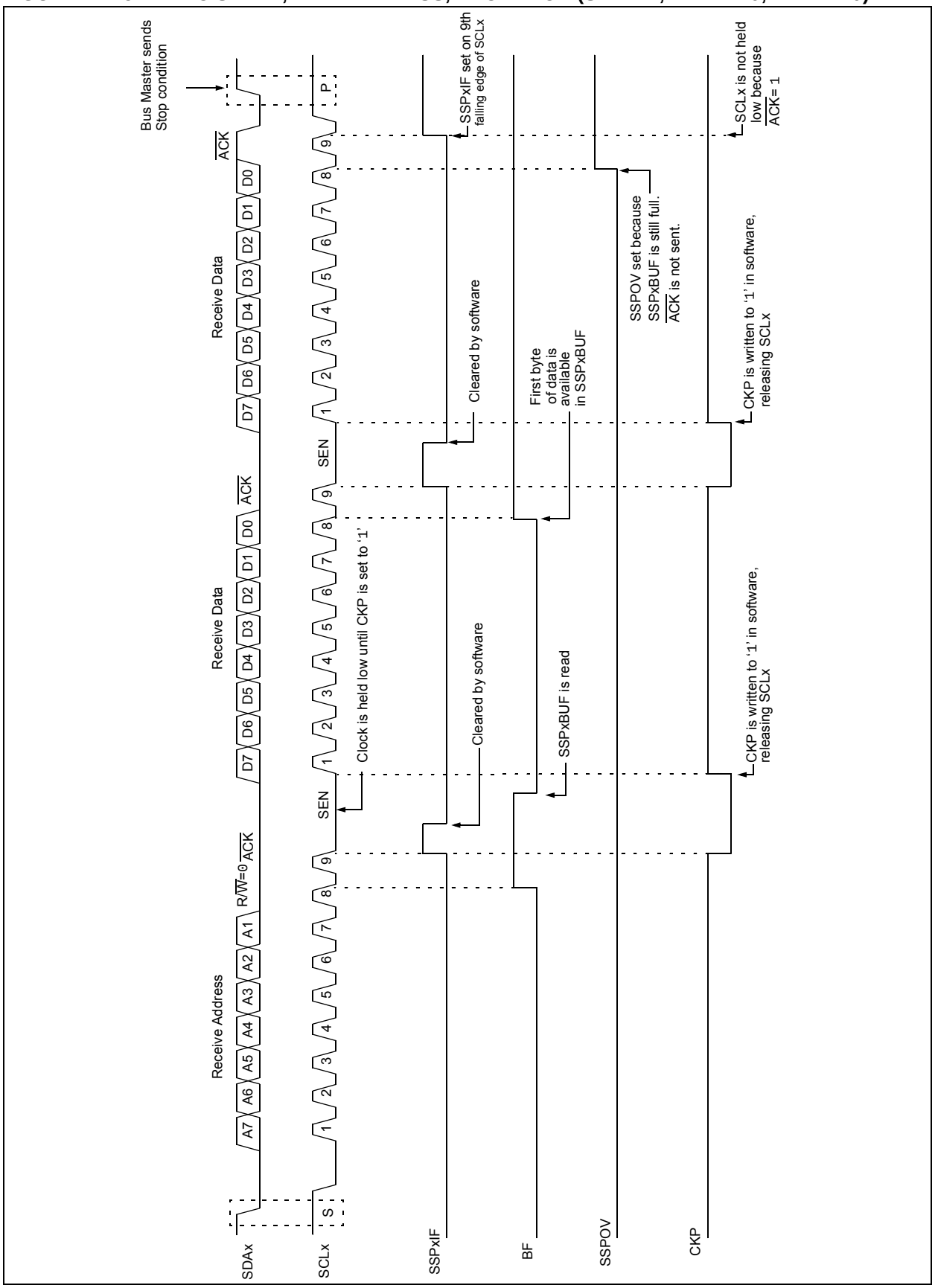


FIGURE 21-15: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)



21.5.8 GENERAL CALL ADDRESS SUPPORT

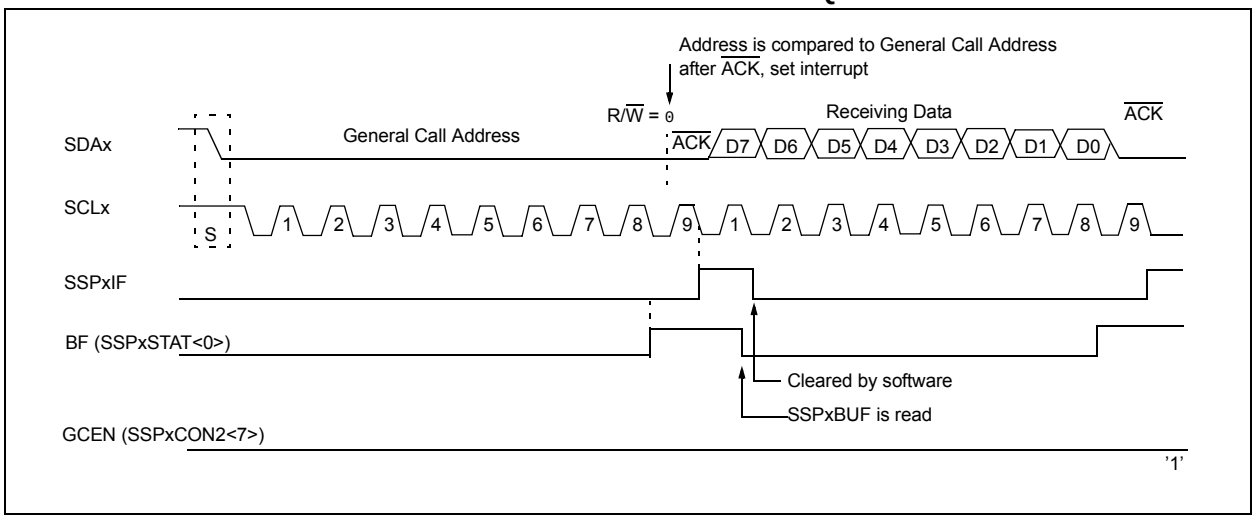
The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in 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 of the SSPxCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address regardless of the value stored in SSPxADD. 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 SSPxBUF and respond. Figure 21-24 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 of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCLx. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 21-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



21.5.9 SSPx MASK REGISTER

An SSPx Mask (SSPxMSK) register (Register 21-5) is available in I^2C Slave mode as a mask for the value held in the SSPxSR register during an address comparison operation. A zero ('0') bit in the SSPxMSK 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 SSPx operation until written with a mask value.

The SSPx Mask 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 SSPx mask has no effect during the reception of the first (high) byte of the address.

21.6.13.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDAx pin has been deasserted and allowed to float high, SDAx is sampled low after the BRG has timed out (Case 1).

b) After the SCLx pin is deasserted, SCLx is sampled low before SDAx goes high (Case 2).

The Stop condition begins with SDAx asserted low. When SDAx is sampled low, the SCLx pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPxADD and counts down to 0. After the BRG times out, SDAx is sampled. If SDAx is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 21-38). If the SCLx pin is sampled low before SDAx is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 21-39).

FIGURE 21-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)

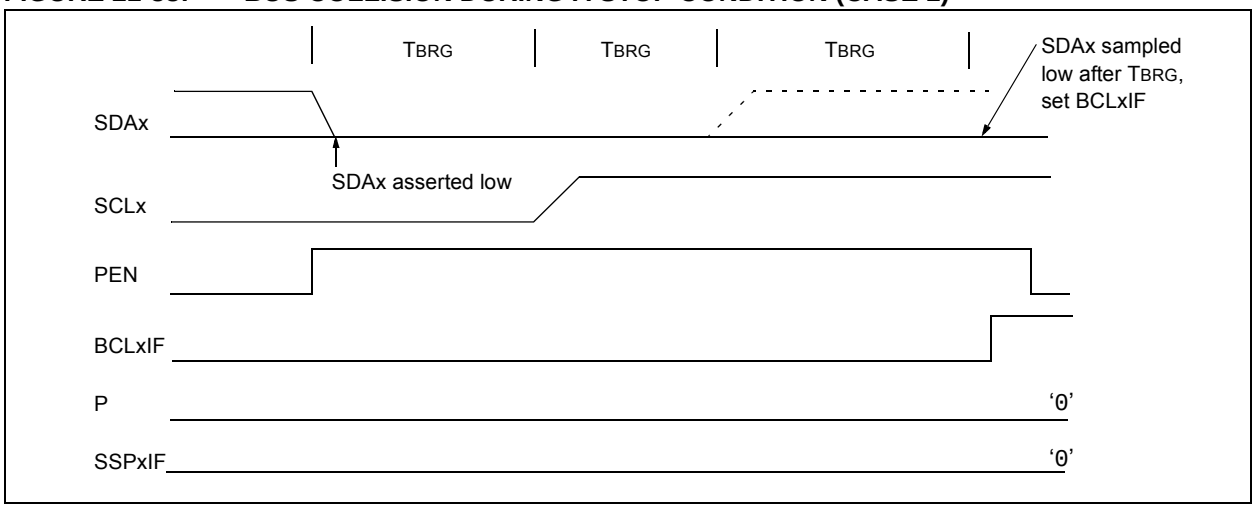
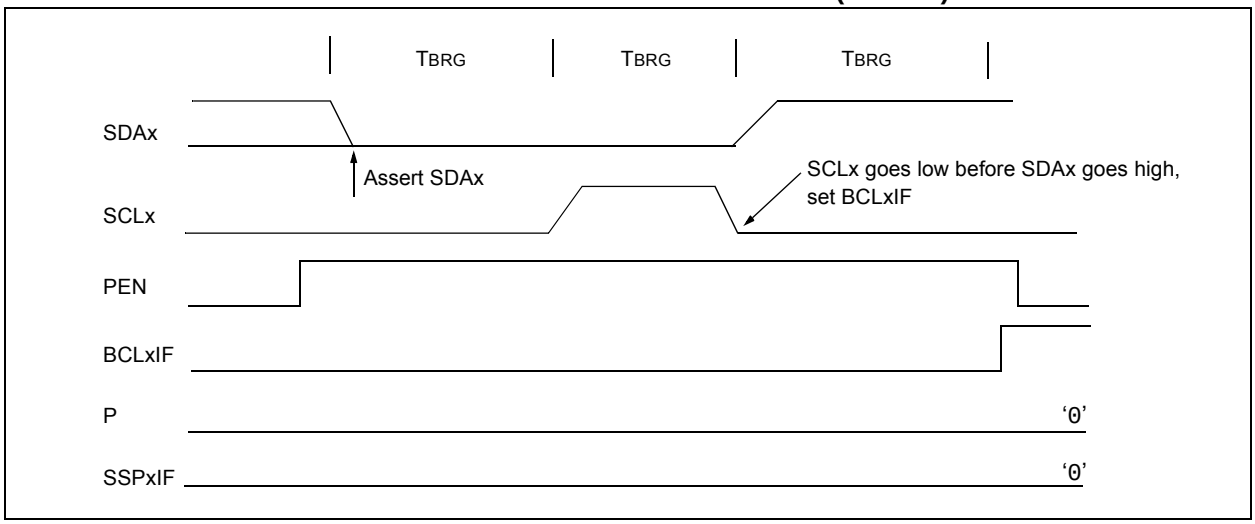


FIGURE 21-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)



REGISTER 21-5: SSPxMSK: SSP MASK REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
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/n = Value at POR and BOR/Value at all other Resets
'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 SSPxADD<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 SSPxADD<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 21-6: SSPxADD: MSSP ADDRESS AND BAUD RATE REGISTER (I^2C MODE)

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	R/W-0/0
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/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

Master mode:

bit 7-0 **ADD<7:0>:** Baud Rate Clock Divider bits
SCLx pin clock period = ((ADD<7:0> + 1) *4)/FOSC

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".

22.1.1.5 TSR Status

The TRMT bit of the TXSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user has to poll this bit to determine the TSR status.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

22.1.1.6 Transmitting 9-Bit Characters

The EUSART supports 9-bit character transmissions. When the TX9 bit of the TXSTA register is set, the EUSART will shift nine bits out for each character transmitted. The TX9D bit of the TXSTA register is the ninth, and Most Significant, data bit. When transmitting 9-bit data, the TX9D data bit must be written before writing the eight Least Significant bits into the TXREG. All nine bits of data will be transferred to the TSR shift register immediately after the TXREG is written.

A special 9-bit Address mode is available for use with multiple receivers. See **Section22.1.2.7 "Address Detection"** for more information on the address mode.

22.1.1.7 Asynchronous Transmission Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section22.4 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If 9-bit transmission is desired, set the TX9 control bit. A set ninth data bit will indicate that the eight Least Significant data bits are an address when the receiver is set for address detection.
4. Set SCKP bit if inverted transmit is desired.
5. Enable the transmission by setting the TXEN control bit. This will cause the TXIF interrupt bit to be set.
6. If interrupts are desired, set the TXIE interrupt enable bit of the PIE1 register. An interrupt will occur immediately provided that the GIE and PEIE bits of the INTCON register are also set.
7. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D data bit.
8. Load 8-bit data into the TXREG register. This will start the transmission.

FIGURE 22-3: ASYNCHRONOUS TRANSMISSION

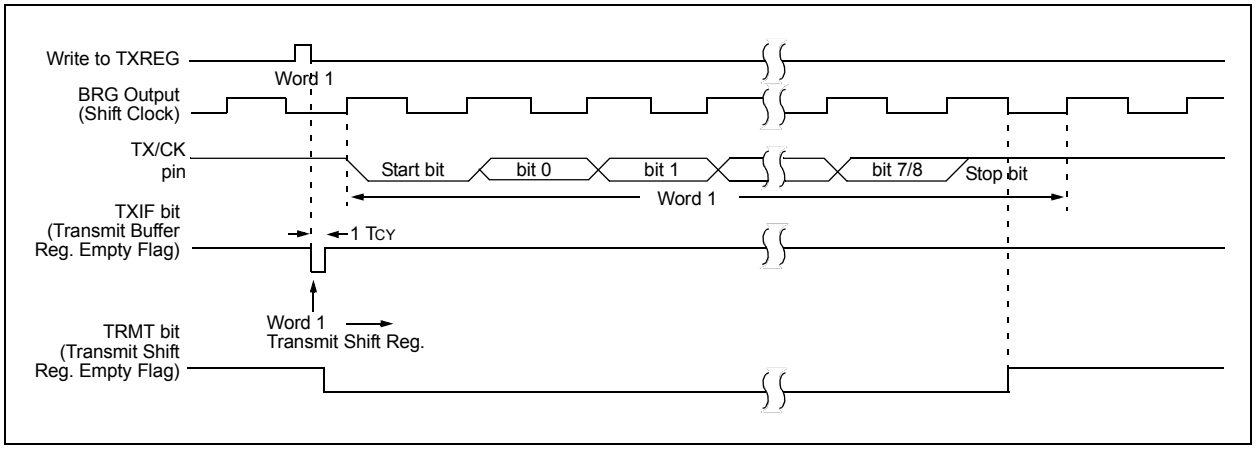
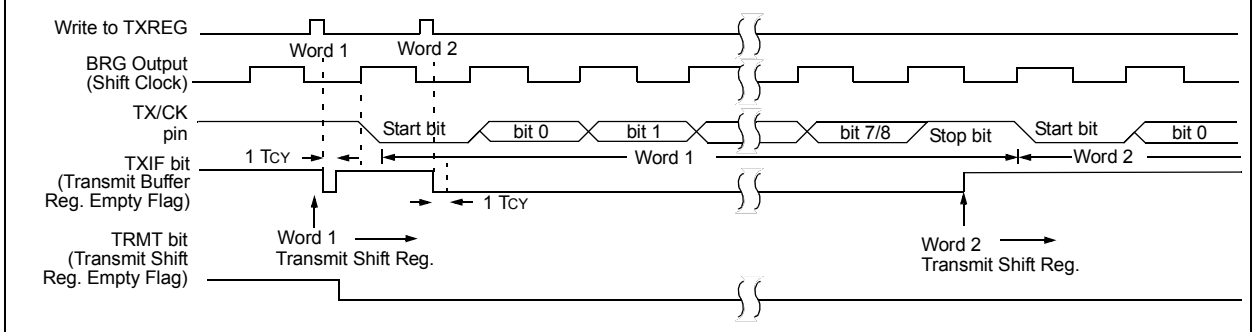
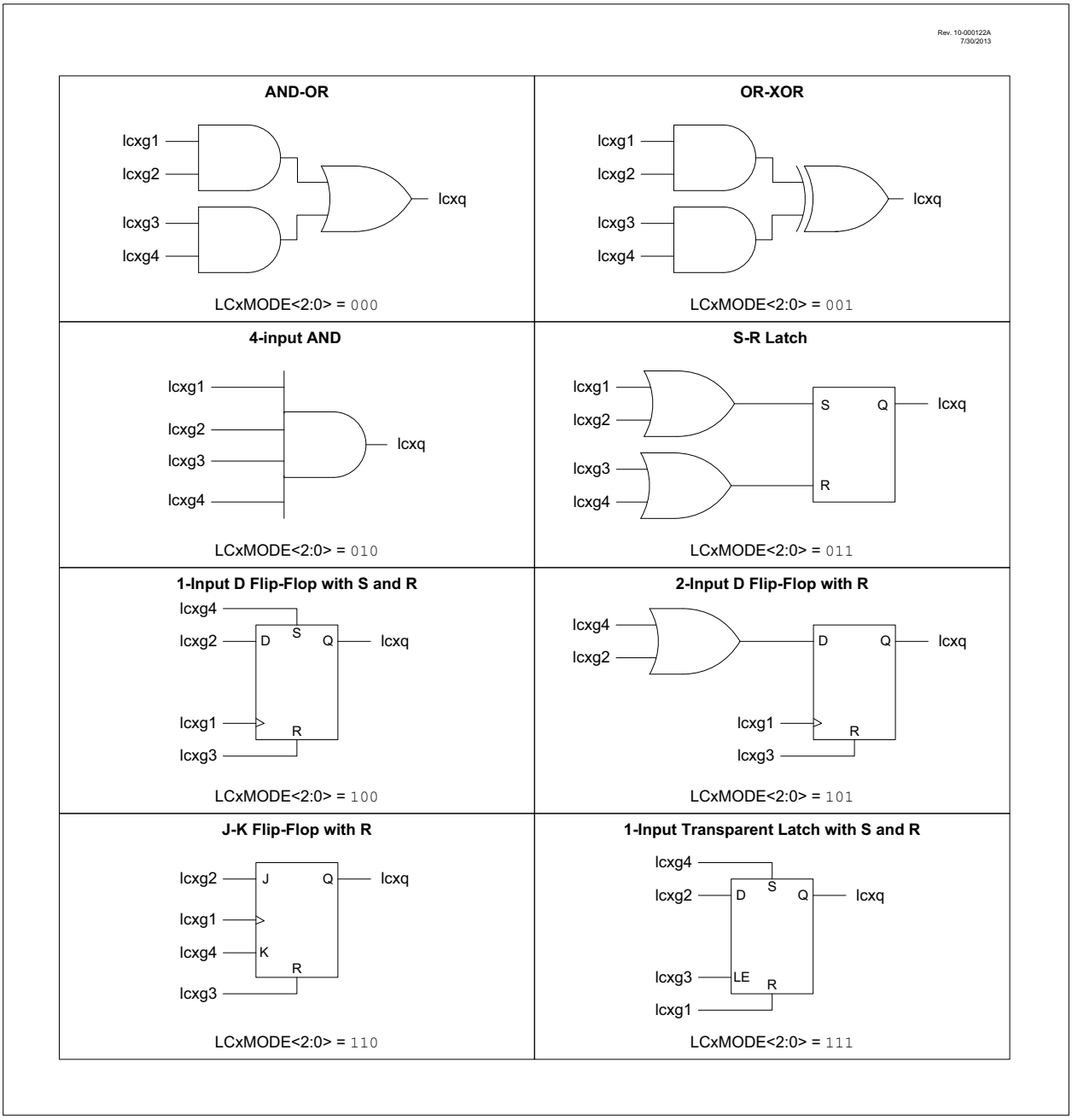


FIGURE 22-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

FIGURE 24-3: PROGRAMMABLE LOGIC FUNCTIONS



REGISTER 24-7: CLCxGLS2: GATE 3 LOGIC SELECT REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
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 **LCxG3D4T:** Gate 3 Data 4 True (non-inverted) bit
1 = lcxd4T is gated into lcxg3
0 = lcxd4T is not gated into lcxg3

bit 6 **LCxG3D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = lcxd4N is gated into lcxg3
0 = lcxd4N is not gated into lcxg3

bit 5 **LCxG3D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = lcxd3T is gated into lcxg3
0 = lcxd3T is not gated into lcxg3

bit 4 **LCxG3D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = lcxd3N is gated into lcxg3
0 = lcxd3N is not gated into lcxg3

bit 3 **LCxG3D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = lcxd2T is gated into lcxg3
0 = lcxd2T is not gated into lcxg3

bit 2 **LCxG3D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = lcxd2N is gated into lcxg3
0 = lcxd2N is not gated into lcxg3

bit 1 **LCxG3D1T:** Gate 3 Data 1 True (non-inverted) bit
1 = lcxd1T is gated into lcxg3
0 = lcxd1T is not gated into lcxg3

bit 0 **LCxG3D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = lcxd1N is gated into lcxg3
0 = lcxd1N is not gated into lcxg3

DECFSZ **Decrement f, Skip if 0**

Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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', then a NOP is executed instead, making it a 2-cycle instruction.

INCFSZ **Increment f, Skip if 0**

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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', a NOP is executed instead, making it a 2-cycle instruction.

GOTO **Unconditional Branch**

Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<6:3> → PC<14:11>
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 2-cycle instruction.

IORLW **Inclusive OR literal with W**

Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

INCF **Increment f**

Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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'.

IORWF **Inclusive OR W with f**

Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
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'.

TABLE 29-2: SUPPLY CURRENT (I_{DD})(1,2) (CONTINUED)

PIC16LF1508/9		Standard Operating Conditions (unless otherwise stated)					
PIC16F1508/9							
Param. No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						V_{DD}	Note
D016		—	215	360	µA	1.8	F_{OSC} = 500 kHz, HFINTOSC
		—	275	480	µA	3.0	
D016		—	270	450	µA	2.3	F_{OSC} = 500 kHz, HFINTOSC
		—	300	500	µA	3.0	
		—	350	620	µA	5.0	
D017*		—	410	660	µA	1.8	F_{OSC} = 8 MHz, HFINTOSC
		—	630	970	µA	3.0	
D017*		—	530	750	µA	2.3	F_{OSC} = 8 MHz, HFINTOSC
		—	660	1100	µA	3.0	
		—	730	1200	µA	5.0	
D018		—	600	940	µA	1.8	F_{OSC} = 16 MHz, HFINTOSC
		—	970	1400	µA	3.0	
D018		—	780	1200	µA	2.3	F_{OSC} = 16 MHz, HFINTOSC
		—	1000	1550	µA	3.0	
		—	1090	1700	µA	5.0	
D019A		—	1030	1500	µA	3.0	F_{OSC} = 20 MHz, External Clock (ECH), High-Power mode
D019A		—	1060	1600	µA	3.0	F_{OSC} = 20 MHz, External Clock (ECH), High-Power mode
		—	1220	1800	µA	5.0	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{SS}; $\overline{MCLR}$ = V_{DD}; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: For RC oscillator configurations, current through R_{EXT} is not included. The current through the resistor can be extended by the formula IR = $V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

TABLE 29-3: POWER-DOWN CURRENTS (I_{PD})(1,2)

PIC16LF1508/9		Operating Conditions: (unless otherwise stated) Low-Power Sleep Mode						
PIC16F1508/9		Low-Power Sleep Mode, VREGPM = 1						
Param. No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							V_{DD}	Note
D022	Base I_{PD}	—	0.020	1.0	8.0	µA	1.8	WDT, BOR, FVR and SOSC disabled, all Peripherals inactive
		—	0.025	2.0	9.0	µA	3.0	
D022	Base I_{PD}	—	0.25	3.0	10	µA	2.3	WDT, BOR, FVR and SOSC disabled, all Peripherals inactive, Low-Power Sleep mode
		—	0.30	4.0	12	µA	3.0	
		—	0.40	6.0	15	µA	5.0	
D022A	Base I_{PD}	—	9.8	16	18	µA	2.3	WDT, BOR, FVR and SOSC disabled, all Peripherals inactive, Normal Power Sleep mode, VREGPM = 0
		—	10.3	18	20	µA	3.0	
		—	11.5	21	26	µA	5.0	
D023		—	0.26	2.0	9.0	µA	1.8	WDT Current
		—	0.44	3.0	10	µA	3.0	
D023		—	0.43	6.0	15	µA	2.3	WDT Current
		—	0.53	7.0	20	µA	3.0	
		—	0.64	8.0	22	µA	5.0	
D023A		—	15	28	30	µA	1.8	FVR Current
		—	18	30	33	µA	3.0	
D023A		—	18	33	35	µA	2.3	FVR Current
		—	19	35	37	µA	3.0	
		—	20	37	39	µA	5.0	
D024		—	6.0	17	20	µA	3.0	BOR Current
D024		—	7.0	17	30	µA	3.0	BOR Current
		—	8.0	20	40	µA	5.0	
D24A		—	0.1	4.0	10	µA	3.0	LPBOR Current
D24A		—	0.35	5.0	14	µA	3.0	LPBOR Current
		—	0.45	8.0	17	µA	5.0	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The peripheral Δ current can be determined by subtracting the base I_{PD} current from this limit. Max. values should be used when calculating total current consumption.

2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{SS}.

3: ADC clock source is FRC.

FIGURE 30-49: V_{OH} vs. I_{OH} OVER TEMPERATURE, V_{DD} = 3.0V

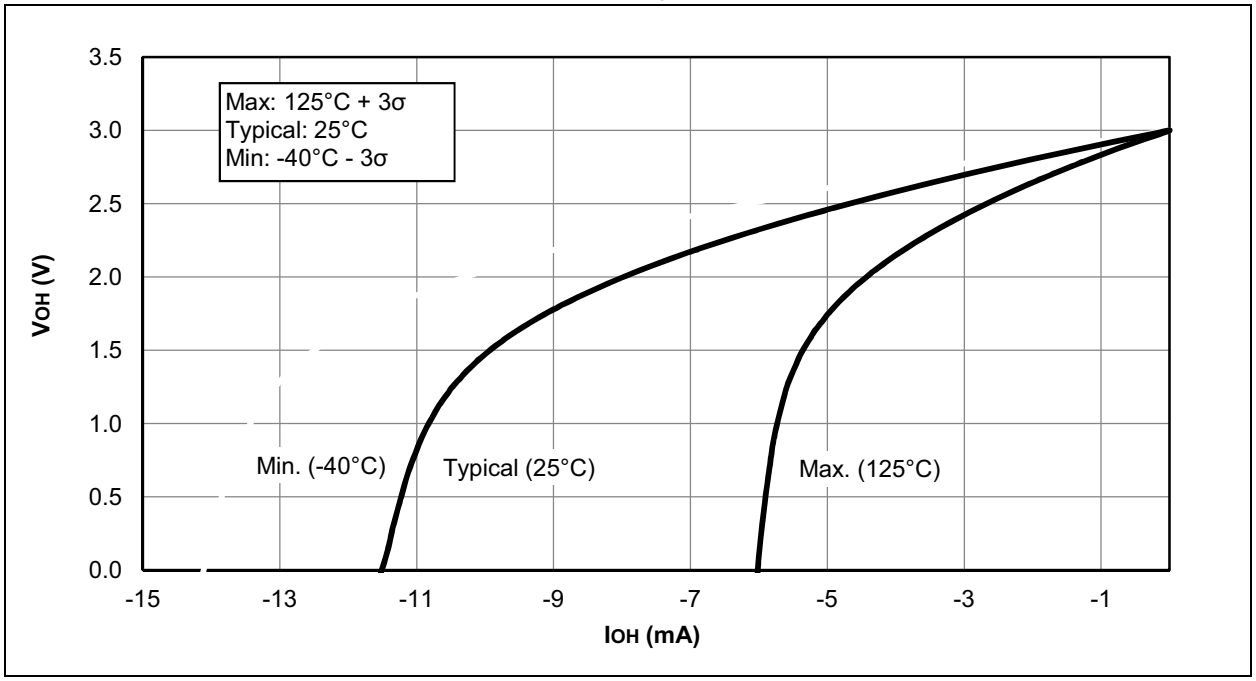


FIGURE 30-50: V_{OL} vs. I_{OL} OVER TEMPERATURE, V_{DD} = 3.0V

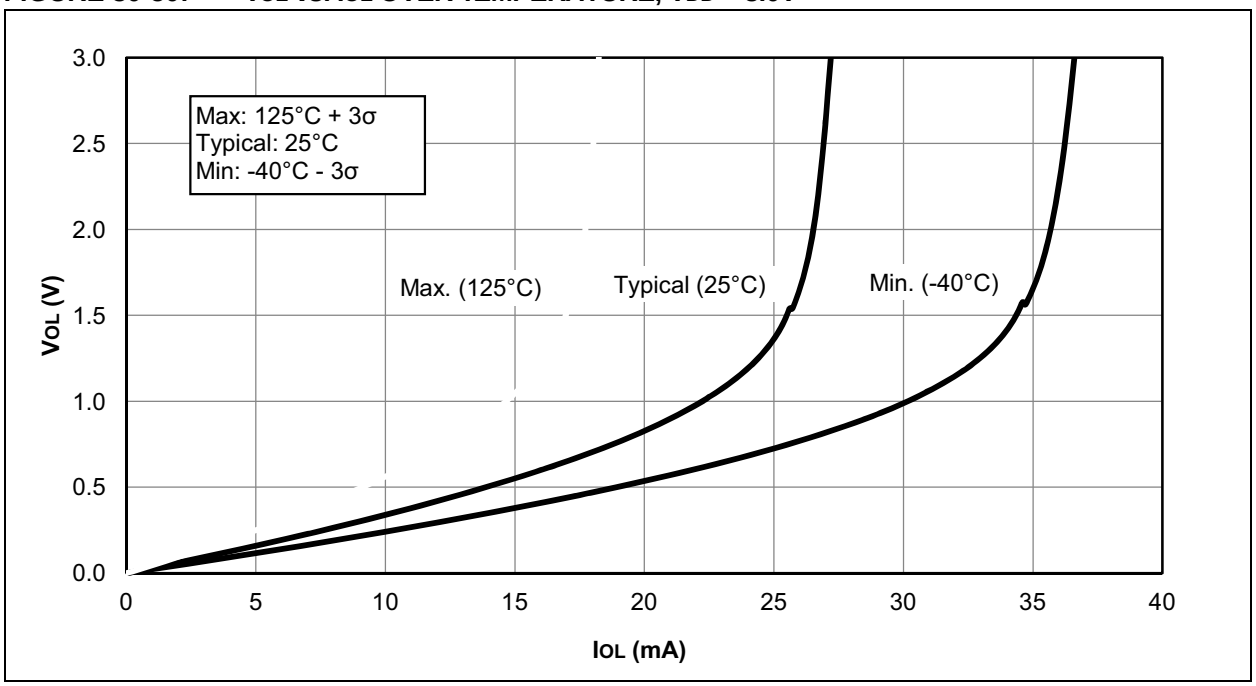


FIGURE 30-62: WDT TIME-OUT PERIOD

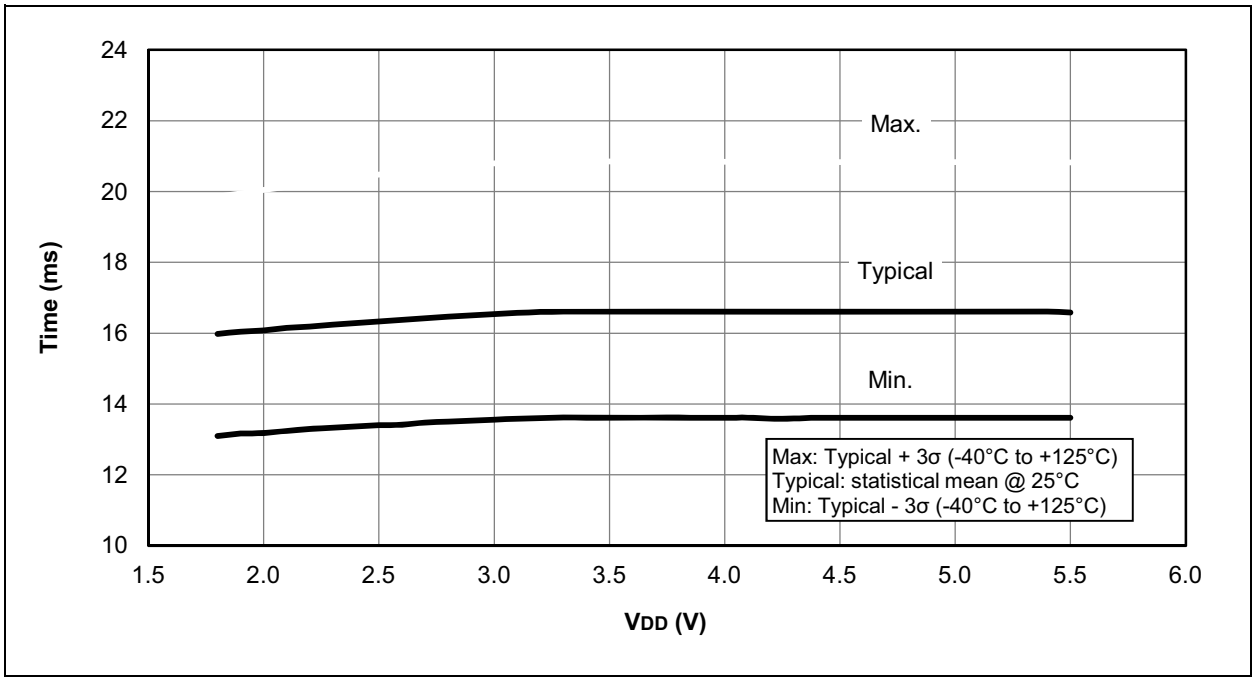


FIGURE 30-63: PWRT PERIOD