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	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
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-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f26k83-i-ss

TABLE 1-1: DEVICE FEATURES

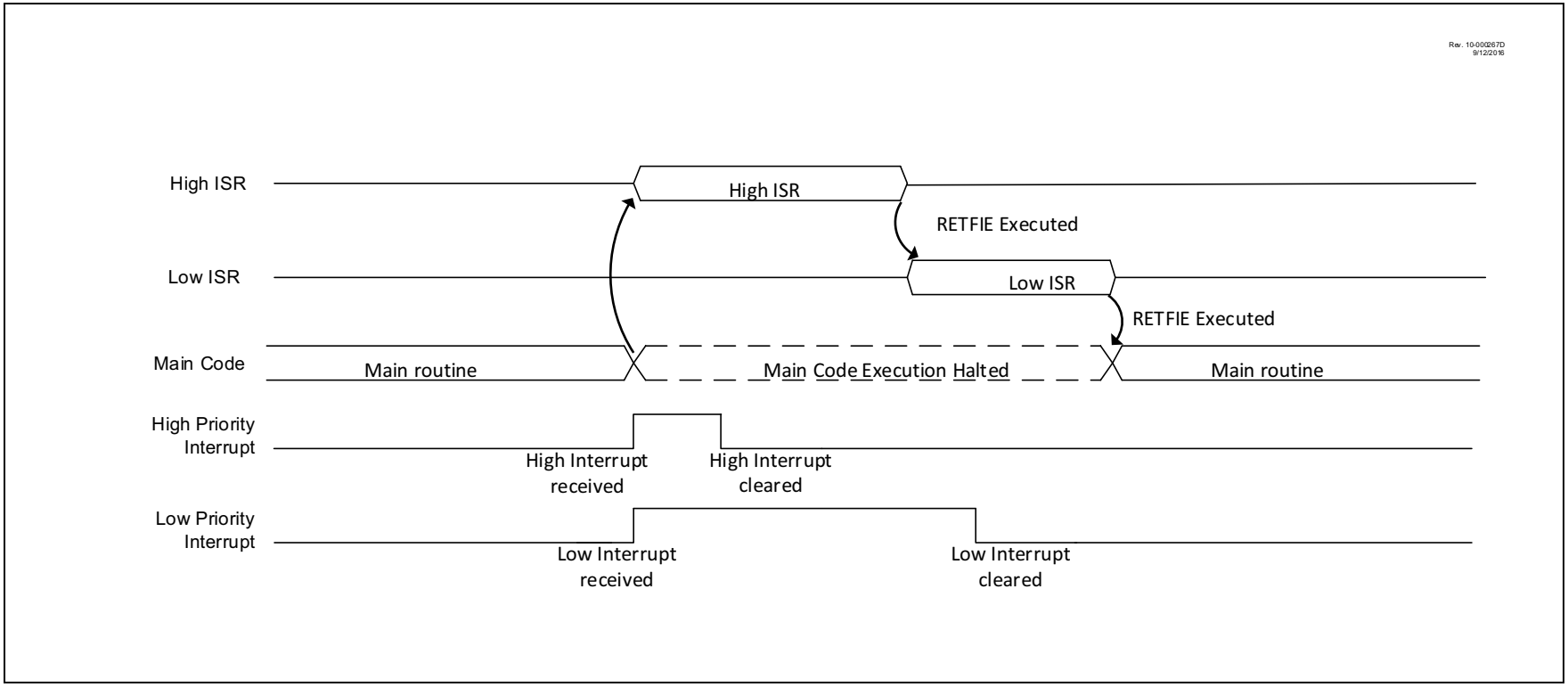
Features	PIC18(L)F25K83	PIC18(L)F26K83
Program Memory (Bytes)	32768	65536
Program Memory (Instructions)	16384	32768
Data Memory (Bytes)	2048	4096
Data EEPROM Memory (Bytes)	1024	1024
Packages	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN
I/O Ports	A,B,C,E(1)	A,B,C,E(1)
12-Bit Analog-to-Digital Conversion Module (ADC^2) with Computation Accelerator	5 internal 24 external	5 internal 24 external
Capture/Compare/PWM Modules (CCP)	4	
10-Bit Pulse-Width Modulator (PWM)	4	
Timers (16-/8-bit)	4/3	
Serial Communications	2 UARTs with DMX/DALI/LIN, 2 I^2C, 1 SPI	
Complementary Waveform Generator (CWG)	3	
Zero-Cross Detect (ZCD)	1	
Data Signal Modulator (DSM)	1	
Signal Measurement Timer (SMT)	2	
5-bit Digital to Analog Converter (DAC)	1	
Numerically Controlled Oscillator (NCO)	1	
Comparator Module	2	
Direct Memory Access (DMA)	2	
Configurable Logic Cell (CLC)	4	
Control Area Network (CAN)	Yes	
Peripheral Module Disable (PMD)	Yes	
16-bit CRC with Scanner	Yes	
Programmable High/Low-Voltage Detect (HLVD)	Yes	
Resets (and Delays)	POR, Programmable BOR, RESET Instruction, Stack Overflow, Stack Underflow (PWRT, OST), $\overline{MCLR}$, WDT, $\overline{MEMV}$	
Instruction Set	81 Instructions; 87 with Extended Instruction Set enabled	
Maximum Operating Frequency	64 MHz	

Note 1: PORTE contains the single RE3 input-only pin.

9.4.4 SIMULTANEOUS LOW AND HIGH PRIORITY INTERRUPTS

When both high and low interrupts are active in the same instruction cycle (i.e., simultaneous interrupt events), both the high and the low priority requests are generated. The high priority ISR is serviced first before servicing the low priority interrupt see Figure 9-5.

FIGURE 9-5: INTERRUPT EXECUTION: SIMULTANEOUS LOW AND HIGH PRIORITY INTERRUPTS



REGISTER 9-32: IPR9: PERIPHERAL INTERRUPT PRIORITY REGISTER 9

U-0	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
—	CLC4IP	CCP4IP	CLC3IP	CWG3IP	CCP3IP	TMR6IP	TMR5GIP
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 **Unimplemented:** Read as '0'

bit 6 **CLC4IP:** CLC4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **CCP4IP:** CCP4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **CLC3IP:** CLC3 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CWG3IP:** CWG3 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP3IP:** CCP3 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR6IP:** TMR6IP Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR5GIP:** TMR5 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 10-2: 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	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	HC = Bit is cleared by hardware

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.

TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
VREGCON(1)	—	—	—	—	—	—	VREGPM	Reserved	166
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			167

Legend: — = unimplemented location, read as '0'. Shaded cells are not used in Power-Down mode.
Note 1: Not present in LF parts.

13.1.4 NVM UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the NVM from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- PFM Row Erase
- Write of PFM write latches to PFM memory
- Write of PFM write latches to User IDs
- Write to Data EEPROM Memory
- Write to Configuration Words

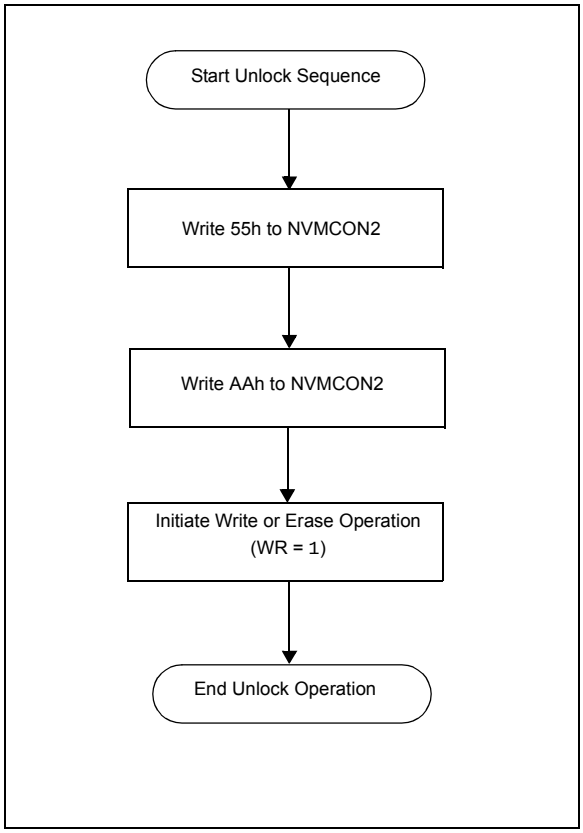
The unlock sequence consists of the following steps and must be completed in order:

- Write 55h to NVMCON2
- Write AAh to NMVCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 13-6: NVM UNLOCK SEQUENCE FLOWCHART



EXAMPLE 13-2: NVM UNLOCK SEQUENCE

```
BCF             INTCON0,GIE             ; Recommended so sequence is not interrupted
BANKSEL         NVMCON1
BSF             NVMCON1,WREN            ; Enable write/erase
MOVLW           55h                     ; Load 55h

MOVWF           NVMCON2                 ; Step 1: Load 55h into NVMCON2
MOVLW           AAh                     ; Step 2: Load W with AAh
MOVWF           NVMCON2                 ; Step 3: Load AAh into NVMCON2
BSF             INTCON1,WR              ; Step 4: Set WR bit to begin write/erase

BSF             INTCON0,GIE             ; Re-enable interrupts
```

Note 1: Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown. If the timing of the steps 1 to 4 is corrupted by an interrupt or a debugger Halt, the action will not take place.

2: Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

15.5.3.1 Clearing the SIRQEN bit

Clearing the SIRQEN bit (DMAxCON1 register) stops the sampling of external start interrupt triggers hence preventing further DMA Message transfers.

An example would be a communications peripheral with a level-triggered interrupt. The peripheral will continue to request data (because its buffer is empty) even though there is no more data to be moved. Disabling the SIRQEN bit prevents the DMA from processing these requests

15.5.3.2 Source/Destination Stop

The SSTP and DSTP bits (DMAxCON0 register) determine whether or not to disable the hardware triggers (SIRQEN = 0) once a DMA message has completed.

When the SSTP bit is set and the DMAxSCNT = 0, then the SIRQEN bit will be cleared. Similarly, when the DSTP bit is set and the DMAxDCNT = 0, the SIRQEN bit will be cleared.

Note:	The SSTP and DSTP bits are independent functions and do not depend on each other. It is possible for a message to be stopped by either counter at message end or both counters at message end.

15.6 Types of Hardware Triggers

The DMA has two different trigger inputs namely the Source trigger and the abort trigger. Each of these trigger sources is user configurable using the DMAxSIRQ and DMAxAIRQ registers.

Based on the source selected for each trigger, there are two types of requests that can be sent to the DMA.

- Edge triggers
- Level triggers

15.6.1 EDGE TRIGGER REQUESTS

An Edge request occurs only once when a given module interrupt requirements are true.

15.6.2 LEVEL TRIGGER REQUESTS

A level request is asserted as long as the condition that causes the interrupt is true.

15.7 Types of Data Transfers

Based on the memory access capabilities of the DMA (See Table 15-1), the following sections discuss the different types of data movement between the Source and Destination Memory regions.

- N: 1

This type of transfer is common when sending predefined data packets (such as strings) through a single interface point (such as communications modules transmit registers).

- N: N

This type of transfer is useful for moving information out of the Program Flash or Data EEPROM to SRAM for manipulation by the CPU or other peripherals.

- 1: N

This type of transfer is common when bridging two different modules data streams together (communications bridge).

- 1: N

This type of transfer is useful for moving information from a single data source into a memory buffer (communications receive registers).

15.8 DMA Interrupts

Each DMA has its own set of four interrupt flags, used to indicate a range of conditions during data transfers. The interrupt flag bits can be accessed using the corresponding PIR registers (Refer to the Interrupt Section).

15.8.1 DMA SOURCE COUNT INTERRUPT

The DMAxSCNTIF source count interrupt flag is set every time the DMAxSCNT<11:0> reaches zero and is reloaded to its starting value.

15.8.2 DMA DESTINATION COUNT INTERRUPT

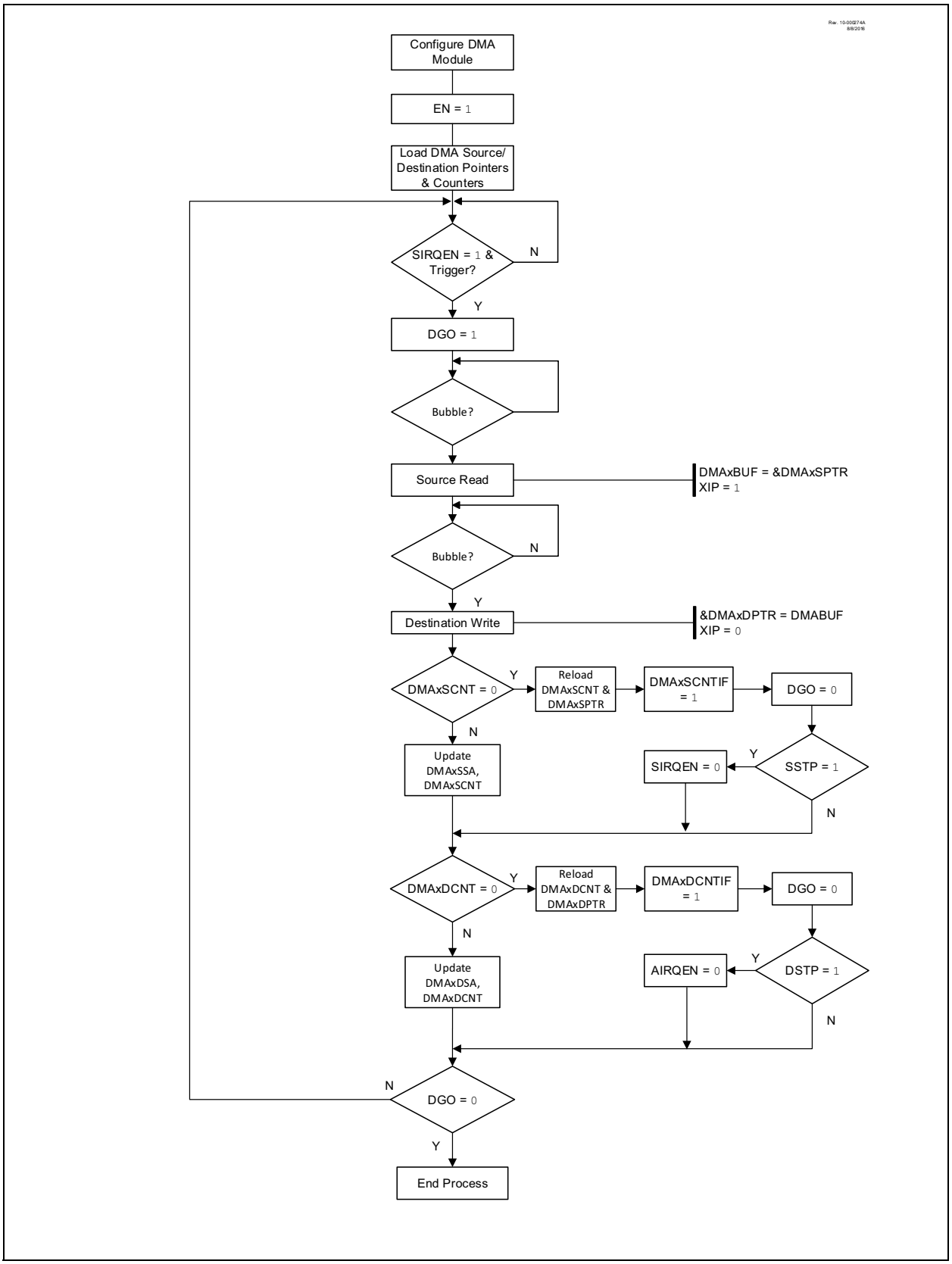
The DMAxDCNTIF destination count interrupt flag is set every time the DMAxDCNT<11:0> reaches zero and is reloaded to its starting value.

The DMA Source Count zero and Destination Count zero interrupts are used in conjunction to determine when to signal the CPU when the DMA Messages are completed.

15.8.3 ABORT INTERRUPT

The DMAxAIF abort interrupt flag is used to signal that the DMA has halted activity due to an abort signal from one of the abort sources. This is used to indicate that the transaction has been halted for some reason.

FIGURE 15-4: DMA OPERATION WITH HARDWARE TRIGGER



REGISTER 22-2: TxRST: TIMER2 EXTERNAL RESET SIGNAL SELECTION 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
—	—	—	RSEL<4:0>				
bit 7							bit 0

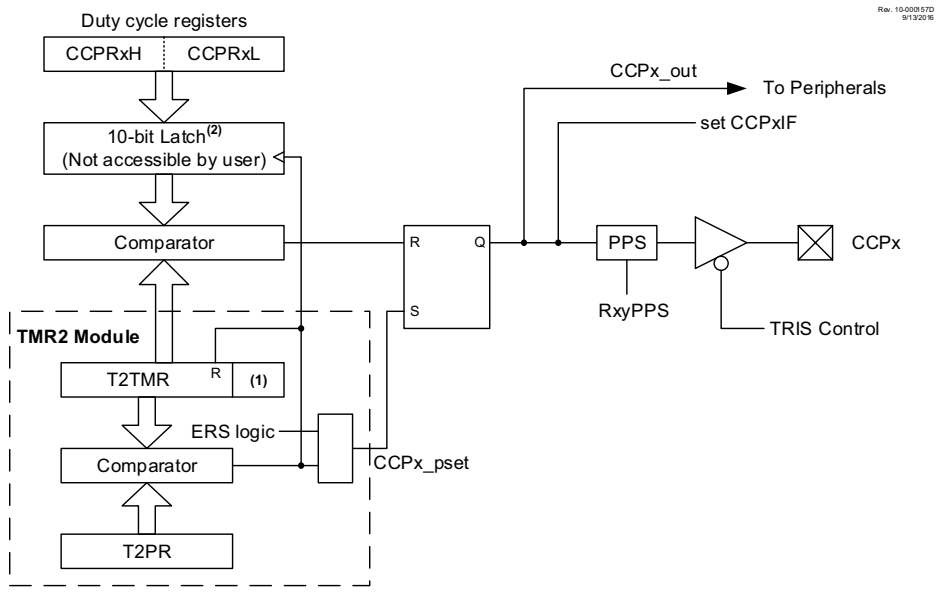
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RSEL<4:0>:** Timer2 External Reset Signal Source Selection bits

RSEL<4:0>	T2TMR	TMR4	TMR6
	Reset Source	Reset Source	Reset Source
11111-11001	Reserved	Reserved	Reserved
11000	UART2_tx_edge	UART2_tx_edge	UART2_tx_edge
10111	UART2_rx_edge	UART2_rx_edge	UART2_rx_edge
10110	UART1_tx_edge	UART1_tx_edge	UART1_tx_edge
10101	UART1_rx_edge	UART1_rx_edge	UART1_rx_edge
10100	CLC4_out	CLC4_out	CLC4_out
10011	CLC3_out	CLC3_out	CLC3_out
10010	CLC2_out	CLC2_out	CLC2_out
10001	CLC1_out	CLC1_out	CLC1_out
10000	ZCD_OUT	ZCD_OUT	ZCD_OUT
01111	CMP2OUT	CMP2OUT	CMP2OUT
01110	CMP1OUT	CMP1OUT	CMP1OUT
01101-01100	Reserved	Reserved	Reserved
01011	PWM8OUT	PWM8OUT	PWM8OUT
01010	PWM7OUT	PWM7OUT	PWM7OUT
01001	PWM6OUT	PWM6OUT	PWM6OUT
01000	PWM5OUT	PWM5OUT	PWM5OUT
00111	CCP4OUT	CCP4OUT	CCP4OUT
00110	CCP3OUT	CCP3OUT	CCP3OUT
00101	CCP2OUT	CCP2OUT	CCP2OUT
00100	CCP1OUT	CCP1OUT	CCP1OUT
00011	TMR6 postscaled	TMR6 postscaled	Reserved
00010	TMR4 postscaled	Reserved	TMR4 postscaled
00001	Reserved	T2TMR postscaled	T2TMR postscaled
00000	Pin selected by T2INPPS	Pin selected by T4INPPS	Pin selected by T6INPPS

FIGURE 23-4: SIMPLIFIED PWM BLOCK DIAGRAM



Notes: 1. 8-bit timer is concatenated with two bits generated by Fosc or two bits of the internal prescaler to create 10-bit time-base.
2. The alignment of the 10 bits from the CCPR register is determined by the CCPxFMT bit.

26.2.3.1 Direction Change in Full-Bridge Mode

In Full-Bridge mode, changing MODE<2:0> controls the forward/reverse direction. Changes to MODE<2:0> change to the new direction on the next rising edge of the modulated input.

A direction change is initiated in software by changing the MODE<2:0> bits of the CWGxCON0 register. The sequence is illustrated in Figure 26-8.

- The associated active output CWGxA and the inactive output CWGxC are switched to drive in the opposite direction.
- The previously modulated output CWGxD is switched to the inactive state, and the previously inactive output CWGxB begins to modulate.
- CWG modulation resumes after the direction-switch dead band has elapsed.

26.2.3.2 Dead-Band Delay in Full-Bridge Mode

Dead-band delay is important when either of the following conditions is true:

1. The direction of the CWG output changes when the duty cycle of the data input is at or near 100%, or
2. The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

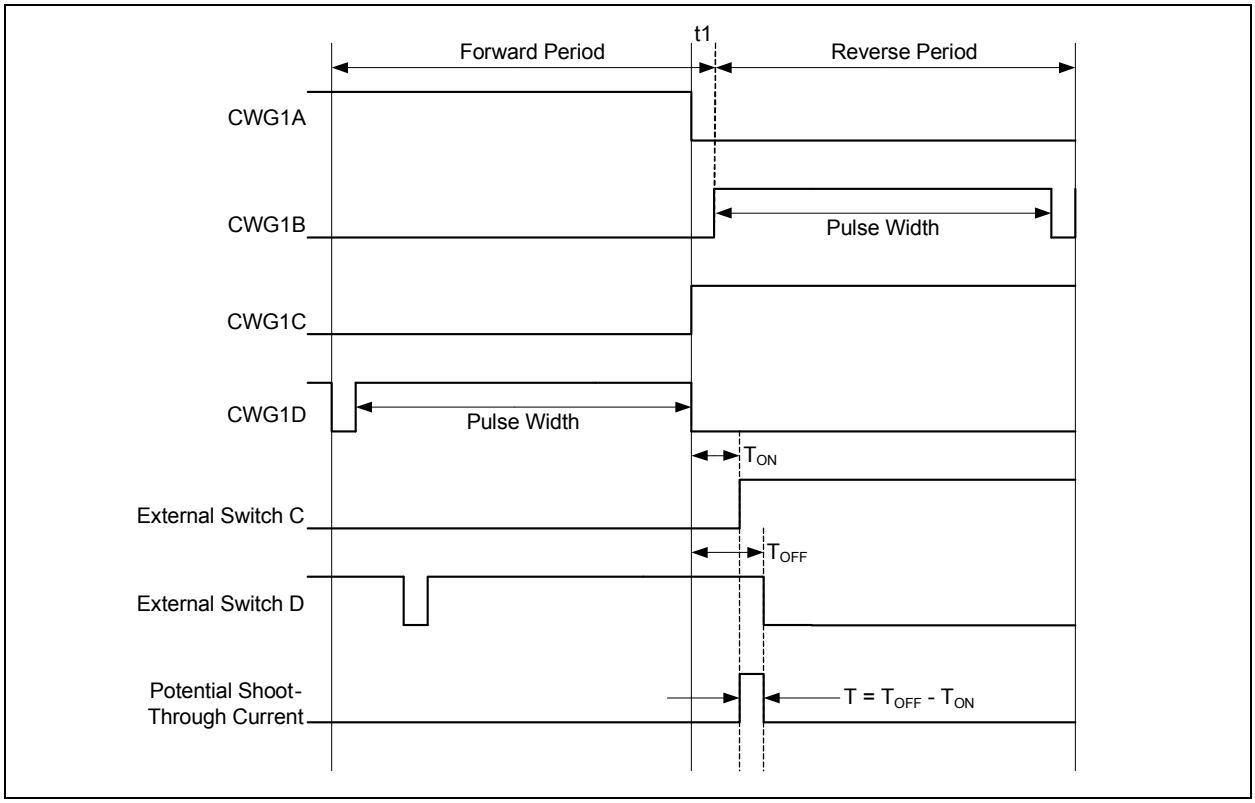
The dead-band delay is inserted only when changing directions, and only the modulated output is affected. The statically-configured outputs (CWGxA and CWGxC) are not afforded dead band, and switch essentially simultaneously.

Figure 26-8 shows an example of the CWG outputs changing directions from forward to reverse, at near 100% duty cycle. In this example, at time t1, the output of CWGxA and CWGxD become inactive, while output CWGxC becomes active. Since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current will flow through power devices QC and QD for the duration of 't'. The same phenomenon will occur to power devices QA and QB for the CWG direction change from reverse to forward.

When changing the CWG direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

1. Reduce the CWG duty cycle for one period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

FIGURE 26-8: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



REGISTER 26-5: CWGxSTR(1): CWG STEERING CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OVRD	OVRC	OVRB	OVRA	STRD(2)	STRC(2)	STRB(2)	STRA(2)
bit 7							bit 0

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

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

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

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

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

bit 3 **STRD:** Steering Enable bit D(2)
- 1 = CWGxD output has the CWG data input waveform with polarity control from POLD bit
- 0 = CWGxD output is assigned to value of OVRD bit

bit 2 **STRC:** Steering Enable bit C(2)
- 1 = CWGxC output has the CWG data input waveform with polarity control from POLC bit
- 0 = CWGxC output is assigned to value of OVRC bit

bit 1 **STRB:** Steering Enable bit B(2)
- 1 = CWGxB output has the CWG data input waveform with polarity control from POLB bit
- 0 = CWGxB output is assigned to value of OVRB bit

bit 0 **STRA:** Steering Enable bit A(2)
- 1 = CWGxA output has the CWG data input waveform with polarity control from POLA bit
- 0 = CWGxA output is assigned to value of OVRA bit

Note 1: The bits in this register apply only when MODE<2:0> = 00x (Register 26-1, Steering modes).

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

29.10 Register Definitions: ZCD Control

REGISTER 29-1: ZCDCON: ZERO-CROSS DETECT CONTROL REGISTER

R/W-0/0	U-0	R-x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
SEN	—	OUT	POL	—	—	INTP	INTN
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 **SEN:** Zero-Cross Detect Software Enable bit
This bit is ignored when ZCDSEN Configuration bit is set.
1= Zero-cross detect is enabled.
0= Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

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

bit 5 **OUT:** Zero-Cross Detect Data Output bit
ZCDPOL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current
ZCDPOL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current

bit 4 **POL:** Zero-Cross Detect Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **INTP:** Zero-Cross Detect Positive-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCD_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCD_output transition

bit 0 **INTN:** Zero-Cross Detect Negative-Going Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCD_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCD_output transition

TABLE 29-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
ZCDCON	SEN	—	OUT	POL	—	—	INTP	INTN	448

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

TABLE 30-2: MD1SRC SELECTION MUX CONNECTIONS

MS<4:0>		Connection
1 1111 - 1 1000	31 - 24	Reserved
1 0111	23	CAN_tx0
1 0110	22	SPI1 SDO
1 0101	21	Reserved
1 0100	20	UART2 TX
1 0011	19	UART1 TX
1 0010	18	CLC4 OUT
1 0001	17	CLC3 OUT
1 0000	16	CLC2 OUT
0 1111	15	CLC1 OUT
0 1110	14	CMP2 OUT
0 1101	13	CMP1 OUT
0 1100	12	NCO1 OUT
0 1011	11	Reserved
0 1010	10	Reserved
0 1001	9	PWM8 OUT
0 1000	8	PWM7 OUT
0 0111	7	PWM6 OUT
0 0110	6	PWM5 OUT
0 0101	5	CCP4 OUT
0 0100	4	CCP3 OUT
0 0011	3	CCP2 OUT
0 0010	2	CCP1 OUT
0 0001	1	DSM1 BIT
0 0000	0	Pin selected by MDSRCPPS

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH DATA SIGNAL MODULATOR MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
MD1CON0	EN	—	OUT	OPOL	—	—	—	BIT	455
MD1CON1	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	456
MD1CARH	—	—	—	—	—	CHS<2:0>			457
MD1CARL	—	—	—	—	—	CLS<2:0>			457
MDSRC	—	—	—	—	SRCS<3:0>				458

Legend: — = unimplemented, read as '0'. Shaded cells are not used in the Data Signal Modulator mode.

33.5.12 MASTER RECEPTION IN 10-BIT ADDRESSING MODE

This section describes the sequence of events for the I^2C module configured as an I^2C master in 10-bit Addressing mode and is receiving data. Figure 33-22 is used as a visual reference for this description.

1. Master software loads high address byte in I2CxADB1 and low address byte in I2CxADB0 for write and sets restart enable (RSTEN) bit.
2. Master software sets START bit.
3. Master hardware waits for BFRE bit to be set; then shifts out start, high address and waits for acknowledge.
4. If slave responds with a NACK, master hardware sends Stop and ends communication.
5. If slave responds with ACK, master hardware shifts out the low address.
6. If the transmit buffer empty flag (TXBE) is set and I2CxCNT! = 0, the clock is stretched on 8th falling SCL edge. Allowing master software writes next data to I2CxTXB.
7. Master hardware sends 9th SCL pulse for ACK from slave and loads the shift register from I2CxTXB.
8. If slave responds with a NACK, master hardware sends Stop and ends communication.
9. If slave responds with an ACK and I2CxCNT = 0, master hardware sets MDR bit, go to Step 11.
10. If slave responds with an ACK and I2CxCNT! = 0, master hardware outputs data in shift register on SDA and waits for ACK from slave. Go to step 4.
11. Master software loads I2CxADB0 for read, and I2CCNT with the number of bytes to be received in the current transaction.
12. Master software sets Start bit.
13. Master hardware shifts out Restart and high address with R/W = 1.
14. Master sends out 9th SCL pulse for ACK from slave.
15. If slave responds with a NACK, master hardware sends Stop or sets MDR (RSEN bit).
16. If slave responds with an ACK, master hardware shifts 7 bits of data into the shift register from the slave.
17. If the receive buffer full flag (RXBF) is set, clock is stretched on seventh falling SCL edge.
18. Master software can clear clock stretching by reading the previous data in the receive buffer.
19. Master hardware shifts 8th bit of data into the shift register from slave and loads it into I2CxRXB.
20. Master software reads data from I2CxRXB register.
21. If I2CxCNT! = 0, master hardware clocks out ACKDT as ACK value to slave.
22. If I2CxCNT = 0, master hardware clocks out ACKCNT as ACK value to slave
23. Go to step 4.

REGISTER 33-12: I2CxADR0: I²C ADDRESS 0 REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
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 HC = Hardware clear

bit 7-0 **ADR<7-0>:** Address 1 bits

MODE<2:0> = 00x | 11x - 7-bit Slave/Multi-Master Modes

ADR0<7:1>:7-bit Slave Address

ADR0<0>: Unused in this mode; bit state is a "don't care"

MODE<2:0> = 01x - 10-bit Slave Modes

ADR0<7:0>: Eight Least Significant bits of 10-bit address 0

REGISTER 33-15: I2CXADR3: I²C ADDRESS 3 REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0
ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8	—
bit 15							bit 8
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	—
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 HC = Hardware clear

bit 7-0 **ADR<7-0>:** Address 3 bits

MODE<2:0> = 000 | 110 - 7-bit Slave/Multi-Master Modes

ADR<7:1>:7-bit Slave Address

ADR<0>: Unused in this mode; bit state is a "don't care"

MODE<2:0> = 001 | 111 - 7-bit Slave/Multi-Master Mode with Masking

MSK1<7:1>:7-bit Slave Address

MSK1<0>: Unused in this mode; bit state is a "don't care"

MODE<2:0> = 010 - 10-Bit Slave Mode

ADR<14-10>: Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, these bit values are compared by hardware to the received data to determine a match. It is up to the user to set these bits as '11110'

ADR<9-8>: Two Most Significant bits of 10-bit address

MODE<2:0> = 011 - 10-Bit Slave Mode with Masking

MSK0<14-8>: The received address byte, bit *n,* is compared to I2CxADR0 to detect I²C address match

34.14 CAN Interrupts

The module has several sources of interrupts. Each of these interrupts can be individually enabled or disabled. The PIR5 register contains interrupt flags. The PIE5 register contains the enables for the eight main interrupts. A special set of read-only bits in the CANSTAT register, the ICODE bits, can be used in combination with a jump table for efficient handling of interrupts.

All interrupts have one source, with the exception of the error interrupt and buffer interrupts in Mode 1 and 2. Any of the error interrupt sources can set the error interrupt flag. The source of the error interrupt can be determined by reading the Communication Status register, COMSTAT. In Mode 1 and 2, there are two interrupt enable/disable and flag bits – one for all transmit buffers and the other for all receive buffers.

The interrupts can be broken up into two categories: receive and transmit interrupts.

The receive related interrupts are:

- Receive Interrupts
- Wake-up Interrupt
- Receiver Overrun Interrupt
- Receiver Warning Interrupt
- Receiver Error-Passive Interrupt

The transmit related interrupts are:

- Transmit Interrupts
- Transmitter Warning Interrupt
- Transmitter Error-Passive Interrupt
- Bus-Off Interrupt

34.14.1 INTERRUPT CODE BITS

To simplify the interrupt handling process in user firmware, the ECAN module encodes a special set of bits. In Mode 0, these bits are ICODE<3:1> in the CANSTAT register. In Mode 1 and 2, these bits are EICODE<4:0> in the CANSTAT register. Interrupts are internally prioritized such that the higher priority interrupts are assigned lower values. Once the highest priority interrupt condition has been cleared, the code for the next highest priority interrupt that is pending (if any) will be reflected by the ICODE bits (see Table 34-4). Note that only those interrupt sources that have their associated interrupt enable bit set will be reflected in the ICODE bits.

In Mode 2, when a receive message interrupt occurs, the EICODE bits will always consist of '10000'. User firmware may use FIFO Pointer bits to actually access the next available buffer.

34.14.2 TRANSMIT INTERRUPT

When the transmit interrupt is enabled, an interrupt will be generated when the associated transmit buffer becomes empty and is ready to be loaded with a new message. In Mode 0, there are separate interrupt enable/disable and flag bits for each of the three dedicated transmit buffers. The TXBnIF bit will be set to indicate the source of the interrupt. The interrupt is cleared by the MCU, resetting the TXBnIF bit to a '0'. In Mode 1 and 2, all transmit buffers share one interrupt enable/disable bit and one flag bit. In Mode 1 and 2, TXBnIE in PIE5 and TXBnIF in PIR5 indicate when a transmit buffer has completed transmission of its message. TXBnIF, TXBnIE and TXBnIP in PIR5, PIE5 and IPR5, respectively, are not used in Mode 1 and 2. Individual transmit buffer interrupts can be enabled or disabled by setting or clearing TXBnIE and B0IE register bits. When a shared interrupt occurs, user firmware must poll the TXREQ bit of all transmit buffers to detect the source of interrupt.

34.14.3 RECEIVE INTERRUPT

When the receive interrupt is enabled, an interrupt will be generated when a message has been successfully received and loaded into the associated receive buffer. This interrupt is activated immediately after receiving the End-of-Frame (EOF) field.

In Mode 0, the RXBnIF bit is set to indicate the source of the interrupt. The interrupt is cleared by the MCU, resetting the RXBnIF bit to a '0'.

In Mode 1 and 2, all receive buffers share RXBnIE, RXBnIF and RXBnIP in PIE5, PIR5 and IPR5, respectively. Individual receive buffer interrupts can be controlled by the TXBnIE and BIE0 registers. In Mode 1, when a shared receive interrupt occurs, user firmware must poll the RXFUL bit of each receive buffer to detect the source of interrupt. In Mode 2, a receive interrupt indicates that the new message is loaded into FIFO. FIFO can be read by using FIFO Pointer bits, FP.

34.15.5 CAN MODULE I/O CONTROL REGISTER

This register controls the operation of the CAN module's I/O pins in relation to the rest of the microcontroller.

REGISTER 34-55: CIOCON: CAN I/O CONTROL REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
TX1SRC	—	—	—	—	—	—	CLKSEL
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 **TX1SRC:** CAN_tx1 Signal Data Source bit
1 = CAN_tx1 signal will output the CAN clock
0 = CAN_tx1 signal will output $\overline{\text{CANTX}}$

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **CLKSEL:** CAN Clock Source Selection bit
1 = CAN clock is sourced by the clock selected by the FEXTOSC Configuration bit field, regardless of system clock(1)
0 = CAN clock is sourced from the system clock

Note 1: When CLKSEL = 1, the clock supplied by FEXTOSC must be less than or equal to the system clock. If the CAN clock is greater than the system clock, unexpected behavior will occur.

36.2.1 CALIBRATION

36.2.1.1 Single-Point Calibration

Single-point calibration is performed by application software using Equation 36-1 and the assumed Mt. A reading of VTSENSE at a known temperature is taken, and the theoretical temperature is calculated by temporarily setting TOFFSET = 0. Then TOFFSET is computed as the difference of the actual and calculated temperatures. Finally, TOFFSET is stored in nonvolatile memory within the device, and is applied to future readings to gain a more accurate measurement.

36.2.1.2 Higher-Order Calibration

If the application requires more precise temperature measurement, additional calibrations steps will be necessary. For these applications, two-point or three-point calibration is recommended.

Note 1:	The TOFFSET value may be determined by the user with a temperature test.
2:	Although the measurement range is -40°C to +125 °C, due to the variations in offset error, the single-point uncalibrated calculated TSENSE value may indicate a temperature from -140°C to +225°C, before the calibration offset is applied.
3:	The User must take into consideration self-heating of the device at different clock frequencies and output pin loading. For package related thermal characteristics information, refer to Table 45-6.

36.2.2 TEMPERATURE RESOLUTION

The resolution of the ADC reading, Ma (°C/count), depends on both the ADC resolution N and the reference voltage used for conversion, as shown in Equation 36-2. It is recommended to use the smallest VREF value, such as 2.048 FVR reference voltage, instead of VDD.

Note:	Refer to Table 45-17 for FVR reference voltage accuracy.

EQUATION 36-2: TEMPERATURE RESOLUTION (°C/LSb)

$$Ma = \frac{V_{REF}}{2^N} \times Mt$$

$$Ma = \frac{\frac{V_{REF}}{2^N}}{Mv}$$

Where:

Mv = sensor voltage sensitivity (V/°C)

VREF = Reference voltage of the ADC module (in Volts)

N = Resolution of the ADC

The typical Mv value for a single diode is approximately -1.267 to -1.32 mV/C.

The typical Mv value for a stack of two diodes (Low Range setting) is approximately -2.533 mV/C.

The typical Mv value for a stack of three diodes (High range setting) is approximately -3.8 mV/C.

36.3 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait a certain time for the ADC value to settle, after the ADC input multiplexer is connected to the temperature indicator output, before the conversion is performed.

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
37A0h	RXF0SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	630
379Fh	CANCON_RO4	CANCON_RO4								603
379Eh	CANSTAT_RO4	CANSTAT_RO4								604
379Dh	B5D7	B5D7								627
379Ch	B5D6	B5D6								627
379Bh	B5D5	B5D5								627
379Ah	B5D4	B5D4								627
3799h	B5D3	B5D3								627
3798h	B5D2	B5D2								627
3797h	B5D1	B5D1								627
3796h	B5D0	B5D0								627
3795h	B5DLC	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	628
3795h	B5DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	629
3794h	B5EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	626
3793h	B5EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	626
3792h	B5SIDL	SID2	SID1	SID0	SRR	EXIDE	—	EID17	EID16	625
3791h	B5SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	624
3790h	B5CON	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	622
3790h	B5CON	TXBIF	TXABT	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	623
378Fh	CANCON_RO5	CANCON_RO5								603
378Eh	CANSTAT_RO5	CANSTAT_RO5								604
378Dh	B4D7	B4D7								627
378Ch	B4D6	B4D6								627
378Bh	B4D5	B4D5								627
378Ah	B4D4	B4D4								627
3789h	B4D3	B4D3								627
3788h	B4D2	B4D2								627
3787h	B4D1	B4D1								627
3786h	B4D0	B4D0								627
3785h	B4DLC	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	628
3785h	B4DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	629
3784h	B4EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	626
3783h	B4EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	626
3782h	B4SIDL	SID2	SID1	SID0	SRR	EXIDE	—	EID17	EID16	625
3781h	B4SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	624
3780h	B4CON	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	622
3780h	B4CON	TXBIF	TXABT	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	623
377Fh	CANCON_RO6	CANCON_RO6								603
377Eh	CANSTAT_RO6	CANSTAT_RO6								604
377Dh	B3D7	B3D7								627
377Ch	B3D6	B3D6								627
377Bh	B3D5	B3D5								627
377Ah	B3D4	B3D4								627
3779h	B3D3	B3D3								627
3778h	B3D2	B3D2								627
3777h	B3D1	B3D1								627
3776h	B3D0	B3D0								627
3775h	B3DLC	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	628
3775h	B3DLC	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	629
3774h	B3EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	626

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