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, POR, PWM, WDT
Number of I/O	18
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	224 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf15344-e-so

TABLE 4-7: PIC16(L)F15324/44 MEMORY MAP, BANKS 56-63

BANK 56		BANK 57		BANK 58		BANK 59		BANK 60		BANK 61		BANK 62		BANK 63	
1C00h	Core Register (Table 4-3)	1C80h	Core Register (Table 4-3)	1D00h	Core Register (Table 4-3)	1D80h	Core Register (Table 4-3)	1E00h	Core Register (Table 4-3)	1E80h	Core Register (Table 4-3)	1F00h	Core Register (Table 4-3)	1F80h	Core Register (Table 4-3)
1C0Bh		1C8Bh		1D0Bh		1D8Bh		1E0Bh		1E8Bh		1F0Bh		1F8Bh	
1C0Ch	—	1C8Ch	—	1D0Ch	—	1D8Ch	—	1E0Ch		1E8Ch		1F0Ch		1F8Ch	
1C0Dh	—	1C8Dh	—	1D0Dh	—	D8Dh1	—	1E0Dh		1E8Dh		1F0Dh		1F8Dh	
1C0Eh	—	1C8Eh	—	1D0Eh	—	1D8Eh	—	1E0Eh		1E8Eh		1F0Eh		1F8Eh	
1C0Fh	—	1C8Fh	—	1D0Fh	—	1D8Fh	—	1E0Fh		1E8Fh		1F0Fh		1F8Fh	
1C10h	—	1C90h	—	1D10h	—	1D90h	—	1E10h		1E90h		1F10h		1F90h	
1C11h	—	1C91h	—	1D11h	—	1D91h	—	1E11h		1E91h		1F11h		1F91h	
1C12h	—	1C92h	—	1D12h	—	1D92h	—	1E12h		1E92h		1F12h		1F92h	
1C13h	—	1C93h	—	1D13h	—	1D93h	—	1E13h		1E93h		1F13h		1F93h	
1C14h	—	1C94h	—	1D14h	—	1D94h	—	1E14h		1E94h		1F14h		1F94h	
1C15h	—	1C95h	—	1D15h	—	1D95h	—	1E15h		1E95h		1F15h		1F95h	
1C16h	—	1C96h	—	1D16h	—	1D96h	—	1E16h		1E96h		1F16h		1F96h	
1C17h	—	1C97h	—	1D17h	—	1D97h	—	1E17h	CLC Controls	1E97h	nnnPPS Controls	1F17h	RxyPPS Controls	1F97h	(See Table 4-8 for register mapping details)
1C18h	—	1C98h	—	1D18h	—	1D98h	—	1E18h	(See Table 4-8 for register mapping details)	1E98h	(See Table 4-8 for register mapping details)	1F18h	(See Table 4-8 for register mapping details)	1F98h	
1C19h	—	1C99h	—	1D19h	—	1D99h	—	1E19h		1E99h		1F19h		1F99h	
1C1Ah	—	1C9Ah	—	1D1Ah	—	1D9Ah	—	1E1Ah		1E9Ah		1F1Ah		1F9Ah	
1C1Bh	—	1C9Bh	—	1D1Bh	—	1D9Bh	—	1E1Bh		1E9Bh		1F1Bh		1F9Bh	
1C1Ch	—	1C9Ch	—	1D1Ch	—	1D9Ch	—	1E1Ch		1E9Ch		1F1Ch		1F9Ch	
1C1Dh	—	1C9Dh	—	1D1Dh	—	1D9Dh	—	1E1Dh		1E9Dh		1F1Dh		1F9Dh	
1C1Eh	—	1C9Eh	—	1D1Eh	—	1D9Eh	—	1E1Eh		1E9Eh		1F1Eh		1F9Eh	
1C1Fh	—	1C9Fh	—	1D1Fh	—	1D9Fh	—	1E1Fh		1E9Fh		1F1Fh		1F9Fh	
1C20h	Unimplemented Read as '0'	1CA0h	Unimplemented Read as '0'	1D20h	Unimplemented Read as '0'	1DA0h	Unimplemented Read as '0'	1E20h		1EA0h		1F20h		1FA0h	
1C6Fh		1CEFh		1D6Fh		1DEFh		1E6Fh		1EEFh		1F6Fh		1FEFh	
1C70h	Common RAM Accesses 70h-7Fh	1CF0h	Common RAM Accesses 70h-7Fh	1D70h	Common RAM Accesses 70h-7Fh	1DF0h	Common RAM Accesses 70h-7Fh	1E70h	Common RAM Accesses 70h-7Fh	1EF0h	Common RAM Accesses 70h-7Fh	1F70h	Common RAM Accesses 70h-7Fh	1FF0h	Common RAM Accesses 70h-7Fh
1C7Fh		1CFFh		1D7Fh		1DFFh		1E7Fh		1EFFh		1F7Fh		1FFFh	

Note 1: Unimplemented locations read as '0'.

2: The banks 24-55 have been omitted from the tables in the data sheet since the banks have unimplemented registers.

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 61											
CPU CORE REGISTERS; see Table 4-3 for specifics											
1E8Ch	—	Unimplemented								—	—
1E8Dh	—	Unimplemented								—	—
1E8Eh	—	Unimplemented								—	—
1E8Fh	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	---- ---0	---- ---0
1E90h	INTPPS	—	—	INTPPS<5:0>						--00 1000	--uu uuuu
1E91h	T0CKIPPS	—	—	T0CKIPPS<5:0>						--00 0100	--uu uuuu
1E92h	T1CKIPPS	—	—	T1CKIPPS<5:0>						--01 0000	--uu uuuu
1E93h	T1GPPS	—	—	T1GPPS<5:0>						--00 1101	--uu uuuu
1E94h — 1E9Bh	—	Unimplemented								—	—
1E9Ch	T2INPPS	—	—	T2INPPS<5:0>						--01 0011	--uu uuuu
1E9Dh — 1EA0h	—	Unimplemented								—	—
1EA1h	CCP1PPS	—	—	CCP1PPS<5:0>						--01 0010	--uu uuuu
1EA2h	CCP2PPS	—	—	CCP2PPS<5:0>						--01 0001	--uu uuuu
1EA3h — 1EB0h	—	Unimplemented								—	—
1EB1h	CWG1PPS	—	—	CWG1PPS<5:0>						--00 1000	--uu uuuu
1EB2h — 1EBAh	—	Unimplemented								—	—
1EBBh	CLCIN0PPS	—	—	CLCIN0PPS<5:0>						--00 0000	--uu uuuu
1EBCh	CLCIN1PPS	—	—	CLCIN1PPS<5:0>						--00 0001	--uu uuuu
1EBDh	CLCIN2PPS	—	—	CLCIN2PPS<5:0>						--00 1110	--uu uuuu
1EBEh	CLCIN3PPS	—	—	CLCIN3PPS<5:0>						--00 1111	--uu uuuu
1EBFh — 1EC2h	—	Unimplemented								—	—
1EC3h	ADACTPPS	—	—	CLCIN3PPS<5:0>						--001100	--uuuuuu
1EC4h	—	Unimplemented								—	—

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: MCLR
Bank 62 (Continued)											
1F38h	ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	--11 1111	--11 1111
1F39h	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--00 0000	--00 0000
1F3Ah	ODCONA	—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0	--00 0000	--00 0000
1F3Bh	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--11 1111	--11 1111
1F3Ch	INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	--11 1111	--11 1111
1F3Dh	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
1F3Eh	IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
1F3Fh	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
1F40h — 1F42h	—	Unimplemented								—	—
1F43h	ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	1111 ----	1111 ----
1F44h	WPUB(1)	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	0000 ----	0000 ----
1F45h	ODCONB(1)	ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—	0000 ----	0000 ----
1F46h	SLRCONB(1)	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	1111 ----	1111 ----
1F47h	INLVLB(1)	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	1111 ----	1111 ----
1F48h	IOCBP(1)	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
1F49h	IOCBN(1)	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
1F4Ah	IOCBF(1)	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
1F4Bh — 1F4Dh	—	Unimplemented								—	—
1F4Eh	ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	1111 1111	1111 1111
1F4Fh	WPUC	WPUC7(1)	WPUC6(1)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	0000 0000	0000 0000
1F50h	ODCONC	ODCC7(1)	ODCC6(1)	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	0000 0000	0000 0000
1F51h	SLRCONC	SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111
1F52h	INLVLC	INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
1F53h	IOCCP	IOCCP7(1)	IOCCP6(1)	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
1F54h	IOCCN	IOCCN7(1)	IOCCN6(1)	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
1F55h	IOCCF	IOCCF7(1)	IOCCF6(1)	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000
1F56h — 1F6Fh	—	Unimplemented								—	—

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.
Note 1: Present only in PIC16(L)F15344.

9.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a precision digitally-controlled internal clock source that produces a stable clock up to 32 MHz. The HFINTOSC can be enabled through one of the following methods:

- Programming the RSTOSC<2:0> bits in Configuration Word 1 to '110' (1 MHz) or '001' (32 MHz) to set the oscillator upon device Power-up or Reset.
- Write to the NOSC<2:0> bits of the OSCCON1 register during run-time.

The HFINTOSC frequency can be selected by setting the HFFRQ<2:0> bits of the OSCFRQ register. The MFINTOSC is an internal clock source within the HFINTOSC that provides two (500 kHz, 32 kHz) constant clock outputs. These constant clock outputs are available for selection to various peripherals, internally.

The NDIV<3:0> bits of the OSCCON1 register allow for division of the HFINTOSC output from a range between 1:1 and 1:512.

9.2.2.2 Internal Oscillator Frequency Adjustment

The internal oscillator is factory-calibrated. This internal oscillator can be adjusted in software by writing to the OSCTUNE register (Register 9-7).

The default value of the OSCTUNE register is 00h. The value is a 6-bit two's complement number. A value of 1Fh will provide an adjustment to the maximum frequency. A value of 20h will provide an adjustment to the minimum frequency.

When the OSCTUNE register is modified, the oscillator frequency will begin shifting to the new frequency. Code execution continues during this shift. There is no indication that the shift has occurred.

OSCTUNE does not affect the LFINTOSC frequency. Operation of features that depend on the LFINTOSC clock source frequency, such as the Power-up Timer (PWRT), Watchdog Timer (WDT), Fail-Safe Clock Monitor (FSCM) and peripherals, are *not* affected by the change in frequency.

9.2.2.3 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a factory-calibrated 31 kHz internal clock source. The LFINTOSC is the clock source for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM). The LFINTOSC can also be used as the system clock, or as a clock or input source to certain peripherals.

The LFINTOSC is selected as the clock source through one of the following methods:

- Programming the RSTOSC<2:0> bits of Configuration Word 1 to enable LFINTOSC.
- Write to the NOSC<2:0> bits of the OSCCON1 register.

9.2.2.4 Oscillator Status and Manual Enable

The 'ready' status of each oscillator is displayed in the OSCSTAT register (Register 9-4). The oscillators can also be manually enabled through the OSCEN register (Register 9-7). Manual enabling makes it possible to verify the operation of the EXTOSC oscillator. This can be achieved by enabling the selected oscillator, then watching the corresponding 'ready' state of the oscillator in the OSCSTAT register.

REGISTER 10-2: PIE0: PERIPHERAL INTERRUPT ENABLE REGISTER 0

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
—	—	TMR0IE	IOCIE	—	—	—	INTE
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 set

bit 7-6 **Unimplemented**: Read as '0'

bit 5 **TMR0IE:** Timer0 Overflow Interrupt Enable bit
- 1 = Enables the Timer0 interrupt
- 0 = Disables the Timer0 interrupt

bit 4 **IOCIE:** Interrupt-on-Change Interrupt Enable bit
- 1 = Enables the IOC change interrupt
- 0 = Disables the IOC change interrupt

bit 3-1 **Unimplemented**: Read as '0'

bit 0 **INTE:** INT External Interrupt Flag bit(1)
- 1 = Enables the INT external interrupt
- 0 = Disables the INT external interrupt

Note 1: The External Interrupt GPIO pin is selected by INTPPS (Register 15-1).

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by PIE1-PIE7. Interrupt sources controlled by the PIE0 register do not require PEIE to be set in order to allow interrupt vectoring (when GIE is set).

REGISTER 11-2: CPUDOZE: DOZE AND IDLE REGISTER

R/W-0/u	R/W/HC/HS-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
IDLEN	DOZEN(1,2)	ROI	DOE	—	DOZE<2:0>		
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 **IDLEN:** Idle Enable bit
1 = A SLEEP instruction inhibits the CPU clock, but not the peripheral clock(s)
0 = A SLEEP instruction places the device into full Sleep mode

bit 6 **DOZEN:** Doze Enable bit(1,2)
1 = The CPU executes instruction cycles according to DOZE setting
0 = The CPU executes all instruction cycles (fastest, highest power operation)

bit 5 **ROI:** Recover-on-Interrupt bit
1 = Entering the Interrupt Service Routine (ISR) makes DOZEN = 0 bit, bringing the CPU to full-speed operation.
0 = Interrupt entry does not change DOZEN

bit 4 **DOE:** Doze on Exit bit
1 = Executing RETFIE makes DOZEN = 1, bringing the CPU to reduced speed operation.
0 = RETFIE does not change DOZEN

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

bit 2-0 **DOZE<2:0>:** Ratio of CPU Instruction Cycles to Peripheral Instruction Cycles
111 =1:256
110 =1:128
101 =1:64
100 =1:32
011 =1:16
010 =1:8
001 =1:4
000 =1:2

Note 1: When ROI = 1 or DOE = 1, DOZEN is changed by hardware interrupt entry and/or exit.
2: Entering ICD overrides DOZEN, returning the CPU to full execution speed; this bit is not affected.

FIGURE 12-1: WATCHDOG TIMER BLOCK DIAGRAM

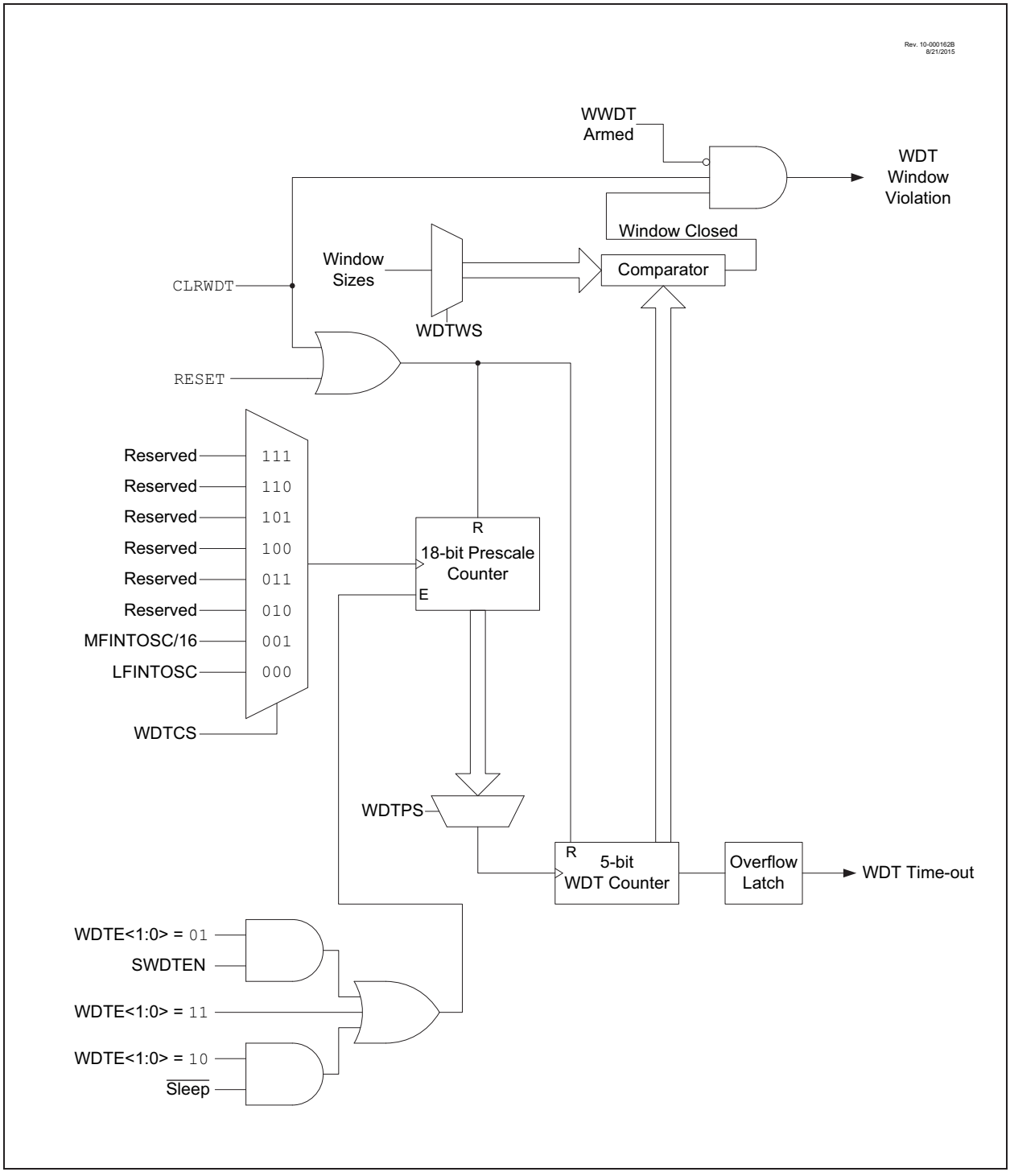


TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		222
ADCON0	CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	235
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		236
DAC1CON0	DAC1EN	—	DAC1OE1	DAC1OE2	DAC1PSS<1:0>		—	DAC1NSS	244

Legend: – = unimplemented locations read as '0'. Shaded cells are not used with the Fixed Voltage Reference.

20.4 Register Definitions: ADC Control

REGISTER 20-1: ADCON0: ADC CONTROL REGISTER 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	R/W-0/0
CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

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

bit 7-2 **CHS<5:0>:** Analog Channel Select bits
- 111111 = FVR Buffer 2 reference voltage(2)
- 111110 = FVR 1Buffer 1 reference voltage(2)
- 111101 = DAC1 output voltage(1)
- 111100 = Temperature sensor output(3)
- 111011 = AVSS (Analog Ground)
- 010111 = RC7(4)
- 010110 = RC6(4)
- 010101 = RC5
- 010100 = RC4
- 010011 = RC3
- 010010 = RC2
- 010001 = RC1
- 010000 = RC0
- 001111 = RB7(4)
- 001110 = RB6(4)
- 001101 = RB5(4)
- 001100 = RB4(4)
- 001011-000110 = Reserved
- 000101 = RA5
- 000100 = RA4
- 000011 = RA3
- 000010 = RA2
- 000001 = RA1
- 000000 = RA0

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

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Note 1: See **Section 21.0 "5-Bit Digital-to-Analog Converter (DAC1) Module"** for more information.
2: See **Section 18.0 "Fixed Voltage Reference (FVR)"** for more information.
3: See **Section 19.0 "Temperature Indicator Module"** for more information.
4: Present only on the PIC16(L)F15344.

REGISTER 20-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
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	

bit 7-0 **ADRES<9:2>**: ADC Result Register bits
Upper eight bits of 10-bit conversion result

REGISTER 20-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<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 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

22.1 NCO OPERATION

The NCO operates by repeatedly adding a fixed value to an accumulator. Additions occur at the input clock rate. The accumulator will overflow with a carry periodically, which is the raw NCO output (NCO_overflow). This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. See Equation 22-1.

The NCO output can be further modified by stretching the pulse or toggling a flip-flop. The modified NCO output is then distributed internally to other peripherals and can be optionally output to a pin. The accumulator overflow also generates an interrupt (NCO_overflow).

The NCO period changes in discrete steps to create an average frequency.

EQUATION 22-1: NCO OVERFLOW FREQUENCY

$$F_{OVERFLOW} = \frac{NCO\ Clock\ Frequency \times Increment\ Value}{2^{20}}$$

22.1.1 NCO CLOCK SOURCES

Clock sources available to the NCO include:

- HFINTOSC
- FOSC
- LC1_out
- LC2_out
- LC3_out
- LC4_out
- MFINTOSC (500 kHz)
- MFINTOSC (32 kHz)
- CLKR

The NCO clock source is selected by configuring the N1CKS<2:0> bits in the NCO1CLK register.

22.1.2 ACCUMULATOR

The accumulator is a 20-bit register. Read and write access to the accumulator is available through three registers:

- NCO1ACCL
- NCO1ACCH
- NCO1ACCU

22.1.3 ADDER

The NCO Adder is a full adder, which operates synchronously from the source clock. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

22.1.4 INCREMENT REGISTERS

The increment value is stored in three registers making up a 20-bit incrementer. In order of LSB to MSB they are:

- NCO1INCL
- NCO1INCH
- NCO1INCU

When the NCO module is enabled, the NCO1INCU and NCO1INCH registers should be written first, then the NCO1INCL register. Writing to the NCO1INCL register initiates the increment buffer registers to be loaded simultaneously on the second rising edge of the NCO_clk signal.

The registers are readable and writable. The increment registers are double-buffered to allow value changes to be made without first disabling the NCO module.

When the NCO module is disabled, the increment buffers are loaded immediately after a write to the increment registers.

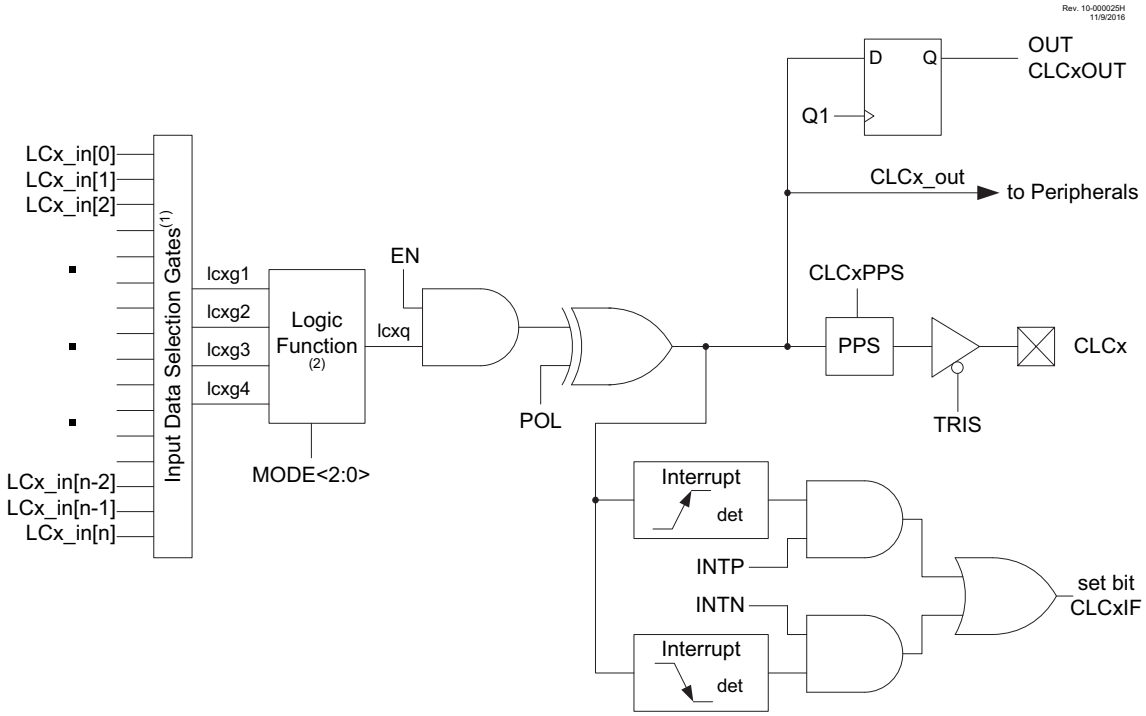
Note: The increment buffer registers are not user-accessible.

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWG1CLKCON	—	—	—	—	—	—	—	CS	357
CWG1ISM	—	—	—	—	IS<3:0>				357
CWG1DBR	—	—	DBR<5:0>						353
CWG1DBF	—	—	DBF<5:0>						353
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			356
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	352
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	354
CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	355
CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	356

Legend: – = unimplemented locations read as '0'. Shaded cells are not used by CWG.

FIGURE 31-1: CLCx SIMPLIFIED BLOCK DIAGRAM



Note 1: See Figure 31-2: Input Data Selection and Gating.

2: See Figure 31-3: Programmable Logic Functions.

FIGURE 32-17: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)

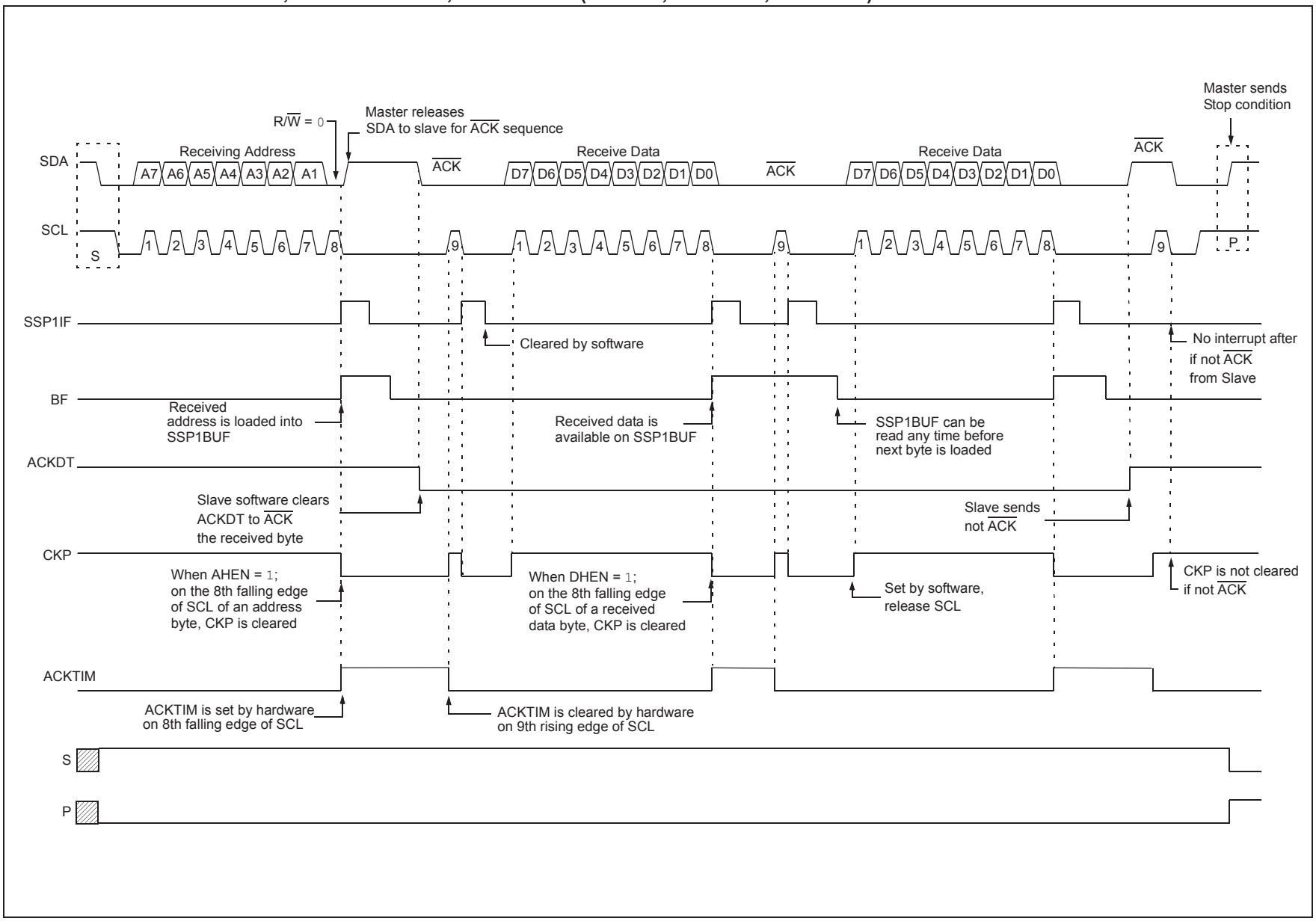
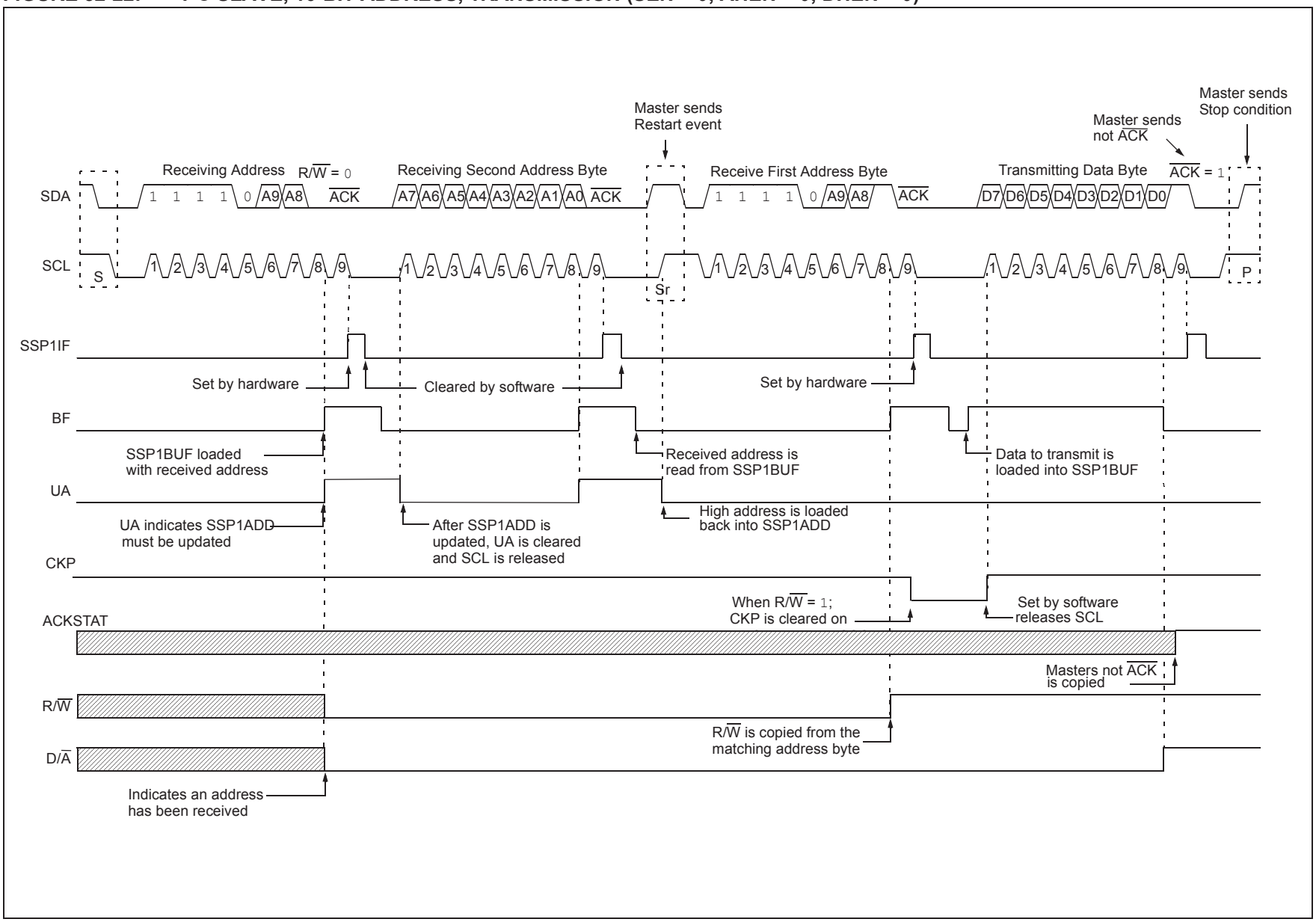


FIGURE 32-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)



33.2 Clock Accuracy with Asynchronous Operation

The factory calibrates the internal oscillator block output (INTOSC). However, the INTOSC frequency may drift as V_{DD} or temperature changes, and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source. See **Section 9.2.2.2 "Internal Oscillator Frequency Adjustment"** for more information.

The other method adjusts the value in the Baud Rate Generator. This can be done automatically with the Auto-Baud Detect feature (see **Section 33.3.1 "Auto-Baud Detect"**). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

33.6 Register Definitions: EUSART Control

REGISTER 33-1: TXxSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Unused in this mode – value ignored
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send SYNCH BREAK on next transmission – Start bit, followed by 12 '0' bits, followed by Stop bit; cleared by hardware upon completion
0 = SYNCH BREAK transmission disabled or completed
Synchronous mode:
Unused in this mode – value ignored

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode – value ignored

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

LSLF **Logical Left Shift**

Syntax: [*label*] LSLF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF **Logical Right Shift**

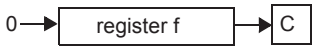
Syntax: [*label*] LSRF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: 0 → dest<7>
(f<7:1>) → dest<6:0>,
(f<0>) → C,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

TABLE 37-2: SUPPLY CURRENT (I_{DD})(1,2,4)

PIC16LF15324/44			Standard Operating Conditions (unless otherwise stated)					
PIC16F15324/44								
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max.	Units	Conditions	
							V_{DD}	Note
D100	$I_{DD_{XT4}}$	XT = 4 MHz	—	360	400	µA	3.0V	
D100	$I_{DD_{XT4}}$	XT = 4 MHz	—	380	450	µA	3.0V	
D101	$I_{DD_{HFO16}}$	HFINTOSC = 16 MHz	—	1.4	1.8	mA	3.0V	
D101	$I_{DD_{HFO16}}$	HFINTOSC = 16 MHz	—	1.5	1.9	mA	3.0V	
D102	$I_{DD_{HFOPLL}}$	HFINTOSC = 32 MHz	—	2.3	3.2	mA	3.0V	
D102	$I_{DD_{HFOPLL}}$	HFINTOSC = 32 MHz	—	2.4	3.2	mA	3.0V	
D103	$I_{DD_{HSPLL32}}$	HS+PLL = 32 MHz	—	2.3	3.2	mA	3.0V	
D103	$I_{DD_{HSPLL32}}$	HS+PLL = 32 MHz	—	2.4	3.2	mA	3.0V	
D104	$I_{DD_{IDLE}}$	IDLE mode, HFINTOSC = 16 MHz	—	1.05	1.5	mA	3.0V	
D104	$I_{DD_{IDLE}}$	IDLE mode, HFINTOSC = 16 MHz	—	1.15	1.5	mA	3.0V	
D105	$I_{DD_{DOZE}}$(3)	DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16	—	1.1	—	mA	3.0V	
D105	$I_{DD_{DOZE}}$(3)	DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16	—	1.2	—	mA	3.0V	

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

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

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

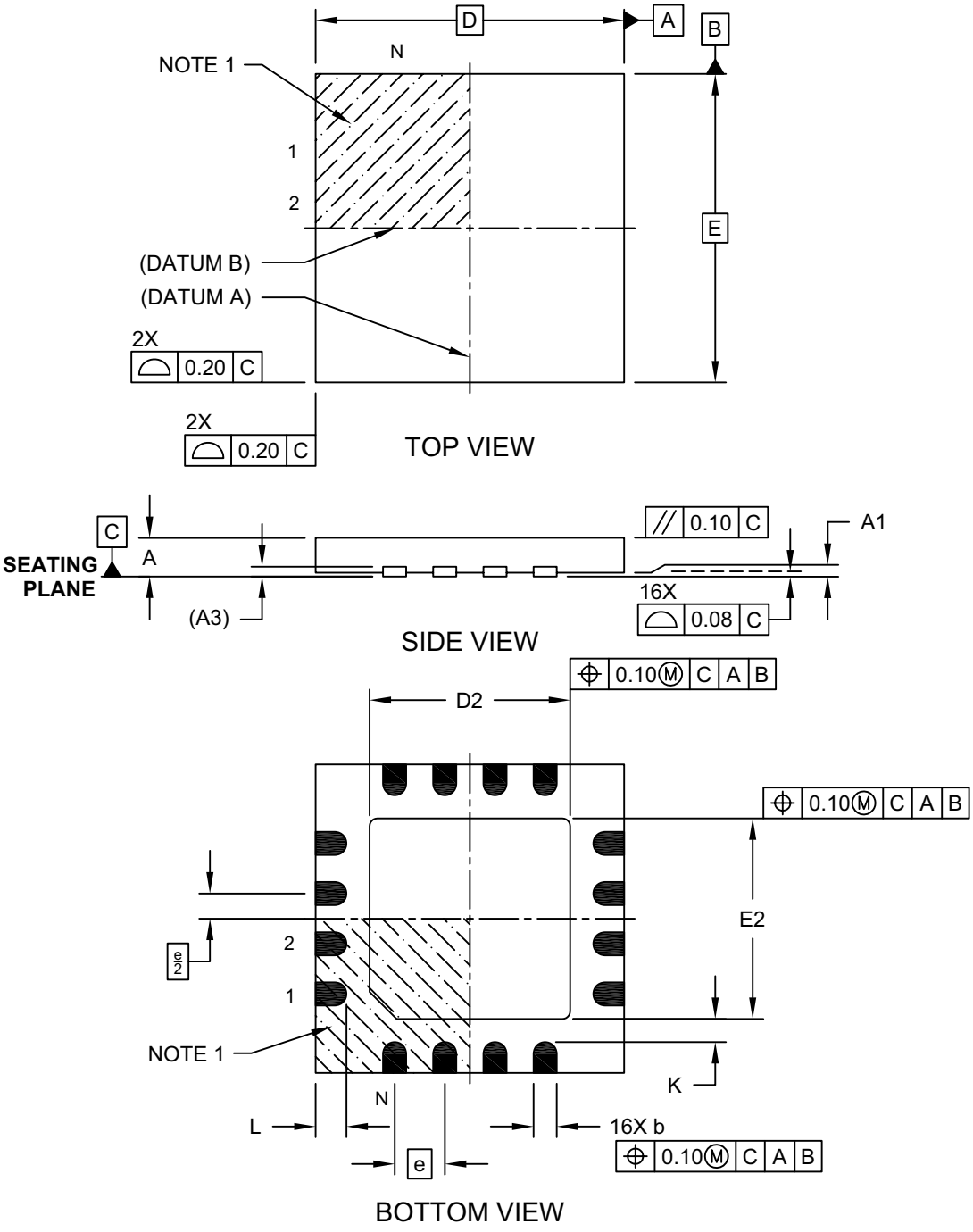
3: $I_{DD_{DOZE}}$ = [$I_{DD_{IDLE}}$*(N-1)/N] + $I_{DD_{HFO}}$16/N where N = DOZE Ratio (Register 11-2).

4: PMD bits are all in the default state, no modules are disabled.

5: = F device

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-257A Sheet 1 of 2