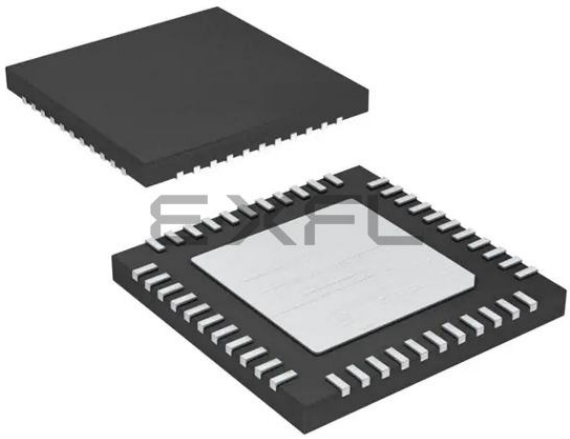


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

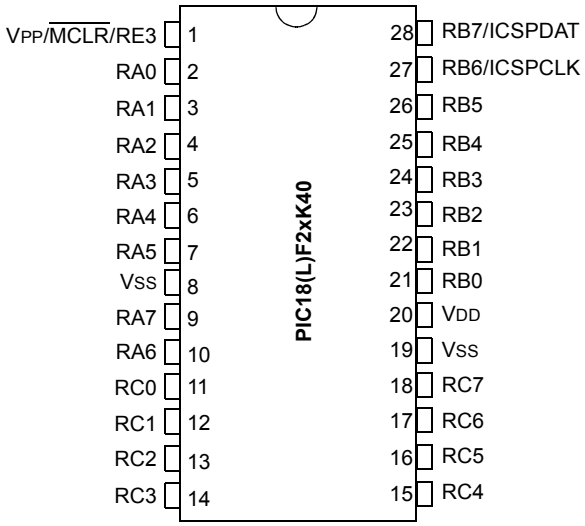
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf45k40-i-ml

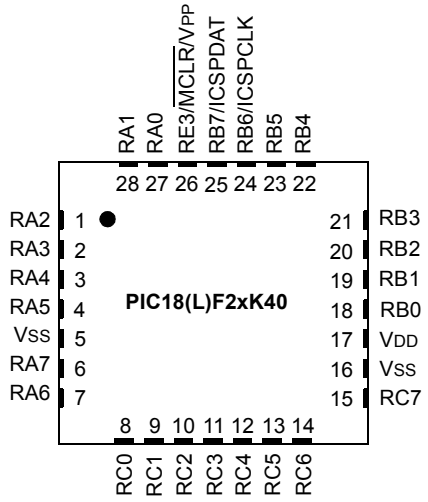
Pin Diagrams

28-pin SPDIP, SOIC, SSOP



Note: See Table 1 for location of all peripheral functions.

28-pin QFN (6x6x0.9mm), UQFN (4x4x0.5mm)



Note 1: See Table 1 for location of all peripheral functions.

2: It is recommended that the exposed bottom pad be connected to VSS, however it must not be the only VSS connection to the device.

REGISTER 3-4: Configuration Word 2H (30 0003h): Supervisor

R/W-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{XINST}}$	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	$\overline{\text{ZCD}}$	BORV<1:0>	
bit 7							bit 0

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

bit 7 **$\overline{\text{XINST}}$:** Extended Instruction Set Enable bit
1 = Extended Instruction Set and Indexed Addressing mode disabled (Legacy mode)
0 = Extended Instruction Set and Indexed Addressing mode enabled

bit 6 **Unimplemented:** Read as '1'

bit 5 **$\overline{\text{DEBUG}}$:** Debugger Enable bit
1 = Background debugger disabled
0 = Background debugger enabled

bit 4 **STVREN:** Stack Overflow/Underflow Reset Enable bit
1 = Stack Overflow or Underflow will cause a Reset
0 = Stack Overflow or Underflow will not cause a Reset

bit 3 **PPS1WAY:** PPSLOCKED bit One-Way Set Enable bit
1 = The PPSLOCKED bit can only be set once after an unlocking sequence is executed; once PPSLOCK is set, all future changes to PPS registers are prevented
0 = The PPSLOCKED bit can be set and cleared as needed (provided an unlocking sequence is executed)

bit 2 **$\overline{\text{ZCD}}$:** ZCD Disable bit
1 = ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of ZCDCON
0 = ZCD always enabled, ZCDMD bit is ignored

bit 1-0 **BORV<1:0>:** Brown-out Reset Voltage Selection bit(1)
PIC18F2x/4xK40 device:
11 = Brown-out Reset Voltage (V_{BOR}) set to 2.45V
10 = Brown-out Reset Voltage (V_{BOR}) set to 2.45V
01 = Brown-out Reset Voltage (V_{BOR}) set to 2.7V
00 = Brown-out Reset Voltage (V_{BOR}) set to 2.85V

PIC18LF2x/4xK40 device:
11 = Brown-out Reset Voltage (V_{BOR}) set to 1.90V
10 = Brown-out Reset Voltage (V_{BOR}) set to 2.45V
01 = Brown-out Reset Voltage (V_{BOR}) set to 2.7V
00 = Brown-out Reset Voltage (V_{BOR}) set to 2.85V

Note 1: The higher voltage setting is recommended for operation at or above 16 MHz.

8.13 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON0 registers are updated to indicate the cause of the Reset. Table 8-3 shows the Reset conditions of these registers.

TABLE 8-3: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register(2,3)	PCON0 Register
Power-on Reset	0	-110 0000	0011 110x
Brown-out Reset	0	-110 0000	0011 11u0
$\overline{\text{MCLR}}$ Reset during normal operation	0	-uuu uuuu	uuuu 0uuu
$\overline{\text{MCLR}}$ Reset during Sleep	0	-10u uuuu	uuuu 0uuu
WDT Time-out Reset	0	-0uu uuuu	uuu0 uuuu
WDT Wake-up from Sleep	PC + 2	-00u uuuu	uuuu uuuu
WWDT Window Violation Reset	0	-uuu uuuu	uu0u uuuu
Interrupt Wake-up from Sleep	PC + 2(1)	-10u 0uuu	uuuu uuuu
RESET Instruction Executed	0	-uuu uuuu	uuuu u0uu
Stack Overflow Reset (STVREN = 1)	0	-uuu uuuu	1uuu uuuu
Stack Underflow Reset (STVREN = 1)	0	-uuu uuuu	u1uu uuuu

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

Note 1: When the wake-up is due to an interrupt and Global Interrupt Enable bit (GIE) is set the return address is pushed on the stack and PC is loaded with the corresponding interrupt vector (depending on source, high or low priority) after execution of PC + 2.

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

3: Status bits Z, C, DC are reset by POR/BOR (Register 10-2).

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH WINDOWED WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PCON0	STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	76
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	118
WDTCON0	—	—	WDTPS<4:0>					SEN	85
WDTCON1	—	WDTCS<2:0>			—	WINDOW<2:0>			86
WDTPSL	PSCNT<7:0>								87
WDTPSH	PSCNT<15:8>								87
WDTTMR	WDTTMR<4:0>					STATE	PSCNT<17:16>		88

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by Windowed Watchdog Timer.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH WINDOWED 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

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Windowed Watchdog Timer.

10.3.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as either two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSb = 0). To maintain alignment with instruction boundaries, the PC increments in steps of two and the LSb will always read '0' (see **Section 10.1.1 "Program Counter"**).

Figure 10-3 shows an example of how instruction words are stored in the program memory.

The CALL and GOTO instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 10-3 shows how the instruction GOTO 0006h is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 35.0 "Instruction Set Summary"** provides further details of the instruction set.

10.3.4 TWO-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: CALL, MOVFF, GOTO and LFSR. In all cases, the second word of the instruction always has '1111' as its four Most Significant bits; the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the 4 MSbs of an instruction specifies a special form of NOP. If the instruction is executed in proper sequence – immediately after the first word – the data in the second word is accessed and used by the instruction sequence. If the first word is skipped for some reason and the second word is executed by itself, a NOP is executed instead. This is necessary for cases when the two-word instruction is preceded by a conditional instruction that changes the PC. Example 10-4 shows how this works.

Note:	See **Section 10.8 "PIC18 Instruction Execution and the Extended Instruction Set"** for information on two-word instructions in the extended instruction set.

FIGURE 10-3: INSTRUCTIONS IN PROGRAM MEMORY

			LSB = 1	LSB = 0	Word Address ↓
Program Memory Byte Locations →					000000h
					000002h
					000004h
					000006h
Instruction 1:	MOVLW	055h	0Fh	55h	000008h
Instruction 2:	GOTO	0006h	EFh	03h	00000Ah
			F0h	00h	00000Ch
Instruction 3:	MOVFF	123h, 456h	C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h

EXAMPLE 10-4: TWO-WORD INSTRUCTIONS

CASE 1:

Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; No, skip this word
1111 0100 0101 0110			; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3	; continue code

CASE 2:

Object Code	Source Code		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; Yes, execute this word
1111 0100 0101 0110			; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3	; continue code

TABLE 10-5: REGISTER FILE SUMMARY FOR PIC18(L)F26/45/46K40 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FFFh	TOSU	—	—	—	Top of Stack Upper byte (TOS<20:16>)					---xxxxx
FFEh	TOSH	Top of Stack High byte (TOS<15:8>)								xxxxxxxx
FFDh	TOSL	Top of Stack Low byte (TOS<7:0>)								xxxxxxxx
FFCh	STKPTR	—	—	—	STKPTR<4:0>					--000000
FFBh	PCLATU	—	—	—	Holding Register for PC<20:16>					---00000
FFAh	PCLATH	Holding Register for PC<15:8>								00000000
FF9h	PCL	PC Low byte (PC<7:0>)								00000000
FF8h	TBLPTRU	—	—	Program Memory Table Pointer (TBLPTR<21:16>)						--000000
FF7h	TBLPTRH	Program Memory Table Pointer (TBLPTR<15:8>)								00000000
FF6h	TBLPTRL	Program Memory Table Pointer (TBLPTR<7:0>)								00000000
FF5h	TABLAT	TABLAT								00000000
FF4h	PRODH	Product Register High byte								xxxxxxxx
FF3h	PRODL	Product Register Low byte								xxxxxxxx
FF2h	INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	000--111
FF1h	—	Unimplemented								—
FF0h	—	Unimplemented								—
FEFh	INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								--------
FEEh	POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								--------
FEDh	POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								--------
FECh	PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								--------
FEBh	PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W								--------
FEAh	FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0 High				----xxxx
FE9h	FSR0L	Indirect Data Memory Address Pointer 0 Low								xxxxxxxx
FE8h	WREG	Working Register								xxxxxxxx
FE7h	INDF1	Uses contents of FSR0 to address data memory – value of FSR1 not changed (not a physical register)								--------
FE6h	POSTINC1	Uses contents of FSR0 to address data memory – value of FSR1 post-incremented (not a physical register)								--------
FE5h	POSTDEC1	Uses contents of FSR0 to address data memory – value of FSR1 post-decremented (not a physical register)								--------
FE4h	PREINC1	Uses contents of FSR0 to address data memory – value of FSR1 pre-incremented (not a physical register)								--------
FE3h	PLUSW1	Uses contents of FSR0 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR0 offset by W								--------

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Not available on LF devices.

2: Not available on PIC18(L)F26K40 (28-pin variants).

3: Not available on PIC18(L)F45K40 devices.

EXAMPLE 11-4: WRITING TO PROGRAM FLASH MEMORY (CONTINUED)

```
WRITE_BYTE_TO_HREGS
             MOVF     POSTINC0, W              ; get low byte of buffer data
             MOVWF    TABLAT                   ; present data to table latch
             TBLWT+*                           ; write data, perform a short write
                                               ; to internal TBLWT holding register.
             DECFSZ   COUNTER                  ; loop until holding registers are full
             BRA      WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
             BCF      NVMCON1, NVMREG0         ; point to Program Flash Memory
             BSF      NVMCON1, NVMREG1         ; point to Program Flash Memory
             BSF      NVMCON1, WREN            ; enable write to memory
             BCF      NVMCON1, FREE            ; enable write to memory
             BCF      INTCON, GIE              ; disable interrupts
             MOVLW    55h
Required     MOVWF    NVMCON2                  ; write 55h
Sequence     MOVLW    0AAh
             MOVWF    NVMCON2                  ; write 0AAh
             BSF      NVMCON1, WR              ; start program (CPU stall)
             DCFSZ    COUNTER2                 ; repeat for remaining write blocks
             BRA      WRITE_BYTE_TO_HREGS
             BSF      INTCON, GIE              ; re-enable interrupts
             BCF      NVMCON1, WREN            ; disable write to memory
```

13.11.7 IN-CIRCUIT DEBUG (ICD) INTERACTION

The scanner freezes when an ICD halt occurs, and remains frozen until user-mode operation resumes. The debugger may inspect the SCANCON0 and SCANLADR registers to determine the state of the scan.

The ICD interaction with each operating mode is summarized in Table 13-4.

TABLE 13-4: ICD AND SCANNER INTERACTIONS

ICD Halt	Scanner Operating Mode		
	Peek	**Concurrent Triggered**	**Burst**
External Halt	If scanner would peek an instruction that is not executed (because of ICD entry), the peek will occur after ICD exit, when the instruction executes.	If external halt is asserted during a scan cycle, the instruction (delayed by scan) may or may not execute before ICD entry, depending on external halt timing.	If external halt is asserted during the BSF(SCANCON.GO), ICD entry occurs, and the burst is delayed until ICD exit. Otherwise, the current NVM-access cycle will complete, and then the scanner will be interrupted for ICD entry.
		If external halt is asserted during the cycle immediately prior to the scan cycle, both scan and instruction execution happen after the ICD exits.	If external halt is asserted during the burst, the burst is suspended and will resume with ICD exit.
PC Breakpoint		Scan cycle occurs before ICD entry and instruction execution happens after the ICD exits.	If PCPB (or single step) is on BSF(SCANCON.GO), the ICD is entered before execution; execution of the burst will occur at ICD exit, and the burst will run to completion. Note that the burst can be interrupted by an external halt.
Data Breakpoint		The instruction with the dataBP executes and ICD entry occurs immediately after. If scan is requested during that cycle, the scan cycle is postponed until the ICD exits.	
Single Step		If a scan cycle is ready after the debug instruction is executed, the scan will read PFM and then the ICD is re-entered.	
SWBP and ICDINST		If scan would stall a SWBP, the scan cycle occurs and the ICD is entered.	If SWBP replaces BSF(SCANCON.GO), the ICD will be entered; instruction execution will occur at ICD exit (from ICDINSTR register), and the burst will run to completion.

13.11.8 PERIPHERAL MODULE DISABLE

Both the CRC and scanner module can be disabled individually by setting the CRCMD and SCANMD bits of the PMD0 register (Register 7-1). The SCANMD can be used to enable or disable to the scanner module only if the SCANE bit of Configuration Word 4 is set. If the SCANE bit is cleared, then the scanner module is not available for use and the SCANMD bit is ignored.

REGISTER 15-8: INLVLx: 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
INLVLx7	INLVLx6	INLVLx5	INLVLx4	INLVLx3	INLVLx2	INLVLx1	INLVLx0
bit 7							bit 0

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

bit 7-0 **INLVLx<7:0>**: Input Level Select on Pins Rx<7:0>, respectively
1 = ST input used for port reads and interrupt-on-change
0 = TTL input used for port reads and interrupt-on-change

TABLE 15-9: INPUT LEVEL PORT REGISTERS

Name	Device		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	28 Pins	40/44 Pins								
INLVLA	X	X	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
INLVLB	X	X	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2(1)	INLVLB1(1)	INLVLB0
INLVLC	X	X	INLVLC7	INLVLC6	INLVLC5	INLVLC4(1)	INLVLC3(1)	INLVLC2	INLVLC1	INLVLC0
INLVLD	X		—	—	—	—	—	—	—	—
		X	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1(1)	INLVLD0(1)
INLVLE	X		—	—	—	—	INLVLE3	—	—	—
		X	—	—	—	—	INLVLE3	INLVLE2	INLVLE1	INLVLE0

Note 1: Pins read the I^2C ST inputs when MSSP inputs select these pins, and I^2C mode is enabled.

REGISTER 17-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits(1)

RxyPPS<4:0>	Pin Rxy Output Source	Device Configuration							
		PIC18(L)F26K40			PIC18(L)F45/46K40				
5’b1 0111	ADGRDB	A	—	C	A	—	C	—	—
5’b1 0110	ADGRDA	A	—	C	A	—	C	—	—
5’b1 0101	DSM	A	—	C	A	—	—	D	—
5’b1 0100	CLKR	—	B	C	—	B	C	—	—
5’b1 0011	TMR0	—	B	C	—	B	C	—	—
5’b1 0010	MSSP2 (SDO/SDA)	—	B	C	—	B	—	D	—
5’b1 0001	MSSP2 (SCK/SCL)	—	B	C	—	B	—	D	—
5’b1 0000	MSSP1 (SDO/SDA)	—	B	C	—	B	C	—	—
5’b0 1111	MSSP1 (SCK/SCL)	—	B	C	—	B	C	—	—
5’b0 1110	CMP2	A	—	C	A	—	—	—	E
5’b0 1101	CMP1	A	—	C	A	—	—	D	—
5’b0 1100	EUSART2 (RX)	—	B	C	—	B	—	D	—
5’b0 1011	EUSART2 (TX)	—	B	C	—	B	—	D	—
5’b0 1010	EUSART1 (RX)	—	B	C	—	B	C	—	—
5’b0 1001	EUSART1 (TX)	—	B	C	—	B	C	—	—
5’b0 1000	PWM4	A	—	C	A	—	C	—	—
5’b0 0111	PWM3	A	—	C	A	—	—	D	—
5’b0 0110	CCP2	—	B	C	—	B	C	—	—
5’b0 0101	CCP1	—	B	C	—	B	C	—	—
5’b0 0100	CWG1D	—	B	C	—	B	—	D	—
5’b0 0011	CWG1C	—	B	C	—	B	—	D	—
5’b0 0010	CWG1B	—	B	C	—	B	—	D	—
5’b0 0001	CWG1A	—	B	C	—	B	C	—	—
5’b0 0000	LATxy	A	B	C	A	B	C	D	E

Note 1: PORTD is present only on the PIC18(L)F45/46K40 devices.

19.8.2 TIMER1/3/5 GATE SOURCE SELECTION

The gate source for Timer1/3/5 can be selected using the GSS<3:0> bits of the TMRxGATE register (Register 19-4). The polarity selection for the gate source is controlled by the TxGPOL bit of the TxGCON register (Register 19-2).

Any of the above mentioned signals can be used to trigger the gate. The output of the CMPx can be synchronized to the Timer1/3/5 clock or left asynchronous. For more information see **Section 32.5.1 "Comparator Output Synchronization"**.

19.8.3 TIMER1/3/5 GATE TOGGLE MODE

When Timer1/3/5 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1/3/5 gate signal, as opposed to the duration of a single level pulse.

The Timer1/3/5 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 19-5 for timing details.

Timer1/3/5 Gate Toggle mode is enabled by setting the GTM bit of the TxGCON register. When the GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note:	Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

19.8.4 TIMER1/3/5 GATE SINGLE-PULSE MODE

When Timer1/3/5 Gate Single-Pulse mode is enabled, it is possible to capture a single-pulse gate event. Timer1/3/5 Gate Single-Pulse mode is first enabled by setting the GSPM bit in the TxGCON register. Next, the GGO/$\overline{\text{DONE}}$ bit in the TxGCON register must be set. The Timer1/3/5 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the GGO/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1/3/5 until the GGO/$\overline{\text{DONE}}$ bit is once again set in software.

Clearing the TxGSPM bit of the TxGCON register will also clear the GGO/$\overline{\text{DONE}}$ bit. See Figure 19-6 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1/3/5 gate source to be measured. See Figure 19-7 for timing details.

19.8.5 TIMER1/3/5 GATE VALUE STATUS

When Timer1/3/5 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the GVAL bit in the TxGCON register. The GVAL bit is valid even when the Timer1/3/5 gate is not enabled (GE bit is cleared).

19.8.6 TIMER1/3/5 GATE EVENT INTERRUPT

When Timer1/3/5 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of GVAL occurs, the TMRxGIF flag bit in the PIR5 register will be set. If the TMRxGIE bit in the PIE5 register is set, then an interrupt will be recognized.

The TMRxGIF flag bit operates even when the Timer1/3/5 gate is not enabled (GE bit is cleared).

For more information on selecting high or low priority status for the Timer1/3/5 Gate Event Interrupt see **Section 14.0 "Interrupts"**.

20.7 Register Definitions: Timer2/4/6 Control

Long bit name prefixes for the Timer2/4/6 peripherals are shown in Table 20-2. Refer to **Section 1.4.2.2 "Long Bit Names"** for more information.

TABLE 20-2:

Peripheral	Bit Name Prefix
Timer2	T2
Timer4	T4
Timer6	T6

REGISTER 25-2: MDCON1: MODULATION CONTROL REGISTER 1

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC
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 7-6 **Unimplemented:** Read as ‘0’

bit 5 **CHPOL:** Modulator High Carrier Polarity Select bit
- 1 = Selected high carrier signal is inverted
- 0 = Selected high carrier signal is not inverted

bit 4 **CHSYNC:** Modulator High Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the high time carrier signal before allowing a switch to the low time carrier
- 0 = Modulator output is not synchronized to the high time carrier signal(1)

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **CLPOL:** Modulator Low Carrier Polarity Select bit
- 1 = Selected low carrier signal is inverted
- 0 = Selected low carrier signal is not inverted

bit 0 **CLSYNC:** Modulator Low Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier
- 0 = Modulator output is not synchronized to the low time carrier signal(1)

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

27.4.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDxCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RXx pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPxBRGH:SPxBRGL register pair. After the ABDOVF bit has been set, the counter continues to count until the fifth rising edge is detected on the RXx pin. Upon detecting the fifth RX edge, the hardware will set the RCxIF interrupt flag and clear the ABDEN bit of the BAUDxCON register. The RCxIF flag can be subsequently cleared by reading the RCxREG register. The ABDOVF flag of the BAUDxCON register can be cleared by software directly.

To terminate the auto-baud process before the RCxIF flag is set, clear the ABDEN bit then clear the ABDOVF bit of the BAUDxCON register. The ABDOVF bit will remain set if the ABDEN bit is not cleared first.

27.4.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDxCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCxIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 27-7), and asynchronously if the device is in Sleep mode (Figure 27-8). The interrupt condition is cleared by reading the RCxREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

27.4.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all ‘0’s. This must be ten or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCxIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCxREG register and discarding its contents.

To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

31.1.5 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 FRC 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 ADIE bit of the PIE1 register and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

31.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bits of the ADCON0 register controls the output format.

Figure 31-3 shows the two output formats.

FIGURE 31-3: 10-BIT ADC CONVERSION RESULT FORMAT

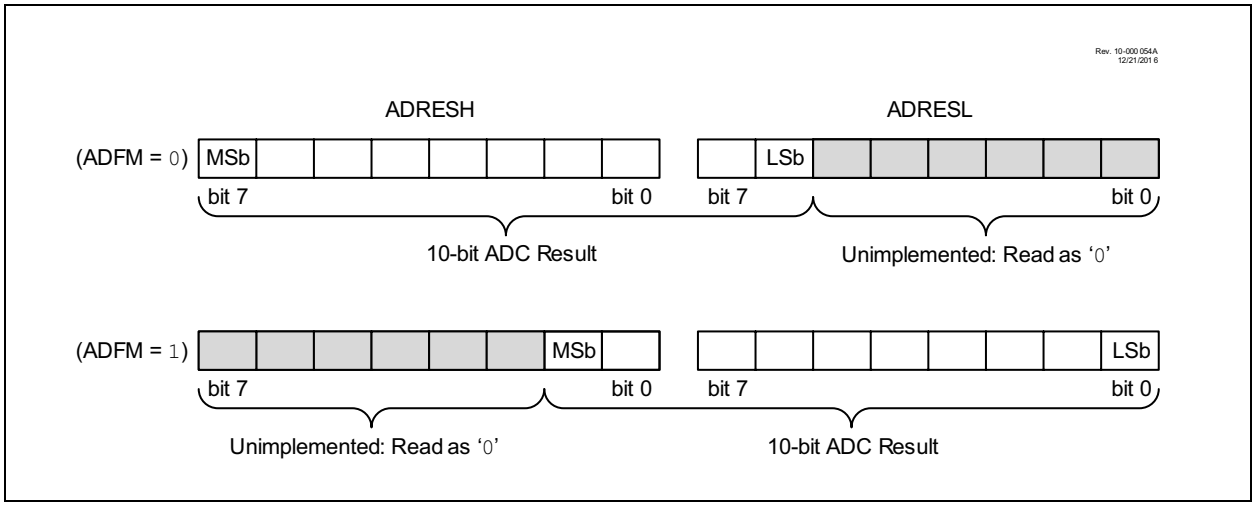


FIGURE 31-9: DIFFERENTIAL CVD WITH GUARD RING OUTPUT WAVEFORM

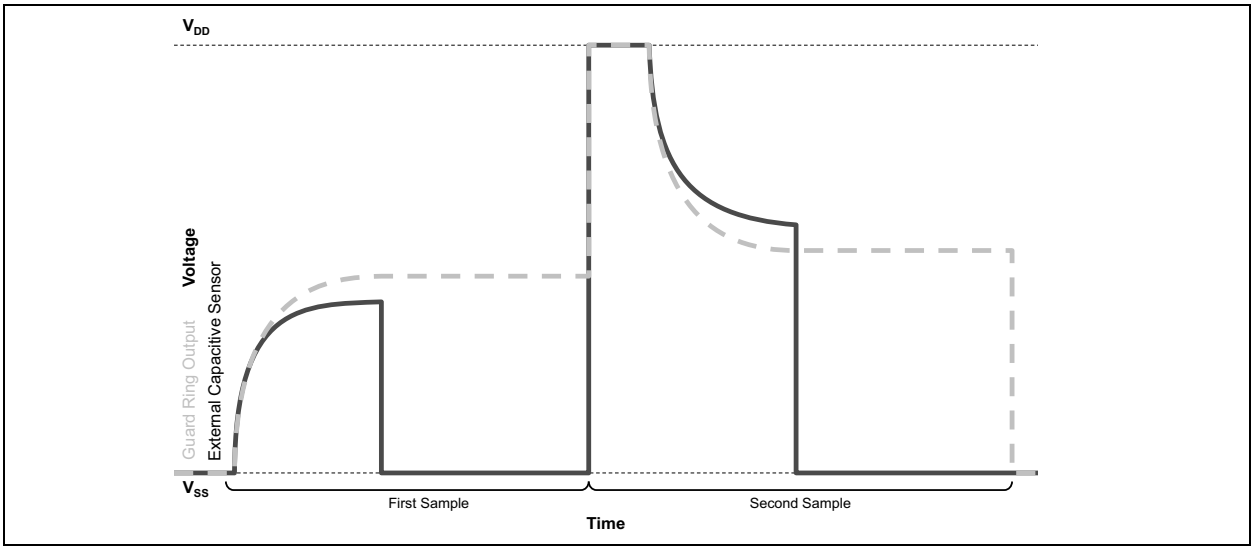
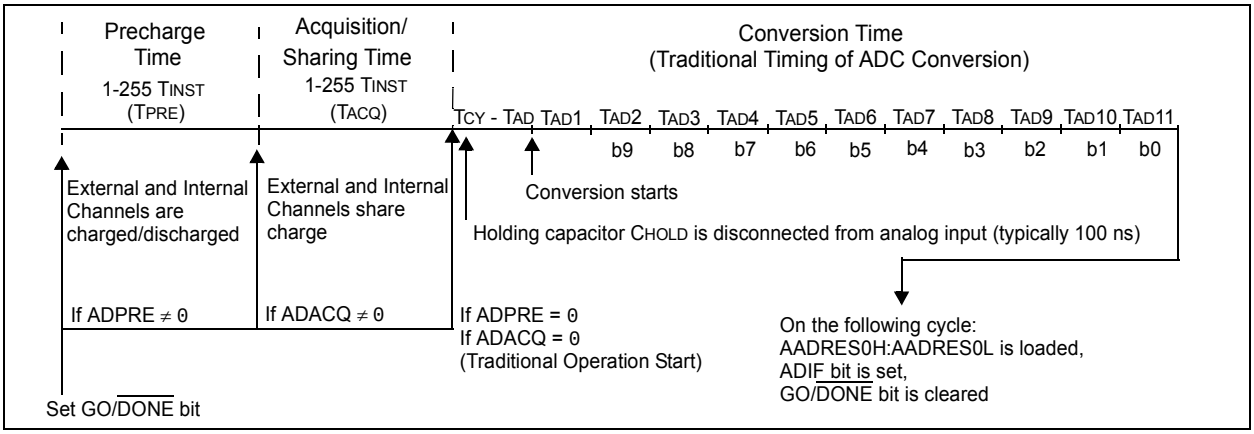


FIGURE 31-10: HARDWARE CVD SEQUENCE TIMING DIAGRAM



31.4.5 ADDITIONAL SAMPLE AND HOLD CAPACITANCE

Additional capacitance can be added in parallel with the internal sample and hold capacitor (CHOLD) by using the ADCAP register. This register selects a digitally programmable capacitance which is added to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion. See Figure 31-11.

REGISTER 31-32: ADACT: ADC AUTO CONVERSION TRIGGER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ADACT<4:0>				
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 7-5 **Unimplemented**: Read as '0'

bit 4-0 **ADACT<4:0>**: Auto-Conversion Trigger Select Bits
11111 = Software write to ADPCH
11110 = Reserved, do not use
11101 = Software read of ADRESH
11100 = Software read of ADERRH
11011 = Reserved, do not use
•
•
•
10000 = Reserved, do not use
01111 = Interrupt-on-change Interrupt Flag
01110 = C2_out
01101 = C1_out
01100 = PWM4_out
01011 = PWM3_out
01010 = CCP2_trigger
01001 = CCP1_trigger
01000 = TMR6_postscaled
00111 = TMR5_overflow
00110 = TMR4_postscaled
00101 = TMR3_overflow
00100 = TMR2_postscaled
00011 = TMR1_overflow
00010 = TMR0_overflow
00001 = Pin selected by ADACTPPS
00000 = External Trigger Disabled

TABLE 35-2: INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
				MSb			LSb		
BIT-ORIENTED OPERATIONS									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4
BTG	f, b, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2
CONTROL OPERATIONS									
BC	n	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	None	
BN	n	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	None	
BNC	n	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	None	
BNN	n	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	None	
BNOV	n	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	None	
BNZ	n	Branch if Not Zero	1 (2)	1110	0001	nnnn	nnnn	None	
BOV	n	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	None	
BRA	n	Branch Unconditionally	2	1101	0nnn	nnnn	nnnn	None	
BZ	n	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	None	
CALL	k, s	Call subroutine 1st word	2	1110	110s	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	$\overline{TO}$, $\overline{PD}$	
DAW	—	Decimal Adjust WREG	1	0000	0000	0000	0111	C	
GOTO	k	Go to address 1st word	2	1110	1111	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
NOP	—	No Operation	1	0000	0000	0000	0000	None	
NOP	—	No Operation	1	1111	xxxx	xxxx	xxxx	None	4
POP	—	Pop top of return stack (TOS)	1	0000	0000	0000	0110	None	
PUSH	—	Push top of return stack (TOS)	1	0000	0000	0000	0101	None	
RCALL	n	Relative Call	2	1101	1nnn	nnnn	nnnn	None	
RESET		Software device Reset	1	0000	0000	1111	1111	All	
RETFIE	s	Return from interrupt enable	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
RETURN	s	Return from Subroutine	2	0000	0000	0001	001s	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0000	0011	$\overline{TO}$, $\overline{PD}$	

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), 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.

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

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

TABLE 37-5: MEMORY PROGRAMMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
Data EEPROM Memory Specifications							
MEM20	E_D	DataEE Byte Endurance	100k	—	—	E/W	-40°C ≤ TA ≤ +85°C
MEM21	T_{D_RET}	Characteristic Retention	—	40	—	Year	Provided no other specifications are violated
MEM22	N_{D_REF}	Total Erase/Write Cycles before Refresh	1M 500k	10M —	— —	E/W	-40°C ≤ TA ≤ +60°C -40°C ≤ TA ≤ +85°C
MEM23	V_{D_RW}	VDD for Read or Erase/Write operation	VDDMIN	—	VDDMAX	V	
MEM24	T_{D_BEW}	Byte Erase and Write Cycle Time	—	4.0	5.0	ms	
Program Flash Memory Specifications							
MEM30	E_P	Flash Memory Cell Endurance	10k	—	—	E/W	-40°C ≤ TA ≤ +85°C (**Note 1**)
MEM32	T_{P_RET}	Characteristic Retention	—	40	—	Year	Provided no other specifications are violated
MEM33	V_{P_RD}	VDD for Read operation	VDDMIN	—	VDDMAX	V	
MEM34	V_{P_REW}	VDD for Row Erase or Write operation	VDDMIN	—	VDDMAX	V	
MEM35	T_{P_REW}	Self-Timed Row Erase or Self-Timed Write	—	2.0	2.5	ms	

† 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: Flash Memory Cell Endurance for the Flash memory is defined as: One Row Erase operation and one Self-Timed Write.

APPENDIX A: REVISION HISTORY

Revision A (9/2015)

Initial Release.

Revision B (5/2016)

Updated Example 11-6; Figures 37-1, 37-2, 37-5; Register 31-5; Sections 1.1.2, 21.4.1, 21.4.2, 22.1.3, 22.1.9, 22.1.10, 37.2; Tables 37-1, 37-2, 37-3, 37-7, 37-8, 37-9, 37-11, 37-13.

Removed Register 5-3.

Added long name bit/short name bits section 1.4 and updated bit names accordingly.

Revision C (9/2016)

Updated Peripheral Module, Memory and Core features descriptions on cover page. Updated the PIC18(L)F2x/4xK40 Family Types Table. Updated Examples 11-1, 11-3, 11-5 and 11-6; Figures 14-1 and 31-2; Registers 4-2, 4-5, 13-18 and 31-6; Sections 1.2, 4.4.1, 4.5, 4.5.4, 17.3, 17.5, 17.7, 18.1, 18.1.1, 18.1.1.1, 18.1.2, 18.1.6, 18.3, 18.4, 18.7, 19.0, 19.8.1, 20.0, 21.3, and 25.3; Tables 4-2, 37-2, 37-3, 37-5, 37-13 and 37-14.

Revision D (4/2017)

Updated Cover page. Updated Example 13-1; Figures 6-1 and 11-11; Registers 3-6, 3-13, 19-1, and 26-9; Sections 1.1.2, 4.3, 13.8, 23.5, 26.5.1, 26.10, 31.1.2, and 31.1.6; Tables 4-1, 10-5, 37-11 and 37-15.

New Timer 2 chapter.

Removed Section 4.4.2 and 31.2.3.

Added Section 23.5.1