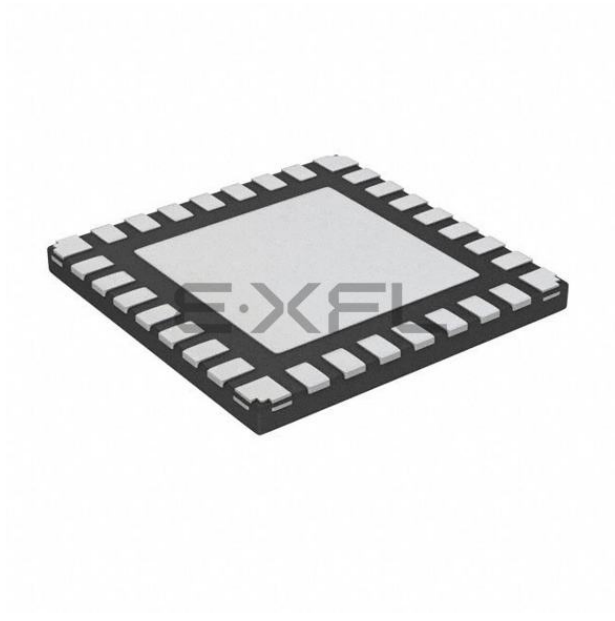


Welcome to E-XFL.COM

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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UQFN Exposed Pad
Supplier Device Package	28-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k83-e-mx

4.3 Register Definitions: Stack Pointer

REGISTER 4-1: TOSU: TOP-OF-STACK UPPER BYTE

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TOS<20:16>				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

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

bit 4-0 **TOS<20:16>:** Top-of-Stack Location bits

REGISTER 4-2: TOSH: TOP-OF-STACK HIGH BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TOS<15:8>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **TOS<15:8>:** Top-of-Stack Location bits

REGISTER 4-3: TOSL: TOP-OF-STACK LOW BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TOS<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **TOS<7:0>:** Top-of-Stack Location bits

5.5 Device ID and Revision ID

The 16-bit device ID word is located at 3F FFFEh and the 16-bit revision ID is located at 3F FFFCh. These locations are read-only and cannot be erased or modified.

Development tools, such as device programmers and debuggers, may be used to read the Device ID, Revision ID and Configuration Words. Refer to **13.0 "Nonvolatile Memory (NVM) Control"** for more information on accessing these locations.

5.6 Register Definitions: Device ID and Revision ID

REGISTER 5-11: DEVICE ID: DEVICE ID REGISTER

R	R	R	R	R	R	R	R
DEV<15:8>							
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	'1' = Bit is set	0' = Bit is cleared	x = Bit is unknown

bit 15-0 **DEV<15:0>:** Device ID bits

Device	Device ID
PIC18F25K83	6EE0h
PIC18F26K83	6EC0h
PIC18LF25K83	6F20h
PIC18LF26K83	6F00h

5.7.1 MICROCHIP UNIQUE IDENTIFIER (MUI)

The PIC18(L)F25/26K83 devices are individually encoded during final manufacturing with a Microchip Unique Identifier, or MUI. The MUI cannot be user-erased. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is a required. It may also be used by the application manufacturer for a number of functions that require unverified unique identification, such as:

- Tracking the device
- Unique serial number

The MUI consists of six program words. When read together, these fields form a unique identifier. The MUI is stored in nine read-only locations, located between 3F0000h to 3F000Fh in the DIA space. Table 5-3 lists the addresses of the identifier words.

Note:	For applications that require verified unique identification, contact your Microchip Technology sales office to create a Serialized Quick Turn Programming^{SM} option.

5.7.2 EXTERNAL UNIQUE IDENTIFIER (EUI)

The EUI data is stored at locations 3F0010h to 3F0023h in the Program Memory region. This region is an optional space for placing application specific information. The data is coded per customer requirements during manufacturing.

Note:	Data is stored in this address range on receiving a request from the customer. The customer may contact the local sales representative, or Field Applications Engineer, and provide them the unique identifier information that is supposed to be stored in this region.

5.7.3 ANALOG-TO-DIGITAL CONVERSION DATA OF THE TEMPERATURE SENSOR

The purpose of the Temperature Sensor module is to provide a temperature-dependent voltage that can be measured by an analog module, see **Section 36.0 “Temperature Indicator Module”**.

The DIA table contains the internal ADC measurement values of the Temperature sensor for Low and High range at fixed points of reference. The values are measured during test and are unique to each device. The measurement data is stored in the DIA memory region as hexadecimal numbers corresponding to the ADC conversion result. The calibration data can be used to plot the approximate sensor output voltage, V_{TSENSE} vs. Temperature curve without having to make calibration measurements in the application. For more information on the operation of the Temperature Sensor, refer to **Section 36.0 “Temperature Indicator Module”**.

- **TSLR1-TSLR3**: Address 3F0024h to 3F0029h store the measurements for the low-range setting of the Temperature Sensor at V_{DD} = 3V.
- **TSHR1-TSHR3**: Address 3F002Ah to 3F002Fh store the measurements for the High Range setting of the Temperature Sensor at V_{DD} = 3V.
- The stored measurements are made by the device ADC using the internal V_{REF} = 2.048V.

9.2 Interrupt Vector Table (IVT)

The interrupt controller supports an Interrupt Vector Table (IVT) that contains the vector address location for each interrupt request source.

The Interrupt Vector Table (IVT) resides in program memory, starting at address location determined by the IVTBASE registers; refer to Registers 9-33 through 9-35 for details. The IVT contains 68 vectors, one for each source of interrupt. Each interrupt vector location contains the starting address of the associated Interrupt Service Routine (ISR).

The MVECEN bit in Configuration Word 2L controls the availability of the vector table.

9.2.1 INTERRUPT VECTOR TABLE BASE ADDRESS (IVTBASE)

The start address of the vector table is user programmable through the IVTBASE registers. The user must ensure the start address is such that it can encompass the entire vector table inside the program memory.

Each vector address is a 16-bit word (or two address locations on PIC18 devices). So for n interrupt sources, there are 2n address locations necessary to hold the table starting from IVTBASE as the first location. So the staring address of IVTBASE should be chosen such that the address range form IVTBASE to (IVTBASE +2n-1) can be encompassed inside the program flash memory.

For example, the K42 devices have the highest vector number: 81. So IVTBASE should be chosen such that (IVTBASE + 0xA1) is less than the last memory location in program flash memory.

A programmable vector table base address is useful in situations to switch between different sets of vector tables, depending on the application. It can also be used when the application program needs to update the existing vector table (vector address values).

Note:	It is required that the user assign an even address to the IVTBASE register for correct operation.

9.2.2 INTERRUPT VECTOR TABLE CONTENTS

MVECEN = 0

When MVECEN = 0, the address location pointed by the IVTBASE registers has a GOTO instruction for a high priority interrupt. Similarly, the corresponding low priority vector location also has a GOTO instruction, which is executed in case of a low priority interrupt.

MVECEN = 1

When MVECEN = 1, the value in the vector table of each interrupt, points to the address location of the first instruction of the interrupt service routine.

ISR Location = Interrupt Vector Table entry << 2.

9.2.3 INTERRUPT VECTOR TABLE (IVT) ADDRESS CALCULATION

MVECEN = 0

When the MVECEN bit in Configuration Word 2L (Register 5-3) is cleared, the address pointed by IVTBASE registers is used as the high priority interrupt vector address. The low priority interrupt vector address is offset eight instruction words from the address in IVTBASE registers.

For PIC18 devices the IVTBASE registers default to 00 0008h, the high priority interrupt vector address will be 00 0008h and the low priority interrupt vector address will be 00 0018h.

MVECEN = 1

Each interrupt has a unique vector number associated with it as defined in Table 9-2. This vector number is used for calculating the location of the interrupt vector for a particular interrupt source.

Interrupt Vector Address = IVTBASE + (2*Vector Number).

This calculated Interrupt Vector Address value is stored in the IVTAD<20:0> registers when an interrupt is received (Registers 9-36 through 9-38).

User-assigned software priority assigned using the IPRx registers does not affect address calculation and is only used to resolve concurrent interrupts.

If for any reason the address of the ISR could not be fetched from the vector table, it will cause the system to reset and clear the memory execution violation flag (MEMV bit) in PCON1 register (Register 6-3). This occurs due to any one of the following:

- The entry for the interrupt in the vector table lies outside the executable PFM area (SAF area is non-executable when SAFEN = 1).
- ISR pointed by the vector table lies outside the executable PFM area (SAF area is non-executable when SAFEN = 1).

14.8 Scanner Module Overview

The Scanner allows segments of the Program Flash Memory or Data EEPROM, to be read out (scanned) to the CRC Peripheral. The Scanner module interacts with the CRC module and supplies it data one word at a time. Data is fetched from the address range defined by SCANLADR registers up to the SCANHADR registers.

The Scanner begins operation when the SGO bit is set (SCANCON0 Register) and ends when either SGO is cleared by the user or when SCANLADR increments past SCANHADR. The SGO bit is also cleared by clearing the EN bit (CRCCON0 register).

14.9 Configuring the Scanner

The scanner module may be used in conjunction with the CRC module to perform a CRC calculation over a range of program memory or Data EEPROM addresses. In order to set up the scanner to work with the CRC, perform the following steps:

1. Set up the CRC module (See **Section 14.7 "Configuring the CRC"**) and enable the Scanner module by setting the EN bit in the SCANCON0 register.
2. Choose which memory region the Scanner module should operate on and set the MREG bit of the SCANCON0 register appropriately.
3. If trigger is used for scanner operation, set the TRIGEN bit of the SCANCON0 register and select the trigger source using SCANTRIG register. Select the trigger source using SCANTRIG register and then set the TRIGEN bit of the SCANCON0 register. See Table 14-1 for Scanner Operation.
4. If Burst mode of operation is desired, set the BURSTMD bit (SCANCON0 register). See Table 14-1 for Scanner Operation.
5. Set the SCANLADRL/H/U and SCANHADRL/H/U registers with the beginning and ending locations in memory that are to be scanned.
6. Select the priority level for the Scanner module (See **Section 3.1 "System Arbitration"**) and lock the priorities (See **Section 3.1.1 "Priority Lock"**).
7. Both CRCEN and CRCGO bits must be enabled to use the scanner. Setting the SGO bit will start the scanner operation.

14.10 Scanner Interrupt

The scanner will trigger an interrupt when the SCANLADR increments past SCANHADR. The SCANIF bit can only be cleared in software.

14.11 Scanning Modes

The interaction of the scanner with the system operation is controlled by the priority selection in the System Arbiter (see **Section 3.2 "Memory Access Scheme"**). Additionally, BURSTMD and TRIGEN also determine the operation of the Scanner.

14.11.1 TRIGEN = 0, BURSTMD = 0

In this case, the memory access request is granted to the scanner if no other higher priority source is requesting access.

All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

14.11.2 TRIGEN = 1, BURSTMD = 0

In this case, the memory access request is generated when the CRC module is ready to accept.

The memory access request is granted to the scanner if no other higher priority source is requesting access. All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.
The memory access request is granted to the scanner if no other higher priority source is requesting access. All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

14.11.3 TRIGEN = x, BURSTMD = 1

In this case, the memory access is always requested by the scanner.

The memory access request is granted to the scanner if no other higher priority source is requesting access. The memory access cycles will not be granted to lower priority sources than the scanner until it completes operation i.e., SGO = 0 (SCANCON0 register)

Note:	If TRIGEN = 1 and BURSTMD = 1, the user should ensure that the trigger source is active for the Scanner operation to complete.

15.8.4 OVERRUN INTERRUPT

When the DMA receives a trigger to start a new message before the current message is completed, then the DMAxORIF Overrun interrupt flag is set.

This condition indicates that the DMA is being requested before its current transaction is finished. This implies that the active DMA may not be able to keep up with the demands from the peripheral module being serviced, which may result in data loss.

The DMAxORIF flag being set does not cause the current DMA transfer to terminate.

The Overrun interrupt is only available for trigger sources that are edge based and not available for sources that are level-based. Therefore a level-based interrupt source does not trigger a DMA overrun error due to the potential latency issues in the system.

An example of an interrupt that could use the overrun interrupt would be a timer overflow (or period match) interrupt. This event only happens every time the timer rolls over and is not dependent on any other system conditions.

An example of an interrupt that does not allow the overrun interrupt would be the UARTTX buffer. The UART will continue to assert the interrupt until the DMA is able to process the MSG. Due to latency issues, the DMA may not be able to service an empty buffer immediately, but the UART continues to assert its transmit interrupt until it is serviced. If overrun was allowed in this case, the overrun would occur almost immediately as the module samples the interrupt sources every instruction cycle.

15.9 DMA Setup and Operation

The following steps illustrate how to configure the DMA for data transfer:

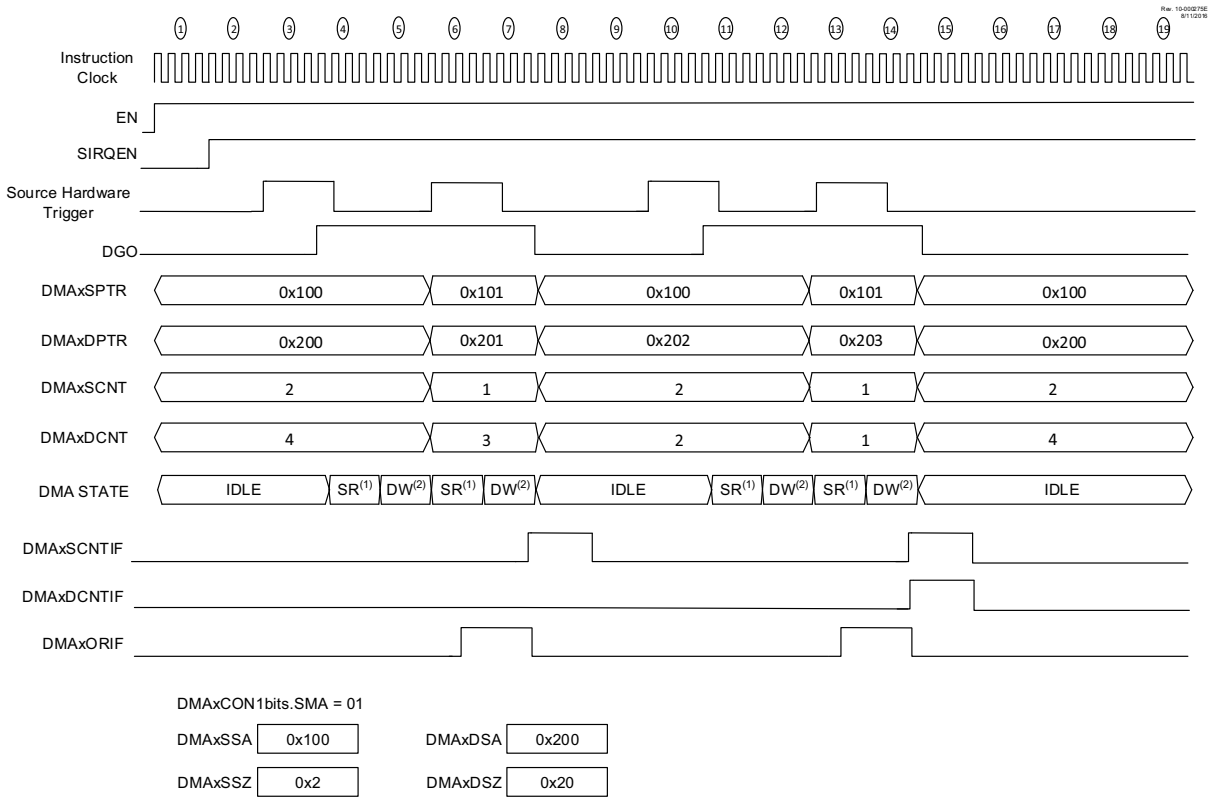
1. Program the appropriate Source and Destination addresses for the transaction into the DMAxSSA and DMAxDSA registers
2. Select the source memory region that is being addressed by DMAxSSA register, using the SMR<1:0> bits.
3. Program the SMODE and DMODE bits to select the addressing mode.
4. Program the Source size DMAxSSZ and Destination size DMAxDSZ registers with the number of bytes to be transferred. It is recommended for proper operation that the size registers be a multiple of each other.
5. If the user desires to disable data transfers once the message has completed, then the SSTP and DSTP bits in DMAxCON0 register need to be set.(see **Section 15.5.3.2 “Source/Destination Stop”**).
6. If using hardware triggers for data transfer, setup the hardware trigger interrupt sources for the starting and aborting DMA transfers (DMAxSIRQ and DMAxAIRQ), and set the corresponding interrupt request enable bits (SIRQEN and AIRQEN).
7. Select the priority level for the DMA (see **Section 3.1 “System Arbitration”**) and lock the priorities (see **Section 3.1.1 “Priority Lock”**)
8. Enable the DMA (DMAxCON1bits. EN = 1)
9. If using software control for data transfer, set the DGO bit, else this bit will be set by the hardware trigger.

Once the DMA is set up, the following flow chart describes the sequence of operation when the DMA uses hardware triggers and utilizes the unused CPU cycles (bubble) for DMA transfers.

15.9.5 OVERRUN INTERRUPT

The Overrun Interrupt flag is set if the DMA receives a trigger to start a new message before the current message is completed.

FIGURE 15-9: OVERRUN INTERRUPT



Note 1: SR – Source Read

2: DW – Destination Write

19.0 PERIPHERAL MODULE DISABLE (PMD)

Sleep, Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume some amount of power. There may be cases where the application needs what these modes do not provide: the ability to allocate limited power resources to the CPU while eliminating power consumption from the peripherals.

The PIC18(L)F25/26K83 family addresses this requirement by allowing peripheral modules to be selectively enabled or disabled, placing them into the lowest possible power mode.

All modules are ON by default following any Reset.

19.1 Disabling a Module

Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset.
- Any SFR becomes “unimplemented”
  - Writing is disabled
  - Reading returns 00h
- I/O functionality is prioritized as per **Section 16.2, I/O Priorities**
- All associated Input Selection registers are also disabled

19.2 Enabling a Module

When the PMD register bit is cleared, the module is re-enabled and will be in its Reset state (Power-on Reset). SFR data will reflect the POR Reset values.

Depending on the module, it may take up to one full instruction cycle for the module to become active. There should be no interaction with the module (e.g., writing to registers) for at least one instruction after it has been re-enabled.

19.3 Effects of a Reset

Following any Reset, each control bit is set to ‘0’, enabling all modules.

19.4 System Clock Disable

Setting SYSCMD (PMD0, Register 19-1) disables the system clock (FOSC) distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit.

REGISTER 19-8: PMD7: PMD CONTROL REGISTER 7

R/W-0/0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
CANMD	—	—	—	—	—	DMA2MD	DMA1MD
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 **CANMD:** Disable CAN Module bit
1 = CAN module disabled
0 = CAN module enabled

bit 6-2 **Unimplemented:** Read as '0'

bit 1 **DMA2MD:** Disable DMA2 Module bit
1 = DMA2 module disabled
0 = DMA2 module enabled

bit 0 **DMA1MD:** Disable DMA1 Module bit
1 = DMA1 module disabled
0 = DMA1 module enabled

TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH PERIPHERAL MODULE DISABLE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PMD0	SYSCMD	FVRMD	HLVDMD	CRCMD	SCANMD	NVMMD	CLKRMD	IOCMD	275
PMD1	NCO1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD	276
PMD2	—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	ZCDMD	277
PMD3	PWM8MD	PWM7MD	PWM6MD	PWM5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	278
PMD4	CWG3MD	CWG2MD	CWG1MD	—	—	—	—	—	279
PMD5	—	—	U2MD	U1MD	—	SPI1MD	I2C2MD	I2C1MD	280
PMD6	—	SMT2MD	SMT1MD	CLC4MD	CLC3MD	CLC2MD	CLC1MD	DSMMD	280
PMD7	CANMD	—	—	—	—	—	DMA2MD	DMA1MD	282

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by peripheral module disable.

REGISTER 22-5: TxCON: TIMERx CONTROL REGISTER

R/W/HC-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
ON	CKPS<2:0>			OUTPS<3:0>			
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 **ON:** Timerx On bit(1)

1 = Timerx is On
0 = Timerx is Off: all counters and state machines are reset

bit 6-4 **CKPS<2:0>:** Timerx-type Clock Prescale Select bits

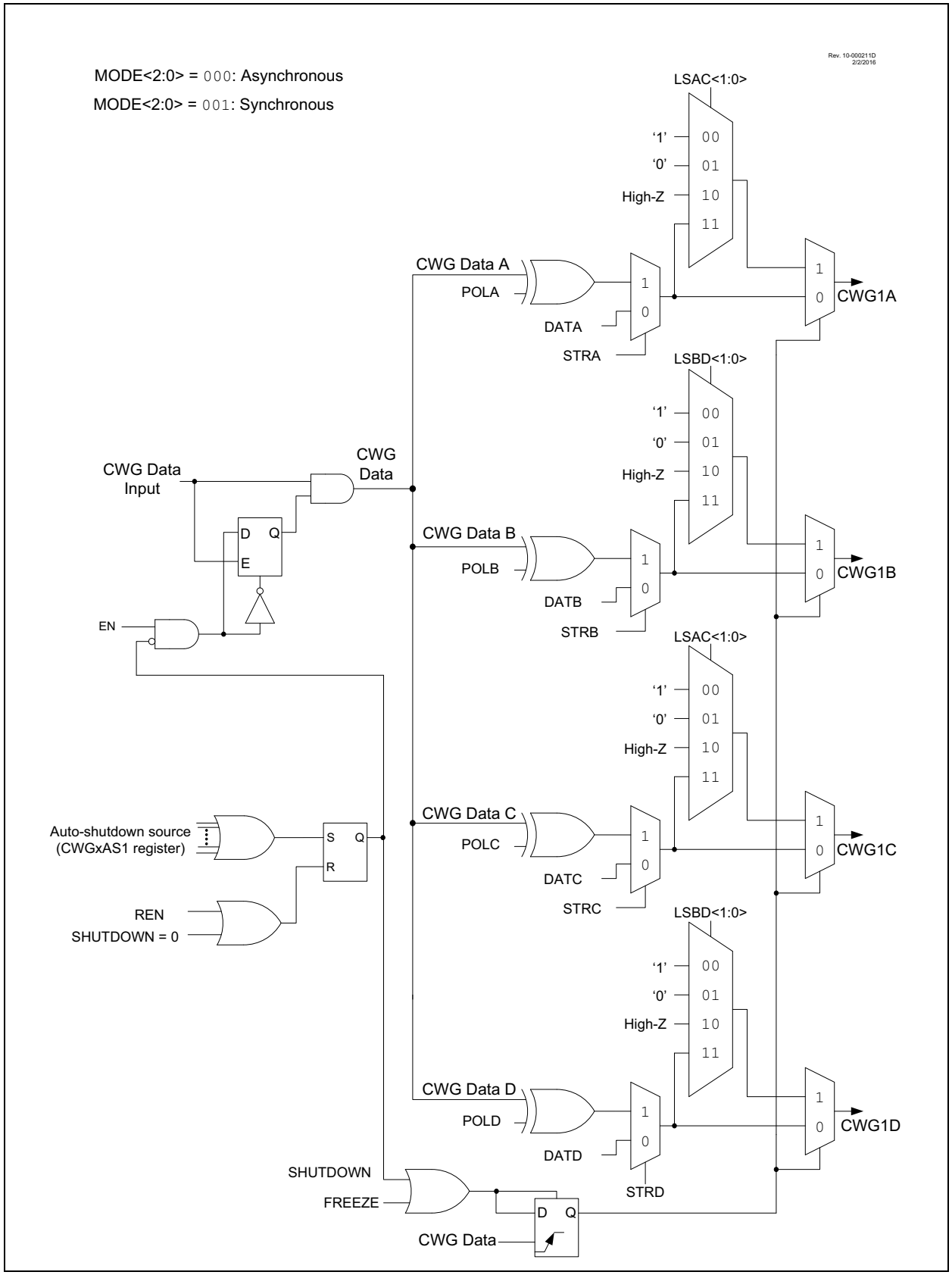
111 = 1:128 Prescaler
110 = 1:64 Prescaler
101 = 1:32 Prescaler
100 = 1:16 Prescaler
011 = 1:8 Prescaler
010 = 1:4 Prescaler
001 = 1:2 Prescaler
000 = 1:1 Prescaler

bit 3-0 **OUTPS<3:0>:** Timerx Output Postscaler Select bits

1111 = 1:16 Postscaler
1110 = 1:15 Postscaler
1101 = 1:14 Postscaler
1100 = 1:13 Postscaler
1011 = 1:12 Postscaler
1010 = 1:11 Postscaler
1001 = 1:10 Postscaler
1000 = 1:9 Postscaler
0111 = 1:8 Postscaler
0110 = 1:7 Postscaler
0101 = 1:6 Postscaler
0100 = 1:5 Postscaler
0011 = 1:4 Postscaler
0010 = 1:3 Postscaler
0001 = 1:2 Postscaler
0000 = 1:1 Postscaler

Note 1: In certain modes, the ON bit will be auto-cleared by hardware. See **Section 22.1.2 "One-Shot Mode"**.

FIGURE 26-11: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



REGISTER 27-7: CLCxGLS0: GATE 0 LOGIC SELECT REGISTER

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
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
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 **G1D4T:** Gate 0 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 0
0 = CLCIN3 (true) is not gated into CLCx Gate 0

bit 6 **G1D4N:** Gate 0 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 0
0 = CLCIN3 (inverted) is not gated into CLCx Gate 0

bit 5 **G1D3T:** Gate 0 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 0
0 = CLCIN2 (true) is not gated into CLCx Gate 0

bit 4 **G1D3N:** Gate 0 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 0
0 = CLCIN2 (inverted) is not gated into CLCx Gate 0

bit 3 **G1D2T:** Gate 0 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 0
0 = CLCIN1 (true) is not gated into l CLCx Gate 0

bit 2 **G1D2N:** Gate 0 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 0
0 = CLCIN1 (inverted) is not gated into CLCx Gate 0

bit 1 **G1D1T:** Gate 0 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 0
0 = CLCIN0 (true) is not gated into CLCx Gate 0

bit 0 **G1D1N:** Gate 0 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 0
0 = CLCIN0 (inverted) is not gated into CLCx Gate 0

29.0 ZERO-CROSS DETECTION (ZCD) MODULE

The ZCD module detects when an A/C signal crosses through the ground potential. The actual zero-crossing threshold is the zero-crossing reference voltage, VCPINV, which is typically 0.75V above ground.

The connection to the signal to be detected is through a series current-limiting resistor. The module applies a current source or sink to the ZCD pin to maintain a constant voltage on the pin, thereby preventing the pin voltage from forward biasing the ESD protection diodes. When the applied voltage is greater than the reference voltage, the module sinks current. When the applied voltage is less than the reference voltage, the module sources current. The current source and sink action keeps the pin voltage constant over the full range of the applied voltage. The ZCD module is shown in the simplified block diagram Figure 29-2.

The ZCD module is useful when monitoring an A/C waveform for, but not limited to, the following purposes:

- A/C period measurement
- Accurate long term time measurement
- Dimmer phase delayed drive
- Low EMI cycle switching

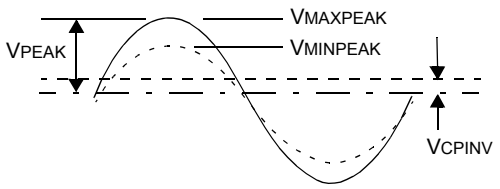
29.1 External Resistor Selection

The ZCD module requires a current-limiting resistor in series with the external voltage source. The impedance and rating of this resistor depends on the external source peak voltage. Select a resistor value that will drop all of the peak voltage when the current through the resistor is nominally 300 µA. Refer to Equation 29-1 and Figure 29-1. Make sure that the ZCD I/O pin internal weak pull-up is disabled so it does not interfere with the current source and sink.

EQUATION 29-1: EXTERNAL RESISTOR

$$R_{SERIES} = \frac{V_{PEAK}}{3\times10^{-4}}$$

FIGURE 29-1: EXTERNAL VOLTAGE



31.17.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit in the UxERRIR register will be set if the baud rate counter overflows before the fifth falling edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the UxBRGH:UxBRGL register pair. After the ABDOVF bit has been set, the state machine continues to search until the fifth falling edge is detected on the RX pin. Upon detecting the fifth falling RX edge, the hardware will set the ABDIF interrupt flag and clear the ABDEN bit in the UxCON0 register. The UxBRGH and UxBRGL register values retain their previous value. The ABDIF flag in the UxUIR register and ABDOVF flag in the UxERRIR register can be cleared by software directly. To generate an interrupt on an auto-baud overflow condition, all the following bits must be set:

- ABDOVE bit in the UxERRIE register
- UxEIE bit in the PIEx register
- PIE and GIE bits in the INTCON register

To terminate the auto-baud process before the ABDIF flag is set, clear the ABDEN bit, then clear the ABDOVF bit in the UxERRIR register.

31.17.3 AUTO-WAKE-UP ON BREAK

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

The Auto-Wake-up feature is enabled by setting both the WUE bit in the UxCON1 register and the UxIE bit in the PIEx register. Once set, the normal receive sequence on RX is disabled, and the UART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a transition out of the Idle state on the RX line. (This coincides with the start of a Break or a wake-up signal character for the LIN protocol.)

The UART module generates a WUIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 31-13), and asynchronously, if the device is in Sleep mode (Figure 31-14). The interrupt condition is cleared by clearing the WUIF bit in the UxUIR register. To generate an interrupt on a wake-up event, all the following bits must be set:

- UxIE bit in the PIEx register
- PIE and GIE bits in the INTCON register

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

31.17.3.1 Special Considerations

Break Character

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

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

Therefore, the initial character of the transmission must be all zeros. This must be eleven or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

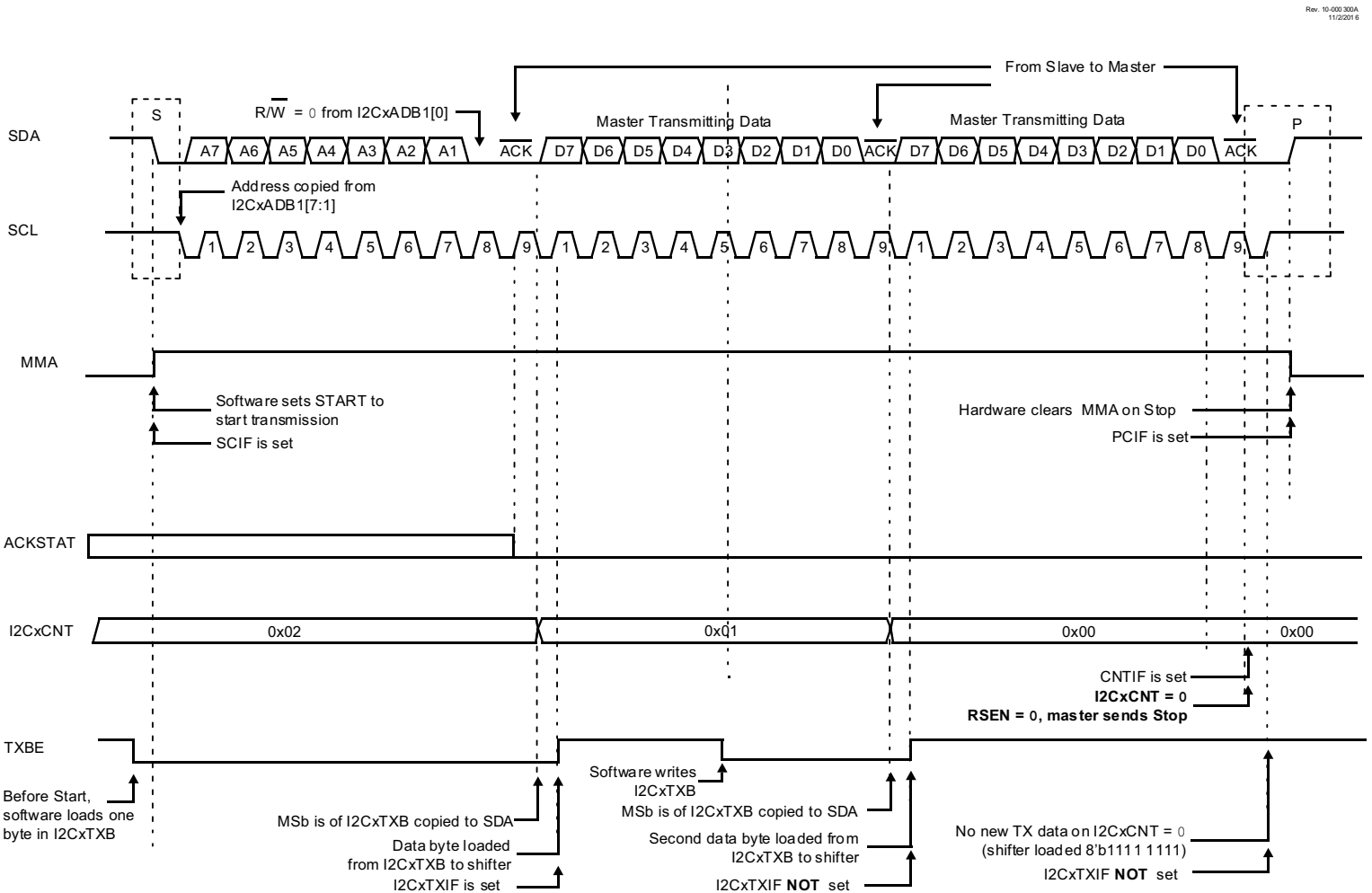
Oscillator Start-up Time

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

WUE Bit

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

FIGURE 33-19: I²C MASTER, 7-BIT ADDRESS, TRANSMISSION WITH STOP



REGISTER 33-10: I2CxPIR: I2CxIF INTERRUPT FLAG REGISTER

R/W/HS-0	R/W/HS-0	U-0	R/W/HS-0	R/W/HS-0	R/W/HS-0	R/W/HS-0	R/W/HS-0
CNTIF	ACKTIF	—	WRIF	ADRIF	PCIF	RSCIF	SCIF
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 **CNTIF:** Byte Count Interrupt Flag bit
1 = When I2CCNT = 0, set by the 9th falling edge of SCL.
0 = I2CCNT condition has not occurred.

bit 6 **ACKTIF:** Acknowledge Status Time Interrupt Flag bit (2) (MODE<2:0> = 0xx OR 11x)
1 = Set by the 9th falling edge of SCL for any byte when addressed as a slave
0 = Acknowledge condition not detected.

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

bit 4 **WRIF:** Data Write Interrupt Flag bit (MODE<2:0> = 0xx OR 11x)
1 = Set the 8th falling edge of SCL for a received data byte.
0 = Data Write condition not detected

bit 3 **ADRIF:** Address Interrupt Flag bit (MODE<2:0> = 0xx OR 11x)
1 = Set the 8th falling edge of SCL for a matching received (high/low) address byte
0 = Address condition not detected

bit 2 **PCIF:** Stop Condition Interrupt Flag
1 = Set on detection of Stop condition
0 = No Stop condition detected

bit 1 **RSCIF:** Restart Condition Interrupt Flag
1 = Set on detection of Restart condition
0 = No Restart condition detected

bit 0 **SCIF:** Start Condition Interrupt Flag
1 = Set on detection of Start condition
0 = No Start condition detected

Note 1: Enabled interrupt flags are OR'd to produce the PIRx<I2CxIF> bit.
2: ACKTIF is not set by a matching, 10-bit, high address byte with the R/W bit clear. It is only set after the matching low address byte is shifted in.

REGISTER 37-27: ADSTPTH: ADC THRESHOLD SETPOINT REGISTER HIGH

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
STPT<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 **STPT<15:8>**: ADC Threshold Setpoint MSB. Upper byte of ADC threshold setpoint, depending on ADCALC, may be used to determine ERR, see Register 37-29 for more details.

REGISTER 37-28: ADSTPTL: ADC THRESHOLD SETPOINT REGISTER LOW

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
STPT<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 **STPT<7:0>**: ADC Threshold Setpoint LSB. Lower byte of ADC threshold setpoint, depending on ADCALC, may be used to determine ERR, see Register 37-30 for more details.

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
3D73h	I2C1CON0	EN	RSEN	S	CSTR	MDR	MODE			562
3D72h	I2C1ADR3	ADR							—	577
3D71h	I2C1ADR2	ADR								576
3D70h	I2C1ADR1	ADR							—	575
3D6Fh	I2C1ADR0	ADR								574
3D6Eh	I2C1ADB1	ADB								579
3D6Dh	I2C1ADB0	ADB								578
3D6Ch	I2C1CNT	CNT								571
3D6Bh	I2C1TXB	TXB								—
3D6Ah	I2C1RXB	RXB								—
3D69h - 3D67h	—	Unimplemented								—
3D66h	I2C2BTO	BTO								567
3D65h	I2C2CLK	CLK								566
3D64h	I2C2PIE	CNTIE	ACKTIE	—	WRIE	ADRIE	PCIE	RSCIE	SCIE	573
3D63h	I2C2PIR	CNTIF	ACKTIF	—	WRIF	ADRIF	PCIF	RSCIF	SCIF	572
3D62h	I2C2STAT1	TXWE	—	TXBE	—	RXRE	CLRBF	—	RXBF	569
3D61h	I2C2STAT0	BFRE	SMA	MMA	R	D	—	—	—	568
3D60h	I2C2ERR	—	BTOIF	BCLIF	NACKIF	—	BTOIE	BCLIE	NACKIE	570
3D5Fh	I2C2CON2	ACNT	GCEN	FME	ABD	SDAHT		BFRET		565
3D5Eh	I2C2CON1	ACKCNT	ACKDT	ACKSTAT	ACKT	—	RXO	TXU	CSD	564
3D5Dh	I2C2CON0	EN	RSEN	S	CSTR	MDR	MODE			562
3D5Ch	I2C2ADR3	ADR							—	577
3D5Bh	I2C2ADR2	ADR								576
3D5Ah	I2C2ADR1	ADR							—	575
3D59h	I2C2ADR0	ADR								574
3D58h	I2C2ADB1	ADB								579
3D57h	I2C2ADB0	ADB								578
3D56h	I2C2CNT	CNT								571
3D55h	I2C2TXB	TXB								—
3D54h	I2C2RXB	RXB								—
3D53h - 3D1Dh	—	Unimplemented								—
3D1Ch	SPI1CLK	CLKSEL								527
3D1Bh	SPI1INTE	SRMTIE	TCZIE	SOSIE	EOSIE	—	RXOIE	TXUIE	—	521
3D1Ah	SPI1INTF	SRMTIF	TCZIF	SOSIF	EOSIF	—	RXOIF	TXUIF	—	520
3D19h	SPI1BAUD	BAUD								523
3D18h	SPI1TWIDTH	—	—	—	—	—	TWIDTH			522
3D17h	SPI1STATUS	TXWE	—	TXBE	—	RXRE	CLRBF	—	RXBF	526
3D16h	SPI1CON2	BUSY	SSFLT	—	—	—	SSET	TXR	RXR	525
3D15h	SPI1CON1	SMP	CKE	CKP	FST	—	SSP	SDIP	SDOP	524
3D14h	SPI1CON0	EN	—	—	—	—	LSBF	MST	BMODE	523
3D13h	SPI1TCNTH	—	—	—	—	—	TCNTH			522
3D12h	SPI1TCNTL	TCNTL								521
3D11h	SPI1TXB	TXB								527
3D10h	SPI1RXB	RXB								526
3D0Fh - 3CFFh	—	Unimplemented								—
3CFEh	MD1CARH	—	—	—	CH					457
3CFDh	MD1CARL	—	—	—	CL					457

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

Note 1: Not present in LF devices.

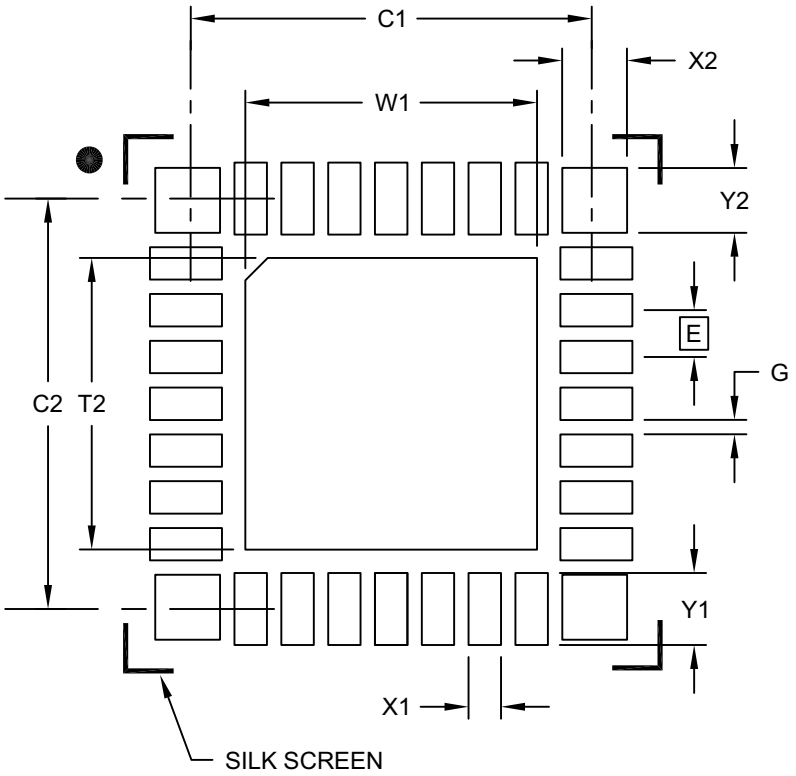
TABLE 43-1: REGISTER FILE SUMMARY FOR PIC18(L)F25/26K83 DEVICES (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3C55h - 3C00h	—	Unimplemented								—
3BFFh	DMA1SIRQ	SIRQ								246
3BFEh	DMA1AIRQ	AIRQ								246
3BFDh	DMA1CON1	DMODE		DSTP	SMR		SMODE		SSTP	239
3BFCh	DMA1CON0	EN	SIRQEN	DGO	—	—	AIRQEN	—	XIP	238
3BFBh	DMA1SSAU	—	—	SSA						241
3BFAh	DMA1SSAH	SSA								240
3BF9h	DMA1SSAL	SSA								240
3BF8h	DMA1SSZH	—	—	—	—	SSZ				242
3BF7h	DMA1SSZL	SSZ								242
3BF6h	DMA1SPTRU	—	—	SPTR						242
3BF5h	DMA1SPTRH	SPTR								241
3BF4h	DMA1SPTRL	SPTR								241
3BF3h	DMA1SCNTH	—	—	—	—	SCNT				243
3BF2h	DMA1SCNTL	SCNT								243
3BF1h	DMA1DSAH	DSA								244
3BF0h	DMA1DSAL	SSA								243
3BEFh	DMA1DSZH	—	—	—	—	DSZ				245
3BEEh	DMA1DSZL	DSZ								245
3BEDh	DMA1DPTRH	DPTR								244
3BECh	DMA1DPTRL	DPTR								244
3BEBh	DMA1DCNTH	—	—	—	—	DCNT				243
3BEAh	DMA1DCNTL	DCNT								245
3BE9h	DMA1BUF	BUF								240
3BE8h - 3BE0h	—	Unimplemented								—
3BDFh	DMA2SIRQ	—	SIRQ							246
3BDEh	DMA2AIRQ	—	AIRQ							246
3BDDh	DMA2CON1	DMODE		DSTP	SMR		SMODE		SSTP	239
3BDCh	DMA2CON0	EN	SIRQEN	DGO	—	—	AIRQEN	—	XIP	238
3BDBh	DMA2SSAU	—	—	SSA						241
3BDAh	DMA2SSAH	SSA								240
3BD9h	DMA2SSAL	SSA								240
3BD8h	DMA2SSZH	—	—	—	—	SSZ				242
3BD7h	DMA2SSZL	SSZ								242
3BD6h	DMA2SPTRU	—	—	SPTR						242
3BD5h	DMA2SPTRH	SPTR								241
3BD4h	DMA2SPTRL	SPTR								241
3BD3h	DMA2SCNTH	—	—	—	—	SCNT				243
3BD2h	DMA2SCNTL	SCNT								243
3BD1h	DMA2DSAH	DSA								244
3BD0h	DMA2DSAL	SSA								243
3BCFh	DMA2DSZH	—	—	—	—	DSZ				245
3BCEh	DMA2DSZL	DSZ								245
3BCDh	DMA2DPTRH	DPTR								244
3BCCh	DMA2DPTRL	DPTR								244
3BCBh	DMA2DCNTH	—	—	—	—	DCNT				243
3BCAh	DMA2DCNTL	DCNT								245
3BC9h	DMA2BUF	BUF								240

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

28-Lead Plastic Quad Flat, No Lead Package (MX) - 6x6 mm Body [UQFN] With 0.60mm Contact Length And Corner Anchors

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W1			4.05
Optional Center Pad Length	T2			4.05
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.00
Corner Pad Width (X4)	X2			0.90
Corner Pad Length (X4)	Y2			0.90
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2209B