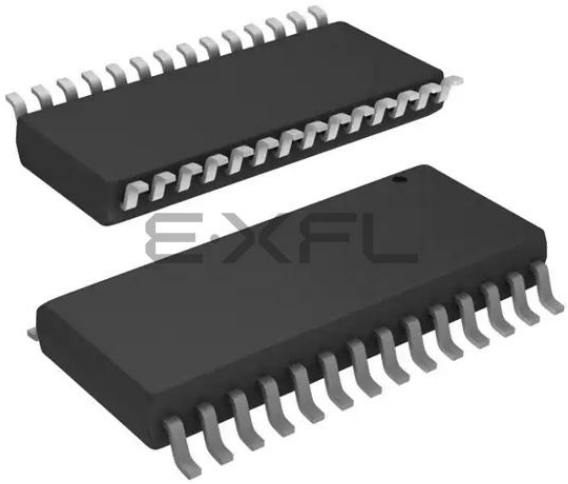


Welcome to E-XFL.COM

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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1938-i-so

TABLE 3-7: PIC16(L)F1938 MEMORY MAP, BANK 15

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	—
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	—
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	—
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	—
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	—
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h – 7EFh	Unimplemented Read as '0'

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

TABLE 3-8: PIC16(L)F1939 MEMORY MAP, BANK 15

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	LCDSE2
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	LCDDATA2
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	LCDDATA5
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	LCDDATA8
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	LCDDATA11
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h – 7EFh	Unimplemented Read as '0'

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

TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 6											
300h(2)	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
301h(2)	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
302h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
303h(2)	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
304h(2)	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
305h(2)	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
306h(2)	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
307h(2)	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
308h(2)	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000
309h(2)	WREG	Working Register								0000 0000	uuuu uuuu
30Ah(1, 2)	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
30Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
30Ch	—	Unimplemented								—	—
30Dh	—	Unimplemented								—	—
30Eh	—	Unimplemented								—	—
30Fh	—	Unimplemented								—	—
310h	—	Unimplemented								—	—
311h	CCPR3L	Capture/Compare/PWM Register 3 (LSB)								xxxx xxxx	uuuu uuuu
312h	CCPR3H	Capture/Compare/PWM Register 3 (MSB)								xxxx xxxx	uuuu uuuu
313h	CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<1:0>				0000 0000	0000 0000
314h	PWM3CON	P3RSEN	P3DC<6:0>							0000 0000	0000 0000
315h	CCP3AS	CCP3ASE	CCP3AS2	CCP3AS1	CCP3AS0	PSS3AC<1:0>		PSS3BD<1:0>		0000 0000	0000 0000
316h	PSTR3CON	—	—	—	STR3SYNC	STR3D	STR3C	STR3B	STR3A	---0 0001	---0 0001
317h	—	Unimplemented								—	—
318h	CCPR4L	Capture/Compare/PWM Register 4 (LSB)								xxxx xxxx	uuuu uuuu
319h	CCPR4H	Capture/Compare/PWM Register 4 (MSB)								xxxx xxxx	uuuu uuuu
31Ah	CCP4CON	—	—	DC4B<1:0>		CCP4M<3:0>				--00 0000	--00 0000
31Bh	—	Unimplemented								—	—
31Ch	CCPR5L	Capture/Compare/PWM Register 5 (LSB)								xxxx xxxx	uuuu uuuu
31Dh	CCPR5H	Capture/Compare/PWM Register 5 (MSB)								xxxx xxxx	uuuu uuuu
31Eh	CCP5CON	—	—	DC5B<1:0>		CCP5M<3:0>				--00 0000	--00 0000
31Fh	—	Unimplemented								—	—

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

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: These registers/bits are not implemented on PIC16(L)F1938 devices, read as '0'.

4: Unimplemented, read as '1'.

5.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the FOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run-time. See **Section 5.3 "Clock Switching"**for more information.

In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators and a dedicated Phase-Lock Loop, HFPLL that can produce one of three internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The HFINTOSC source is generated from the 500 kHz MFINTOSC source and the dedicated Phase-Lock Loop, HFPLL. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

5.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of multiple frequencies derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.7 "Internal Oscillator Clock Switch Timing"** for more information.

The HFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<2:0> = `100`, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '`1x`'.

A fast startup oscillator allows internal circuits to power up and stabilize before switching to HFINTOSC.

The High Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running.

The High Frequency Internal Oscillator Status Locked bit (HFIOFL) of the OSCSTAT register indicates when the HFINTOSC is running within 2% of its final value.

The High Frequency Internal Oscillator Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

5.2.2.2 MFINTOSC

The Medium-Frequency Internal Oscillator (MFINTOSC) is a factory calibrated 500 kHz internal clock source. The frequency of the MFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the MFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of nine frequencies derived from the MFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.7 "Internal Oscillator Clock Switch Timing"** for more information.

The MFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<2:0> = `100`, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '`1x`'

The Medium Frequency Internal Oscillator Ready bit (MFIOFR) of the OSCSTAT register indicates when the MFINTOSC is running.

5.6 Register Definitions: Oscillator Control

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF<3:0>				—	SCS<1:0>	
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 **SPLLEN:** Software PLL Enable bit
If PLLEN in Configuration Words = 1:
SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements)
If PLLEN in Configuration Words = 0:
1 = 4x PLL Is enabled
0 = 4x PLL is disabled

bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits
1111 = 16 MHz HF
1110 = 8 MHz or 32 MHz HF (see **Section 5.2.2.1 "HFINTOSC"**)
1101 = 4 MHz HF
1100 = 2 MHz HF
1011 = 1 MHz HF
1010 = 500 kHz HF(1)
1001 = 250 kHz HF(1)
1000 = 125 kHz HF(1)
0111 = 500 kHz MF (default upon Reset)
0110 = 250 kHz MF
0101 = 125 kHz MF
0100 = 62.5 kHz MF
0011 = 31.25 kHz HF(1)
0010 = 31.25 kHz MF
000x = 31 kHz LF

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

bit 1-0 **SCS<1:0>:** System Clock Select bits
1x = Internal oscillator block
01 = Timer1 oscillator
00 = Clock determined by FOSC<2:0> in Configuration Words.

Note 1: Duplicate frequency derived from HFINTOSC.

7.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

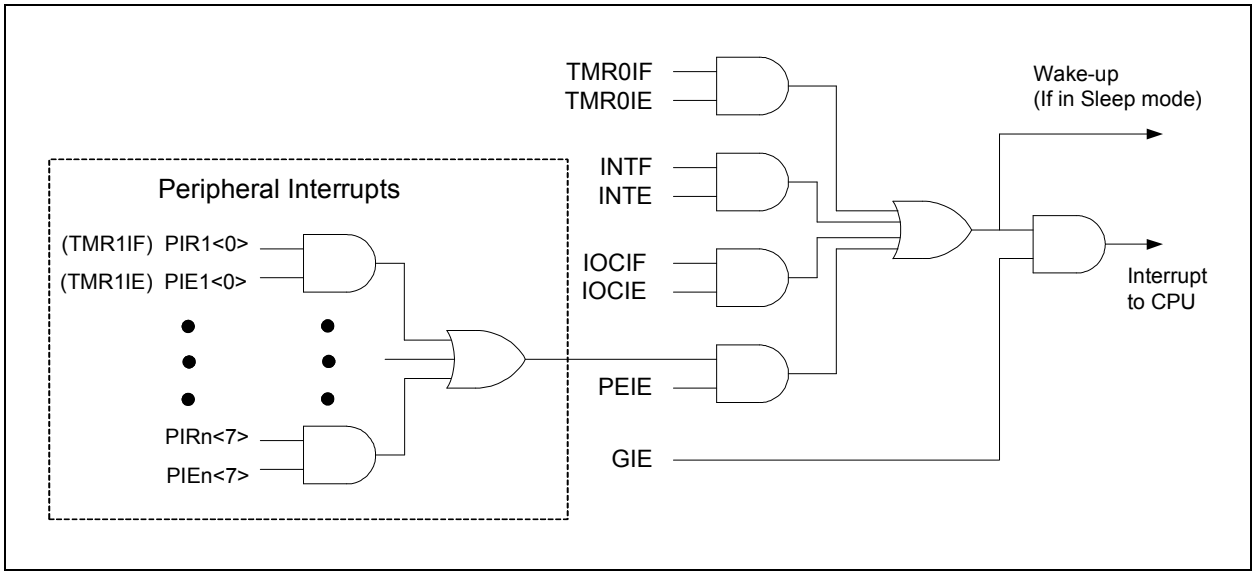
This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Automatic Context Saving

Many peripherals produce Interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 7-1.

FIGURE 7-1: INTERRUPT LOGIC



7.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to the **Section 9.0 "Power-Down Mode (Sleep)"** for more details.

7.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

7.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{TO}$ and $\overline{PD}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

REGISTER 11-5: EECON1: EEPROM CONTROL 1 REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **EEPGD:** Flash Program/Data EEPROM Memory Select bit
- 1 = Accesses program space Flash memory
- 0 = Accesses data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
- 1 = Accesses Configuration, User ID and Device ID Registers
- 0 = Accesses Flash Program or data EEPROM Memory

bit 5 **LWLO:** Load Write Latches Only bit

If CFGS = 1 (Configuration space) **OR** CFGS = 0 and EEPGD = 1 (program Flash):
- 1 = The next WR command does not initiate a write; only the program memory latches are updated.
- 0 = The next WR command writes a value from EEDATH:EEDATL into program memory latches and initiates a write of all the data stored in the program memory latches.

If CFGS = 0 and EEPGD = 0: (Accessing data EEPROM)
LWLO is ignored. The next WR command initiates a write to the data EEPROM.

bit 4 **FREE:** Program Flash Erase Enable bit

If CFGS = 1 (Configuration space) **OR** CFGS = 0 and EEPGD = 1 (program Flash):
- 1 = Performs an erase operation on the next WR command (cleared by hardware after completion of erase).
- 0 = Performs a write operation on the next WR command.

If EEPGD = 0 and CFGS = 0: (Accessing data EEPROM)
FREE is ignored. The next WR command will initiate both a erase cycle and a write cycle.

bit 3 **WRERR:** EEPROM Error Flag bit
- 1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write '1') of the WR bit).
- 0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1 **WR:** Write Control bit
- 1 = Initiates a program Flash or data EEPROM program/erase operation. The operation is self-timed and the bit is cleared by hardware once operation is complete. The WR bit can only be set (not cleared) in software.
- 0 = Program/erase operation to the Flash or data EEPROM is complete and inactive.

bit 0 **RD:** Read Control bit
- 1 = Initiates an program Flash or data EEPROM read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
- 0 = Does not initiate a program Flash or data EEPROM data read.

12.0 I/O PORTS

Each port has three standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

TABLE 12-1: PORT AVAILABILITY PER DEVICE

Device	PORTA	PORTB	PORTC
PIC16F1933	•	•	
PIC16F1934	•	•	•

TABLE 12-2: PORT AVAILABILITY PER DEVICE

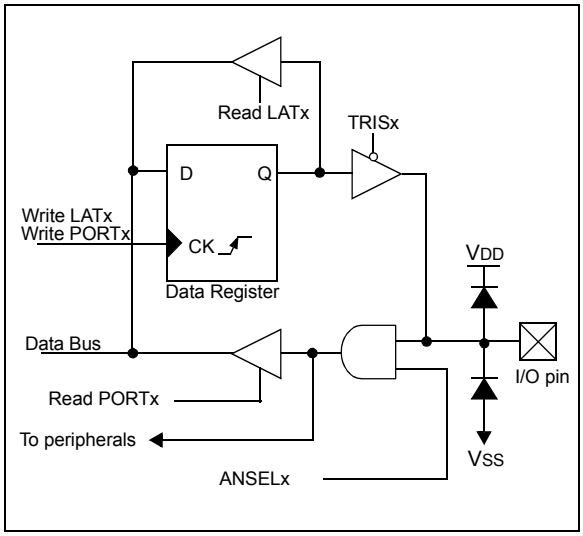
Device	PORTA	PORTB	PORTC	PORTD	PORTE
PIC16(L)F1938	•	•	•		•
PIC16(L)F1939	•	•	•	•	•

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 12-1.

FIGURE 12-1: GENERIC I/O PORT OPERATION



EXAMPLE 12-1: INITIALIZING PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL  PORTA        ;
CLRF     PORTA        ;Init PORTA
BANKSEL  LATA         ;Data Latch
CLRF     LATA         ;
BANKSEL  ANSELA       ;
CLRF     ANSELA       ;digital I/O
BANKSEL  TRISA        ;
MOVLW    B'00111000'  ;Set RA<5:3> as inputs
MOVWF    TRISA        ;and set RA<2:0> as
                      ;outputs
```

FIGURE 17-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM

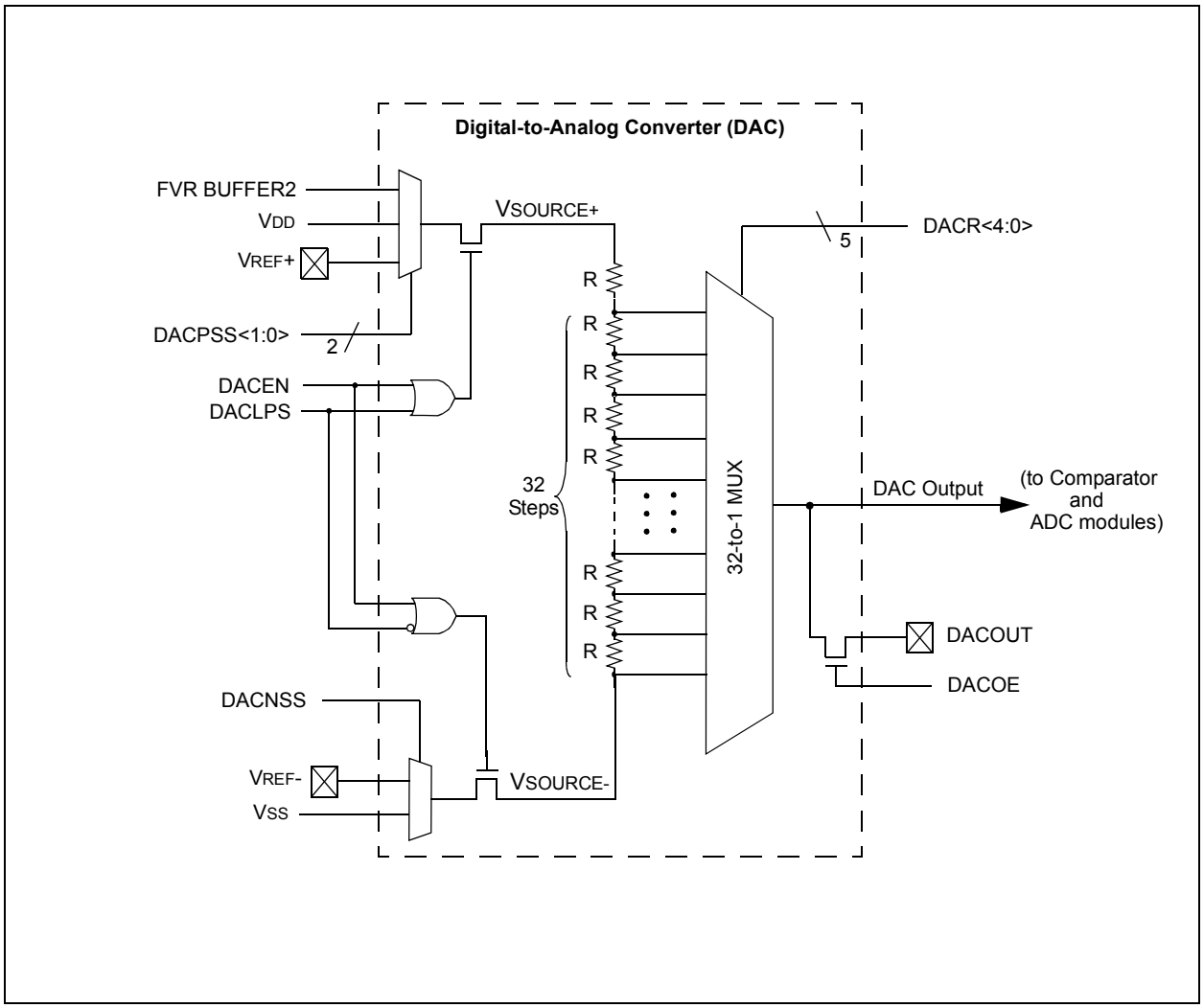


FIGURE 17-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

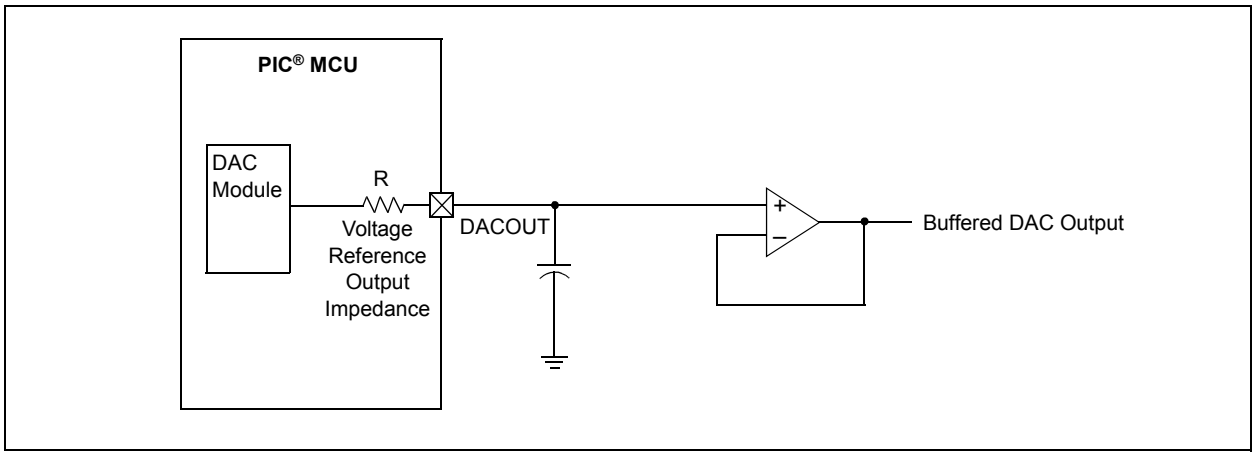


FIGURE 24-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

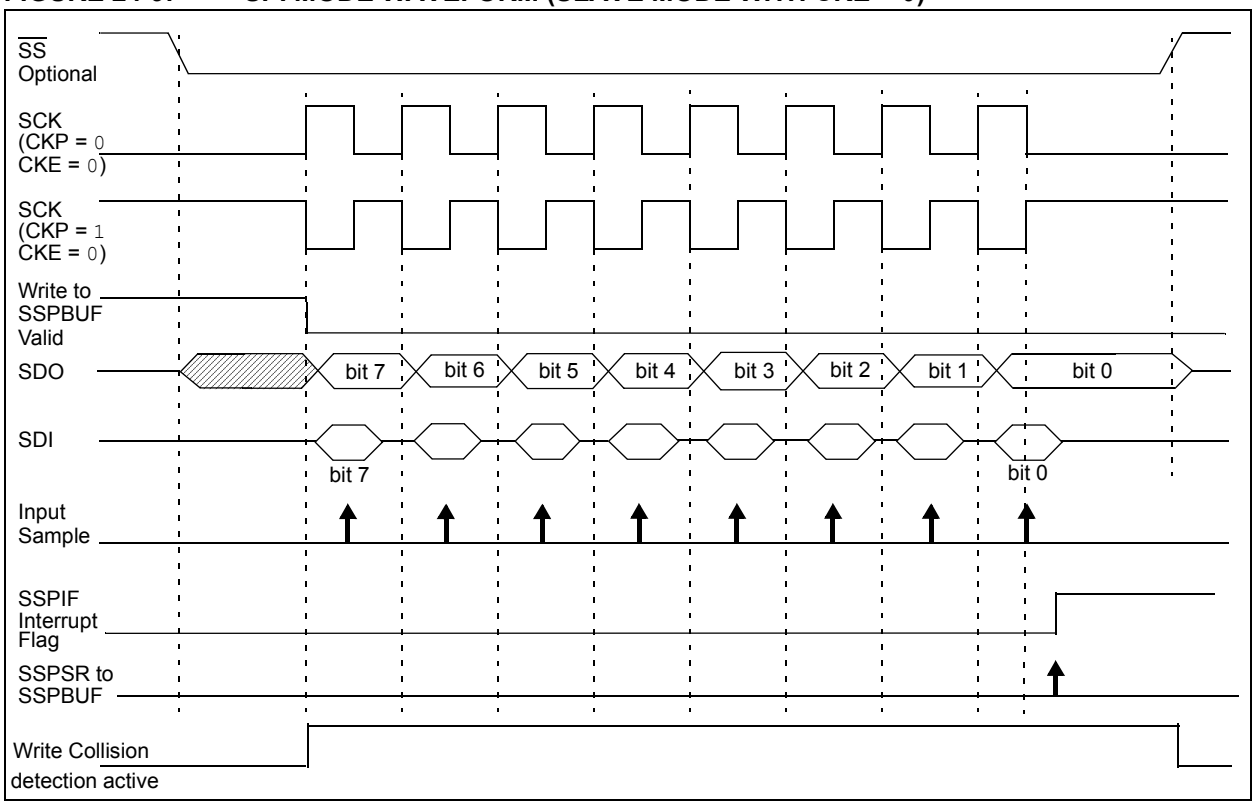


FIGURE 24-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)

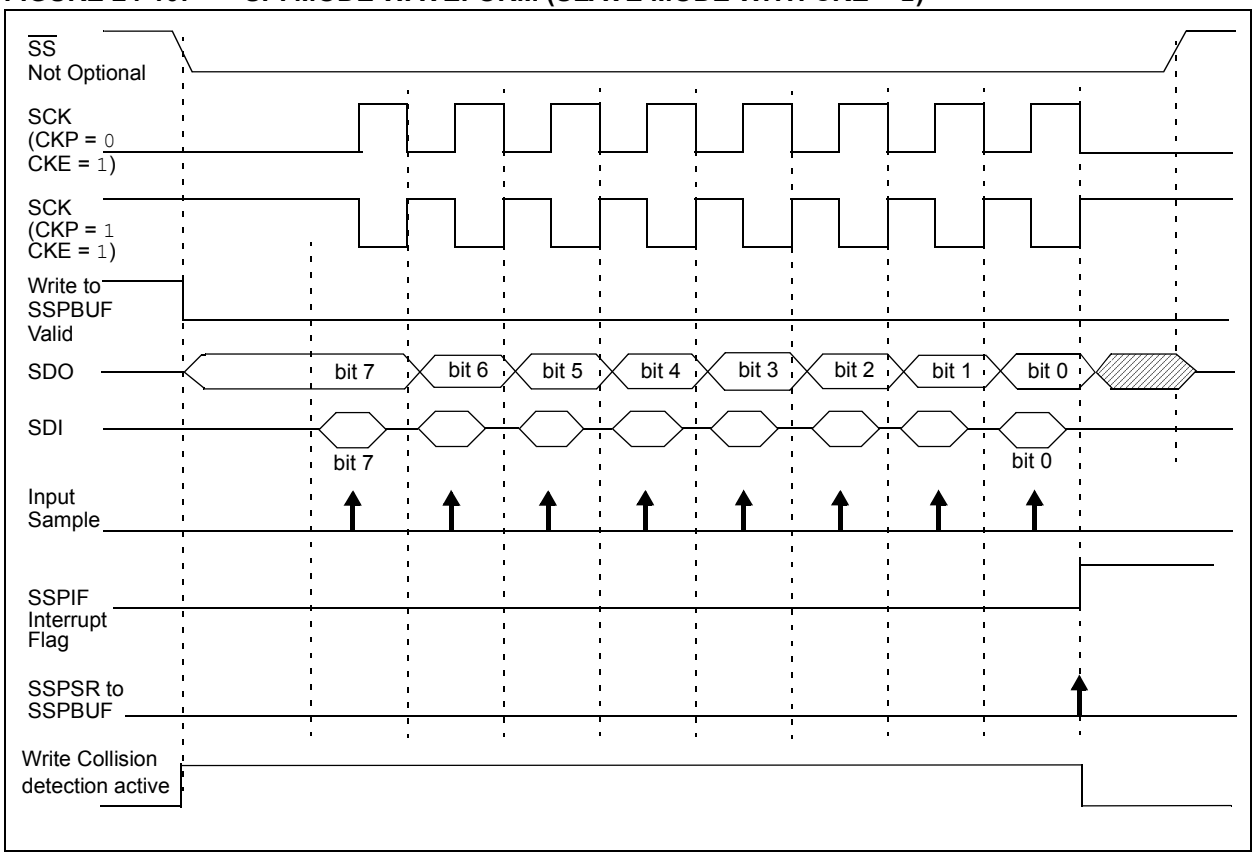
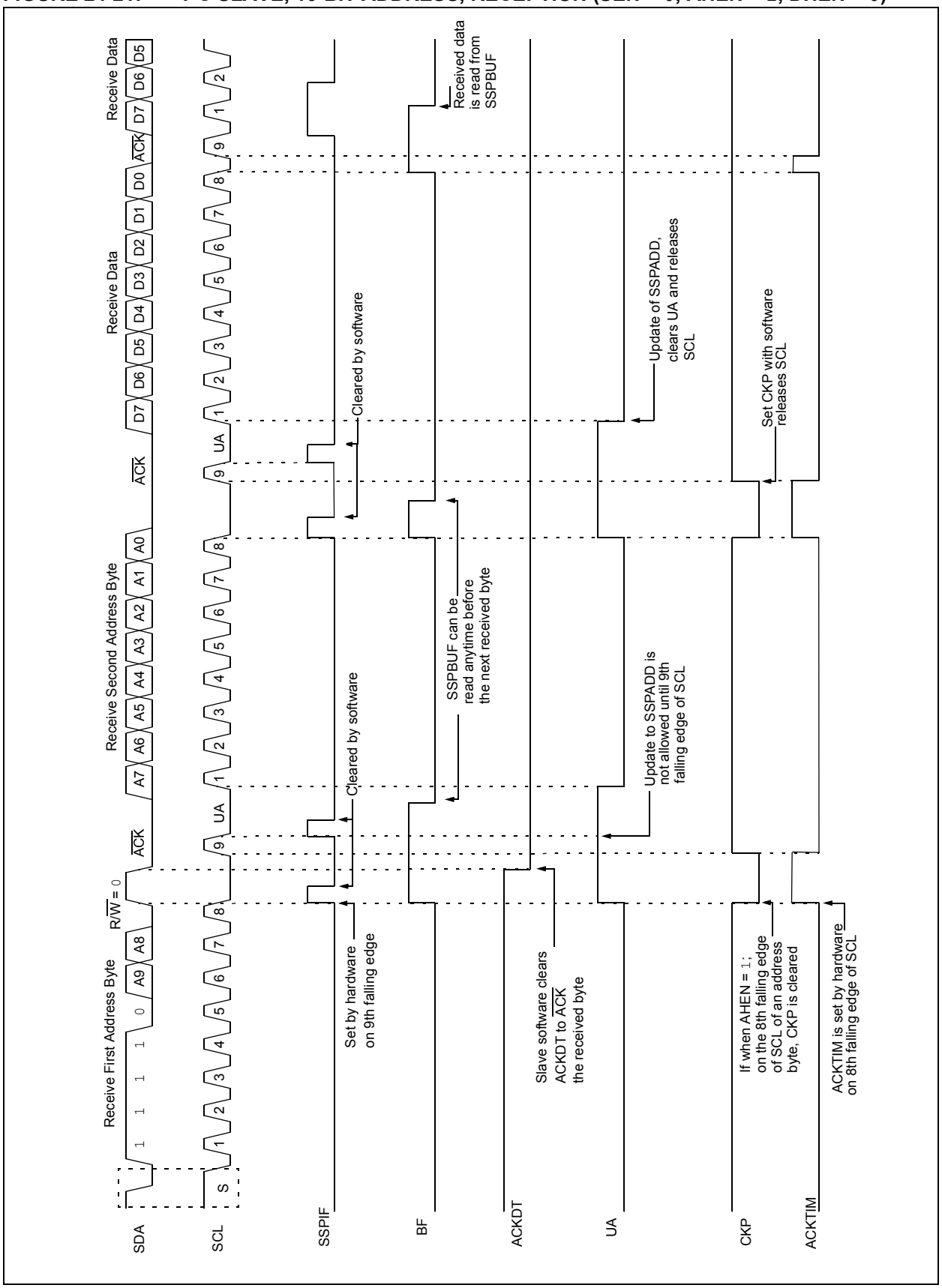


FIGURE 24-21: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)



24.6.13.3 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 zero. 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 24-38). 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 24-39).

FIGURE 24-38: BUS COLLISION DURING A STOP CONDITION (CASE 1)

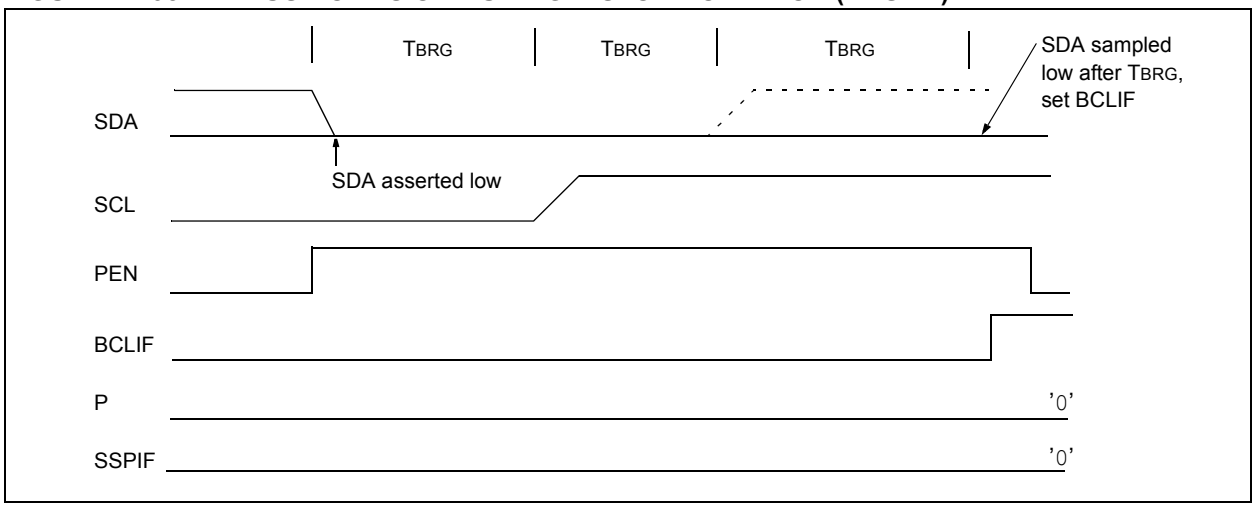
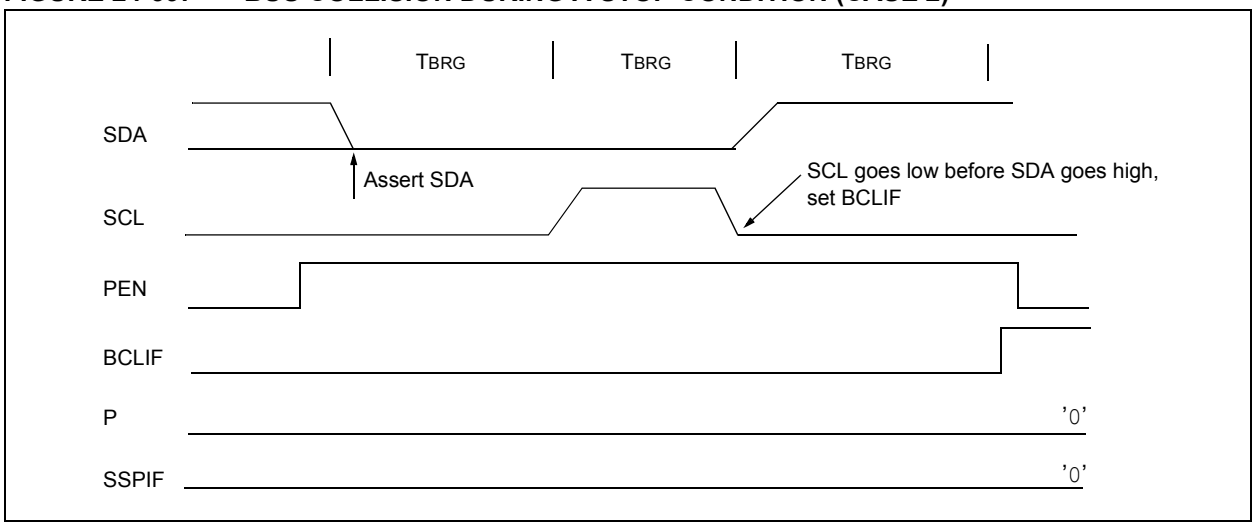


FIGURE 24-39: BUS COLLISION DURING A STOP CONDITION (CASE 2)



REGISTER 24-2: SSPCON1: SSP CONTROL REGISTER 1

R/C/HS-0/0	R/C/HS-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
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/n = Value at POR and BOR/Value at all other Resets	
'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 I²C 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)
In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register (must be cleared in software).
0 = No overflow
In I²C mode:
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
In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and $\overline{SS}$ as the source of the serial port pins(2)
0 = Disables serial port and configures these pins as I/O port pins
In I²C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins(3)
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I²C Slave mode:
SCL release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I²C Master mode:
Unused in this mode

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = FOSC/4
0001 = SPI Master mode, clock = FOSC/16
0010 = SPI Master mode, clock = FOSC/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0110 = I²C Slave mode, 7-bit address
0111 = I²C Slave mode, 10-bit address
1000 = I²C Master mode, clock = FOSC / (4 * (SSPADD+1))(4)
1001 = Reserved
1010 = SPI Master mode, clock = FOSC/(4 * (SSPADD+1))(5)
1011 = I²C firmware controlled Master mode (Slave idle)
1100 = Reserved
1101 = Reserved
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = I²C 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, these pins must be properly configured as input or output.
3: When enabled, the SDA and SCL pins must be configured as inputs.
4: SSPADD values of 0, 1 or 2 are not supported for I²C mode.
5: SSPADD value of '0' is not supported. Use SSPM = 0000 instead.

TABLE 29-3: PIC16(L)F193X ENHANCED 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	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	–	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
BYTE ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
BIT-ORIENTED SKIP OPERATIONS									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLP	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

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'.

TRIS **Load TRIS Register with W**

Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) → TRIS register 'f'
Status Affected:	None
Description:	Move data from W register to TRIS register. When 'f' = 5, TRISA is loaded. When 'f' = 6, TRISB is loaded. When 'f' = 7, TRISC is loaded.

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:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (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'.

30.4 DC Characteristics: PIC16(L)F1938/39-I/E

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O PORT:					
D032		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D032A			—	—	0.15 VDD	V	1.8V ≤ VDD ≤ 4.5V
D033		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
		with I²C™ levels	—	—	0.3 VDD	V	
		with SMBus levels	—	—	0.8	V	2.7V ≤ VDD ≤ 5.5V
D034		$\overline{MCLR}$, OSC1 (RC mode)(1)	—	—	0.2 VDD	V	
D034A		OSC1 (HS mode)	—	—	0.3 VDD	V	
	VIH	**Input High Voltage**					
		I/O ports:					
D040		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD ≤ 4.5V
D041		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
		with I²C™ levels	0.7 VDD	—	—	V	
		with SMBus levels	2.1	—	—	V	2.7V ≤ VDD ≤ 5.5V
D042		$\overline{MCLR}$	0.8 VDD	—	—	V	
D043A		OSC1 (HS mode)	0.7 VDD	—	—	V	
D043B		OSC1 (RC mode)	0.9 VDD	—	—	V	**(Note 1)** VDD > 2.0V
	IIL	**Input Leakage Current(2)**					
D060		I/O ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance @ 85°C
				± 5	± 1000	nA	125°C
D061		$\overline{MCLR}$(3)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD @ 85°C
	IPUR	**Weak Pull-up Current**					
D070*			25	100	200	μA	VDD = 3.3V, VPIN = VSS
			25	140	300		VDD = 5.0V, VPIN = VSS
	VOL	**Output Low Voltage(4)**					
D080		I/O ports	—	—	0.6	V	IOL = 8mA, VDD = 5V IOL = 6mA, VDD = 3.3V IOL = 1.8mA, VDD = 1.8V
	VOH	**Output High Voltage(4)**					
D090		I/O ports	VDD - 0.7	—	—	V	IOH = 3.5mA, VDD = 5V IOH = 3mA, VDD = 3.3V IOH = 1mA, VDD = 1.8V

* 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: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

2: Negative current is defined as current sourced by the pin.

3: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

4: Including OSC2 in CLKOUT mode.

FIGURE 31-5: IDD TYPICAL, XT AND EXTRC OSCILLATOR, PIC16F1938/9 ONLY

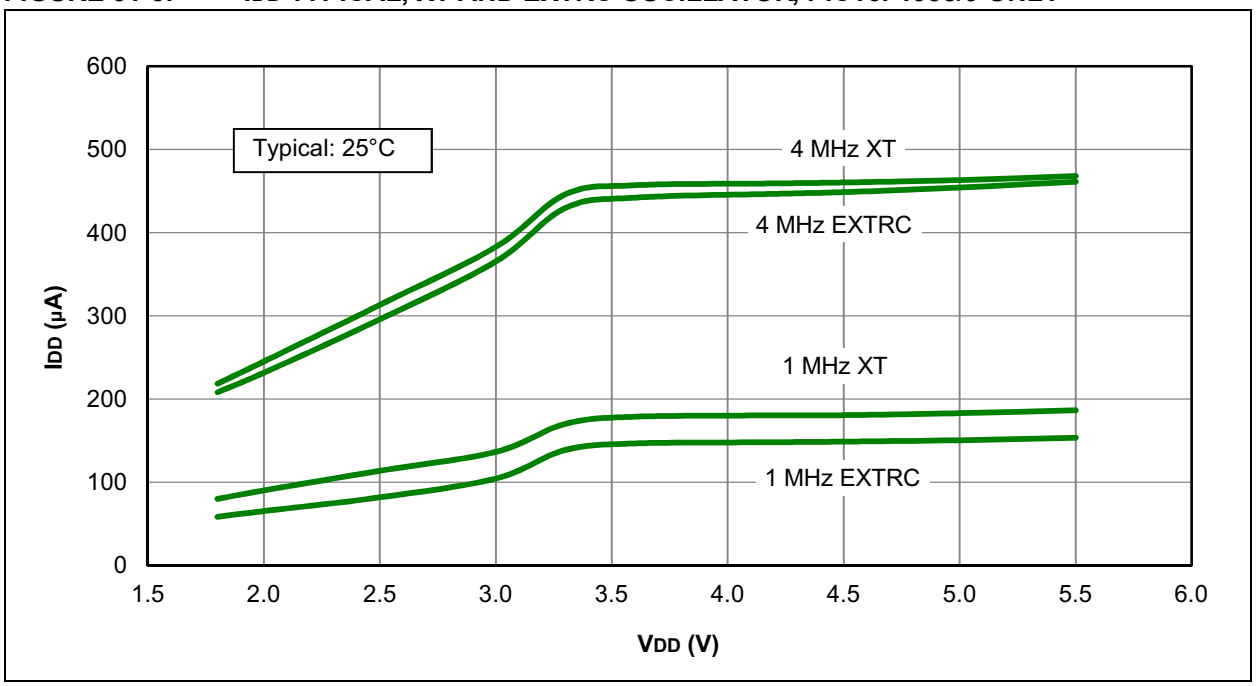


FIGURE 31-6: IDD MAXIMUM, XT AND EXTRC OSCILLATOR, PIC16F1938/9 ONLY

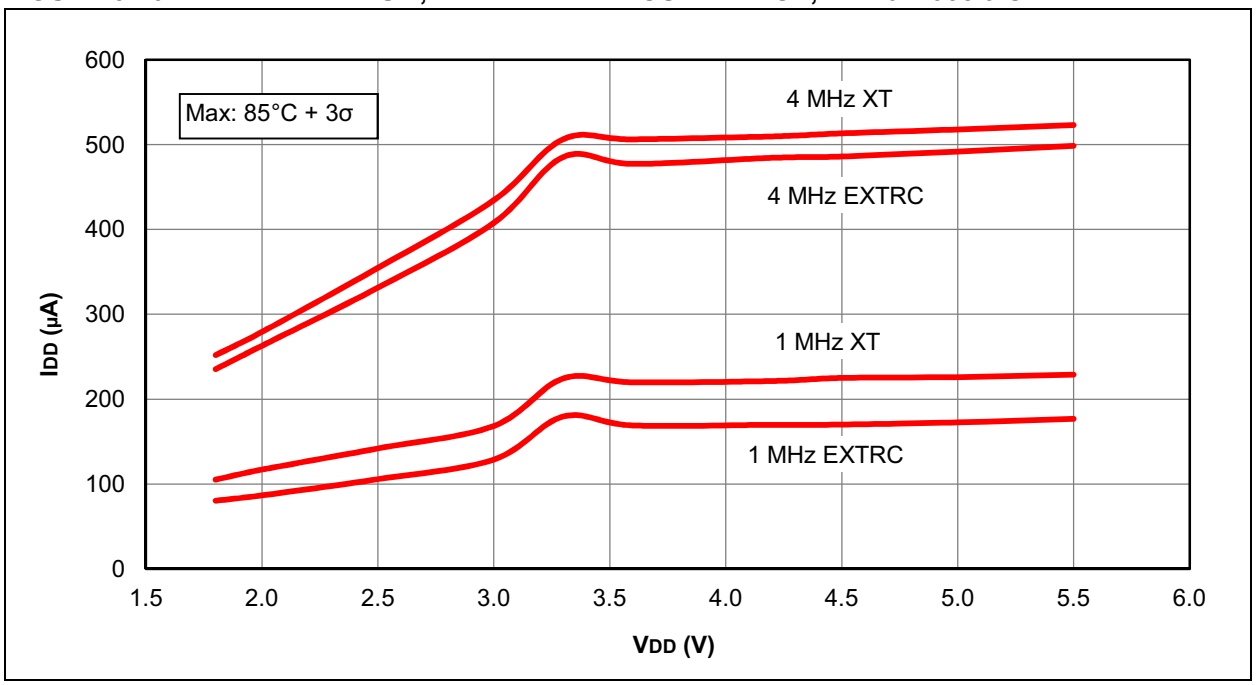


FIGURE 31-53: VOH VS. IOH OVER TEMPERATURE, VDD = 3.0V

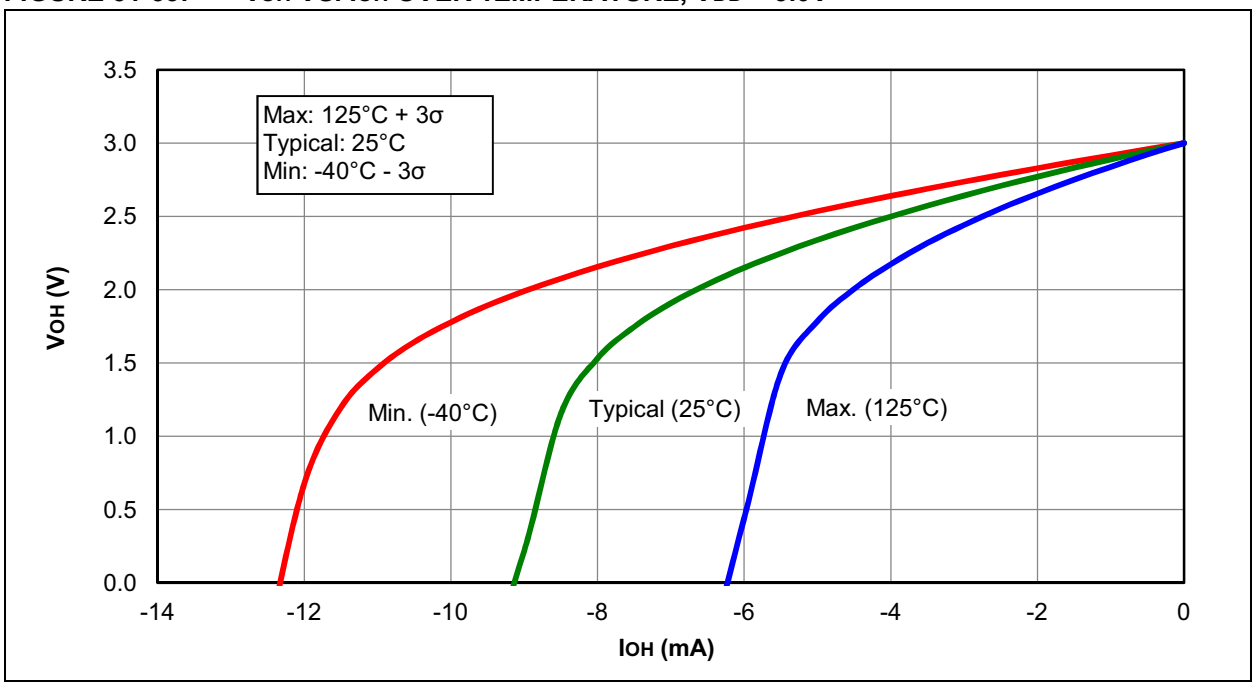


FIGURE 31-54: VOL VS. IOL OVER TEMPERATURE, VDD = 3.0V

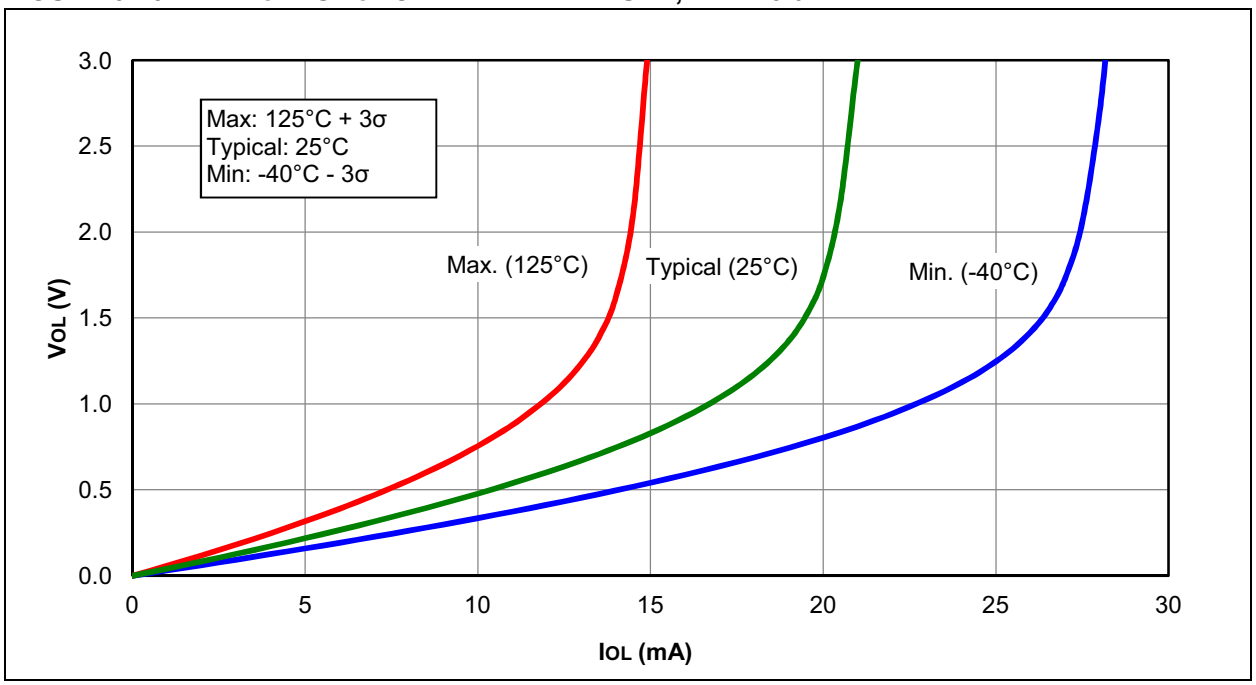


FIGURE 31-55: VOH VS. IOH OVER TEMPERATURE, VDD = 1.8V

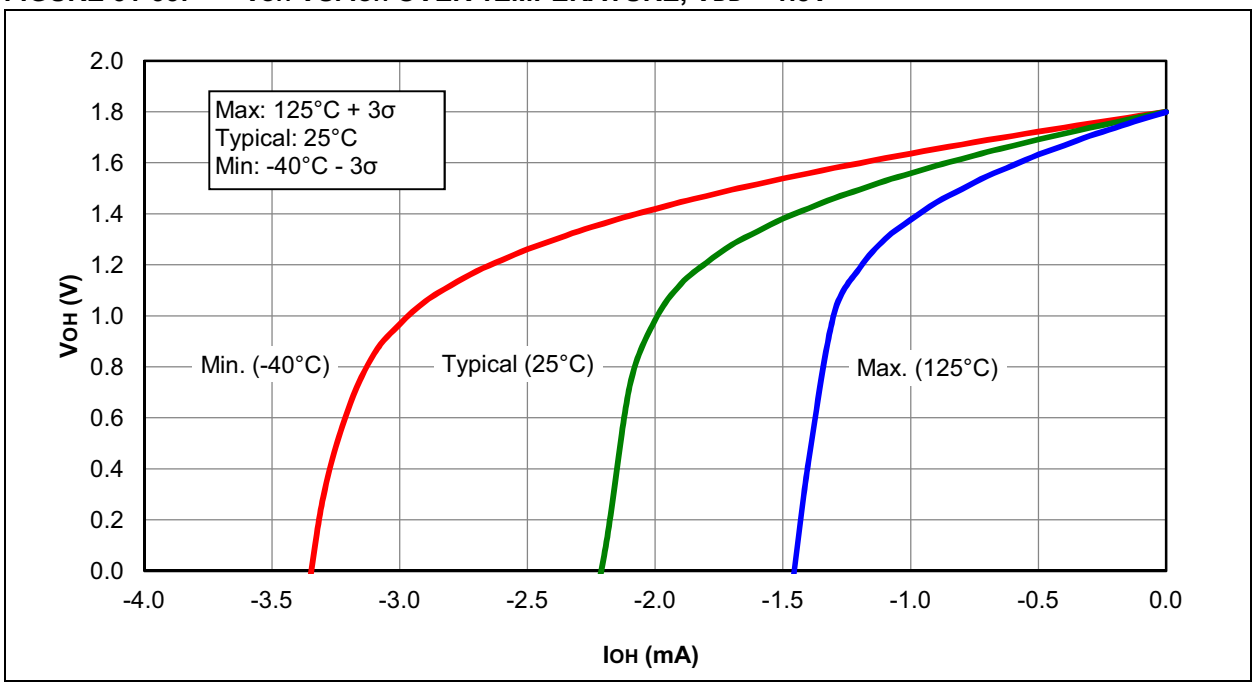


FIGURE 31-56: VOL VS. IOL OVER TEMPERATURE, VDD = 1.8V

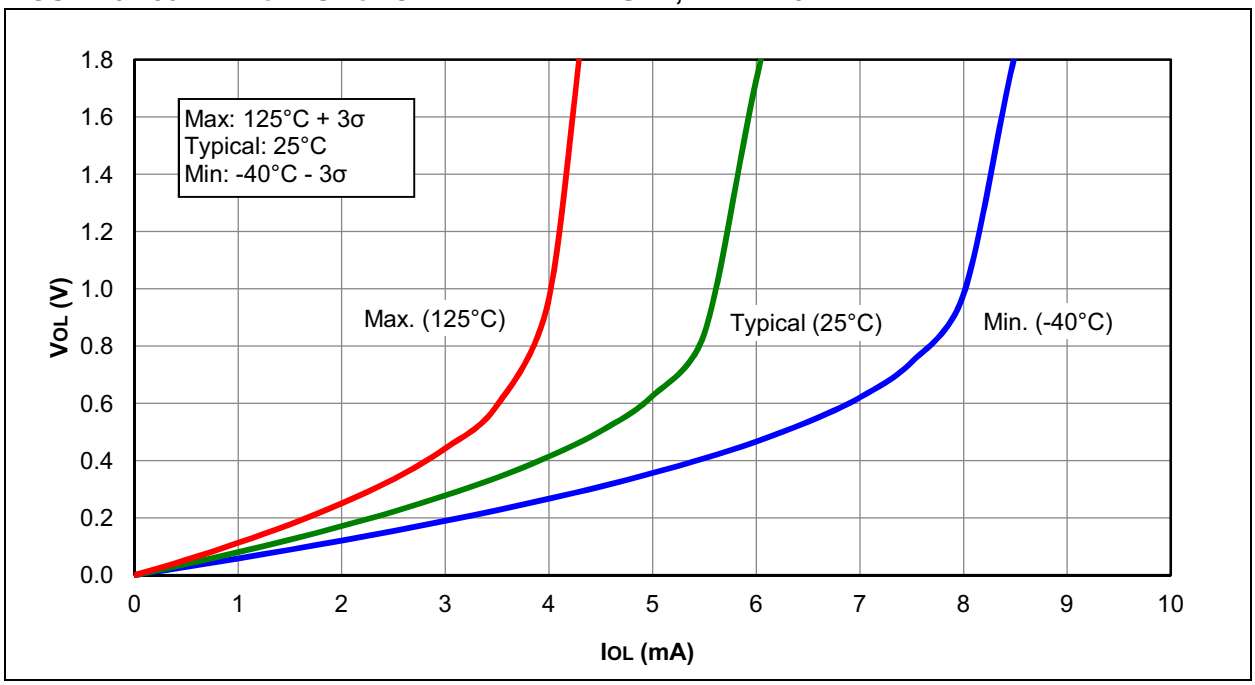


FIGURE 31-65: COMPARATOR HYSTERESIS, NORMAL-POWER MODE (CxSP = 1, CxHYS = 1)

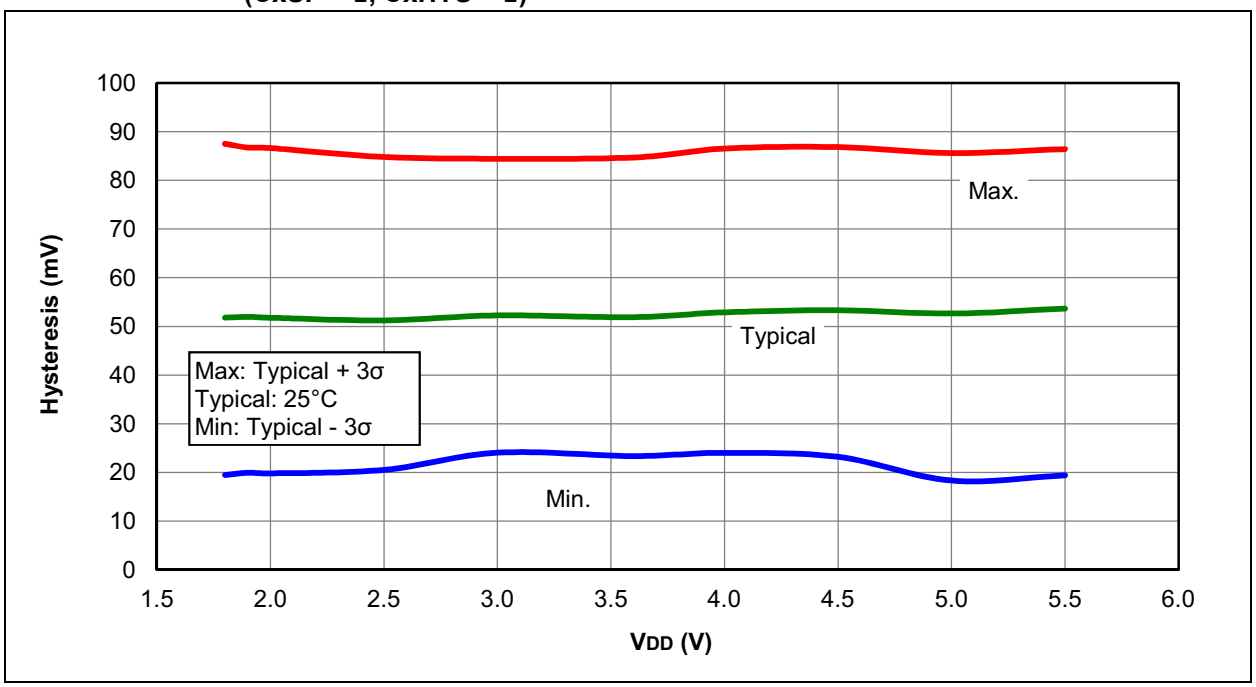


FIGURE 31-66: COMPARATOR HYSTERESIS, LOW-POWER MODE (CxSP = 0, CxHYS = 1)