



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	4KB (1.375K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f04kl101-i-p

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



Register(s) Name	Description
W0 through W15	Working Register Array
PC	23-Bit Program Counter
SR	ALU STATUS Register
SPLIM	Stack Pointer Limit Value Register
TBLPAG	Table Memory Page Address Register
PSVPAG	Program Space Visibility Page Address Register
RCOUNT	REPEAT Loop Counter Register
CORCON	CPU Control Register

4.2.5 SOFTWARE STACK

In addition to its use as a Working register, the W15 register in PIC24F devices is also used as a Software Stack Pointer. The pointer always points to the first available free word and grows from lower to higher addresses. It predecrements for stack pops and post-increments for stack pushes, as shown in Figure 4-4.

Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

Note:	A PC push during exception processing					
	will concatenate the SRL register to the					
	MSB of the PC prior to the push.					

The Stack Pointer Limit Value (SPLIM) register, associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' as all stack operations must be word-aligned. Whenever an EA is generated, using W15 as a source or destination pointer, the resulting address is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal, and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation.

Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address, 0DF6, in RAM, initialize the SPLIM with the value, 0DF4.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0800h. This prevents the stack from interfering with the Special Function Register (SFR) space.

Note: A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

FIGURE 4-4: CALL STACK FRAME



4.3 Interfacing Program and Data Memory Spaces

The PIC24F architecture uses a 24-bit wide program space and 16-bit wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Apart from the normal execution, the PIC24F architecture provides two methods by which the program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space, PSV

Table instructions allow an application to read or write small areas of the program memory. This makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data. It can only access the least significant word (lsw) of the program word.

4.3.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Memory Page Address register (TBLPAG) is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full 24-bit program space address. In this format, the Most Significant bit (MSb) of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = 0) or the configuration memory (TBLPAG<7> = 1).

For remapping operations, the 8-bit Program Space Visibility Page Address register (PSVPAG) is used to define a 16K word page in the program space. When the MSb of the EA is '1', PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike the table operations, this limits remapping operations strictly to the user memory area.

Table 4-20 and Figure 4-5 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> bits refer to a program space word, whereas the D<15:0> bits refer to a data space word.

R/SO-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0		
WR	WREN	WRERR	PGMONLY		—		_		
bit 15		•					bit 8		
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
_	ERASE	NVMOP5 ⁽¹⁾	NVMOP4 ⁽¹⁾	NVMOP3 ⁽¹⁾	NVMOP2 ⁽¹⁾	NVMOP1 ⁽¹⁾	NVMOP0 ⁽¹⁾		
bit 7							bit 0		
Legend:		HC = Hardware	e Clearable bit	U = Unimpler	mented bit, rea	ad as '0'			
R = Readable	bit	W = Writable bi	t	SO = Settable	e Only bit				
-n = Value at I	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown		
bit 15	 bit 15 WR: Write Control bit (program or erase) 1 = Initiates a data EEPROM erase or write cycle (can be set but not cleared in software) 0 = Write cycle is complete (cleared automatically by hardware) 								
bit 14	WREN: Write 1 = Enables a 0 = No operat	Enable bit (eras in erase or progr ion allowed (dev	e or program) am operation rice clears this b	it on completic	on of the write/	erase operatio	n)		
bit 13	WRERR: Flas	h Error Flag bit							
	1 = A write of operation0 = The write	operation is prei	maturely termin leted successfu	ated (any MC	CLR or WDT	Reset during	programming		
bit 12	PGMONLY: P	rogram Only En	able bit						
 1 = Write operation is executed without erasing target address(es) first 0 = Automatic erase-before-write; write operations are preceded automatically by an erase of target address(es) 									
bit 11-7 Unimplemented: Read as '0'									
bit 6	ERASE: Eras	e Operation Sele	ect bit						
	 1 = Performs an erase operation when WR is set 0 = Performs a write operation when WR is set 								
bit 5-0	-0 NVMOP<5:0>: Programming Operation Command Byte bits ⁽¹⁾								
	Erase Operations (when ERASE bit is '1'):								
	011010 = Erases & Words 011001 = Erases 4 words								
	011000 = Erases 1 word								
	0100xx = Era	ases entire data	EEPROM						
	0100xx = Erases entire data EEPROM <u>Programming Operations (when ERASE bit is '0'):</u> 001xxx = Writes 1 word								

REGISTER 6-1: NVMCON: NONVOLATILE MEMORY CONTROL REGISTER

Note 1: These NVMOP configurations are unimplemented on PIC24F04KL10X and PIC24F08KL20X devices.

REGISTER 8-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0		
—	_	—	_		—	—	DC ⁽¹⁾		
bit 15		-			•		bit 8		
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0		
IPL2 ^(2,3)	IPL1 ^(2,3)	IPL0 ^(2,3)	RA ⁽¹⁾	N ⁽¹⁾	OV ⁽¹⁾	Z ⁽¹⁾	C ⁽¹⁾		
bit 7	·				•		bit 0		
Legend:									
R = Readabl	R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'				
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clear		ared x = Bit is unkn			
bit 15-9	Unimplemen	ted: Read as 'o)'						
bit 7-5	IPL<2:0>: CF	PU Interrupt Pric	ority Level Stat	us bits ^(2,3)					
	111 = CPU lr	nterrupt Priority	Level is 7 (15)	: user interrupt	s disabled				
	110 = CPII Interrupt Priority Level is 6 (14)								
	101 = CPU Interrupt Priority Level is 5 (13)								
	100 = CPU Interrupt Priority Level is 4 (12)								
	011 = CPU Interrupt Priority evel is 3 (11)								
	011 = CPU Interrupt Priority Level is 3 (11) $010 = CPU Interrupt Priority Level is 2 (10)$								
	0.01 = CPU Ir	nterrunt Priority	Level is 1 (9)						

- 000 = CPU Interrupt Priority Level is 0 (8)
- **Note 1:** See Register 3-1 for the description of these bits, which are not dedicated to interrupt control functions.
 - **2:** The IPL bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the Interrupt Priority Level if IPL3 = 1.
 - 3: The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

Note: Bit 8 and bits 4 through 0 are described in Section 3.0 "CPU".

REGISTER 8-27: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	U2ERIP2 ⁽¹⁾	U2ERIP1 ⁽¹⁾	U2ERIP0 ⁽¹⁾
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	U1ERIP2 ⁽¹⁾	U1ERIP1 ⁽¹⁾	U1ERIP0 ⁽¹⁾	—	—	—	—
bit 7							bit 0

Legend:									
R = Readable	e 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 15-11	1 Unimplemented: Read as '0'								
bit 10-8	U2ERIP<2:0>	-: UART2 Error Interrupt Pri	ority bits ⁽¹⁾						
	111 = Interru	pt is Priority 7 (highest prior	ty interrupt)						
	•								
	•								
	001 = Interru	pt is Priority 1							
	000 = Interru	pt source is disabled							
bit 7	Unimplemen	ted: Read as '0'							
bit 6-4	U1ERIP<2:0	-: UART1 Error Interrupt Pri	ority bits ⁽¹⁾						
	111 = Interru	pt is Priority 7 (highest prior	ty interrupt)						
	•								
	•								
	• 001 = Interru	ot is Priority 1							
	000 = Interru	pt source is disabled							
bit 3-0	Unimplemen	ted: Read as '0'							
	•								

Note 1: These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

9.3 Control Registers

The operation of the oscillator is controlled by three Special Function Registers (SFRs):

- OSCCON
- CLKDIV
- OSCTUN

The OSCCON register (Register 9-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The Clock Divider register (Register 9-2) controls the features associated with Doze mode, as well as the postscaler for the FRC oscillator.

The FRC Oscillator Tune register (Register 9-3) allows the user to fine-tune the FRC oscillator. OSCTUN functionality has been provided to help customers compensate for temperature effects on the FRC frequency over a wide range of temperatures. The tuning step-size is an approximation and is neither characterized nor tested.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0, HSC	R-0, HSC	R-0, HSC	U-0	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0, HSC	U-0	R-0, HSC ⁽²⁾	U-0	R/CO-0, HS	R/W-0 ⁽³⁾	R/W-0	R/W-0
CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit				
HS = Hardware Settable bit	CO = Clearable Only bit	SO = Settable Only bit			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'			
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown		

bit 15 Unimplemented: Read as '0'

bit 14-12 COSC<2:0>: Current Oscillator Selection bits

- 111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = 8 MHz FRC Oscillator (FRC)
- bit 11 Unimplemented: Read as '0'

bit 10-8 NOSC<2:0>: New Oscillator Selection bits⁽¹⁾

- 111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = 8 MHz FRC Oscillator (FRC)

Note 1: Reset values for these bits are determined by the FNOSC<2:0> Configuration bits.

- 2: Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.
- **3:** When SOSC is selected to run from a digital clock input rather than an external crystal (SOSCSRC = 0), this bit has no effect.

9.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

Note: The Primary Oscillator mode has three different submodes (XT, HS and EC), which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

9.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in the FOSC Configuration register must be programmed to '0'. (Refer to **Section 23.0** "**Special Features**" for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and FSCM function are disabled; this is the default setting.

The NOSCx control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSCx bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled; it is held at '0' at all times.

9.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

- 1. If desired, read the COSCx bits (OSCCON<14:12>) to determine the current oscillator source.
- 2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
- 3. Write the appropriate value to the NOSCx bits (OSCCON<10:8>) for the new oscillator source.
- 4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
- 5. Set the OSWEN bit to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically, as follows:

- 1. The clock switching hardware compares the COSCx bits with the new value of the NOSCx bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
- If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and CF (OSCCON<3>) bits are cleared.
- The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = 1).
- 4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
- 5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSCx bits value is transferred to the COSCx bits.
- The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM, with LPRC as a clock source, are enabled) or SOSC (if SOSCEN remains enabled).

Note 1: The processor will continue to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.





'1' = Bit is set

REGISTER 17-8: SSPxADD: MSSPx SLAVE ADDRESS/BAUD RATE GENERATOR REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
			ADE)<7:0>			
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable bi	t	U = Unimplen	nented bit, read	d as '0'	

'0' = Bit is cleared

bit 15-8 Unimplemented: Read as '0'

-n = Value at POR

bit 7-0 ADD<7:0>: Slave Address/Baud Rate Generator Value bits SPI Master and I²C[™] Master modes: Reloads value for Baud Rate Generator. Clock period is (([SPxADD] + 1) *2)/Fosc. I²C Slave modes: Represents 7 or 8 bits of the slave address, depending on the addressing mode used: 7-Bit mode: Address is ADD<7:1>; ADD<0> is ignored. 10-Bit LSb mode: ADD<7:0> are the Least Significant bits of the address. 10-Bit MSb mode: ADD<2:1> are the two Most Significant bits of the address; ADD<7:3> are always '11110' as a specification requirement, ADD<0> is ignored.

REGISTER 17-9: SSPxMSK: I²C[™] SLAVE ADDRESS MASK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
—	—	_	—	—	—	_	—	
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	R/W-1	
			MSK<	:7:0>(1)				
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable I	bit	U = Unimplem	nented bit, read			
-n = Value at POR '1' = Bit is set			'0' = Bit is cleared x = Bit		x = Bit is unkr	is unknown		
L								
bit 15-8	Unimpleme	nted: Read as 'o)'					

bit 7-0 MSK<7:0>: Slave Address Mask Select bits⁽¹⁾

1 = Masking of corresponding bit of SSPxADD is enabled

0 = Masking of corresponding bit of SSPxADD is disabled

Note 1: MSK0 is not used as a mask bit in 7-bit addressing.

x = Bit is unknown

REGISTER 19-1: AD1CON1: A/D CONTROL REGISTER 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON ⁽¹⁾	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HSC	R-0, HSC
SSRC2	SSRC1	SSRC0	—	_	ASAM	SAMP	DONE
bit 7							bit 0

Legend:	HSC = Hardware Settable/0	Clearable bit						
R = Readable	e bit W = Writable bit	U = Unimplemented bit, rea	d as '0'					
-n = Value at	POR '1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown					
bit 15	ADON: A/D Operating Mode bit ⁽¹⁾							
	1 = A/D Converter module is operating0 = A/D Converter is off							
bit 14	Unimplemented: Read as '0'							
bit 13	ADSIDL: A/D Stop in Idle Mode bit							
	1 = Discontinues module operation when0 = Continues module operation in Idle m	device enters Idle mode ode						
bit 12-10	Unimplemented: Read as '0'							
bit 9-8	FORM<1:0>: Data Output Format bits							
	11 = Signed fractional (sddd dddd dd00 0000) 10 = Fractional (dddd dddd dd00 0000) 01 = Signed integer (ssss sssd dddd dddd) 00 = Integer (0000 00dd dddd dddd)							
bit 7-5	SSRC<2:0>: Conversion Trigger Source S	Select bits						
	<pre>111 = Internal counter ends sampling and starts conversion (auto-convert) 110 = Reserved 101 = Reserved 100 = Reserved 011 = Reserved 010 = Timer1 compare ends sampling and starts conversion 001 = Active transition on INT0 pin ends sampling and starts conversion</pre>							
bit 4-3	Unimplemented: Read as '0'	5						
bit 2	ASAM: A/D Sample Auto-Start bit							
	 1 = Sampling begins immediately after the 0 = Sampling begins when the SAMP bit is 	e last conversion completes; S s set	AMP bit is auto-set					
bit 1	SAMP: A/D Sample Enable bit							
	 1 = A/D Sample-and-Hold amplifier is sam 0 = A/D Sample-and-Hold amplifier is hold 	pling input ing						
bit 0	DONE: A/D Conversion Status bit							
	1 = A/D conversion is done 0 = A/D conversion is not done							

Note 1: Values of ADC1BUFx registers will not retain their values once the ADON bit is cleared. Read out the conversion values from the buffer before disabling the module.

REGISTER 23-3: FOSCSEL: OSCILLATOR SELECTION CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-0	R/P-0	R/P-1
IESO	LPRCSEL	SOSCSRC	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

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

bit 7	IESO: Internal External Switchover bit
	 1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled) 0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)
bit 6	LPRCSEL: Internal LPRC Oscillator Power Select bit
	1 = High-Power/High-Accuracy mode0 = Low-Power/Low-Accuracy mode
bit 5	SOSCSRC: Secondary Oscillator Clock Source Configuration bit
	 1 = SOSC analog crystal function is available on the SOSCI/SOSCO pins 0 = SOSC crystal is disabled; digital SCLKI function is selected on the SOSCO pin
bit 4-3	Unimplemented: Read as '0'
bit 2-0	FNOSC<2:0>: Oscillator Selection bits
	111 = 8 MHz FRC Oscillator with Divide-by-N (FRCDIV)
	110 = 500 kHz Low-Power FRC Oscillator with Divide-by-N (LPFRCDIV)
	101 = Low-Power RC Oscillator (LPRC)
	100 = Secondary Oscillator (SOSC)
	011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
	010 = Primary Oscillator (XT, HS, EC)
	001 = 8 MHz FRC Oscillator with Divide-by-N with PLL module (FRCDIV+PLL)

000 = 8 MHz FRC Oscillator (FRC)

REGISTER 23-5: FWDT: WATCHDOG TIMER CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	
FWDTEN1	WINDIS	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0	
bit 7	I				L		bit 0	
Legend:								
R = Readable	bit	P = Programm	able bit	U = Unimplemented bit, read as '0'				
-n = Value at POR		'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown	
bit 7,5 FWDTEN<1:0>: Watchdog Timer Enable bits 11 = WDT is enabled in hardware 10 = WDT is controlled with the SWDTEN bit setting 01 = WDT is enabled only while device is active; WDT is disabled in Sleep, SWDTEN bit is disabled 00 = WDT is disabled in hardware; SWDTEN bit is disabled								
bit 6	 WINDIS: Windowed Watchdog Timer Disable bit 1 = Standard WDT is selected; windowed WDT is disabled 0 = Windowed WDT is enabled; note that executing a CLRWDT instruction while the WDT is disabled in hardware and software (FWDTEN<1:0> = 00 and SWDTEN (RCON<5> = 0) will not cause a device Reset 							
bit 4	FWPSA: WD	T Prescaler bit						
	1 = WDT pres	scaler ratio of 1:	128					
	0 = WD1 pres	scaler ratio of 1:	32					
bit 3-0	WDTPS<3:0> 1111 = 1:32,7 1110 = 1:16,7 1101 = 1:8,19 1000 = 1:4,09 1011 = 1:2,04 1010 = 1:1,02 1001 = 1:512 1000 = 1:256 0111 = 1:128 0110 = 1:64 0101 = 1:32 0100 = 1:16 0011 = 1:8 0010 = 1:4 0001 = 1:2 0000 = 1:1	▶: Watchdog Tin 768 384 92 96 48 24	ner Postscale	Select bits				

23.3 Unique ID

A read-only Unique ID value is stored at addresses, 800802h through 800808h. This factory programmed value is unique to each microcontroller produced in the PIC24F16KL402 family. To access this region, use Table Read instructions or Program Space Visibility. To ensure a globally Unique ID across other Microchip microcontroller families, the "Unique ID" value should be further concatenated with the family and Device ID values stored at address, FF0000h.

REGISTER 23-8: DEVID: DEVICE ID REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

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

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits 01001011 = PIC24F16KL402 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits 00000001 = PIC24F04KL100

00000010 = PIC24F04KL101

00000101 = PIC24F08KL200 00000110 = PIC24F08KL201

00001010 = PIC24F08KL301 00000000 = PIC24F08KL302

00001110 = PIC24F08KL401 00000100 = PIC24F08KL402 00011110 = PIC24F16KL401 00010100 = PIC24F16KL402

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	—	—	—	—	—	—	—
bit 23							bit 16
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
		—	—	—	—	—	—
bit 15	•		•	•		•	bit 8
U-0	U-0	U-0	U-0	R	R	R	R
		—	—	REV3	REV2	REV1	REV0
bit 7	•		•	•		•	bit 0
Legend:							
R = Readable	le 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			nown

REGISTER 23-9: DEVREV: DEVICE REVISION REGISTER

bit 23-4 Unimplemented: Read as '0'

bit 3-0 **REV<3:0>:** Revision Identifier bits

Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
ADD	ADD	f	f = f + WREG	1	1	C, DC, N, OV, Z
	ADD	f,WREG	WREG = f + WREG	1	1	C, DC, N, OV, Z
	ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C, DC, N, OV, Z
	ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C, DC, N, OV, Z
	ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C, DC, N, OV, Z
ADDC	ADDC	f	f = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C, DC, N, OV, Z
AND	AND	f	f = f .AND. WREG	1	1	N, Z
	AND	f,WREG	WREG = f .AND. WREG	1	1	N, Z
	AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N, Z
	AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N, Z
	AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N, Z
ASR	ASR	£	f = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C, N, OV, Z
	ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N, Z
	ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N, Z
BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
	BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
	BRA	GE, Expr	Branch if Greater than or Equal	1	1 (2)	None
	BRA	GEU, Expr	Branch if Unsigned Greater than or Equal	1	1 (2)	None
	BRA	GT, Expr	Branch if Greater than	1	1 (2)	None
	BRA	GTU, Expr	Branch if Unsigned Greater than	1	1 (2)	None
	BRA	LE, Expr	Branch if Less than or Equal	1	1 (2)	None
	BRA	LEU, Expr	Branch if Unsigned Less than or Equal	1	1 (2)	None
	BRA	LT, Expr	Branch if Less than	1	1 (2)	None
	BRA	LTU, Expr	Branch if Unsigned Less than	1	1 (2)	None
	BRA	N,Expr	Branch if Negative	1	1 (2)	None
	BRA	NC, Expr	Branch if Not Carry	1	1 (2)	None
	BRA	NN, Expr	Branch if Not Negative	1	1 (2)	None
	BRA	NOV, Expr	Branch if Not Overflow	1	1 (2)	None
	BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
	BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
	BRA	Expr	Branch Unconditionally	1	2	None
	BRA	Z,Expr	Branch if Zero	1	1 (2)	None
	BRA	Wn	Computed Branch	1	2	None
BSET	BSET	f,#bit4	Bit Set f	1	1	None
	BSET	Ws,#bit4	Bit Set Ws	1	1	None
BSW	BSW.C	Ws,Wb	Write C bit to Ws <wb></wb>	1	1	None
	BSW.Z	Ws,Wb	Write Z bit to Ws <wb></wb>	1	1	None
BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
	BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
	BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

TABLE 25-2:	INSTRUCTION SET	OVERVIEW

Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
GOTO	GOTO	Expr	Go to Address	2	2	None
	GOTO	Wn	Go to Indirect	1	2	None
INC	INC	£	f = f + 1	1	1	C, DC, N, OV, Z
	INC	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2	£	f = f + 2	1	1	C, DC, N, OV, Z
	INC2	f,WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2	Ws,Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR	f	f = f .IOR. WREG	1	1	N, Z
	IOR	f,WREG	WREG = f .IOR. WREG	1	1	N, Z
	IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N, Z
	IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N, Z
	IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK	#lit14	Link Frame Pointer	1	1	None
LSR	LSR	f	f = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	f,WREG	WREG = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C, N, OV, Z
	LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N, Z
	LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N, Z
MOV	MOV	f,Wn	Move f to Wn	1	1	None
	MOV	[Wns+Slit10],Wnd	Move [Wns+Slit10] to Wnd	1	1	None
	MOV	f	Move f to f	1	1	N, Z
	MOV	f,WREG	Move f to WREG	1	1	None
	MOV	#litl6,Wn	Move 16-bit Literal to Wn	1	1	None
	MOV.b	#lit8,Wn	Move 8-bit Literal to Wn	1	1	None
	MOV	Wn,f	Move Wn to f	1	1	None
	MOV	Wns,[Wns+Slit10]	Move Wns to [Wns+Slit10]	1	1	None
	MOV	Wso,Wdo	Move Ws to Wd	1	1	None
	MOV	WREG, f	Move WREG to f	1	1	None
	MOV.D	Wns,Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
	MOV.D	Ws,Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
	MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
	MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
	MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
	MUL.SU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
	MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
	MUL	f	W3:W2 = f * WREG	1	1	None
NEG	NEG	f	$f = \overline{f} + 1$	1	1	C, DC, N, OV, Z
	NEG	f,WREG	WREG = \overline{f} + 1	1	1	C, DC, N, OV, Z
	NEG	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
NOP	NOP		No Operation	1	1	None
	NOPR		No Operation	1	1	None
POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
	POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
	POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd+1)	1	2	None
	POP.S		Pop Shadow Registers	1	1	All
PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
	PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
	PUSH.D	Wns	Push W(ns):W(ns+1) to Top-of-Stack (TOS)	1	2	None
	PUSH.S		Push Shadow Registers	1	1	None

TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)

DC CHARACTERISTICS			$\begin{array}{llllllllllllllllllllllllllllllllllll$					
Parameter No.	Typical ⁽¹⁾	Max	Units Conditions					
Module Differential Current (△IPD)								
DC71	0.21	0.65	μA	1.8V	+82°C			
	0.45	0.95	μA	3.3V	+00 C	Watchdog Timer Current:		
	—	1.30	μA	1.8V	±125°C	∆WDT ^(2,3)		
	—	1.50	μA	3.3V	+125 C			
DC72	0.69	1.50	μA	1.8V	+85°C	32 kHz Crystal with Timer1:		
	1.00	1.50	μA	3.3V	+05 C	\triangle SOSC (SOSCSEL = 0) ⁽²⁾		
DC75	5.24	—	μA	1.8V	+85°C			
	5.16	11.00	μA	3.3V	105 0	<u>лні \/D(2,3)</u>		
	_	12.00	μA	1.8V	+125°C			
	—	15.00	μA	3.3V	123 0			
DC76	4.15	9.00	μA	3.3V	+85°C	ABOD(2,3)		
		11.0	μA	3.3V	+125°C	ABOR		
DC78	0.03	0.20	μA	1.8V	+85°C			
	0.03	0.20	μA	3.3V	700 C			
		0.40	μA	1.8V	+125°C			
		0.40	μA	3.3V	123 0			

TABLE 26-9: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

Note 1: Data in the Typical column is at 3.3V, +25°C unless otherwise stated.

2: The ∆ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

3: This current applies to Sleep only.

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



RECOMMENDED LAND PATTERN

	Units			S
Dimension	Dimension Limits		NOM	MAX
Contact Pitch	Е	1.27 BSC		
Contact Pad Spacing	С		9.40	
Contact Pad Width (X20)	Х			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2094A

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



	MILLIMETERS			
Dimension	MIN	NOM	MAX	
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	С		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
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. C04-2073A

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length





	MILLIMETERS			
Dimension	MIN	NOM	MAX	
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
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. C04-2105A