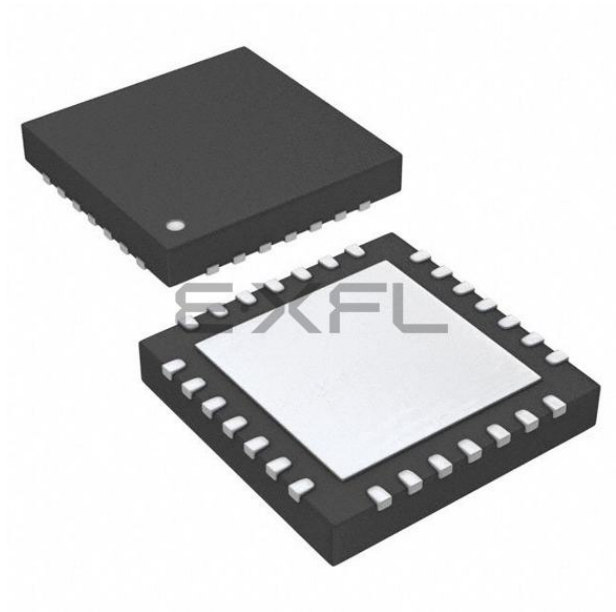


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k42-e-mv

1.4 Register and Bit naming conventions

1.4.1 REGISTER NAMES

When there are multiple instances of the same peripheral in a device, the peripheral control registers will be depicted as the concatenation of a peripheral identifier, peripheral instance, and control identifier. The control registers section will show just one instance of all the register names with an 'x' in the place of the peripheral instance number. This naming convention may also be applied to peripherals when there is only one instance of that peripheral in the device to maintain compatibility with other devices in the family that contain more than one.

1.4.2 BIT NAMES

There are two variants for bit names:

- Short name: Bit function abbreviation
- Long name: Peripheral abbreviation + short name

1.4.2.1 Short Bit Names

Short bit names are an abbreviation for the bit function. For example, some peripherals are enabled with the EN bit. The bit names shown in the registers are the short name variant.

Short bit names are useful when accessing bits in C programs. The general format for accessing bits by the short name is *RegisterName*bits.*ShortName*. For example, the enable bit, EN, in the COG1CON0 register can be set in C programs with the instruction `COG1CON0bits.EN = 1`.

Short names are generally not useful in assembly programs because the same name may be used by different peripherals in different bit positions. When this occurs, during the include file generation, all instances of that short bit name are appended with an underscore plus the name of the register in which the bit resides to avoid naming contentions.

1.4.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.

Long bit names are useful in both C and assembly programs. For example, in C the COG1CON0 enable bit can be set with the `G1EN = 1` instruction. In assembly, this bit can be set with the `BSF COG1CON0,G1EN` instruction.

1.4.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the mode control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

```
COG1CON0bits.MD = 0x5;
```

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name MD2 and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode:

Example 1:

```
MOVLW  ~(1<<G1MD1)
ANDWF  COG1CON0,F
MOVLW  1<<G1MD2 | 1<<G1MD0
IORWF  COG1CON0,F
```

Example 2:

```
BSF    COG1CON0,G1MD2
BCF    COG1CON0,G1MD1
BSF    COG1CON0,G1MD0
```

1.4.3 REGISTER AND BIT NAMING EXCEPTIONS

1.4.3.1 Status, Interrupt, and Mirror Bits

Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

4.6 Register Definitions: Status Registers

REGISTER 4-2: STATUS: STATUS REGISTER

U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C
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 **Unimplemented:** Read as '0'

bit 6 **$\overline{TO}$:** Time-Out bit
1 = Set at power-up or by execution of CLRWDT or SLEEP instruction
0 = A WDT time-out occurred

bit 5 **$\overline{PD}$:** Power-Down bit
1 = Set at power-up or by execution of CLRWDT instruction
0 = Set by execution of the SLEEP instruction

bit 4 **N:** Negative bit used for signed arithmetic (2's complement); indicates if the result is negative, (ALU MSb = 1).
1 = The result is negative
0 = The result is positive

bit 3 **OV:** Overflow bit used for signed arithmetic (2's complement); indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit 7) to change state.
1 = Overflow occurred for current signed arithmetic operation
0 = No overflow occurred

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{Borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)**(1)**
1 = A carry-out from the 4th low-order bit of the result occurred
0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{Borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)**(1,2)**
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note 1: For $\overline{Borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand.
2: For Rotate (RRF, RLF) instructions, this bit is loaded with either the high or low-order bit of the Source register.

REGISTER 5-3: CONFIGURATION WORD 2L (30 0002h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
BOREN<1:0>		$\overline{\text{LPBOREN}}$	IVT1WAY	MVECEN	PWRTS<1:0>		MCLRE
bit 7							bit 0

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

bit 7-6 **BOREN<1:0>:** Brown-out Reset Enable bits

When enabled, Brown-out Reset Voltage (V_{BOR}) is set by the BORV bit.
11 = Brown-out Reset is enabled, SBOREN bit is ignored
10 = Brown-out Reset is enabled while running, disabled in Sleep; SBOREN is ignored
01 = Brown-out Reset is enabled according to SBOREN
00 = Brown-out Reset is disabled

bit 5 **$\overline{\text{LPBOREN}}$:** Low-Power BOR Enable bit

1 = Low-Power BOR is disabled
0 = Low-Power BOR is enabled

bit 4 **IVT1WAY:** IVTLOCK bit One-Way Set Enable bit

1 = IVTLOCK bit can be cleared and set only once; IVT registers remain locked after one clear/set cycle
0 = IVTLOCK bit can be set and cleared repeatedly (subject to the unlock sequence)

bit 3 **MVECEN:** Multi-vector Enable bit

1 = Multi-vector enabled; Vector table used for interrupts
0 = Multi-vector disabled; Vector table not used for interrupts

bit 2-1 **PWRTS<1:0>:** Power-up Timer Selection bits

11 = PWRT is disabled
10 = PWRT set at 64 ms
01 = PWRT set at 16 ms
00 = PWRT set at 1 ms

bit 0 **MCLRE:** Master Clear ($\overline{\text{MCLR}}$) Enable bit

If LVP = 1:
RE3 pin function is $\overline{\text{MCLR}}$.
If LVP = 0:
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$
0 = $\overline{\text{MCLR}}$ pin function is a port defined function

10.1 Clock Source

The input to the reference clock output can be selected using the CLKRCLK register.

10.1.1 CLOCK SYNCHRONIZATION

Once the reference clock enable (EN) is set, the module is ensured to be glitch-free at start-up.

When the reference clock output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled, but glitches may occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the CLKREN is clear.

10.2 Programmable Clock Divider

The module takes the clock input and divides it based on the value of the DIV<2:0> bits of the CLKRCON register (Register 10-1).

The following configurations can be made based on the DIV<2:0> bits:

- Base FOSC value
- FOSC divided by 2
- FOSC divided by 4
- FOSC divided by 8
- FOSC divided by 16
- FOSC divided by 32
- FOSC divided by 64
- FOSC divided by 128

The clock divider values can be changed while the module is enabled; however, in order to prevent glitches on the output, the DIV<2:0> bits should only be changed when the module is disabled (EN = 0).

10.3 Selectable Duty Cycle

The DC<1:0> bits of the CLKRCON register can be used to modify the duty cycle of the output clock. A duty cycle of 25%, 50%, or 75% can be selected for all clock rates, with the exception of the undivided base FOSC value.

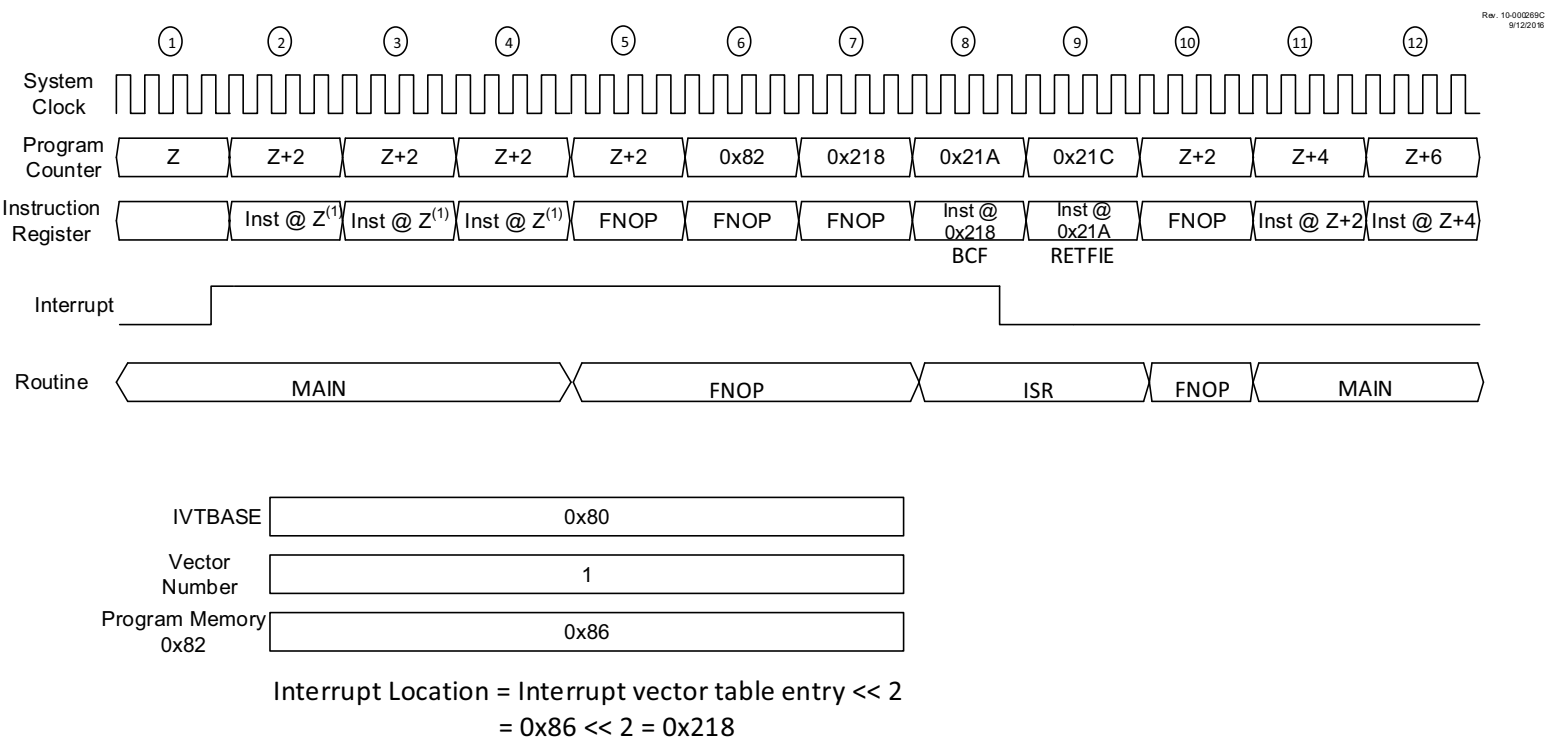
The duty cycle can be changed while the module is enabled; however, in order to prevent glitches on the output, the DC<1:0> bits should only be changed when the module is disabled (EN = 0).

Note:	The DC1 bit is reset to '1'. This makes the default duty cycle 50% and not 0%.

10.4 Operation in Sleep Mode

The reference clock output module clock is based on the system clock. When the device goes to Sleep, the module outputs will remain in their current state. This will have a direct effect on peripherals using the reference clock output as an input signal. No change should occur in the module from entering or exiting from Sleep.

FIGURE 11-9: INTERRUPT TIMING DIAGRAM - THREE CYCLE INSTRUCTION



Note 1: Instruction @ Z is a Three-cycle instruction.

REGISTER 11-11: PIR8: PERIPHERAL INTERRUPT REGISTER 8

R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0	U-0	U-0
TMR5GIF	TMR5IF	—	—	—	—	—	—
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 = Bit is set in hardware

bit 7 **TMR5GIF:** TMR5 Gate Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 6 **TMR5IF:** TMR5 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

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

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 11-12: PIR9: PERIPHERAL INTERRUPT REGISTER 9

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	CLC3IF	CWG3IF	CCP3IF	TMR6IF
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 7-4 **Unimplemented:** Read as '0'

bit 3 **CLC3IF:** CLC3 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 2 **CWG3IF:** CWG3 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 1 **CCP3IF:** CCP3 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 0 **TMR6IF:** TMR6 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 11-32: IPR7: PERIPHERAL INTERRUPT Priority REGISTER 7

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
—	—	INT2IP	CLC2IP	CWG2IP	—	CCP2IP	TMR4IP
bit 7							bit 0

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

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

bit 5 **INT2IP:** External Interrupt 2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **CLC2IP:** CLC2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CWG2IP:** CWG2 Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 1 **CCP2IP:** CRC Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR4IP:** TMR4 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 11-33: IPR8: PERIPHERAL INTERRUPT Priority REGISTER 8

R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0	U-0	U-0
TMR5GIP	TMR5IP	—	—	—	—	—	—
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 **TMR5GIP:** TMR5 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **TMR5IP:** TMR5 Interrupt Priority bit
1 = High priority
0 = Low priority

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

15.1.3 READING THE PROGRAM FLASH MEMORY

The TBLRD instruction retrieves data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

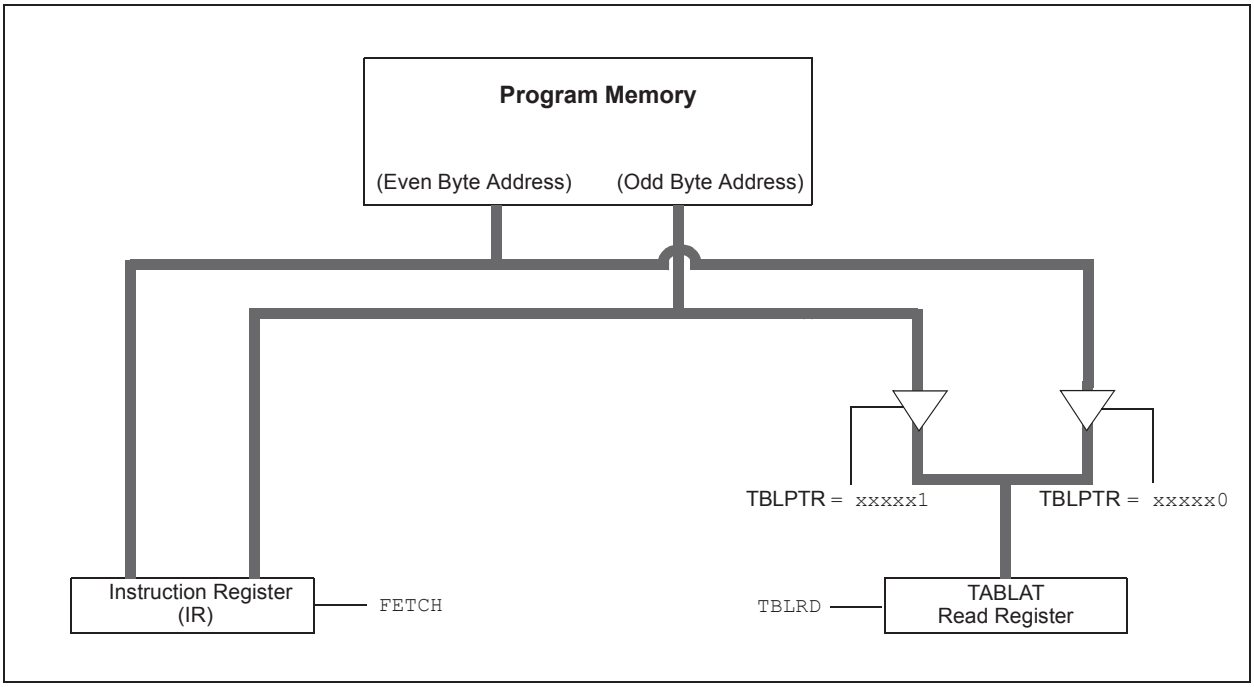
TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The CPU operation is suspended during the read, and it resumes immediately after. From the user point of view, TABLAT is valid in the next instruction cycle.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word.

Figure 15-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 15-4: READS FROM PROGRAM FLASH MEMORY



EXAMPLE 15-1: READING A PROGRAM FLASH MEMORY WORD

```
            BCF       NVMCON1, REG0          ; point to Program Flash Memory
            BSF       NVMCON1, REG1          ; access Program Flash Memory
            MOVLW     CODE_ADDR_UPPER        ; Load TBLPTR with the base
            MOVWF     TBLPTRU                ; address of the word
            MOVLW     CODE_ADDR_HIGH
            MOVWF     TBLPTRH
            MOVLW     CODE_ADDR_LOW
            MOVWF     TBLPTRL
READ_WORD
            TBLRD*+                          ; read into TABLAT and increment
            MOVF      TABLAT, W              ; get data
            MOVWF     WORD_EVEN
            TBLRD*+                          ; read into TABLAT and increment
            MOVFW     TABLAT, W              ; get data
            MOVF      WORD_ODD
```

EXAMPLE 15-4: WRITING TO PROGRAM FLASH MEMORY (CONTINUED)

```
WRITE_BYTE_TO_HREGS
              MOVF      POSTINC0, W              ; get low byte of buffer data
              MOVWF     TABLAT                   ; present data to table latch
              TBLWT+*                            ; write data, perform a short write
                                                 ; to internal TBLWT holding register.
              DECFSZ    COUNTER                  ; loop until holding registers are full
              BRA       WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
              BCF       NVMCON1, REG0            ; point to Program Flash Memory
              BSF       NVMCON1, REG1            ; point to Program Flash Memory
              BSF       NVMCON1, WREN            ; enable write to memory
              BCF       NVMCON1, FREE            ; enable write to memory
              BCF       INTCON0, GIE             ; disable interrupts
              MOVLW     55h
Required      MOVWF     NVMCON2                  ; write 55h
Sequence      MOVLW     0AAh
              MOVWF     NVMCON2                  ; write 0AAh
              BSF       NVMCON1, WR              ; start program (CPU stall)
              DCFSZ     COUNTER2                 ; repeat for remaining write blocks
              BRA       WRITE_BYTE_TO_HREGS
              BSF       INTCON0, GIE             ; re-enable interrupts
              BCF       NVMCON1, WREN            ; disable write to memory
```

REGISTER 15-2: NVMCON2: NONVOLATILE MEMORY CONTROL 2 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMCON2<7:0>							
bit 7							bit 0

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

bit 7-0 **NVMCON2<7:0>:**
Refer to **Section 15.1.4 "NVM Unlock Sequence"**.

Note 1: This register always reads zeros, regardless of data written.

Register 15-3: NVMADRL: Data EEPROM Memory Address Low

R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0	R/W-x/0
ADR<7:0>							
bit 7							bit 0

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

bit 7-0 **ADR<7:0>:** EEPROM Read Address bits

REGISTER 15-4: NVMADRH: DATA EEPROM MEMORY ADDRESS HIGH(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADR<9:8>	
bit 7							bit 0

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

bit 7-2 **Unimplemented:** Read as '0'
bit 1-0 **ADR<9:8>:** EEPROM Read Address bits

Note 1: The NVMADRH register is not implemented on PIC18(L)F24/25K42.

TABLE 16-2: SCANNER OPERATING MODES(1)

TRIGEN	BURSTMD	Scanner Operation
0	0	Memory access is requested when the CRC module is ready to accept data; the request is granted if no other higher priority source request is pending.
1	0	Memory access is requested when the CRC module is ready to accept data and trigger selection is true; the request is granted if no other higher priority source request is pending.
x	1	Memory access is always requested, the request is granted if no other higher priority source request is pending.

Note 1: See **Section 3.1 "System Arbitration"** for Priority selection and **Section 3.2 "Memory Access Scheme"** for Memory Access Scheme.

REGISTER 16-12: SCANLADRU: SCAN LOW ADDRESS UPPER BYTE REGISTER

U-0	U-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
—	—	LADR<21:16>(1,2)					
bit 7							bit 0

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

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

bit 5-0 **LADR<21:16>:** Scan Start/Current Address bits(1,2)
Upper bits of the current address to be fetched from, value increments on each fetch of memory.

Note 1: Registers SCANLADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SGO = 0 (SCANCON0 register).
2: While SGO = 1 (SCANCON0 register), writing to this register is ignored.

REGISTER 16-13: SCANLADRH: SCAN LOW ADDRESS HIGH BYTE 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
LADR<15:8>(1, 2)							
bit 7							bit 0

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

bit 7-0 **LADR<15:8>**: Scan Start/Current Address bits(1, 2)
Most Significant bits of the current address to be fetched from, value increments on each fetch of memory.

Note 1: Registers SCANLADRU/H/L form a 22-bit value, but are not guarded for atomic or asynchronous access; registers should only be read or written while SGO = 0 (SCANCON0 register).
2: While SGO = 1 (SCANCON0 register), writing to this register is ignored.

FIGURE 17-3: DMA COUNTERS BLOCK DIAGRAM

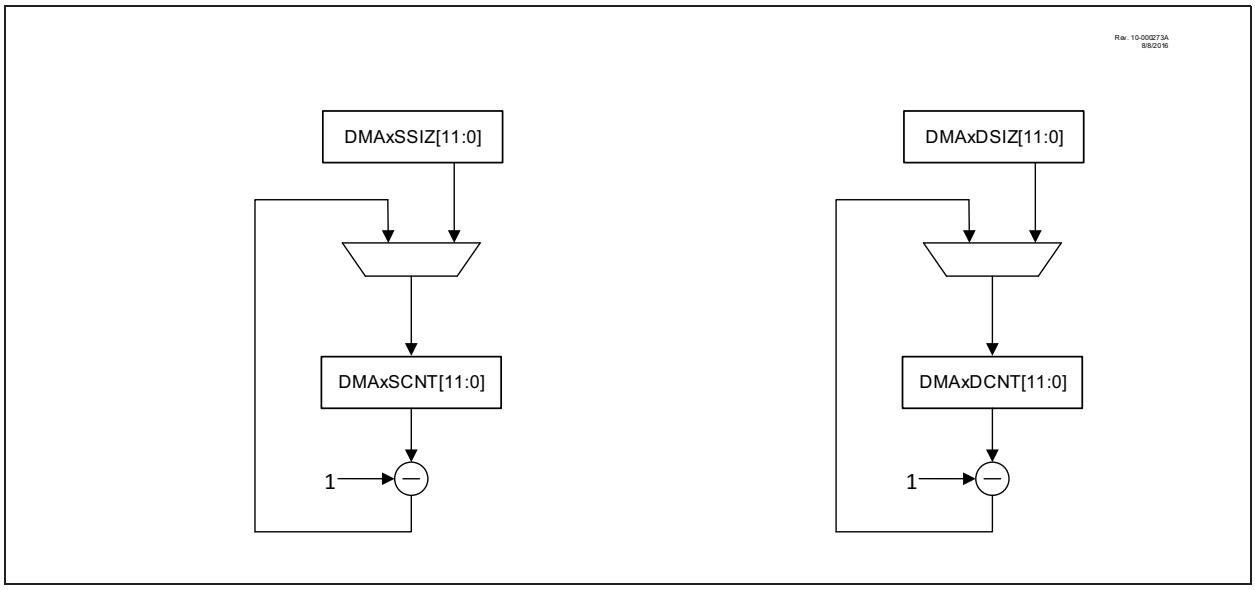


Table 17-2 has a few examples of configuring DMA Message sizes.

TABLE 17-2: EXAMPLE MESSAGE SIZE TABLE

Operation	Example	SCNT	DCNT	Comments
Read from single SFR location to RAM	U1RXB	1	N	N equals the number of bytes desired in the destination buffer. N >= 1.
Write to single SFR location from RAM	U1TXB	N	1	N equals the number of bytes desired in the source buffer. N >= 1.
Read from multiple SFR location	ADRES[H:L]	2	2*N	N equals the number of ADC results to be stored in memory. N>= 1
	TMR1[H:L]	2	2*N	N equals the number of TMR1 Acquisition results to be stored in memory. N>= 1
	SMT1CPR[U:H:L]	3	3*N	N equals the number of Capture Pulse Width measurements to be stored in memory. N>= 1
Write to Multiple SFR registers	PWMDC[H:L]	2*N	2	N equals the number of PWM duty cycle values to be loaded from a memory table. N>= 1
	All ADC registers	N*31	31	Using the DMA to transfer a complete ADC context from RAM to the ADC registers.N>= 1

23.12 Register Definitions: Timer1/3/5

Long bit name prefixes for the Timer1/3/5 are shown in Table 24-2. Refer to **Section 1.4.2.2 "Long Bit Names"** for more information.

TABLE 23-3:

Peripheral	Bit Name Prefix
Timer1	T1
Timer3	T3
Timer5	T5

REGISTER 23-1: TXCON: TIMERx CONTROL REGISTER

U-0	U-0	R/W-0/u	R/W-0/u	U-0	R/W-0/u	R/W-0/0	R/W-0/u
—	—	CKPS<1:0>		—	$\overline{\text{SYNC}}$	RD16	ON
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	u = unchanged

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

bit 5-4 **CKPS<1:0>:** Timerx Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

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

bit 2 **$\overline{\text{SYNC}}$:** Timerx External Clock Input Synchronization Control bit
TMRxCLK = FOSC/4 or FOSC:
This bit is ignored. Timer1 uses the incoming clock as is.
Else:
1 = Do not synchronize external clock input
0 = Synchronize external clock input with system clock

bit 1 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timerx in one 16-bit operation
0 = Enables register read/write of Timerx in two 8-bit operation

bit 0 **ON:** Timerx On bit
1 = Enables Timerx
0 = Disables Timerx

TABLE 27-3: SUMMARY OF REGISTERS ASSOCIATED WITH SMT1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		399
SMT1CON1	GO	REPEAT	—	—	MODE<3:0>				400
SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	401
SMT1CLK	—	—	—	—	—	CSEL<2:0>			402
SMT1SIG	—	—	—	SSEL<4:0>					404
SMT1WIN	—	—	—	WSEL<4:0>					403
SMT1TMRL	TMR<7:0>								405
SMT1TMRH	TMR<15:8>								405
SMT1TMRU	TMR<23:16>								405
SMT1CPRL	CPR<7:0>								406
SMT1CPRH	CPR<15:8>								406
SMT1CPRU	CPR<23:16>								406
SMT1CPWL	CPW<7:0>								407
SMT1CPWH	CPW<15:8>								407
SMT1CPWU	CPW<23:16>								407
SMT1PRL	PR<7:0>								408
SMT1PRH	PR<15:8>								408
SMT1PRU	PR<23:16>								408

Legend: — = unimplemented read as '0'. Shaded cells are not used for SMT1 module.

30.8 NCO Control Registers

REGISTER 30-1: NCO1CON: NCO CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
EN	—	OUT	POL	—	—	—	PFM
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 **EN:** NCO1 Enable bit
1 = NCO1 module is enabled
0 = NCO1 module is disabled

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

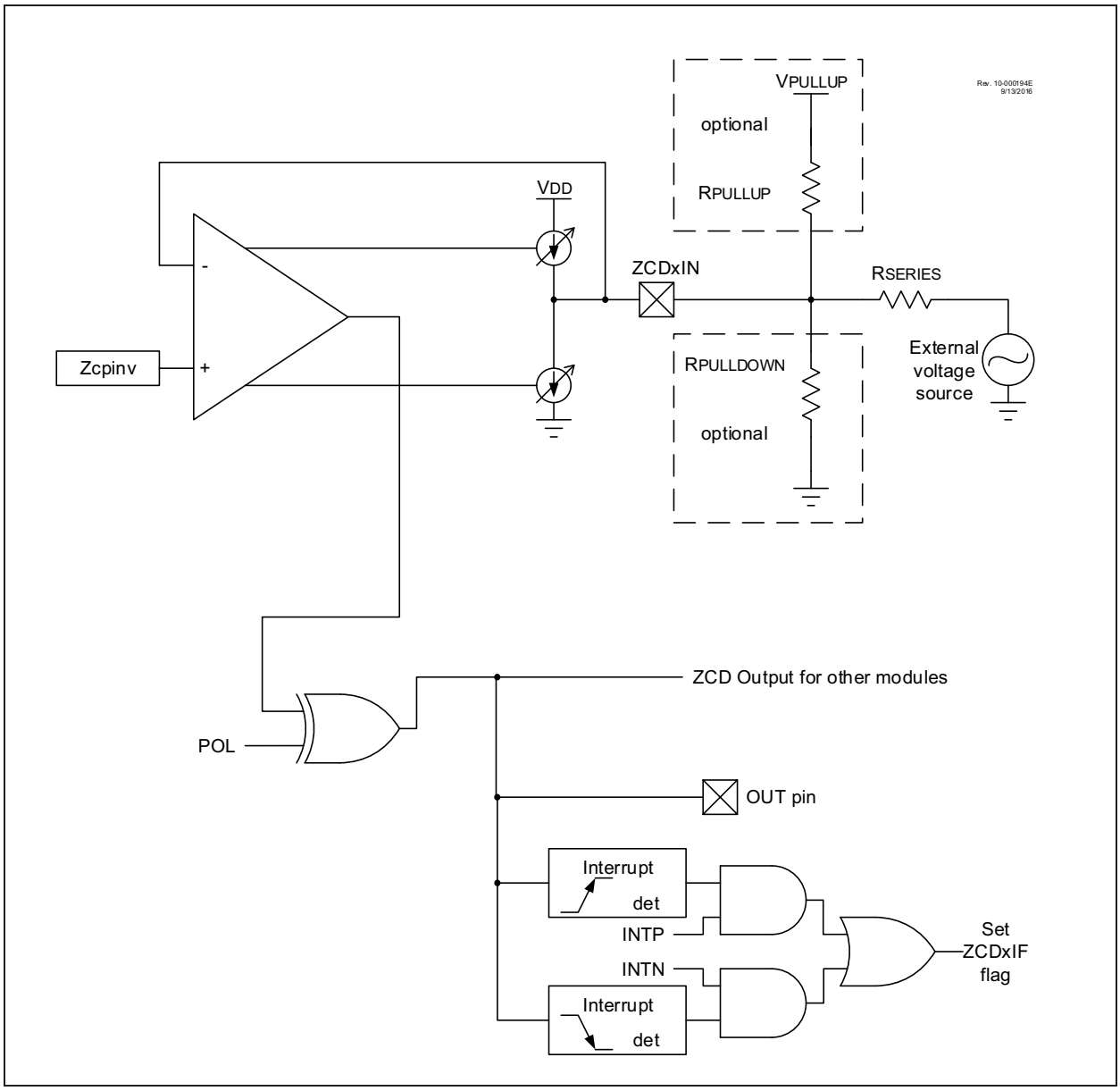
bit 5 **OUT:** NCO1 Output bit
Displays the current output value of the NCO1 module.

bit 4 **POL:** NCO1 Polarity
1 = NCO1 output signal is inverted
0 = NCO1 output signal is not inverted

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

bit 0 **PFM:** NCO1 Pulse Frequency Mode bit
1 = NCO1 operates in Pulse Frequency mode
0 = NCO1 operates in Fixed Duty Cycle mode, divide by 2

FIGURE 31-2: SIMPLIFIED ZCD BLOCK DIAGRAM



31.2 ZCD Logic Output

The ZCD module includes a Status bit, which can be read to determine whether the current source or sink is active. The OUT bit of the ZCDCON register is set when the current sink is active, and cleared when the current source is active. The OUT bit is affected by the polarity bit, even if the module is disabled.

The OUT signal can also be used as input to other modules. This is controlled by the registers of the corresponding module. OUT can be used as follows:

- Gate source for TMR1/3/5
- Clock source for TMR2/4/6
- Reset source for TMR2/4/6

31.3 ZCD Logic Polarity

The POL bit of the ZCDCON register inverts the OUT bit relative to the current source and sink output. When the POL bit is set, a OUT high indicates that the current source is active, and a low output indicates that the current sink is active.

The POL bit affects the ZCD interrupts.

32.0 DATA SIGNAL MODULATOR (DSM) MODULE

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical “AND” operation of both the carrier and modulator signals and then provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals. A carrier high (CARH) signal and a carrier low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the carrier high signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the carrier low signal with the modulator signal.

Using this method, the DSM can generate the following types of Key Modulation schemes:

- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Programmable Modulator Data
- Modulated Output Polarity Select
- Peripheral Module Disable, which provides the ability to place the DSM module in the lowest power consumption mode

Figure 32-1 shows a Simplified Block Diagram of the Data Signal Modulator peripheral.

35.4.1.1 7-bit Addresses Mode

In this mode, the LSb of the received data byte is ignored when determining if there is an address match. All four I2CxADR registers are independently compared to the received address byte.

35.4.1.2 7-bit Addresses with Masking

In this mode, the value in I2CxADR0 is masked with the value in I2CxADR1 to determine if an address match occurred. A second address and mask are also compared from I2CxADR2/3. When Mode<2:0> = 001 or 111, the I2CxADR1/3 registers serve as the mask value for I2CxADR0/2. All seven bits of the address can be masked

35.4.1.3 10-bit Addresses

In this mode, the values stored in I2CxADR0 and I2CxADR1 registers are used to create a 10-bit address. A second 10-bit compare address is formed from I2CxADR2 and I2CxADR3.

35.4.1.4 10-bit Address with Masking

In this mode, the I2CxADR0/1 registers are used to form a 10-bit address, and the I2CxADR2/3 registers are used to form a 10-bit mask for that address. When MODE<2:0> = 011, the I2CxADR2/3 registers serve as the mask value for the 10-bit address stored in I2CxADR0/1.

Note:	Even though 10-bit addressing calls out only 10-bits used in the address comparison, all 15 address bits in I2CxADR0/1 are compared in these modes.

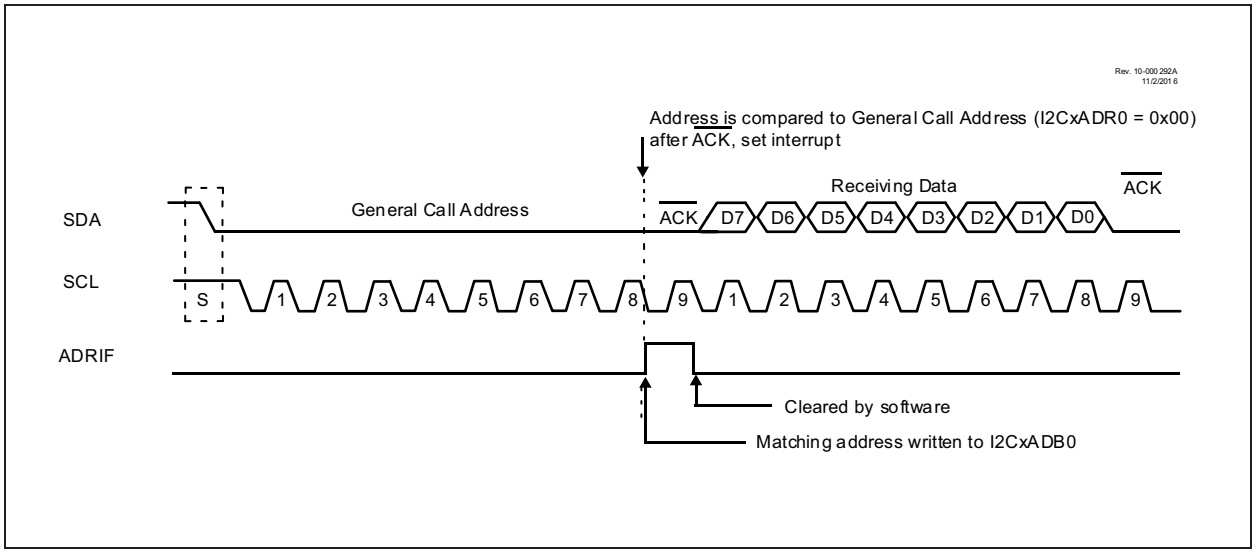
35.4.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an ACK. The general call address is a reserved address in the I^2C protocol, defined as address 0x00. In order for the slave hardware to ACK this address, it must be enabled by setting the GCEN bit in the I2CxCON2 register. Setting one of the I2CxADR0/1/2/3 registers to 0x00 is not required. Figure 35-5 shows a General Call reception sequence.

If the ADRIE bit is set, the module will clock stretch after the eighth SCL pulse just like any other address match.

Note:	General Call addressing is supported in only 7-bit Addressing modes

FIGURE 35-5: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



REGISTER 35-13: I2CxADR1 – I²C ADDRESS 1 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 7							bit 0
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 7-1 **ADR[7-1]:** Address or Divider 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 modes w/Masking

MSK0<7:1>:7-bit Slave Address

MSK0<0>: Unused in this mode; bit state is a don't care

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

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

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

40.7 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Table 46-15 and Table 46-17 for more details.

40.8 Analog Input Connection Considerations

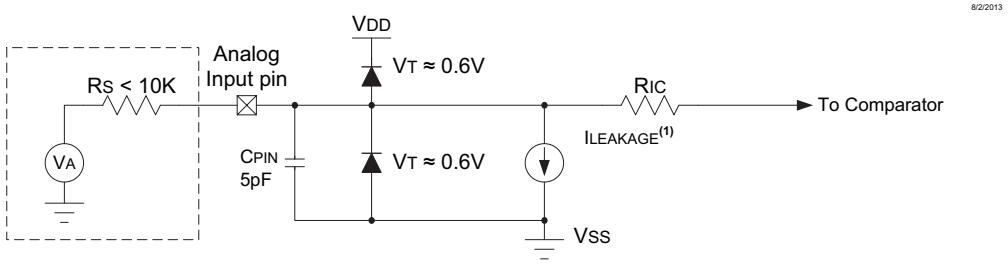
A simplified circuit for an analog input is shown in Figure 40-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 40-3: ANALOG INPUT MODEL



Legend:
C_{PIN}	=	Input Capacitance
$I_{LEAKAGE}$	=	Leakage Current at the pin due to various junctions
R_{IC}	=	Interconnect Resistance
R_S	=	Source Impedance
V_A	=	Analog Voltage
V_T	=	Threshold Voltage

Note 1: See **Section 46.0 "Electrical Specifications"**.