



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	24
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08kl302t-i-mq

Email: info@E-XFL.COM

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

Pin Diagrams: PIC24FXXKL302/402



3.0 CPU

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the CPU, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"CPU"** (DS39703).

The PIC24F CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set and a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M instructions of user program memory space. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the REPEAT instructions, which are interruptible at any point.

PIC24F devices have sixteen, 16-bit Working registers in the programmer's model. Each of the Working registers can act as a data, address or address offset register. The 16th Working register (W15) operates as a Software Stack Pointer (SSP) for interrupts and calls.

The upper 32 Kbytes of the data space memory map can optionally be mapped into program space at any 16K word boundary of either program memory or data EEPROM memory, defined by the 8-bit Program Space Visibility Page Address (PSVPAG) register. The program to data space mapping feature lets any instruction access program space as if it were data space.

The Instruction Set Architecture (ISA) has been significantly enhanced beyond that of the PIC18, but maintains an acceptable level of backward compatibility. All PIC18 instructions and addressing modes are supported, either directly, or through simple macros. Many of the ISA enhancements have been driven by compiler efficiency needs.

The core supports Inherent (no operand), Relative, Literal, Memory Direct and three groups of addressing modes. All modes support Register Direct and various Register Indirect modes. Each group offers up to seven addressing modes. Instructions are associated with predefined addressing modes depending upon their functional requirements. For most instructions, the core is capable of executing a data (or program data) memory read, a Working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing trinary operations (i.e., A + B = C) to be executed in a single cycle.

A high-speed, 17-bit by 17-bit multiplier has been included to significantly enhance the core arithmetic capability and throughput. The multiplier supports Signed, Unsigned and Mixed mode, 16-bit by 16-bit or 8-bit by 8-bit integer multiplication. All multiply instructions execute in a single cycle.

The 16-bit ALU has been enhanced with integer divide assist hardware that supports an iterative non-restoring divide algorithm. It operates in conjunction with the REPEAT instruction looping mechanism and a selection of iterative divide instructions to support 32-bit (or 16-bit), divided by a 16-bit integer signed and unsigned division. All divide operations require 19 cycles to complete, but are interruptible at any cycle boundary.

The PIC24F has a vectored exception scheme, with up to eight sources of non-maskable traps and up to 118 interrupt sources. Each interrupt source can be assigned to one of seven priority levels.

A block diagram of the CPU is illustrated in Figure 3-1.

3.1 Programmer's Model

Figure 3-2 displays the programmer's model for the PIC24F. All registers in the programmer's model are memory mapped and can be manipulated directly by instructions.

Table 3-1 provides a description of each register. All registers associated with the programmer's model are memory mapped.



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				

TABLE 4-20:	PROGRAM SPACE ADDRESS CONSTRUCTION
-------------	------------------------------------

	Access	Program Space Address						
Access Type	Space	<23>	<22:16>	<15>	<14:1>	<0>		
Instruction Access	User	0 PC<22:1>				0		
(Code Execution)		0xx xxxx xxxx xxxx xxxx xxx0						
TBLRD/TBLWT (Byte/Word Read/Write)	User	User TBLPAG<7:0>		Data EA<15:0>				
		0xxx xxxx xxxx				xxx		
	Configuration	TBLPAG<7:0>		Data EA<15:0>				
		12	1xxx xxxx xx					
Program Space Visibility (Block Remap/Read)	User	0	0 PSVPAG<7:0>		0> ⁽²⁾ Data EA<14:0> ⁽¹⁾			
		0	xxxx xxxx		XXX XXXX XXXX XXXX			

Note 1: Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is PSVPAG<0>.

2: PSVPAG can have only two values ('00' to access program memory and FF to access data EEPROM) on PIC24F16KL402 family devices.





EXAMPLE 5-2: ERASING A PROGRAM MEMORY ROW – 'C' LANGUAGE CODE

// C example using MPLAB C30	
<pre>intattribute ((space(auto_psv))) progAddr = &progAddr unsigned int offset;</pre>	// Global variable located in Pgm Memory
//Set up pointer to the first memory location to be written	
TBLPAG =builtin_tblpage(&progAddr); offset = &progAddr & 0xFFFF;	// Initialize PM Page Boundary SFR // Initialize lower word of address
<pre>builtin_tblwtl(offset, 0x0000);</pre>	<pre>// Set base address of erase block // with dummy latch write</pre>
NVMCON = 0x4058;	// Initialize NVMCON
asm("DISI #5");	<pre>// Block all interrupts for next 5 // instructions</pre>
builtin_write_NVM();	// Instructions // C30 function to perform unlock // sequence and set WR

EXAMPLE 5-3: LOADING THE WRITE BUFFERS – ASSEMBLY LANGUAGE CODE

;	Set up NVMCO	N for row programming operati	ons	
	MOV	#0x4004, W0	;	
	MOV	W0, NVMCON	;	Initialize NVMCON
;	Set up a poi	nter to the first program mem	ory	location to be written
;	program memo	ry selected, and writes enabl	ed	
	MOV	#0x0000, W0	;	
	MOV	W0, TBLPAG	;	Initialize PM Page Boundary SFR
	MOV	#0x6000, W0	;	An example program memory address
;	Perform the	TBLWT instructions to write t	he i	latches
;	0th_program_	word		
	MOV	#LOW_WORD_0, W2	;	
	MOV	<pre>#HIGH_BYTE_0, W3</pre>	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
;	lst_program_	word		
	MOV	#LOW_WORD_1, W2	;	
	MOV	#HIGH_BYTE_1, W3	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
;	2nd_program	_word		
	MOV	#LOW_WORD_2, W2	;	
	MOV	<pre>#HIGH_BYTE_2, W3</pre>	;	
	TBLWTL	W2, [W0]	;	Write PM low word into program latch
	TBLWTH	W3, [W0++]	;	Write PM high byte into program latch
	•			
	•			
	•	1		
,	3∠na_program	_WOLA		
	MOV	$\# UW WUKD_{31}, WZ$,	
		HUTRU RITE ST' MS	;	White DM los word into measure lately
	TRTMLT	₩2, [WU] W2 [W0]	;	Write PM IOW Word Into program latch
	IBTMIH	W3, [WU]	'	write PM high byte into program latch

REGISTER 8-19: IPC2: INTERRUPT PRIORITY CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1RXIP2	U1RXIP1	U1RXIP0		—		—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
	—	—	—	—	T3IP2	T3IP1	T3IP0
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplem	nented bit, read	d as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	iown
bit 15	Unimplemen	ted: Read as '	כי				
bit 14-12	U1RXIP<2:0>	: UART1 Rece	eiver Interrupt F	Priority bits			
	111 = Interrup	pt is Priority 7(highest priority	interrupt)			
	•						
	•						
	001 = Interrup	pt is Priority 1					
	000 = Interru	pt source is dis	abled				
bit 11-3	Unimplemen	ted: Read as '	כי				
bit 2-0	T3IP<2:0>: Ti	imer3 Interrupt	Priority bits				
	111 = Interrup	pt is Priority 7 (highest priority	interrupt)			
	•						
	•						
	001 = Interru	ot is Priority 1					
	000 = Interrup	ot source is dis	abled				

REGISTER 8-21: IPC4: INTERRUPT PRIORITY CONTROL REGISTER 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0		
_	CNIP2	CNIP1	CNIP0	_	CMIP2	CMIP1	CMIP0		
bit 15			1			1	bit 8		
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0		
	BCL1IP2	BCL1IP1	BCL1IP0	—	SSP1IP2	SSP1IP1	SSP1IP0		
bit 7							bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimplem	nented bit, read	l as '0'			
-n = Value at F	POR	'1' = Bit is set		0' = Bit is clear	ared	x = Bit is unkr	nown		
bit 1E	Unimploment	had. Dood oo '	, '						
		eu: Reau as	, otification Inter	rupt Drigrity bit	2				
DIL 14-12		iput Change N	bigbest priority	interrunt)	5				
	•		nightest phoney	interrupt)					
	•								
	•								
	001 = Interrup	ot source is dis	abled						
bit 11	Unimplement	ted: Read as ')'						
bit 10-8	CMIP<2:0>: (Comparator Inte	- errupt Priority b	oits					
	111 = Interrup	ot is Priority 7 (highest priority	interrupt)					
	•	, , , , , , , , , , , , , , , , , , ,	0 1 9	• •					
	•								
	• 001 = Interrur	ot is Priority 1							
	000 = Interrup	ot source is dis	abled						
bit 7	Unimplement	ted: Read as '	כי						
bit 6-4	BCL1IP<2:0>	: MSSP1 I ² C™	Bus Collision	Interrupt Priori	ty bits				
	111 = Interrup	ot is Priority 7 (highest priority	interrupt)					
	•								
	•								
	001 = Interrup	ot is Priority 1							
	000 = Interrup	ot source is dis	abled						
	Unimplemented: Read as '0'								
bit 3	SSP1IP<2:0>: MSSP1 SPI/I ² C Event Interrupt Priority bits								
bit 3 bit 2-0	SSP1IP<2:0>	: MSSP1 SPI/I	² C Event Inter	upt Priority bits)				
bit 3 bit 2-0	SSP1IP<2:0> 111 = Interrup	: MSSP1 SPI/I ot is Priority 7(² C Event Intern highest priority	interrupt)	,				
bit 3 bit 2-0	SSP1IP<2:0> 111 = Interrup •	: MSSP1 SPI/I ot is Priority 7 (² C Event Interi highest priority	interrupt)	•				
bit 3 bit 2-0	SSP1IP<2:0> 111 = Interrup •	: MSSP1 SPI/I ot is Priority 7(² C Event Intern highest priority	interrupt)	,				
bit 3 bit 2-0	SSP1IP<2:0> 111 = Interrup • • • • • •	: MSSP1 SPI/I ot is Priority 7(ot is Priority 1	² C Event Interr highest priority	interrupt)	,				

The following code sequence for a clock switch is recommended:

- 1. Disable interrupts during the OSCCON register unlock and write sequence.
- Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8>, in two back-to-back instructions.
- 3. Write the new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
- Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0>, in two back-to-back instructions.
- 5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
- 6. Continue to execute code that is not clock-sensitive (optional).
- 7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
- 8. Check to see if OSWEN is '0'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in Example 9-1.

EXAMPLE 9-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING

;Place the new oscillator selection in WO
;OSCCONH (high byte) Unlock Sequence
MOV #OSCCONH, w1
MOV #0x78, w2
MOV #0x9A, w3
MOV.b w2, [w1]
MOV.b w3, [w1]
;Set new oscillator selection
MOV.b WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV #OSCCONL, w1
MOV #0x46, w2
MOV #0x57, w3
MOV.b w2, [w1]
MOV.b w3, [w1]
;Start oscillator switch operation
BSET OSCCON, #0

9.5 Reference Clock Output

In addition to the CLKO output (Fosc/2) available in certain oscillator modes, the device clock in the PIC24F16KL402 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 9-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the primary modes (EC, HS or XT). Therefore, if the ROSEL bit is also not set, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

REGISTER 11-1: ANSA: PORTA ANALOG SELECTION REGISTER

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/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

Legend:

bit 3-0

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 15-4 Unimplemented: Read as '0'

ANSA<3:0>: Analog Select Control bits

1 = Digital input buffer is not active (use for analog input)

0 = Digital input buffer is active

REGISTER 11-2: ANSB: PORTB ANALOG SELECTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
ANSB15	ANSB14	ANSB13 ⁽¹⁾	ANSB12 ⁽¹⁾	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	—	—	ANSB4	ANSB3 ⁽²⁾	ANSB2 ⁽¹⁾	ANSB1 ⁽¹⁾	ANSB0 ⁽¹⁾
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 15-12	ANSB<15:12>: Analog Select Control bits ⁽¹⁾ 1 = Digital input buffer is not active (use for analog input) 0 = Digital input buffer is active
bit 11-5	Unimplemented: Read as '0'
bit 4-0	<pre>ANSB<4:0>: Analog Select Control bits⁽²⁾ 1 = Digital input buffer is not active (use for analog input) 0 = Digital input buffer is active</pre>

Note 1: ANSB<13:12,2:0> are unimplemented on 14-pin devices.

2: ANSB<3> is unimplemented on 14-pin and 20-pin devices.

REGISTE	ER 16-2: CCP	1CON: ECCP	I CONTROL	REGISTER (ECCP MODU	ILES ONLY)	')
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	_	—	_	_	_	—
bit 15							bit 8
R/W-U	0 R/VV-0	R/W-0	R/W-U	R/VV-U	R/W-U	R/W-U	R/W-U
PIN'I	PMU	DC1B1	DC1B0	CCP1M3	CCP1MZ-	CCP1MIT-	
							Dit U
Legend:							
R = Read	able bit	W = Writable	bit	U = Unimplem	nented bit, read	l as '0'	
-n = Value	e at POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	iown
bit 15-8	Unimpleme	nted: Read as ')'				
bit 7-6	PM<1:0>: Er	nhanced PWM (Dutput Configu	ration bits			
	If CCP1M<3	:2> = 00, 01, 10	<u>.</u>				
	xx = P1A is a	assigned as a ca	pture input or c	ompare output;	P1B, P1C and	P1D are assign	ed as port pins
	<u>If CCP1M<3</u>	: <u>2> = 11:</u>	no: D1D in mo	dulatad: D1C ia		ad D1D are inc.	ativ o
	10 = Half-bri	idae output: P	IA. P1B are	modulated wit	h dead-band	control: P1C	and P1D are
	assigne	ed as port pins					
	01 = Full-bridge output forward: P1D is modulated; P1A is active; P1B, P1C are inactive						
		putput: P1A, P1E	3, P1C and P1L) are controlled	by steering		
dit 5-4	DC1B<1:0>:	PWM Duty Cyc	le bit 1 and bit	0 for CCP1 Mc	dule bits		
	<u>Capture and</u> Unused	Compare mode	<u>s</u> :				
	PWM mode:						
	These bits a	re the two Leas	t Significant bi	ts (bit 1 and bit	0) of the 10-b	it PWM duty cy	cle. The eight
	Most Signific	ant bits (DC1B<	:9:2>) of the du	uty cycle are fou	und in CCPR1L		
bit 3-0	CCP1M<3:0	>: ECCP1 Modu	Ile Mode Selec	ct bits ⁽²⁾			
	1111 = PWN	/I mode: P1A an	d P1C are act	ve-low; P1B an	id P1D are acti	ve-low	
	1100 = PWN	/ mode: P1A an	d P1C are act	ive-high: P1B a	nd P1D are act	tive-low	
	1100 = PWN	/I mode: P1A an	d P1C are act	ve-high; P1B a	nd P1D are ac	tive-high	
	1011 = Com	pare mode: Spe	cial Event Trig	ger; resets time	er on CCP1 ma	atch (CCPxIF bi	it is set)
	1010 = Com	ipare mode: Ger cts I/O state)	nerates softwa	re interrupt on c	compare match	(CCP1IF bit is	set, CCP1 pin
	1001 = Com	pare mode: Initia	alizes CCP1 pi	n high; on comp	are match, for	es CCP1 pin lo	w (CCP1IF bit
	is se	t)					,
	1000 = Com	pare mode: Initi	alizes CCP1 p	oin low; on com	pare match, fo	rces CCP1 pin	high (CCP1IF
	0111 = Cap	ture mode: Ever	v 16th rising e	dae			
	0110 = Capt	ure mode: Ever	y 4th rising ed	ge			
	0101 = Capi	ure mode: Ever	y rising edge				
	0100 = Capi	ure mode: Ever	y falling edge				
	0010 = Com	pare mode: Too	gles output on	match (CCP1I	F bit is set)		
	0001 = Rese	erved	• • • • •	v	/		
	0000 = Capi	ture/Compare/P	WM is disable	d (resets CCP1	module)		
Note 1:	This register is ir configured as Re	nplemented only egister 16-1.	on PIC24FX	KL40X/30X de	evices. For all c	ther devices, C	CP1CON is

2: CCP1M<3:0> = 1011 will only reset the timer and not start the A/D conversion on a CCP1 match.

REGISTER 18-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0, HSC	R-1, HSC	
UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	
bit 15 bit 8								

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
HS = Hardware Settable bit	C = Clearable bit	HSC = Hardware Settable/Cle	earable 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 15,13 UTXISEL<1:0>: UARTx Transmission Interrupt Mode Selection bits

- 11 = Reserved; do not use
- 10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR), and as a result, the transmit buffer becomes empty
- 01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operations are completed
- 00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least one character open in the transmit buffer)

bit 14	UTXINV:	IrDA [®] Encoder	Transmit	Polarity	Inversion	bit
--------	---------	---------------------------	----------	----------	-----------	-----

Sit 11	
	<u>If IREN = 0:</u>
	1 = UxTX Idle '0'
	0 = UxTX Idle '1'
	<u>If IREN = 1:</u>
	1 = UxTX Idle '1'
	0 = UxTX Idle '0'
bit 12	Unimplemented: Read as '0'
bit 11	UTXBRK: UARTx Transmit Break bit
	1 = Sends Sync Break on next transmission – Start bit, followed by twelve '0' bits; followed by Stop bit; cleared by hardware upon completion
	0 = Sync Break transmission is disabled or completed
bit 10	UTXEN: UARTx Transmit Enable bit
	 1 = Transmit is enabled; UxTX pin is controlled by UARTx
	0 = Transmit is disabled; any pending transmission is aborted and the buffer is reset. UxTX pin is controlled by the PORT register.
bit 9	UTXBF: UARTx Transmit Buffer Full Status bit (read-only)
	1 = Transmit buffer is full
	0 = Transmit buffer is not full, at least one more character can be written
bit 8	TRMT: Transmit Shift Register Empty bit (read-only)
	1 = Transmit Shift Register is empty and the transmit buffer is empty (the last transmission has completed)
	0 = Transmit Shift Register is not empty; a transmission is in progress or queued
bit 7-6	URXISEL<1:0>: UARTx Receive Interrupt Mode Selection bits
	 11 = Interrupt is set on the RSR transfer, making the receive buffer full (i.e., has 2 data characters) 10 = Reserved 01 = Reserved

00 = Interrupt is set when any character is received and transferred from the RSR to the receive buffer; receive buffer has one or more characters

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0			
CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0			
bit 15							bit 8			
R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0			
CH0NA				CH0SA3	CH0SA2	CH0SA1	CH0SA0			
bit 7							bit 0			
Legend:										
R = Readab	le bit	W = Writable b	it	U = Unimplen	nented bit, read	l as '0'				
-n = Value a	t POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown			
bit 15 bit 14-12	CH0NB: Char 1 = Channel (0 = Channel (Unimplement	CHONB: Channel 0 Negative Input Select for MUX B Multiplexer Setting bit 1 = Channel 0 negative input is AN1 0 = Channel 0 negative input is VR- Unimplemented: Road as for								
hit 11_8	CH0SB<3:0>: Channel 0 Positive Input Select for MLIX B Multiplexer Setting bits									
	1111 = AN15 1110 = AN14 1101 = AN13 1100 = AN12 1011 = AN11 1010 = AN10 1001 = AN9 1000 = Upper 0111 = Lower 0110 = Intern 0101 = Reser 0100 = AN4 ⁽¹⁾ 0011 = AN3 ⁽¹⁾ 0010 = AN2 ⁽¹⁾ 0010 = AN1 0000 = AN0	(1) (1) r guardband rail r guardband rail lal band gap refe rved; do not use l) l)	(0.785 * VDD (0.215 * VDD erence (VBG))						
bit 7	CH0NA: Char 1 = Channel (0 = Channel (nnel 0 Negative) negative input) negative input	Input Select i is AN1 is VR-	for MUX A Multi	iplexer Setting	bit				
bit 6-4	Unimplemen	ted: Read as '0	,							
bit 3-0	CH0SA<3:0>	: Channel 0 Pos	sitive Input Se	elect for MUX A	Multiplexer Set	tting bits				
	BIT COMDINATIO	ons are identical	to those for (∠низв<3:0> (a	DOVE).					

REGISTER 19-4: AD1CHS: A/D INPUT SELECT REGISTER

Note 1: Unimplemented on 14-pin devices; do not use.



23.0 SPECIAL FEATURES

- Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the Watchdog Timer, High-Level Device Integration and Programming Diagnostics, refer to the individual sections of the "dsPIC33/PIC24 Family Reference Manual" provided below:
 "Watchdog Timer (WDT)" (DS39697)
 - "High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)" (DS39725)
 - "Programming and Diagnostics" (DS39716)

PIC24F16KL402 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection
- In-Circuit Serial Programming[™] (ICSP[™])
- In-Circuit Emulation
- Factory Programmed Unique ID

23.1 Code Protect Security Options

The Boot Segment (BS) and General Segment (GS) are two segments on this device with separate programmable security levels. The Boot Segment, configured via the FBS Configuration register, can have three possible levels of security:

- No Security (BSS = 111): The Boot Segment is not utilized and all addresses in program memory are part of the General Segment (GS).
- Standard Security (BSS = 110 or 101): The Boot Segment is enabled and code-protected, preventing ICSP reads of the Flash memory. Standard security also prevents Flash reads and writes of the BS from the GS. The BS can still read and write to itself.
- High Security (BSS = 010 or 001): The Boot Segment is enabled with all of the security provided by Standard Security mode. In addition, in High-Security mode, there are program flow change restrictions in place. While executing from the GS, program flow changes that attempt to enter the BS (e.g., branch (BRA) or CALL instructions) can only enter the BS at one of the first 32 instruction locations (0x200 to 0x23F). Attempting to jump into the BS at an instruction higher than this will result in an Illegal Opcode Reset.

The General Segment, configured via the FGS Configuration register, can have two levels of security:

- No Security (GSS0 = 1): The GS is not code-protected and can be read in all modes.
- Standard Security (GSS0 = 0): The GS is code-protected, preventing ICSP reads of the Flash memory.

For more detailed information on these Security modes, refer to the *"dsPlC33/PlC24 Family Reference Manual"*, **"CodeGuard™ Security"** (DS70199).

23.2 Configuration Bits

The Configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped starting at program memory location, F80000h. A complete list is provided in Table 23-1. A detailed explanation of the various bit functions is provided in Register 23-1 through Register 23-7.

The address, F80000h, is beyond the user program memory space. In fact, it belongs to the configuration memory space (800000h-FFFFFh), which can only be accessed using Table Reads and Table Writes.

TABLE 23-1:	CONFIGURATION REGISTERS
	LOCATIONS

Configuration Register	Address
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E

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

					r –	
Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
	BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
BTST	BTST	f,#bit4	Bit Test f	1	1	Z
	BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	С
	BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
	BTST.C	Ws,Wb	Bit Test Ws <wb> to C</wb>	1	1	С
	BTST.Z	Ws,Wb	Bit Test Ws <wb> to Z</wb>	1	1	Z
BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
	BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	С
	BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
CALL	CALL	lit23	Call Subroutine	2	2	None
	CALL	Wn	Call Indirect Subroutine	1	2	None
CLR	CLR	f	f = 0x0000	1	1	None
	CLR	WREG	WREG = 0x0000	1	1	None
	CLR	Ws	Ws = 0x0000	1	1	None
CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO, Sleep
COM	COM	f	f = f	1	1	N, Z
	COM	f,WREG	WREG = f	1	1	N. Z
	COM	Wa Wd	$Wd = \overline{Ws}$	1	1	N Z
CP	CP	f	Compare f with WREG	1	1	C DC N OV Z
01	CP	wb #li+5	Compare Wh with lit5	1	1	C, DC, N, OV, Z
	CP	WD We	Compare Wb with Ws (Wb $-$ Ws)	1	1	
CPO	CPO	f	Compare f with 0x0000	1	1	
010	CPO	- Wa	Compare Ws with 0x0000	1	1	C, DC, N, OV, Z
CPB	CPB	f	Compare f with WREG with Borrow	1	1	C, DC, N, OV, Z
010	CPB	wb #li+5	Compare Wh with lit5 with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb Ws	Compare Wb with Ws with Borrow	1	1	C, DC, N, OV, Z
	CID	10,10	$(Wb - Ws - \overline{C})$			0, 20, 11, 01, 2
CPSEQ	CPSEQ	Wb,Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
CPSGT	CPSGT	Wb,Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
CPSLT	CPSLT	Wb,Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
CPSNE	CPSNE	Wb,Wn	Compare Wb with Wn, Skip if \neq	1	1 (2 or 3)	None
DAW	DAW.B	Wn	Wn = Decimal Adjust Wn	1	1	С
DEC	DEC	f	f = f -1	1	1	C, DC, N, OV, Z
	DEC	f,WREG	WREG = f –1	1	1	C, DC, N, OV, Z
	DEC	Ws,Wd	Wd = Ws - 1	1	1	C, DC, N, OV, Z
DEC2	DEC2	f	f = f - 2	1	1	C, DC, N, OV, Z
	DEC2	f,WREG	WREG = f – 2	1	1	C, DC, N, OV, Z
	DEC2	Ws,Wd	Wd = Ws - 2	1	1	C, DC, N, OV, Z
DISI	DISI	#lit14	Disable Interrupts for k Instruction Cycles	1	1	None
DIV	DIV.SW	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UW	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
EXCH	EXCH	Wns, Wnd	Swap Wns with Wnd	1	1	None
FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	С
FF1R	FF1R	Ws.Wnd	Find First One from Right (LSb) Side	1	1	С

TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)

26.1 DC Characteristics





FIGURE 26-2: PIC24F16KL402 FAMILY VOLTAGE-FREQUENCY GRAPH (EXTENDED)



TABLE 26-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Тур	Max	Unit
Operating Junction Temperature Range	TJ	-40	_	+140	°C
Operating Ambient Temperature Range	TA	-40		+125	°C
$\begin{array}{l} \mbox{Power Dissipation:} \\ \mbox{Internal Chip Power Dissipation:} \\ \mbox{PINT} = V \mbox{DD } x \ (\mbox{IDD} - \Sigma \ \mbox{IOH}) \\ \mbox{I/O Pin Power Dissipation:} \\ \mbox{PI/O} = \Sigma \ (\{\mbox{VDD} - \mbox{VOH}\} \ x \ \mbox{IOH}) + \Sigma \ (\mbox{VOL } x \ \mbox{IOL}) \end{array}$	PD	Pint + Pi/o		W	
Maximum Allowed Power Dissipation	PDMAX	(TJ — TA)/θJ	A	W

TABLE 26-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Тур	Max	Unit	Notes
Package Thermal Resistance, 20-Pin PDIP	θJA	62.4		°C/W	1
Package Thermal Resistance, 28-Pin SPDIP	θJA	60	—	°C/W	1
Package Thermal Resistance, 20-Pin SSOP	θJA	108	—	°C/W	1
Package Thermal Resistance, 28-Pin SSOP	θJA	71	—	°C/W	1
Package Thermal Resistance, 20-Pin SOIC	θJA	75	—	°C/W	1
Package Thermal Resistance, 28-Pin SOIC	θJA	80.2	—	°C/W	1
Package Thermal Resistance, 20-Pin QFN	θJA	43	—	°C/W	1
Package Thermal Resistance, 28-Pin QFN	θJA	32	—	°C/W	1
Package Thermal Resistance, 14-Pin PDIP	θJA	62.4	—	°C/W	1
Package Thermal Resistance, 14-Pin TSSOP	θJA	108	_	°C/W	1

Note 1: Junction to ambient thermal resistance, Theta-JA (θ JA) numbers are achieved by package simulations.

TABLE 26-3: DC CHARACTERISTICS: TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CH	ARACTER	$\begin{array}{l} \mbox{Standard Operating Conditions: } 1.8V \mbox{ to } 3.6V \\ \mbox{Operating temperature} & -40^\circ \mbox{C} \leq T \mbox{A} \leq +85^\circ \mbox{C for Industrial} \\ -40^\circ \mbox{C} \leq T \mbox{A} \leq +125^\circ \mbox{C for Extended} \end{array}$					
Para m No.	Symbol	Characteristic	Min Typ ⁽¹⁾ Max Units Conditions				Conditions
DC10	Vdd	Supply Voltage	1.8		3.6	V	
DC12	Vdr	RAM Data Retention Voltage ⁽²⁾	1.5	-	_	V	
DC16	VPOR	VDD Start Voltage to Ensure Internal Power-on Reset Signal	Vss	_	0.7	V	
DC17	SVDD	VDD Rise Rate to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	0-3.3V in 0.1s 0-2.5V in 60 ms
	Vbg	Band Gap Voltage Reference	1.14	1.2	1.26	V	

Note 1: Data in "Typ" column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: This is the limit to which VDD can be lowered without losing RAM data.

DC CHA	ARACTE	RISTICS	Standard Operating Conditions: 1.8V to 3.6VOperating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					
Param No.	Sym	Characteristic	Min	Typ ⁽¹⁾	Max	Units	Con	ditions
	Vol	Output Low Voltage						
DO10		All I/O Pins	—	—	0.4	V	IOL = 4.0 mA	VDD = 3.6V
			—	—	0.4	V	IOL = 3.5 mA	VDD = 2.0V
DO16		OSC2/CLKO	—	—	0.4	V	IOL = 1.2 mA	VDD = 3.6V
			—	—	0.4	V	IOL = 0.4 mA	VDD = 2.0V
	Vон	Output High Voltage						
DO20		All I/O Pins	3	—	—	V	Iон = -3.0 mA	VDD = 3.6V
			1.6	—	—	V	Iон = -1.0 mA	VDD = 2.0V
DO26		OSC2/CLKO	3	—	—	V	Юн = -1.0 mA	VDD = 3.6V
			1.6	—	—	V	Iон = -0.5 mA	VDD = 2.0V

TABLE 26-11: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

Note 1: Data in "Typ" column is at +25°C unless otherwise stated.

TABLE 26-12: DC CHARACTERISTICS: PROGRAM MEMORY

DC CHA	ARACTE	Standard Operating Conditions:1.8V to 3.6VOperating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial $-40^{\circ}C \le TA \le +125^{\circ}C$ for Extended					.8V to 3.6V $A \le +85^{\circ}C$ for Industrial $A \le +125^{\circ}C$ for Extended
Param No.	Sym	Characteristic	Min	Conditions			
-		Program Flash Memory					
D130	Eр	Cell Endurance	10,000 ⁽²⁾	_	—	E/W	
D131	VPR	VDD for Read	VMIN	_	3.6	V	VMIN = Minimum operating voltage
D133A	Tiw	Self-Timed Write Cycle Time	_	2	—	ms	
D134	TRETD	Characteristic Retention	40	_	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current During Programming	_	10	—	mA	

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

2: Self-write and block erase.

28-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

	N	ILLIMETER	S	
Dimensior	Dimension Limits			MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	С		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A