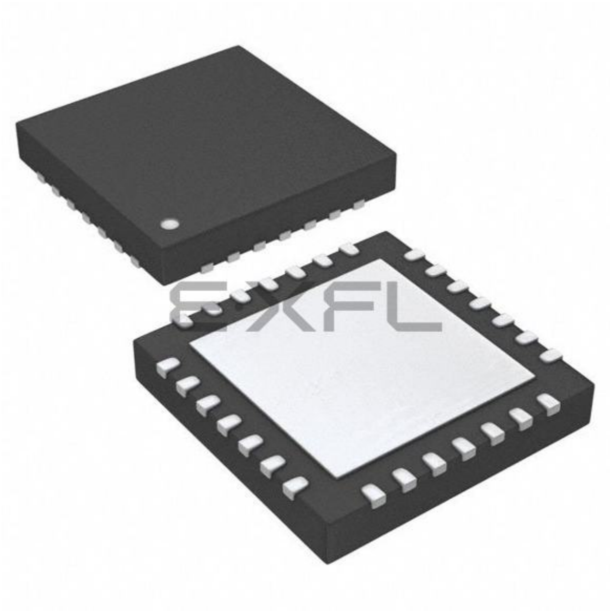
Microchip Technology - PIC16LF15355T-I/ML Datasheet



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	224 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf15355t-i-ml

3.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 10.5 "Automatic Context Saving"** for more information.

3.2 16-Level Stack with Overflow and Underflow

These devices have a hardware stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and if enabled, will cause a software Reset. See **Section 4.5 "Stack"** for more details.

3.3 File Select Registers

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers and program memory, which allows one Data Pointer for all memory. When an FSR points to program memory, there is one additional instruction cycle in instructions using INDF to allow the data to be fetched. General purpose memory can also be addressed linearly, providing the ability to access contiguous data larger than 80 bytes. See **Section 4.6 "Indirect Addressing"** for more details.

3.4 Instruction Set

There are 48 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 36.0 "Instruction Set Summary"** for more details.

4.3.3 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes of the data banks 0-59 and 100 bytes of the data banks 60-63, after the core registers.

The SFRs associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

4.3.4 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank.

4.3.4.1 Linear Access to GPR

The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 4.6.2 “Linear Data Memory”** for more information.

4.3.5 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

4.3.6 DEVICE MEMORY MAPS

The memory maps are as shown in Table 4-4 through Table 4-9.

TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (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: MCLR
Bank 2											
CPU CORE REGISTERS; see Table 4-3 for specifics											
10Ch — 118h	—	Unimplemented								—	—
119h	RC1REG	EUSART Receive Data Register								0000 0000	0000 0000
11Ah	TX1REG	EUSART Transmit Data Register								0000 0000	0000 0000
11Bh	SP1BRGL	SP1BRG<7:0>								0000 0000	0000 0000
11Ch	SP1BRGH	SP1BRG<15:8>								0000 0000	0000 0000
11Dh	RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 0000	0000 0000
11Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
11Fh	BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00

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

TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (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: $\overline{\text{MCLR}}$
Bank 21-59											
CPU CORE REGISTERS; see Table 4-3 for specifics											
x0Ch/ x8Ch — x1Fh/ x9Fh	—	Unimplemented								—	—

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

7.0 DEVICE CONFIGURATION INFORMATION

The Device Configuration Information (DCI) is a dedicated region in the Program Flash Memory mapped from 8200h to 821Fh. The data stored in the DCI memory is hard-coded into the device during manufacturing.

Refer to Table 7-1: Device Configuration Information for PIC16(L)F15354/55 Devices for the complete DCI table address and description. The DCI holds information about the device which is useful for programming and bootloader applications. These locations are read-only and cannot be erased or modified.

TABLE 7-1: DEVICE CONFIGURATION INFORMATION FOR PIC16(L)F15354/55 DEVICES

ADDRESS	Name	DESCRIPTION	VALUE		UNITS
			PIC16(L)F15354	PIC16(L)F15355	
8200h	ERSIZ	Erase Row Size	32	32	Words
8201h	WLSIZ	Number of write latches	32	32	Latches
8202h	URSIZ	Number of User Rows	128	256	Rows
8203h	EESIZ	EE Data memory size	0	0	Bytes
8204h	PCNT	Pin Count	28	28	Pins

7.1 DIA and DCI Access

The DIA and DCI data are read-only and cannot be erased or modified. See **13.3.6 "NVMREG Access to Device Information Area, Device Configuration Area, User ID, Device ID and Configuration Words"** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the DIA and DCI regions, similar to the Device ID and Revision ID.

REGISTER 9-5: OSCEN: OSCILLATOR MANUAL ENABLE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EXTOEN	HFOEN	MFOEN	LFOEN	SOSCEN	ADOEN	—	—
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 **EXTOEN:** External Oscillator Manual Request Enable bit[1]
1 = EXTOSC is explicitly enabled, operating as specified by FEXTOSC
0 = EXTOSC could be enabled by some modules

bit 6 **HFOEN:** HFINTOSC Oscillator Manual Request Enable bit
1 = HFINTOSC is explicitly enabled, operating as specified by OSCFRQ
0 = HFINTOSC could be enabled by another module

bit 5 **MFOEN:** MFINTOSC Oscillator Manual Request Enable bit
1 = MFINTOSC is explicitly enabled
0 = MFINTOSC could be enabled by another module

bit 4 **LFOEN:** LFINTOSC (31 kHz) Oscillator Manual Request Enable bit
1 = LFINTOSC is explicitly enabled
0 = LFINTOSC could be enabled by another module

bit 3 **SOSCEN:** Secondary (Timer1) Oscillator Manual Request bit
1 = Secondary oscillator is explicitly enabled, operating as specified by SOSCPWR
0 = Secondary oscillator could be enabled by another module

bit 2 **ADOEN:** FRC Oscillator Manual Request Enable bit
1 = FRC is explicitly enabled
0 = FRC could be enabled by another module

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

REGISTER 10-13: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF
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 = Hardware clearable

bit 7 **RC2IF:** EUSART2 Receive Interrupt Enable bit

1 = The EUSART2 receive buffer is not empty (contains at least one byte)
0 = The EUSART2 receive buffer is empty

bit 6 **TX2IF:** EUSART2 Transmit Interrupt Enable bit

1 = The EUSART2 transmit buffer contains at least one unoccupied space
0 = The EUSART2 transmit buffer is currently full. The application firmware should not write to TXxREG

bit 5 **RC1IF:** EUSART1 Receive Interrupt Flag (read-only) bit [(1)]

1 = The EUSART1 receive buffer is not empty (contains at least one byte)
0 = The EUSART1 receive buffer is empty

bit 4 **TX1IF:** EUSART1 Transmit Interrupt Flag (read-only) bit[(2)]

1 = The EUSART1 transmit buffer contains at least one unoccupied space
0 = The EUSART1 transmit buffer is currently full. The application firmware should not write to TXxREG again, until more room becomes available in the transmit buffer.

bit 3 **BCL2IF:** MSSP2 Bus Collision Interrupt Flag bit

1 = A bus collision was detected (must be cleared in software)
0 = No bus collision was detected

bit 2 **SSP2IF:** Synchronous Serial Port (MSSP2) Interrupt Flag bit

1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)
0 = Waiting for the Transmission/Reception/Bus Condition in progress

bit 1 **BCL1IF:** MSSP1 Bus Collision Interrupt Flag bit

1 = A bus collision was detected (must be cleared in software)
0 = No bus collision was detected

bit 0 **SSP1IF:** Synchronous Serial Port (MSSP1) Interrupt Flag bit

1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)
0 = Waiting for the Transmission/Reception/Bus Condition in progress

Note 1: The RC1IF flag is a read-only bit. To clear the RC1IF flag, the firmware must read from RCxREG enough times to remove all bytes from the receive buffer.

2: The TX1IF flag is a read-only bit, indicating if there is room in the transmit buffer. To clear the TX1IF flag, the firmware must write enough data to TXxREG to completely fill all available bytes in the buffer. The TX1IF flag does not indicate transmit completion (use TRMT for this purpose instead).

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, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 14-7: SLRCONA: PORTA SLEW RATE CONTROL 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
SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0
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 7-0 **SLRA<7:0>:** PORTA Slew Rate Enable bits
For RA<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 14-8: INLVLA: PORTA INPUT LEVEL CONTROL 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
INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
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 7-0 **INLVLA<7:0>:** PORTA Input Level Select bits
For RA<7:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	173
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	173
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	174
ANSELA	ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	174
WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	175
ODCONA	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	175
SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0	176
INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	176

Legend: x = unknown, u = unchanged, – = unimplemented locations read as ‘0’. Shaded cells are not used by PORTA.

Note 1: Unimplemented, read as ‘1’.

23.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 23-1) contains Control and Status bits for the following:

- Enable
- Output
- Output polarity
- Hysteresis enable
- Timer1 output synchronization

The CMxCON1 register (see Register 23-2) contains Control bits for the following:

- Interrupt on positive/negative edge enables
- The CMxNSEL and CMxPSEL (Register 23-3 and Register 23-4) contain control bits for the following:
  - Positive input channel selection
  - Negative input channel selection

23.2.1 COMPARATOR ENABLE

Setting the CxON bit of the CMxCON0 register enables the comparator for operation. Clearing the CxON bit disables the comparator resulting in minimum current consumption.

23.2.2 COMPARATOR OUTPUT

The output of the comparator can be monitored by reading either the CxOUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register.

The comparator output can also be routed to an external pin through the RxyPPS register (Register 15-2). The corresponding TRIS bit must be clear to enable the pin as an output.

Note 1: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

23.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the CxPOL bit of the CMxCON0 register. Clearing the CxPOL bit results in a non-inverted output.

Table 23-2 shows the output state versus input conditions, including polarity control.

TABLE 23-2: COMPARATOR OUTPUT STATE VS. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

REGISTER 23-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
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 7-2 **Unimplemented:** Read as ‘0’

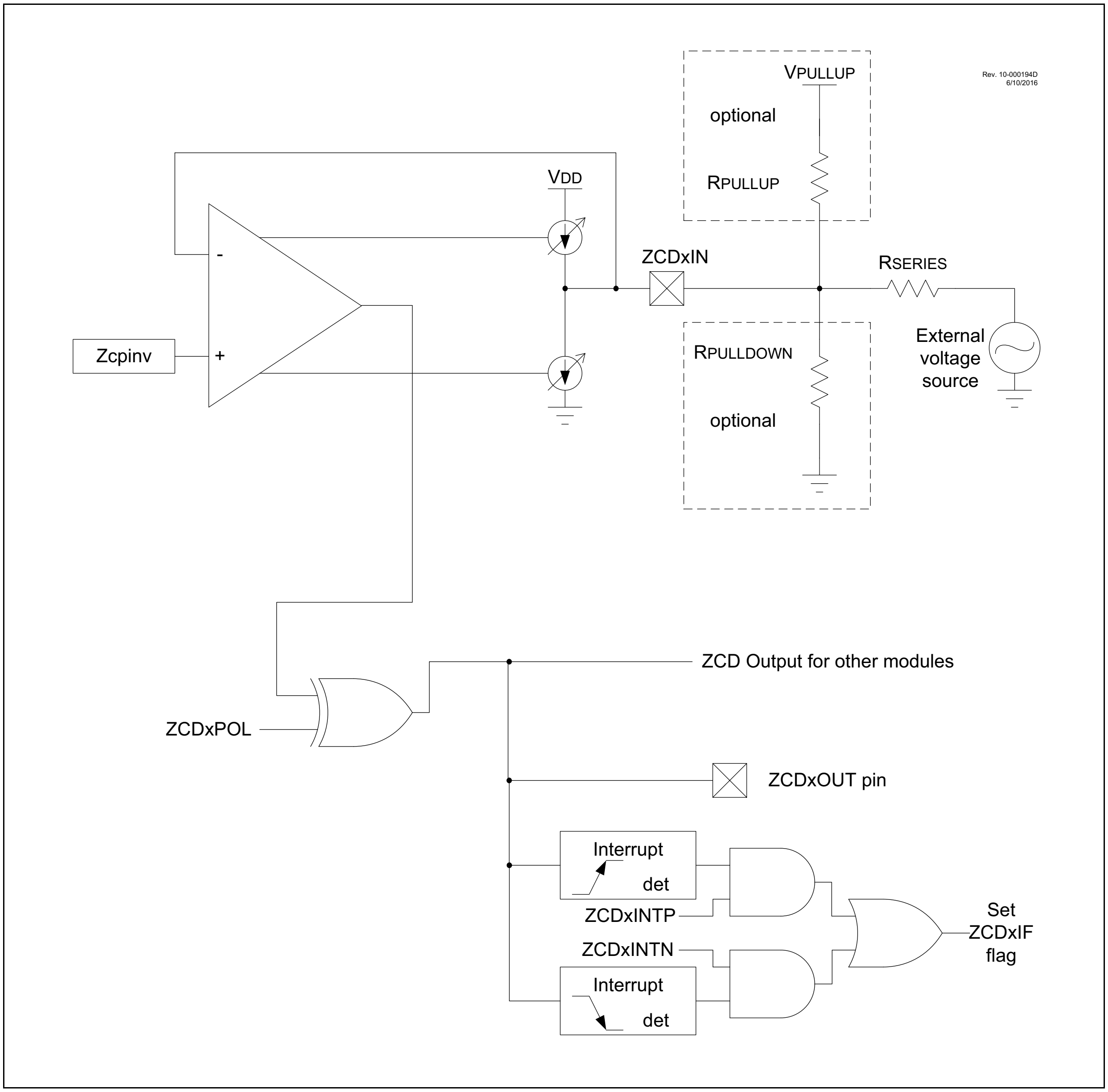
bit 1 **INTP:** Comparator Interrupt on Positive-Going Edge Enable bits

1 = The CxIF interrupt flag will be set upon a positive-going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive-going edge of the CxOUT bit

bit 0 **INTN:** Comparator Interrupt on Negative-Going Edge Enable bits

1 = The CxIF interrupt flag will be set upon a negative-going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative-going edge of the CxOUT bit

FIGURE 24-2: SIMPLIFIED ZCD BLOCK DIAGRAM



REGISTER 30-7: CWG1STR: CWG1 STEERING CONTROL REGISTER(1)

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
OVRD	OVRC	OVRB	OVRA	STRD(2)	STRC(2)	STRB(2)	STRA(2)
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	q = Value depends on condition

bit 7 **OVRD:** Steering Data D bit

bit 6 **OVRC:** Steering Data C bit

bit 5 **OVRB:** Steering Data B bit

bit 4 **OVRA:** Steering Data A bit

bit 3 **STRD:** Steering Enable D bit(2)
1 = CWG1D output has the CWG1_data waveform with polarity control from POLD bit
0 = CWG1D output is assigned the value of OVRD bit

bit 2 **STRC:** Steering Enable C bit(2)
1 = CWG1C output has the CWG1_data waveform with polarity control from POLC bit
0 = CWG1C output is assigned the value of OVRC bit

bit 1 **STRB:** Steering Enable B bit(2)
1 = CWG1B output has the CWG1_data waveform with polarity control from POLB bit
0 = CWG1B output is assigned the value of OVRB bit

bit 0 **STRA:** Steering Enable A bit(2)
1 = CWG1A output has the CWG1_data waveform with polarity control from POLA bit
0 = CWG1A output is assigned the value of OVRA bit

Note 1: The bits in this register apply only when MODE<2:0> = 00x.

2: This bit is effectively double-buffered when MODE<2:0> = 001.

32.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 32-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 32-6, Figure 32-8, Figure 32-9 and Figure 32-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)
- Timer2 output/2
- FOSC/(4 * (SSPxADD + 1))

Figure 32-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

FIGURE 32-6: SPI MODE WAVEFORM (MASTER MODE)

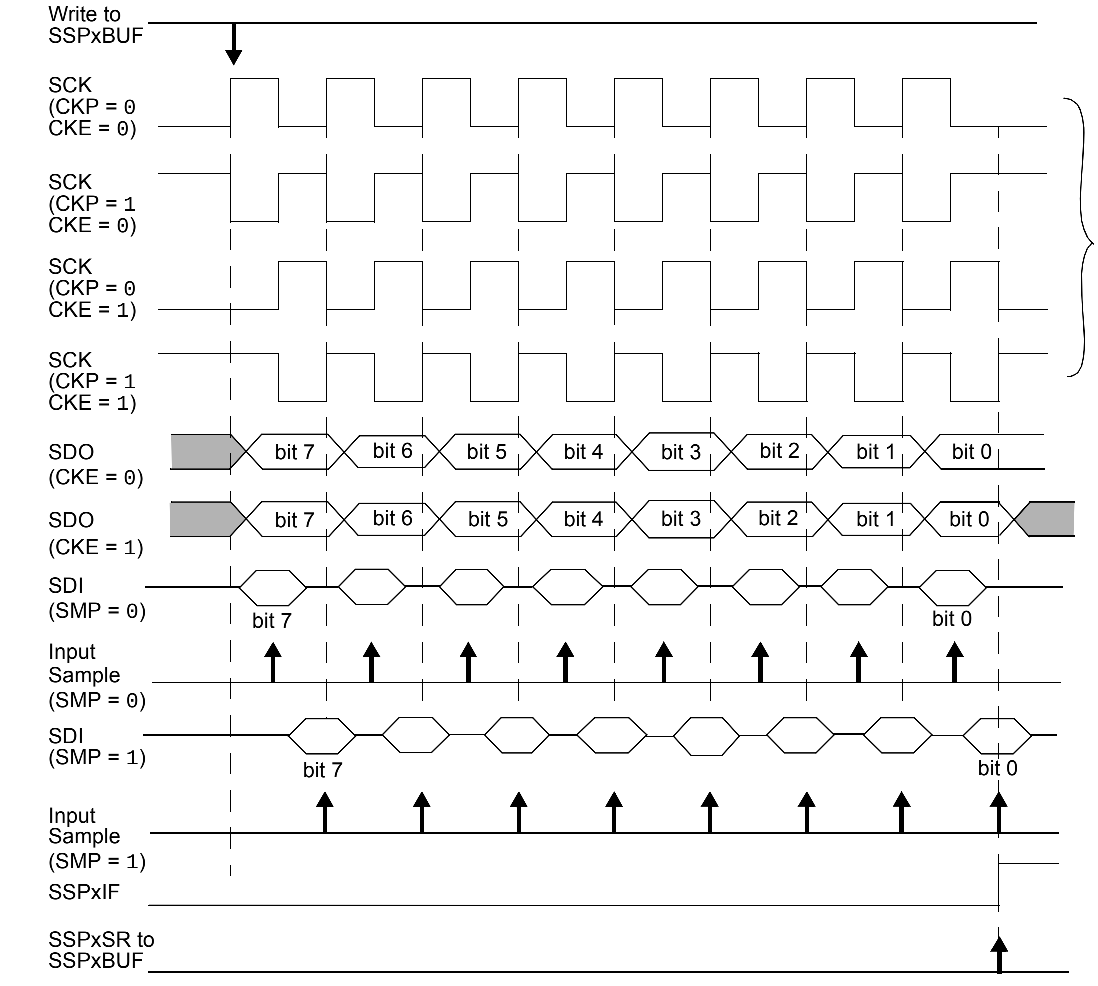
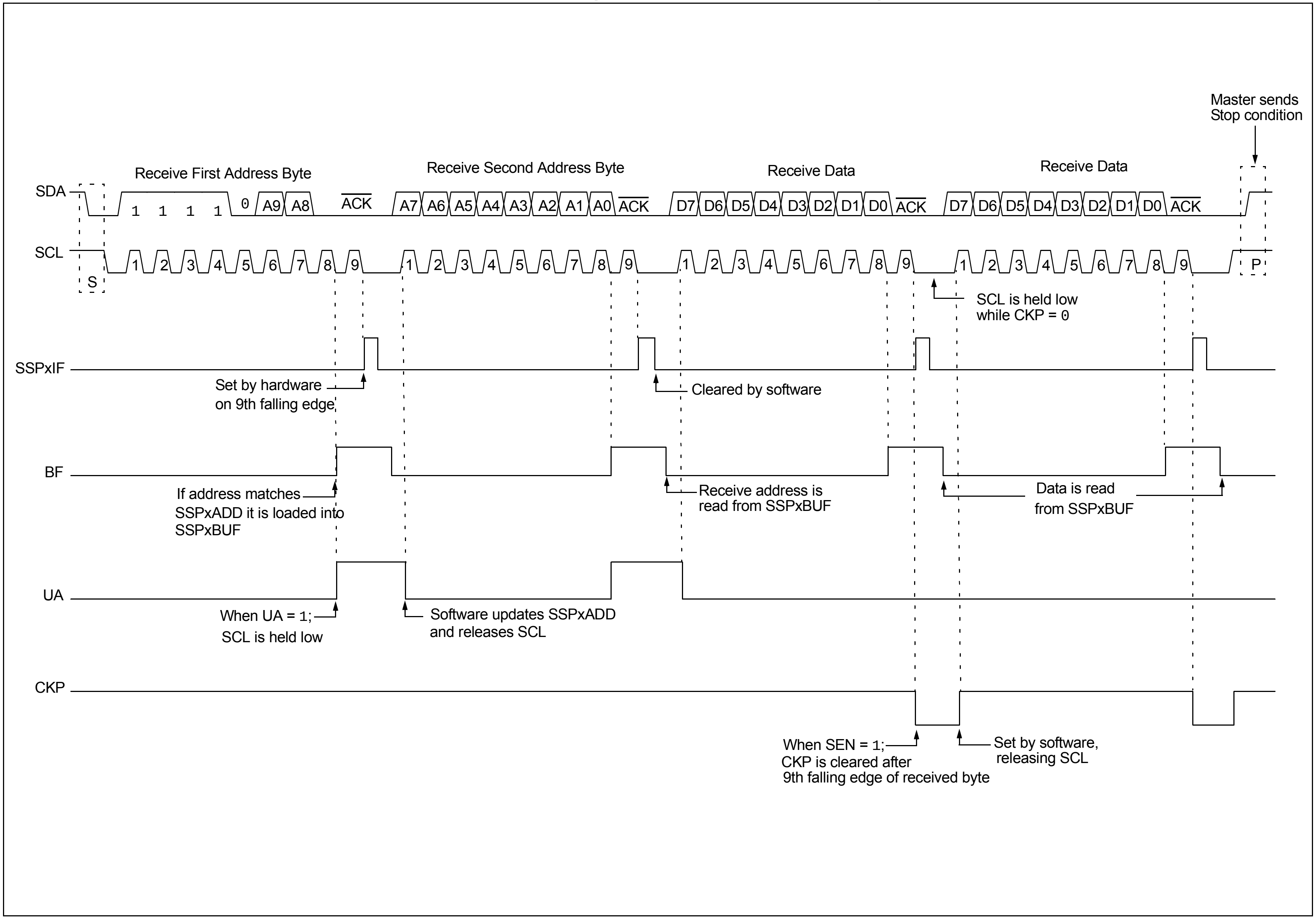


FIGURE 32-20: I^{2}C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)



32.6.5 I^{2}C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition (Figure 32-27) occurs when the RSEN bit of the SSPxCON2 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 (TBRG). 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 TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPxCON2 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 of the SSPxSTAT register will be set. The SSPxIF 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 32-27: REPEATED START CONDITION WAVEFORM

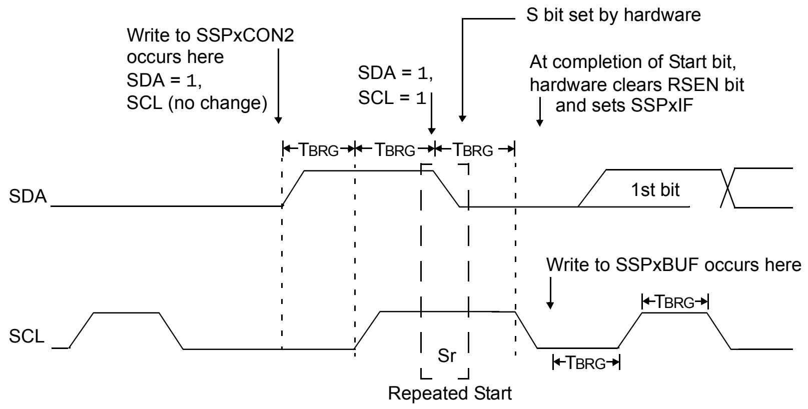
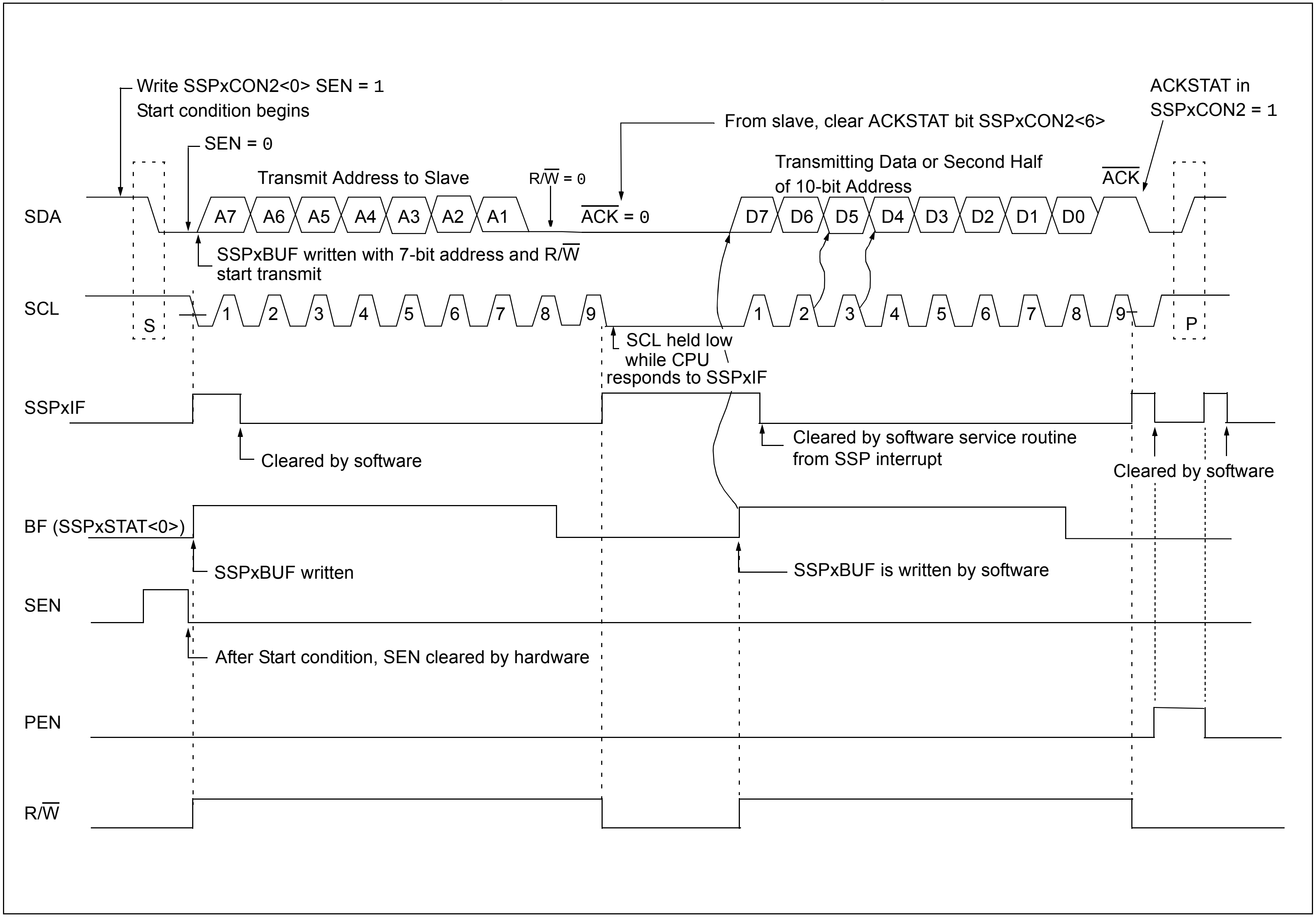

S bit set by hardware
Write to SSPxCON2 occurs here
SDA = 1,
SCL (no change)
SDA = 1,
SCL = 1
At completion of Start bit, hardware clears RSEN bit and sets SSPxIF
TBRG
TBRG
TBRG
SDA
1st bit
Write to SSPxBUF occurs here
TBRG
SCL
Sr
TBRG
Repeated Start

FIGURE 32-28: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



MOVIW Move INDFn to W

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

MOVLB Move literal to BSR

Syntax: [*label*] MOVLB k

Operands: 0 ≤ k ≤

Operation: k → BSR

Status Affected: None

Description: The 6-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP Move literal to PCLATH

Syntax: [*label*] MOVLP k

Operands: 0 ≤ k ≤ 127

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW Move literal to W

Syntax: [*label*] MOVLW k

Operands: 0 ≤ k ≤ 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 ≤ f ≤ 127

Operation: (W) → (f)

Status Affected: None

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example: `MOVWF LATA`

Before Instruction
LATA = 0xFF
W = 0x4F
After Instruction
LATA = 0x4F
W = 0x4F

37.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage:	VDDMIN ≤ VDD ≤ VDDMAX
Operating Temperature:	TA_MIN ≤ TA ≤ TA_MAX

VDD — Operating Supply Voltage[(1)]

PIC16LF15354/55

VDDMIN (Fosc ≤ 16 MHz)	+1.8V
VDDMIN (Fosc ≤ 32 MHz)	+2.5V
VDDMAX	+3.6V

PIC16F15354/55

VDDMIN (Fosc ≤ 16 MHz)	+2.3V
VDDMIN (Fosc ≤ 32 MHz)	+2.5V
VDDMAX	+5.5V

TA — Operating Ambient Temperature Range

Industrial Temperature

TA_MIN	-40°C
TA_MAX	+85°C

Extended Temperature

TA_MIN	-40°C
TA_MAX	+125°C

Note 1: See Parameter Supply Voltage, DS Characteristics: Supply Voltage.

TABLE 37-6: THERMAL CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)					
Param. No.	**Sym.**	**Characteristic**	**Typ.**	**Units**	**Conditions**
TH01	θ_{JA}	Thermal Resistance Junction to Ambient	60	°C/W	28-pin SPDIP package
			80	°C/W	28-pin SOIC package
			90	°C/W	28-pin SSOP package
			48	°C/W	28-pin UQFN 4x4mm package
TH02	θ_{JC}	Thermal Resistance Junction to Case	31.4	°C/W	28-pin SPDIP package
			24	°C/W	28-pin SOIC package
			24	°C/W	28-pin SSOP package
			12	°C/W	28-pin UQFN 4x4mm package
TH03	T_{JMAX}	Maximum Junction Temperature	150	°C	
TH04	P_D	Power Dissipation	—	W	$P_D = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$(1)
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	P_{DER}	Derated Power	—	W	$P_{DER} = PD_{MAX} (T_J - T_A)/\theta_{JA}$(2)

Note 1: I_{DD} is current to run the chip alone without driving any load on the output pins.

2: T_A = Ambient Temperature, T_J = Junction Temperature

TABLE 37-18: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)
Operating Temperature -40°C ≤ TA ≤ +125°C

Param. No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
40*	TT0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period		Greater of: 20 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	Greater of: 30 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value
			Asynchronous	60	—	—	ns	
48	FT1	Secondary Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

* 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.