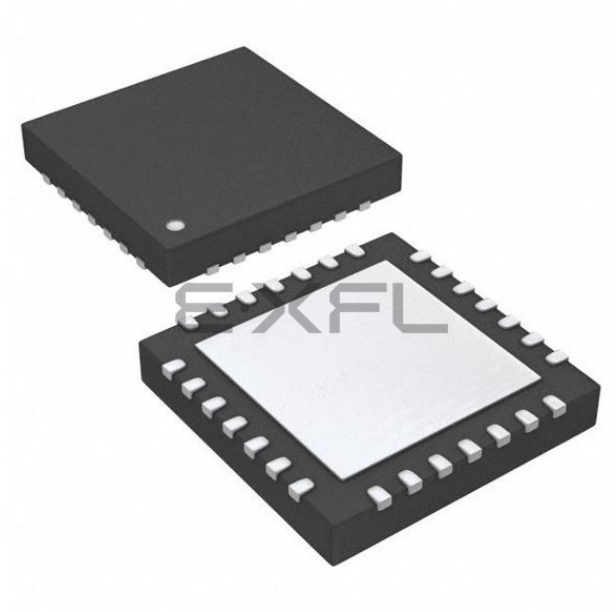


Welcome to E-XFL.COM

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	CANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; 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/pic18lf25k83t-i-ml

REGISTER 7-4: OSCSTAT: OSCILLATOR STATUS REGISTER 1

R-q/q	R-q/q	R-q/q	R-q/q	R-q/q	R-q/q	U-0	R-q/q
EXTOR	HFOR	MFOR	LFOR	SOR	ADOR	—	PLLR
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 = Reset value is determined by hardware

bit 7 **EXTOR:** EXTOSC (external) Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 6 **HFOR:** HFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 5 **MFOR:** MFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 4 **LFOR:** LFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

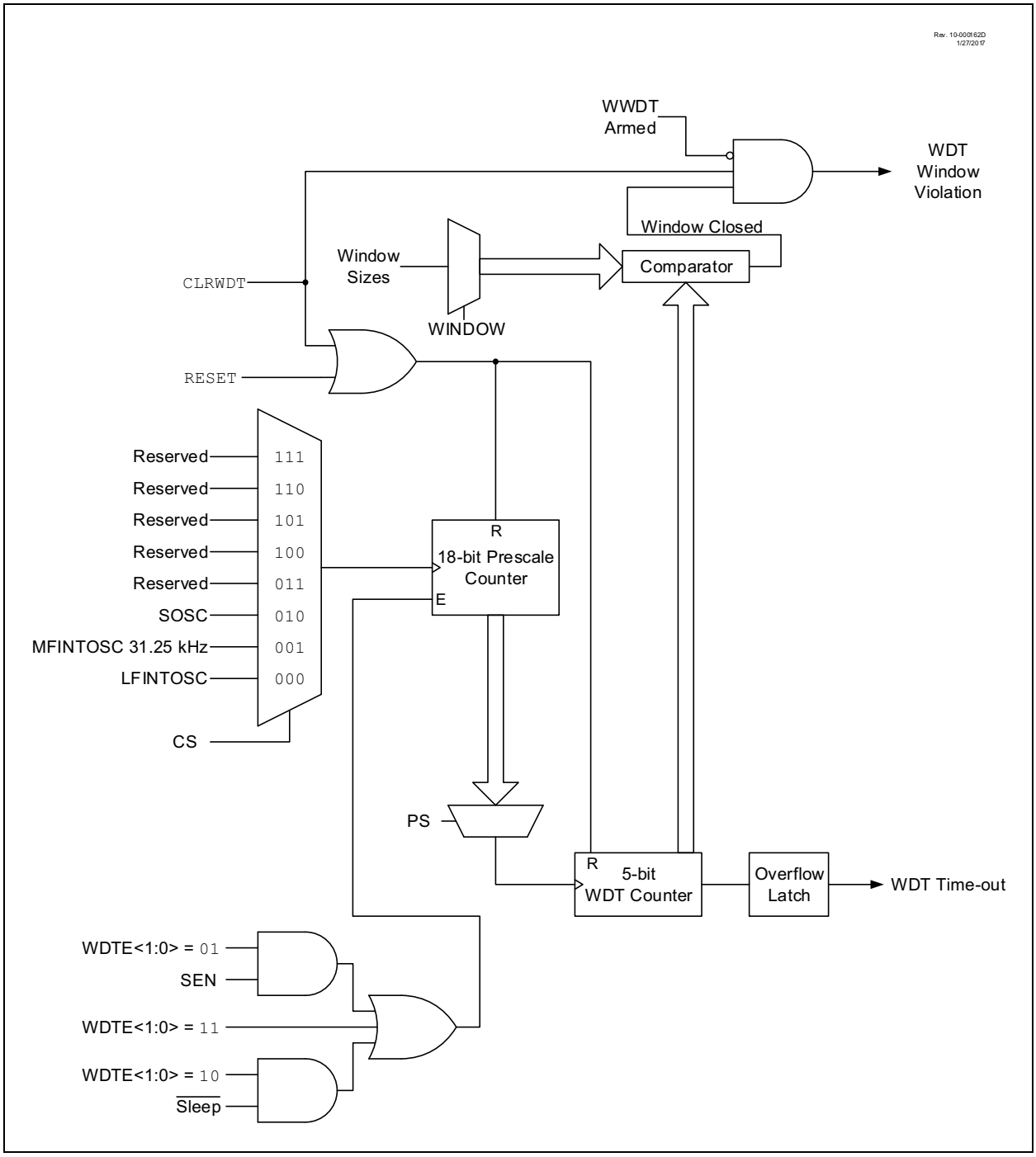
bit 3 **SOR:** Secondary (Timer1) Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 2 **ADOR:** ADC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **PLLR:** PLL is Ready bit
1 = The PLL is ready to be used
0 = The PLL is not enabled, the required input source is not ready, or the PLL is not locked.

FIGURE 11-1: WINDOWED WATCHDOG TIMER BLOCK DIAGRAM



13.1.5 ERASING PROGRAM FLASH MEMORY

The minimum erase block is 64 words (refer to Table 5-4). Only through the use of an external programmer, or through ICSP™ control, can larger blocks of program memory be bulk erased. Word erase in the program memory array is not supported.

For example, when initiating an erase sequence from a microcontroller with erase row size of 64 words, a block of 64 words (128 bytes) of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. The TBLPTR<5:0> bits are ignored.

The NVMCON1 register commands the erase operation. The REG<1:0> bits must be set to point to the Program Flash Memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

The NVM unlock sequence described in **Section 13.1.4 "NVM Unlock Sequence"** should be used to guard against accidental writes. This is sometimes referred to as a long write.

A long write is necessary for erasing program memory. Instruction execution is halted during the long write cycle. The long write is terminated by the internal programming timer.

13.1.5.1 Program Flash Memory Erase Sequence

The sequence of events for erasing a block of internal program memory is:

1. REG bits of the NVMCON1 register point to PFM
2. Set the FREE and WREN bits of the NVMCON1 register
3. Perform the unlock sequence as described in **Section 13.1.4 "NVM Unlock Sequence"**

If the PFM address is write-protected, the WR bit will be cleared and the erase operation will not take place, WRERR is signaled in this scenario.

The operation erases the memory row indicated by masking the LSBs of the current TBLPTR.

While erasing PFM, CPU operation is suspended and it resumes when the operation is complete. Upon completion the WR bit is cleared in hardware, the NVMIF is set and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations and WREN will remain unchanged.

Note 1:	If a write or erase operation is terminated by an unexpected event, WRERR bit will be set which the user can check to decide whether a rewrite of the location(s) is needed.
2:	WRERR is set if WR is written to '1' while TBLPTR points to a write-protected address.
3:	WRERR is set if WR is written to '1' while TBLPTR points to an invalid address location (Table 13-1).

13.4 Register Definitions: Nonvolatile Memory

REGISTER 13-1: NVMCON1: NONVOLATILE MEMORY CONTROL 1 REGISTER

R/W-0/0	R/W-0/0	U-0	R/S/HC-0/0	R/W/HS-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
REG<1:0>		—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	HC = Bit is cleared by hardware
x = Bit is unknown	-n = Value at POR	S = Bit can be set by software, but not cleared
'0' = Bit is cleared	'1' = Bit is set	U = Unimplemented bit, read as '0'

bit 7-6 **REG<1:0>:** NVM Region Selection bit
10 =Access PFM Locations
x1 = Access User IDs, Configuration Bits, DIA, DCI, Rev ID and Device ID
00 = Access Data EEPROM Memory Locations

bit 5 **Unimplemented:** Read as '0'

bit 4 **FREE:** Program Flash Memory Erase Enable bit(1)
1 = Performs an erase operation on the next WR command
0 = The next WR command performs a write operation

bit 3 **WRERR:** Write-Reset Error Flag bit(2,3,4)
1 = A write operation was interrupted by a Reset (hardware set),
or WR was written to 1'b1 when an invalid address is accessed (Table 4-1, Table 13-1)
or WR was written to 1'b1 when REG<1:0> and address do not point to the same region
or WR was written to 1'b1 when a write-protected address is accessed (Table 4-2).
0 = All write operations have completed normally

bit 2 **WREN:** Program/Erase Enable bit
1 = Allows program/erase and refresh cycles
0 = Inhibits programming/erasing and user refresh of NVM

bit 1 **WR:** Write Control bit(5,6,7)
When REG points to a Data EEPROM Memory location:
1 = Initiates an erase/program cycle at the corresponding Data EEPROM Memory location
When REG points to a PFM location:
1 = Initiates the PFM write operation with data from the holding registers
0 = NVM program/erase operation is complete and inactive

bit 0 **RD:** Read Control bit(8)
1 = Initiates a read at address pointed by REG and NVMADR, and loads data into NVMDAT
0 = NVM read operation is complete and inactive

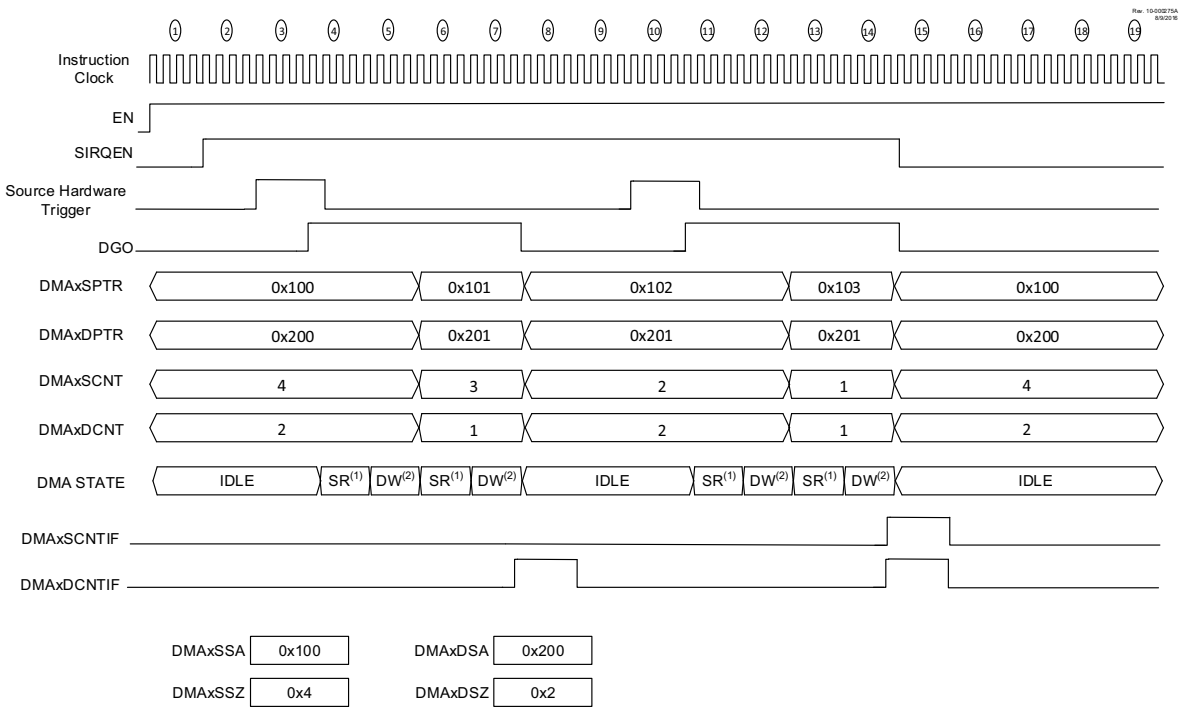
Note 1: This can only be used with PFM.
2: This bit is set when WR = 1 and clears when the internal programming timer expires or the write is completed successfully.
3: Bit must be cleared by the user; hardware will not clear this bit.
4: Bit may be written to '1' by the user in order to implement test sequences.
5: This bit can only be set by following the unlock sequence of **Section 13.1.4 "NVM Unlock Sequence"**.
6: Operations are self-timed and the WR bit is cleared by hardware when complete.
7: Once a write operation is initiated, setting this bit to zero will have no effect.
8: The bit can only be set in software. The bit is cleared by hardware when the operation is complete.

The following sections describe with visual reference the sequence of events for different configurations of the DMA module

15.9.1 SOURCE STOP

When the Source Stop bit is set (SSTP = 1) and the DMAxSCNT register reloads, the DMA clears the SIRQEN bit to stop receiving new start interrupt request signals and sets the DMAxSCNTIF flag.

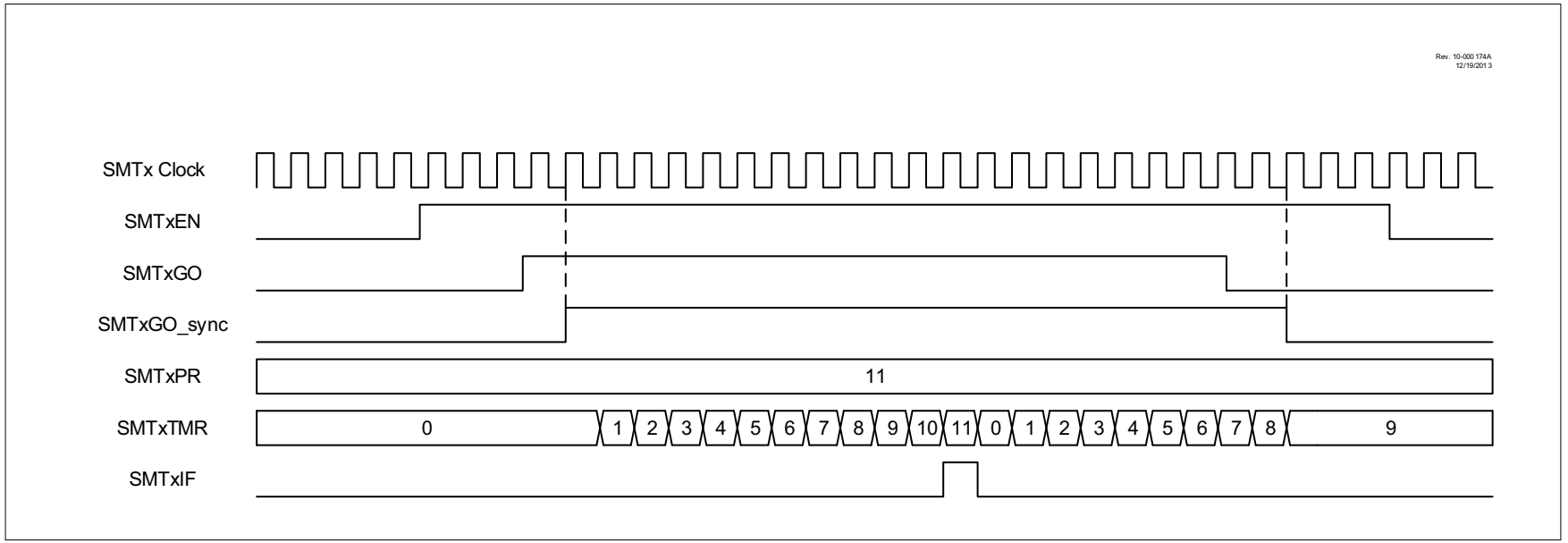
FIGURE 15-5: GPR-GPR TRANSACTIONS WITH HARDWARE TRIGGERS, SSTP = 1



Note 1: SR – Source Read

2: DW – Destination Write

FIGURE 25-3: TIMER MODE TIMING DIAGRAM



REGISTER 25-5: SMTxWIN: SMTx WINDOW INPUT SELECT 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
—	—	—	WSEL<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	q = Value depends on condition

bit 7-5 **Unimplemented**: Read as '0'

bit 4-0 **WSEL<4:0>:** SMTx Window Selection bits

11111 = Reserved
•
•
•
11011 = Reserved
11010 = CLC4_out
11001 = CLC3_out
11000 = CLC2_out
10111 = CLC1_out
10110 = ZCD1_out
10101 = CMP2_out
10100 = CMP1_out
10011 = NCO1_out
10010 = Reserved
10001 = Reserved
10000 = PWM8_out
01111 = PWM7_out
01110 = PWM6_out
01101 = PWM5_out
01100 = CCP4_out
01011 = CCP3_out
01010 = CCP2_out
01001 = CCP1_out
01000 = TMR6_postscaled
00111 = TMR4_postscaled
00110 = TMR2_postscaled
00101 = TMR0_overflow
00100 = CLKREF
00011 = SOSC
00010 = MFINTOSC/16 (32 kHz)
00001 = LFINTOSC
00000 = SMTxWINPPS

REGISTER 25-16: SMTxPRL: SMT PERIOD REGISTER – LOW BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<7:0>							
bit 7							bit 0

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

bit 7-0 **SMTxPR<7:0>**: Significant bits of the SMT Timer Value for Period Match – Low Byte

REGISTER 25-17: SMTxPRH: SMT PERIOD REGISTER – HIGH BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<15:8>							
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 **SMTxPR<15:8>**: Significant bits of the SMT Timer Value for Period Match – High Byte

REGISTER 25-18: SMTxPRU: SMT PERIOD REGISTER – UPPER BYTE

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<23:16>							
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 **SMTxPR<23:16>**: Significant bits of the SMT Timer Value for Period Match – Upper Byte

REGISTER 27-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
POL	—	—	—	G4POL	G3POL	G2POL	G1POL
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 **POL:** CLCxOUT Output Polarity Control bit
1 = The output of the logic cell is inverted
0 = The output of the logic cell is not inverted

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **G4POL:** Gate 3 Output Polarity Control bit
1 = The output of gate 3 is inverted when applied to the logic cell
0 = The output of gate 3 is not inverted

bit 2 **G3POL:** Gate 2 Output Polarity Control bit
1 = The output of gate 2 is inverted when applied to the logic cell
0 = The output of gate 2 is not inverted

bit 1 **G2POL:** Gate 1 Output Polarity Control bit
1 = The output of gate 1 is inverted when applied to the logic cell
0 = The output of gate 1 is not inverted

bit 0 **G1POL:** Gate 0 Output Polarity Control bit
1 = The output of gate 0 is inverted when applied to the logic cell
0 = The output of gate 0 is not inverted

FIGURE 31-7: MANCHESTER TIMING

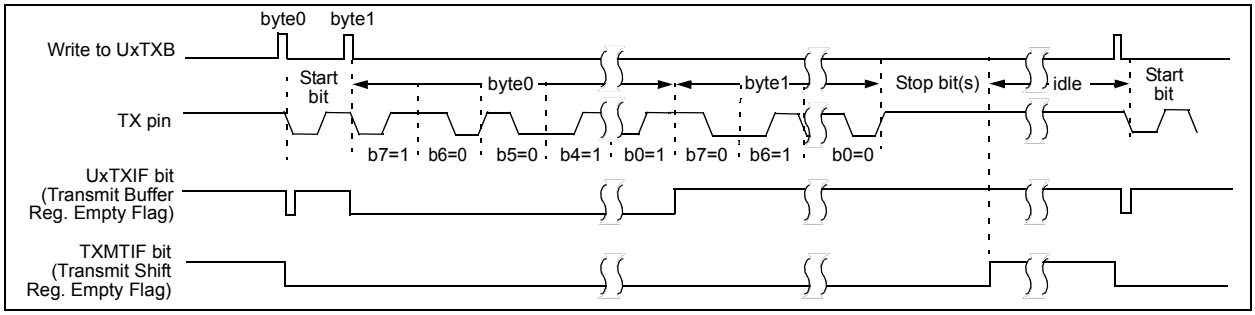


FIGURE 31-8: DALI FRAME TIMING

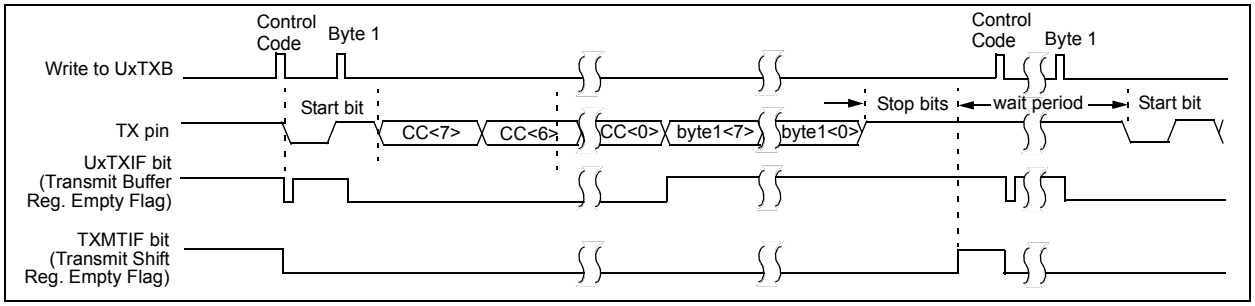
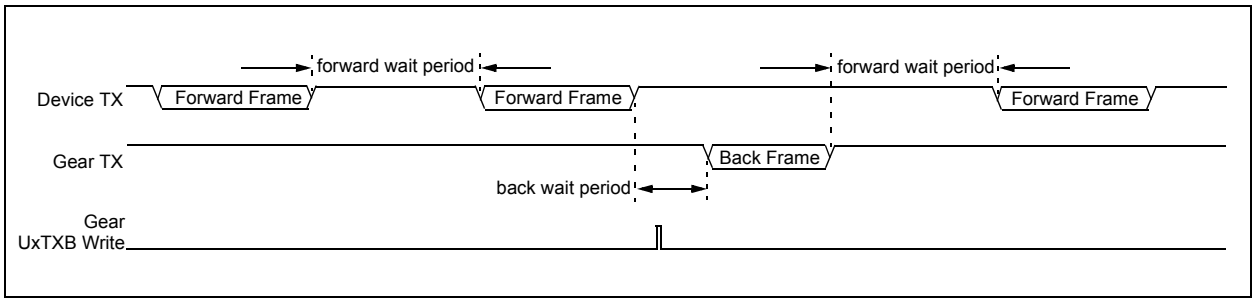


FIGURE 31-9: DALI FORWARD/BACK FRAME TIMING



33.3.2 BYTE FORMAT

All communication in I^2C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent by the receiver. After the 8th falling edge of the SCL line, the device transmitting data on the SDA line releases control of that pin to an input, and reads in an acknowledge value on the next clock pulse. The clock signal is provided by the master. Data is valid to change while the SCL line is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define Start and Stop conditions on the bus which are explained further in the chapter.

33.3.3 SDA AND SCL PINS

The user must configure these pins as open-drain outputs. This is done by clearing the appropriate TRIS bits and setting the appropriate ODCON bits. The user may also select the input threshold, slew-rate and internal pull-up settings using the RxyI2C control registers (Register 16-9).

33.3.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT<1:0> bits of the I2CxCON2 register. Hold time is the time SDA is held valid after the falling edge of SCL. A longer hold time setting may help on buses with large capacitance.

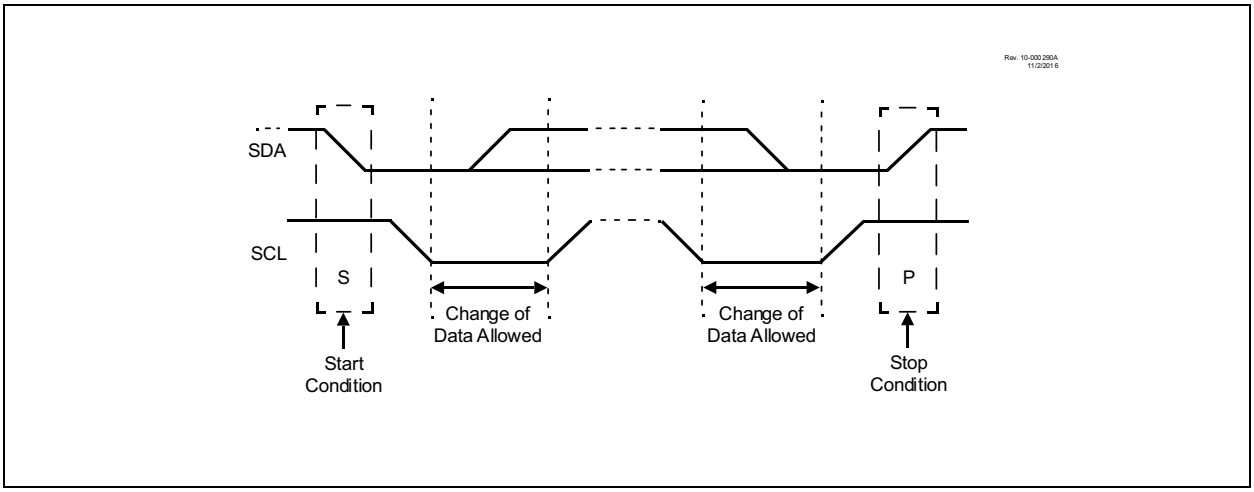
33.3.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDA line from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 33-3 shows waveforms for Start conditions. Master hardware waits for the BFRE bit of I2CxSTAT0 to be set, before asserting a Start condition on the SCL and SDA lines. If two masters assert a start at the same time, a collision will occur during the addressing phase.

33.3.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low to high while the SCL line is high. Figure 33-3 shows waveforms for Stop conditions.

FIGURE 33-3: START AND STOP CONDITIONS



Note:	At least one SCL low time must appear before a Stop is valid. Therefore, if the SDA line goes low then high again while the SCL line is high, only the Start condition is detected.

REGISTER 34-1: CANCON: CAN CONTROL REGISTER

Mode 0	R/W-1	R/W-0	R/W-0	R/S-0	R/W-0	R/W-0	R/W-0	U-0
	REQOP2	REQOP1	REQOP0	ABAT	WIN2	WIN1	WIN0	—
Mode 1	R/W-1	R/W-0	R/W-0	R/S-0	U0	U-0	U-0	U-0
	REQOP2	REQOP1	REQOP0	ABAT	—	—	—	—
Mode 2	R/W-1	R/W-0	R/W-0	R/S-0	R-0	R-0	R-0	R-0
	REQOP2	REQOP1	REQOP0	ABAT	FP3	FP2	FP1	FP0
	bit 7							bit 0

Legend:	S = Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-5 **REQOP<2:0>:** Request CAN Operation Mode bits
1xx = Requests Configuration mode
011 = Requests Listen Only mode
010 = Requests Loopback mode
001 = Disabled/Sleep mode
000 = Requests Normal mode

bit 4 **ABAT:** Abort All Pending Transmissions bit
1 = Abort all pending transmissions (in all transmit buffers)(1)
0 = Transmissions proceeding as normal

bit 3-1 Mode 0:
WIN<2:0>: Window Address bits
These bits select which of the CAN buffers to switch into the Access Bank area. This allows access to the buffer registers from any data memory bank. After a frame has caused an interrupt, the ICODE<3:0> bits can be copied to the WIN<2:0> bits to select the correct buffer. See Example 34-2 for a code example.
111 = Receive Buffer 0
110 = Receive Buffer 0
101 = Receive Buffer 1
100 = Transmit Buffer 0
011 = Transmit Buffer 1
010 = Transmit Buffer 2
001 = Receive Buffer 0
000 = Receive Buffer 0

bit 0 Mode 0:
Unimplemented: Read as ‘0’

bit 4-0 Mode 1:
Unimplemented: Read as ‘0’
Mode 2:
FP<3:0>: FIFO Read Pointer bits
These bits point to the message buffer to be read.
0000 = Receive Message Buffer 0
0001 = Receive Message Buffer 1
0010 = Receive Message Buffer 2
0011 = Receive Message Buffer 3
0100 = Receive Message Buffer 4
0101 = Receive Message Buffer 5
0110 = Receive Message Buffer 6
0111 = Receive Message Buffer 7
1000:1111 Reserved

Note 1: This bit will clear when all transmissions are aborted.

34.15.3.1 Programmable TX/RX and Auto-RTR Buffers

The ECAN module contains six message buffers that can be programmed as transmit or receive buffers. Any of these buffers can also be programmed to automatically handle RTR messages.

Note:	These registers are not used in Mode 0.

REGISTER 34-22: BnCON: TX/RX BUFFER 'n' CONTROL REGISTERS IN RECEIVE MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 0](1)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
RXFUL(2)	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0
bit 7							bit 0

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

bit 7 **RXFUL:** Receive Full Status bit(2)
- 1 = Receive buffer contains a received message
- 0 = Receive buffer is open to receive a new message

bit 6 **RXM1:** Receive Buffer Mode bit
- 1 = Receive all messages including partial and invalid (acceptance filters are ignored)
- 0 = Receive all valid messages as per acceptance filters

bit 5 **RXRTRRO:** Read-Only Remote Transmission Request for Received Message bit
- 1 = Received message is a remote transmission request
- 0 = Received message is not a remote transmission request

bit 4-0 **FILHIT<4:0>:** Filter Hit bits
These bits indicate which acceptance filter enabled the last message reception into this buffer.
- 01111 = Acceptance Filter 15 (RXF15)
- 01110 = Acceptance Filter 14 (RXF14)
- ...
- 00001 = Acceptance Filter 1 (RXF1)
- 00000 = Acceptance Filter 0 (RXF0)

Note 1: These registers are available in Mode 1 and 2 only.

2: This bit is set by the CAN module upon receiving a message and must be cleared by software after the buffer is read. As long as RXFUL is set, no new message will be loaded and the buffer will be considered full.

REGISTER 34-57: PIE5: PERIPHERAL INTERRUPT ENABLE REGISTER 5

Mode 0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	IRXIE	WAKIE	ERRIE	TXB2IE	TXB1IE(1)	TXB0IE(1)	RXB1IE	RXB0IE

Mode 1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	IRXIE	WAKIE	ERRIE	TXBnIE	TXB1IE(1)	TXB0IE(1)	RXBnIE	FIFOWMIE
	bit 7							bit 0

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

bit 7 **IRXIE:** CAN Bus Error Message Received Interrupt Enable bit
1 = Enable invalid message received interrupt
0 = Disable invalid message received interrupt

bit 6 **WAKIE:** CAN bus Activity Wake-up Interrupt Enable bit
1 = Enable bus activity wake-up interrupt
0 = Disable bus activity wake-up interrupt

bit 5 **ERRIE:** CAN bus Error Interrupt Enable bit
1 = Enable CAN module error interrupt
0 = Disable CAN module error interrupt

bit 4 When CAN is in Mode 0:
TXB2IE: CAN Transmit Buffer 2 Interrupt Enable bit
1 = Enable Transmit Buffer 2 interrupt
0 = Disable Transmit Buffer 2 interrupt
When CAN is in Mode 1 or 2:
TXBnIE: CAN Transmit Buffer Interrupts Enable bit
1 = Enable transmit buffer interrupt; individual interrupt is enabled by TXBIE and BIE0
0 = Disable all transmit buffer interrupts

bit 3 **TXB1IE:** CAN Transmit Buffer 1 Interrupt Enable bit(1)
1 = Enable Transmit Buffer 1 interrupt
0 = Disable Transmit Buffer 1 interrupt

bit 2 **TXB0IE:** CAN Transmit Buffer 0 Interrupt Enable bit(1)
1 = Enable Transmit Buffer 0 interrupt
0 = Disable Transmit Buffer 0 interrupt

bit 1 When CAN is in Mode 0:
RXB1IE: CAN Receive Buffer 1 Interrupt Enable bit
1 = Enable Receive Buffer 1 interrupt
0 = Disable Receive Buffer 1 interrupt
When CAN is in Mode 1 or 2:
RXBnIE: CAN Receive Buffer Interrupts Enable bit
1 = Enable receive buffer interrupt; individual interrupt is enabled by BIE0
0 = Disable all receive buffer interrupts

bit 0 When CAN is in Mode 0:
RXB0IE: CAN Receive Buffer 0 Interrupt Enable bit
1 = Enable Receive Buffer 0 interrupt
0 = Disable Receive Buffer 0 interrupt
When CAN is in Mode 1:
Unimplemented: Read as '0'
When CAN is in Mode 2:
FIFOWMIE: FIFO Watermark Interrupt Enable bit
1 = Enable FIFO watermark interrupt
0 = Disable FIFO watermark interrupt

Note 1: In CAN Mode 1 and 2, these bits are forced to '0'.

REGISTER 34-58: IPR5: PERIPHERAL INTERRUPT PRIORITY REGISTER 5

Mode 0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	IRXIP	WAKIP	ERRIP	TXB2IP	TXB1IP(1)	TXB0IP(1)	RXB1IP	RXB0IP

Mode 1,2	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	IRXIP	WAKIP	ERRIP	TXBnIP	TXB1IP(1)	TXB0IP(1)	RXBnIP	FIFOWMIP
	bit 7							bit 0

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

bit 7 **IRXIP:** CAN Bus Error Message Received Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **WAKIP:** CAN Bus Activity Wake-up Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **ERRIP:** CAN Module Error Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 When CAN is in Mode 0:
TXB2IP: CAN Transmit Buffer 2 Interrupt Priority bit
1 = High priority
0 = Low priority
When CAN is in Mode 1 or 2:
TXBnIP: CAN Transmit Buffer Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **TXB1IP:** CAN Transmit Buffer 1 Interrupt Priority bit(1)
1 = High priority
0 = Low priority

bit 2 **TXB0IP:** CAN Transmit Buffer 0 Interrupt Priority bit(1)
1 = High priority
0 = Low priority

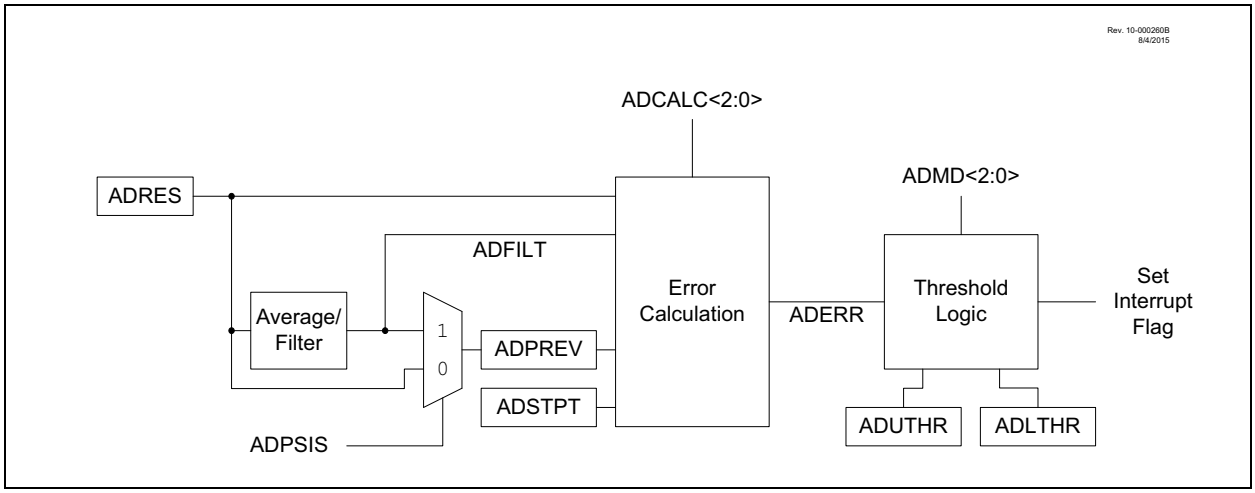
bit 1 When CAN is in Mode 0:
RXB1IP: CAN Receive Buffer 1 Interrupt Priority bit
1 = High priority
0 = Low priority
When CAN is in Mode 1 or 2:
RXBnIP: CAN Receive Buffer Interrupts Priority bit
1 = High priority
0 = Low priority

Note 1: In CAN Mode 1 and 2, these bits are forced to '0'.

37.6 Computation Operation

The ADC module hardware is equipped with post conversion computation features. These features provide data post-processing functions that can be operated on the ADC conversion result, including digital filtering/averaging and threshold comparison functions.

FIGURE 37-10: COMPUTATIONAL FEATURES SIMPLIFIED BLOCK DIAGRAM



The operation of the ADC computational features is controlled by ADMD <2:0> bits in the ADCON2 register.

The module can be operated in one of five modes:

• **Basic**: In this mode, ADC conversion occurs on single (ADDSEN = 0) or double (ADDSEN = 1) samples. ADIF is set after all the conversion are complete.

• **Accumulate**: With each trigger, the ADC conversion result is added to accumulator and CNT increments. ADIF is set after each conversion. ADTIF is set according to the calculation mode.

• **Average**: With each trigger, the ADC conversion result is added to the accumulator. When the RPT number of samples have been accumulated, a threshold test is performed. Upon the next trigger, the accumulator is cleared. For the subsequent tests, additional RPT samples are required to be accumulated.

• **Burst Average**: At the trigger, the accumulator is cleared. The ADC conversion results are then collected repetitively until RPT samples are accumulated and finally the threshold is tested.

• **Low-Pass Filter (LPF)**: With each trigger, the ADC conversion result is sent through a filter. When RPT samples have occurred, a threshold test is performed. Every trigger after that the ADC conversion result is sent through the filter and another threshold test is performed.

The five modes are summarized in Table 37-2 below.

REGISTER 37-36: ADCP: ADC CHARGE PUMP CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	U-0	U-0	U-0	R-0/0
CPON	—	—	—	—	—	—	CPRDY
bit 7							bit 0

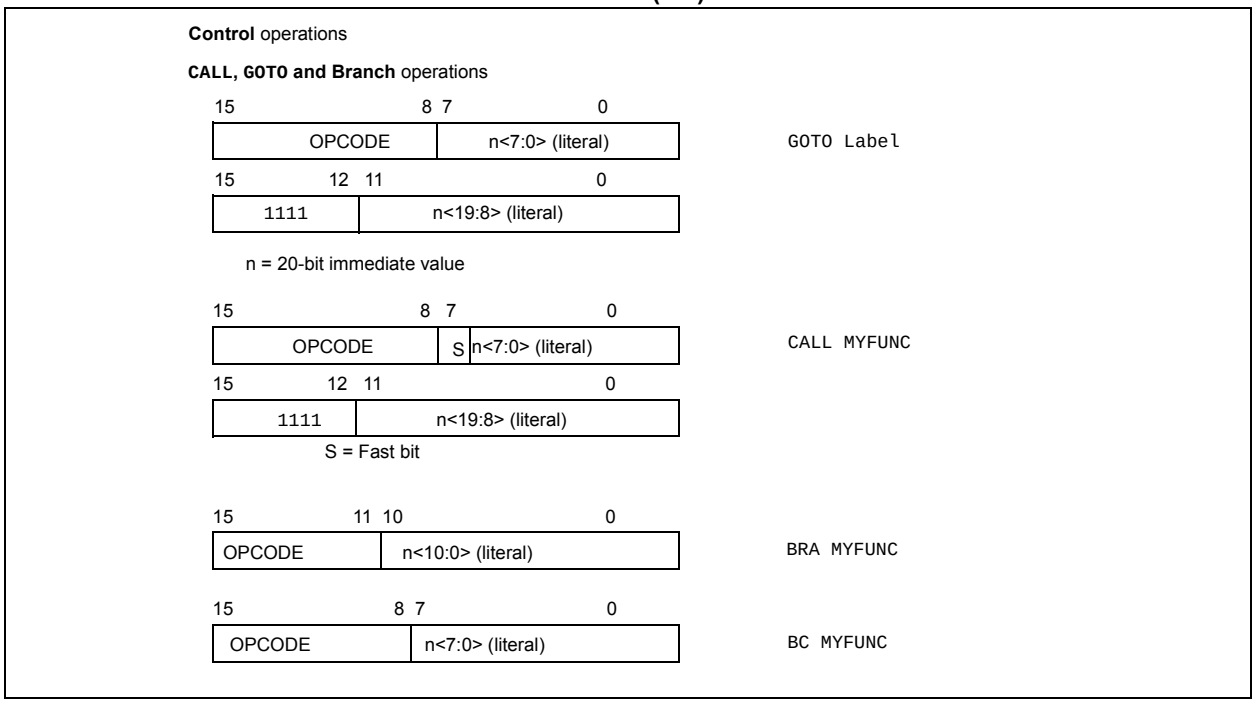
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HS= Hardware set

bit 7 **CPON**: Charge Pump On Control bit
1 = Charge Pump On when requested by the ADC
0 = Charge Pump Off

bit 6-1 **Unimplemented:** Read as ‘0’

bit 0 CPRDY: Charge Pump Ready Status bit
1 = Charge Pump is ready
0 = Charge Pump is not ready (or never started)

FIGURE 42-2: General Format for Instructions (2/2)



ADDWFC **ADD W and CARRY bit to f**

Syntax: ADDWFC f {,d {,a}}

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (W) + (f) + (C) → dest

Status Affected: N,OV, C, DC, Z

Encoding:

0010	00da	ffff	ffff

Description: Add W, the CARRY flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 42.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: ADDWFC REG, 0, 1

Before Instruction
CARRY bit = 1
REG = 02h
W = 4Dh
After Instruction
CARRY bit = 0
REG = 02h
W = 50h

ANDLW **AND literal with W**

Syntax: ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. k → W

Status Affected: N, Z

Encoding:

0000	1011	kkkk	kkkk

Description: The contents of W are AND'ed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: ANDLW 05Fh

Before Instruction
W = A3h
After Instruction
W = 03h

TABLE 43-1: REGISTER FILE SUMMARY FOR PIC18(L)F25/26K83 DEVICES (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3F6Ah	PWM6CON	EN	—	OUT	POL	—	—	—	—	344
3F69h	PWM6DCH	DC9		DC7	DC6	DC5	DC4	DC3	DC2	346
3F68h	PWM6DCL	DC1	DC0	—	—	—	—	—	—	346
3F67h	—	Unimplemented								—
3F66h	PWM7CON	EN	—	OUT	POL	—	—	—	—	344
3F65h	PWM7DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	346
3F64h	PWM7DCL	DC1	DC0	—	—	—	—	—	—	346
3F63h	—	Unimplemented								—
3F62h	PWM8CON	EN	—	OUT	POL	—	—	—	—	344
3F61h	PWM8DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	346
3F60h	PWM8DCL	DC1	DC0	—	—	—	—	—	—	346
3F5Fh	CCPTMRS1	P8TSEL		P7TSEL		P6TSEL		P5TSEL		345
3F5Eh	CCPTMRS0	C4TSEL		C3TSEL		C2TSEL		C1TSEL		345
3F5Dh - 3F5Bh	—	Unimplemented								—
3F5Ah	CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	414
3F59h	CWG1AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	416
3F58h	CWG1AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	415
3F57h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	411
3F56h	CWG1CON0	EN	LD	—	—	—	MODE			410
3F55h	CWG1DBF	—	—	DBF						417
3F54h	CWG1DBR	—	—	DBR						417
3F53h	CWG1ISM	—	—	—	—	IS				413
3F52h	CWG1CLK	—	—	—	—	—	—	—	CS	412
3F51h	CWG2STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	414
3F50h	CWG2AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	416
3F4Fh	CWG2AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	415
3F4Eh	CWG2CON1	—	—	IN	—	POLD	POLC	POLB	POLA	411
3F4Dh	CWG2CON0	EN	LD	—	—	—	MODE			410
3F4Ch	CWG2DBF	—	—	DBF						417
3F4Bh	CWG2DBR	—	—	DBR						417
3F4Ah	CWG2ISM	—	—	—	—	IS				413
3F49h	CWG2CLK	—	—	—	—	—	—	—	CS	412
3F48h	CWG3STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	414
3F47h	CWG3AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	416
3F46h	CWG3AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	415
3F45h	CWG3CON1	—	—	IN	—	POLD	POLC	POLB	POLA	411
3F44h	CWG3CON0	EN	LD	—	—	—	MODE			410
3F43h	CWG3DBF	—	—	DBF						417
3F42h	CWG3DBR	—	—	DBR						417
3F41h	CWG3ISM	—	—	—	—	IS				413
3F40h	CWG3CLK	—	—	—	—	—	—	—	CS	412
3F3Fh	NCO1CLK	PWS			—	CKS				440
3F3Eh	NCO1CON	EN	—	OUT	POL	—	—	—	PFM	439
3F3Dh	NCO1INCU	INC								443
3F3Ch	NCO1INCH	INC								442
3F3Bh	NCO1INCL	INC								442
3F3Ah	NCO1ACCU	ACC								442
3F39h	NCO1ACCH	ACC								441
3F38h	NCO1ACCL	ACC								441

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not present in LF devices.