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	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	21
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga102-i-ss

PIC24FJ64GA104 FAMILY

28/44-Pin, 16-Bit General Purpose Flash Microcontrollers with nanoWatt XLP Technology

Power Management Modes:

- Selectable Power Management modes with nanoWatt XLP Technology for Extremely Low Power:
  - Deep Sleep mode allows near total power-down (20 nA typical and 500 nA with RTCC or WDT), along with the ability to wake-up on external triggers, or self-wake on programmable WDT or RTCC alarm
  - Extreme low-power DSBOR for Deep Sleep, LPBOR for all other modes
  - Sleep mode shuts down peripherals and core for substantial power reduction, fast wake-up
  - Idle mode shuts down the CPU and peripherals for significant power reduction, down to 4.5 μA typical
  - Doze mode enables CPU clock to run slower than peripherals
  - Alternate Clock modes allow on-the-fly switching to a lower clock speed for selective power reduction during Run mode, down to 15 μA typical

High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with:
  - 4x PLL option
  - Multiple divide options
- 17-Bit x 17-Bit Single-Cycle Hardware Fractional/integer Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture:
  - 76 base instructions
  - Flexible addressing modes
- Linear Program Memory Addressing, up to 12 Mbytes
- Linear Data Memory Addressing, up to 64 Kbytes
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

Special Microcontroller Features:

- Operating Voltage Range of 2.0V to 3.6V
- Self-Reprogrammable under Software Control
- 5.5V Tolerant Input (digital pins only)
- High-Current Sink/Source (18 mA/18 mA) on All I/O pins

Special Microcontroller Features (continued):

- Flash Program Memory:
  - 10,000 erase/write cycle endurance (minimum)
  - 20-year data retention minimum
  - Selectable write protection boundary
- Fail-Safe Clock Monitor Operation:
  - Detects clock failure and switches to on-chip FRC Oscillator
- On-Chip 2.5V Regulator
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Two Flexible Watchdog Timers (WDT) for Reliable Operation:
  - Standard programmable WDT for normal operation
  - Extreme low-power WDT with programmable period of 2 ms to 26 days for Deep Sleep mode
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins
- JTAG Boundary Scan Support

Analog Features:

- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter:
  - 500 ksps conversion rate
  - Conversion available during Sleep and Idle
- Three Analog Comparators with Programmable Input/Output Configuration
- Charge Time Measurement Unit (CTMU):
  - Supports capacitive touch sensing for touch screens and capacitive switches
  - Provides high-resolution time measurement and simple temperature sensing

PIC24FJ Device	Pins	Program Memory (Bytes)	SRAM (Bytes)	Remappable Peripherals						I²C™	10-Bit A/D (ch)	Comparators	PMP/PSP	RTCC	CTMU
				Remappable Pins	Timers 16-Bit	Capture Input	Compare/PWM Output	UART w/ IrDA®	SPI						
32GA102	28	32K	8K	16	5	5	5	2	2	2	10	3	Y	Y	Y
64GA102	28	64K	8K	16	5	5	5	2	2	2	10	3	Y	Y	Y
32GA104	44	32K	8K	26	5	5	5	2	2	2	13	3	Y	Y	Y
64GA104	44	64K	8K	26	5	5	5	2	2	2	13	3	Y	Y	Y

TABLE 1-2: PIC24FJ64GA104 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/ SOIC/SSOP	28-Pin QFN	44-Pin QFN/ TQFP			
V_{CAP}	20	17	7	P	—	External Filter Capacitor Connection (regulator enabled).
V_{DD}	13, 28	10, 25	28, 40	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins.
V_{DDCORE}	20	17	7	P	—	Positive Supply for Microcontroller Core Logic (regulator disabled).
V_{REF-}	3	28	20	I	ANA	A/D and Comparator Reference Voltage (low) Input.
V_{REF+}	2	27	19	I	ANA	A/D and Comparator Reference Voltage (high) Input.
V_{SS}	8, 27	5, 24	29, 39	P	—	Ground Reference for Logic and I/O Pins.

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

2.7 Configuration of Analog and Digital Pins During ICSP Operations

If an ICSP compliant emulator is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as "digital" pins. Depending on the particular device, this is done by setting all bits in the ADnPCFG register(s), or clearing all bit in the ANSx registers.

All PIC24F devices will have either one or more ADnPCFG registers or several ANSx registers (one for each port); no device will have both. Refer to **Section 21.0 "10-Bit High-Speed A/D Converter"**) for more specific information.

The bits in these registers that correspond to the A/D pins that initialized the emulator must not be changed by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain A/D pins as analog input pins during the debug session, the user application must modify the appropriate bits during initialization of the ADC module, as follows:

- For devices with an ADnPCFG register, clear the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.
- For devices with ANSx registers, set the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.

When a Microchip debugger/emulator is used as a programmer, the user application firmware must correctly configure the ADnPCFG or ANSx registers. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all A/D pins being recognized as analog input pins, resulting in the port value being read as a logic '0', which may affect user application functionality.

2.8 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

3.3.2 DIVIDER

The divide block supports signed and unsigned integer divide operations with the following data sizes:

1. 32-bit signed/16-bit signed divide
2. 32-bit unsigned/16-bit unsigned divide
3. 16-bit signed/16-bit signed divide
4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned `DIV` instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided below in Table 3-2.

TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE AND MULTI-BIT SHIFT OPERATION

Instruction	Description
ASR	Arithmetic shift right source register by one or more bits.
SL	Shift left source register by one or more bits.
LSR	Logical shift right source register by one or more bits.

5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user to erase blocks of eight rows (512 instructions) at a time and to program one row at a time. It is also possible to program single words.

The 8-row erase blocks and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

When data is written to program memory using `TBLWT` instructions, the data is not written directly to memory. Instead, data written using table writes is stored in holding latches until the programming sequence is executed.

Any number of `TBLWT` instructions can be executed and a write will be successfully performed. However, 64 `TBLWT` instructions are required to write the full row of memory.

To ensure that no data is corrupted during a write, any unused addresses should be programmed with FFFFFFh. This is because the holding latches reset to an unknown state, so if the addresses are left in the Reset state, they may overwrite the locations on rows which were not rewritten.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of `TBLWT` instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

Note:	Writing to a location multiple times without erasing is *not* recommended.

All of the table write operations are single-word writes (2 instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

5.3 JTAG Operation

The PIC24F family supports JTAG boundary scan. Boundary scan can improve the manufacturing process by verifying pin to PCB connectivity.

5.4 Enhanced In-Circuit Serial Programming

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

5.5 Control Registers

There are two SFRs used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. Refer to **Section 5.6 “Programming Operations”** for further details.

5.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

7.3 Interrupt Control and Status Registers

The PIC24FJ64GA104 family of devices implements the following registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS4
- IEC0 through IEC4
- IPC0 through IPC20 (except IPC13, IPC14 and IPC17)
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the Alternate Interrupt Vector Table.

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit which is set by the respective peripherals, or an external signal, and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the interrupt priority level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the order of their vector numbers, as shown in Table 7-2. For example, the INT0 (External Interrupt 0) is shown as having a vector number and a natural order priority of 0. Thus, the INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU control registers contain bits that control interrupt functionality. The ALU STATUS Register (SR) contains the IPL<2:0> bits (SR<7:5>); these indicate the current CPU interrupt priority level. The user may change the current CPU priority level by writing to the IPL bits.

The CORCON register contains the IPL3 bit, which, together with IPL<2:0>, indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

The interrupt controller has the Interrupt Controller Test Register (INTTREG) that displays the status of the interrupt controller. When an interrupt request occurs, its associated vector number and the new interrupt priority level are latched into INTTREG.

This information can be used to determine a specific interrupt source if a generic ISR is used for multiple vectors – such as when ISR remapping is used in bootloader applications. It also could be used to check if another interrupt is pending while in an ISR.

All interrupt registers are described in Register 7-1 through Register 7-32, on the following pages.

REGISTER 7-29: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
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 15 **Unimplemented:** Read as '0'

bit 14-12 **CRCIP<2:0>:** CRC Generator Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **U2ERIP<2:0>:** UART2 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

bit 6-4 **U1ERIP<2:0>:** UART1 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

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

REGISTER 7-30: IPC18: INTERRUPT PRIORITY CONTROL REGISTER 18

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	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0
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 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **LVDIP<2:0>:** Low-Voltage Detect Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-31: IPC19: INTERRUPT PRIORITY CONTROL REGISTER 19

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—
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 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **CTMUIP<2:0>:** CTMU Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as ‘0’

NOTES:

10.3 Input Change Notification

The input change notification function of the I/O ports allows the PIC24FJ64GA104 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 31 external inputs that may be selected (enabled) for generating an interrupt request on a Change-of-State.

Registers, CNEN1 and CNEN2, contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin has a weak pull-up connected to it. The pull-up acts as a current source that is connected to the pin. This eliminates the need for external resistors when push button or keypad devices are connected. The pull-ups are separately enabled using the CNPU1 and CNPU2 registers (for pull-ups). Each CN pin has individual control bits for its pull-up. Setting a control bit enables the weak pull-up for the corresponding pin.

When the internal pull-up is selected, the pin pulls up to V_{DD} – 0.7V (typical). Make sure that there is no external pull-up source when the internal pull-ups are enabled, as the voltage difference can cause a current path.

Note:	Pull-ups on change notification pins should always be disabled whenever the port pin is configured as a digital output.

10.4 Peripheral Pin Select (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features while minimizing the conflict of features on I/O pins. In an application that needs to use more than one peripheral multiplexed on a single pin, inconvenient work arounds in application code or a complete redesign may be the only option.

The Peripheral Pin Select feature provides an alternative to these choices by enabling the user's peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of any one of many digital peripherals to any one of these I/O pins. Peripheral Pin Select is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

10.4.1 AVAILABLE PINS

The Peripheral Pin Select feature is used with a range of up to 25 pins, depending on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the designation "RPn" in their full pin designation, where "n" is the remappable pin number.

See Table 1-2 for a summary of pinout options in each package offering.

10.4.2 AVAILABLE PERIPHERALS

The peripherals managed by the Peripheral Pin Select are all digital only peripherals. These include general serial communications (UART and SPI), general purpose timer clock inputs, timer related peripherals (input capture and output compare) and external interrupt inputs. Also included are the outputs of the comparator module, since these are discrete digital signals.

Peripheral Pin Select is not available for I^{2}C™ change notification inputs, RTCC alarm outputs or peripherals with analog inputs.

A key difference between pin select and non pin select peripherals is that pin select peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non pin select peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

10.4.2.1 Peripheral Pin Select Function Priority

Pin-selectable peripheral outputs (for example, OC and UART transmit) take priority over any general purpose digital functions permanently tied to that pin, such as PMP and port I/O. Specialized digital outputs, such as USB functionality, take priority over PPS outputs on the same pin. The pin diagrams at the beginning of this data sheet list peripheral outputs in order of priority. Refer to them for priority concerns on a particular pin.

Unlike devices with fixed peripherals, pin-selectable peripheral inputs never take ownership of a pin. The pin's output buffer is controlled by the pin's TRIS bit setting, or by a fixed peripheral on the pin. If the pin is configured in Digital mode, then the PPS input will operate correctly, reading the input. If an analog function is enabled on the same pin, the pin-selectable input will be disabled.

REGISTER 12-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER(3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32(1)	—	TCS(2)	—
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 15 **TON:** Timerx On bit
When TxCON<3> = 1:
1 = Starts 32-bit Timerx/y
0 = Stops 32-bit Timerx/y
When TxCON<3> = 0:
1 = Starts 16-bit Timerx
0 = Stops 16-bit Timerx

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as ‘0’

bit 6 **TGATE:** Timerx Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timerx Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **T32:** 32-Bit Timer Mode Select bit(1)
1 = Timerx and Timery form a single 32-bit timer
0 = Timerx and Timery act as two 16-bit timers
In 32-bit mode, T3CON control bits do not affect 32-bit timer operation.

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timerx Clock Source Select bit(2)
1 = External clock from pin, TxCK (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: In 32-bit mode, the T3CON or T5CON control bits do not affect 32-bit timer operation.
2: If TCS = 1, RPINRx (TxCK) must be configured to an available RPn pin. For more information, see **Section 10.4 “Peripheral Pin Select (PPS)”**.
3: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

REGISTER 15-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER (CONTINUED)

bit 1	**SPITBF:** SPIx Transmit Buffer Full Status bit 1 = Transmit not yet started; SPIxTXB is full 0 = Transmit started; SPIxTXB is empty In Standard Buffer mode: Automatically set in hardware when CPU writes SPIxBUF location, loading SPIxTXB. Automatically cleared in hardware when SPIx module transfers data from SPIxTXB to SPIxSR. In Enhanced Buffer mode: Automatically set in hardware when CPU writes SPIxBUF location, loading the last available buffer location. Automatically cleared in hardware when a buffer location is available for a CPU write.
bit 0	**SPIRBF:** SPIx Receive Buffer Full Status bit 1 = Receive is complete, SPIxRXB is full 0 = Receive is not complete, SPIxRXB is empty In Standard Buffer mode: Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB. Automatically cleared in hardware when core reads SPIxBUF location, reading SPIxRXB. In Enhanced Buffer mode: Automatically set in hardware when SPIx transfers data from SPIxSR to buffer, filling the last unread buffer location. Automatically cleared in hardware when a buffer location is available for a transfer from SPIxSR.

Note 1: If SPIEN = 1, these functions must be assigned to available RPn pins before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

17.1 UART Baud Rate Generator (BRG)

The UART module includes a dedicated 16-bit Baud Rate Generator. The UxBRG register controls the period of a free-running, 16-bit timer. Equation 17-1 shows the formula for computation of the baud rate with BRGH = `0`.

EQUATION 17-1: UART BAUD RATE WITH BRGH = `0`[1,2]

$$\text{Baud Rate} = \frac{F_{CY}}{16 \bullet (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{16 \bullet \text{Baud Rate}} - 1$$

Note 1: F_{CY} denotes the instruction cycle clock frequency ($F_{OSC}/2$).

2: Based on $F_{CY} = F_{OSC}/2$, Doze mode and PLL are disabled.

Example 17-1 shows the calculation of the baud rate error for the following conditions:

- F_{CY} = 4 MHz
- Desired Baud Rate = 9600

The maximum baud rate (BRGH = `0`) possible is $F_{CY}/16$ (for UxBRG = `0`) and the minimum baud rate possible is $F_{CY}/(16 * 65536)$.

Equation 17-2 shows the formula for computation of the baud rate with BRGH = `1`.

EQUATION 17-2: UART BAUD RATE WITH BRGH = `1`[1,2]

$$\text{Baud Rate} = \frac{F_{CY}}{4 \bullet (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{4 \bullet \text{Baud Rate}} - 1$$

Note 1: F_{CY} denotes the instruction cycle clock frequency.

2: Based on $F_{CY} = F_{OSC}/2$, Doze mode and PLL are disabled.

The maximum baud rate (BRGH = `1`) possible is $F_{CY}/4$ (for UxBRG = `0`) and the minimum baud rate possible is $F_{CY}/(4 * 65536)$.

Writing a new value to the UxBRG register causes the BRG timer to be reset (cleared). This ensures the BRG does not wait for a timer overflow before generating the new baud rate.

EXAMPLE 17-1: BAUD RATE ERROR CALCULATION (BRGH = `0`)[1]

Desired Baud Rate = $F_{CY}/(16\ (\text{UxBRG} + 1))$

Solving for UxBRG Value:

UxBRG = $((F_{CY}/\text{Desired Baud Rate})/16) - 1$
UxBRG = ((4000000/9600)/16) – 1
UxBRG = 25

Calculated Baud Rate = 4000000/(16 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate) / Desired Baud Rate
= (9615 – 9600)/9600
= 0.16%

Note 1: Based on $F_{CY} = F_{OSC}/2$, Doze mode and PLL are disabled.

FIGURE 19-2: ALARM MASK SETTINGS

Alarm Mask Setting (AMASK<3:0>)	Day of the Week	Month	Day	Hours	Minutes	Seconds
`0000` - Every half second `0001` - Every second						
`0010` - Every 10 seconds						s
`0011` - Every minute						s s
`0100` - Every 10 minutes					m	s s
`0101` - Every hour					m m	s s
`0110` - Every day				h h	m m	s s
`0111` - Every week	d			h h	m m	s s
`1000` - Every month			d d	h h	m m	s s
`1001` - Every year(1)		m m	d d	h h	m m	s s

Note 1: Annually, except when configured for February 29.

REGISTER 21-5: AD1PCFG: A/D PORT CONFIGURATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0(1)	R/W-0	R/W-0	R/W-0	R/W-0(1)
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

R/W-0(1)	R/W-0(1)	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
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 15 **PCFG15:** A/D Input Band Gap Reference Enable bit
1 = Internal band gap (V_{BG}) reference channel is disabled
0 = Internal band gap reference channel is enabled

bit 14 **PCFG14:** A/D Input Half Band Gap Reference Enable bit
1 = Internal half band gap ($V_{BG}/2$) reference channel is disabled
0 = Internal half band gap reference channel is enabled

bit 13 **PCFG13:** A/D Input Voltage Regulator Output Reference Enable bit
1 = Internal voltage regulator output (V_{DDCORE}) reference channel is disabled
0 = Internal voltage regulator output reference channel is enabled

bit 12-0 **PCFG<12:0>:** Analog Input Pin Configuration Control bits(1)
1 = Pin for corresponding analog channel is configured in Digital mode; I/O port read is enabled
0 = Pin is configured in Analog mode; I/O port read is disabled, A/D samples pin voltage

Note 1: Analog channels, AN6, AN7, AN8 and AN12, are unavailable on 28-pin devices; leave these corresponding bits set.

REGISTER 21-6: AD1CSSL: A/D INPUT SCAN SELECT REGISTER

R/W-0	R/W-0	R/W-0	R/W-0(1)	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8(1)
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
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
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 15 **CSSL15:** A/D Input Band Gap Scan Enable bit
1 = Internal band gap (VBG) channel is enabled for input scan
0 = Analog channel is disabled from input scan

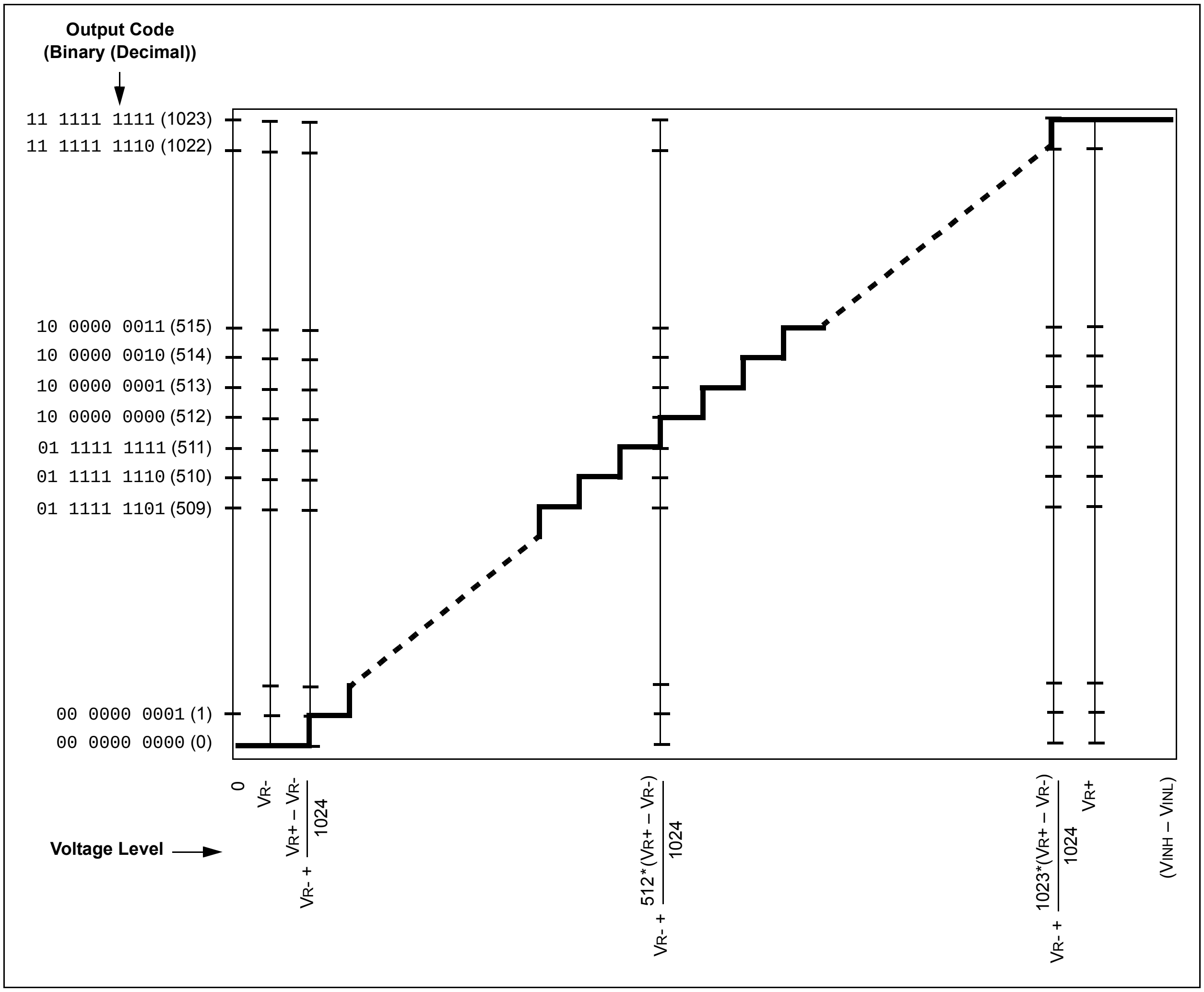
bit 14 **CSSL14:** A/D Input Half Band Gap Scan Enable bit
1 = Internal half band gap (VBG/2) channel is enabled for input scan
0 = Analog channel is disabled from input scan

bit 13 **CSSL13:** A/D Input Voltage Regulator Output Scan Enable bit
1 = Internal voltage regulator output (VDDCORE) is enabled for input scan
0 = Analog channel is disabled from input scan

bit 12-0 **CSSL<12:0>:** A/D Input Pin Scan Selection bits(1)
1 = Corresponding analog channel is selected for input scan
0 = Analog channel is omitted from input scan

Note 1: Analog channels, AN6, AN7, AN8 and AN12, are unavailable on 28-pin devices; leave these corresponding bits cleared.

FIGURE 21-3: A/D TRANSFER FUNCTION



27.0 INSTRUCTION SET SUMMARY

Note:	This chapter is a brief summary of the PIC24F instruction set architecture, and is not intended to be a comprehensive reference source.

The PIC24F instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from previous PIC MCU instruction sets. Most instructions are a single program memory word. Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction. The instruction set is highly orthogonal and is grouped into four basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

Table 27-1 shows the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 27-2 lists all of the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register ‘Wb’ without any address modifier
- The second source operand, which is typically a register ‘Ws’ with or without an address modifier
- The destination of the result, which is typically a register ‘Wd’ with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value, ‘f’
- The destination, which could either be the file register, ‘f’, or the W0 register, which is denoted as ‘WREG’

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of ‘Ws’ or ‘f’)
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register, ‘Wb’)

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of ‘k’)
- The W register or file register where the literal value is to be loaded (specified by ‘Wb’ or ‘f’)

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register ‘Wb’ without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register ‘Wd’ with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSbs are ‘0’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes, and RETURN/RETFIE instructions, which are single-word instructions but take two or three cycles.

Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

TABLE 28-9: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Sym	Characteristic	Min	Typ(1)	Max	Units	Conditions
	VOL	**Output Low Voltage**					
DO10		I/O Ports	—	—	0.4	V	IOL = 8.5 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2.0V
DO16		I/O Ports	—	—	0.4	V	IOL = 8.0 mA, VDD = 3.6V, 125°C
			—	—	0.4	V	IOL = 4.5 mA, VDD = 2.0V, 125°C
	VOH	**Output High Voltage**					
DO20		I/O Ports	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.65	—	—	V	IOH = -1.0 mA, VDD = 2.0V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2.0V
DO26		I/O Ports	3.0	—	—	V	IOH = -2.5 mA, VDD = 3.6V, 125°C
			1.65	—	—	V	IOH = -0.5 mA, VDD = 2.0V, 125°C

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.

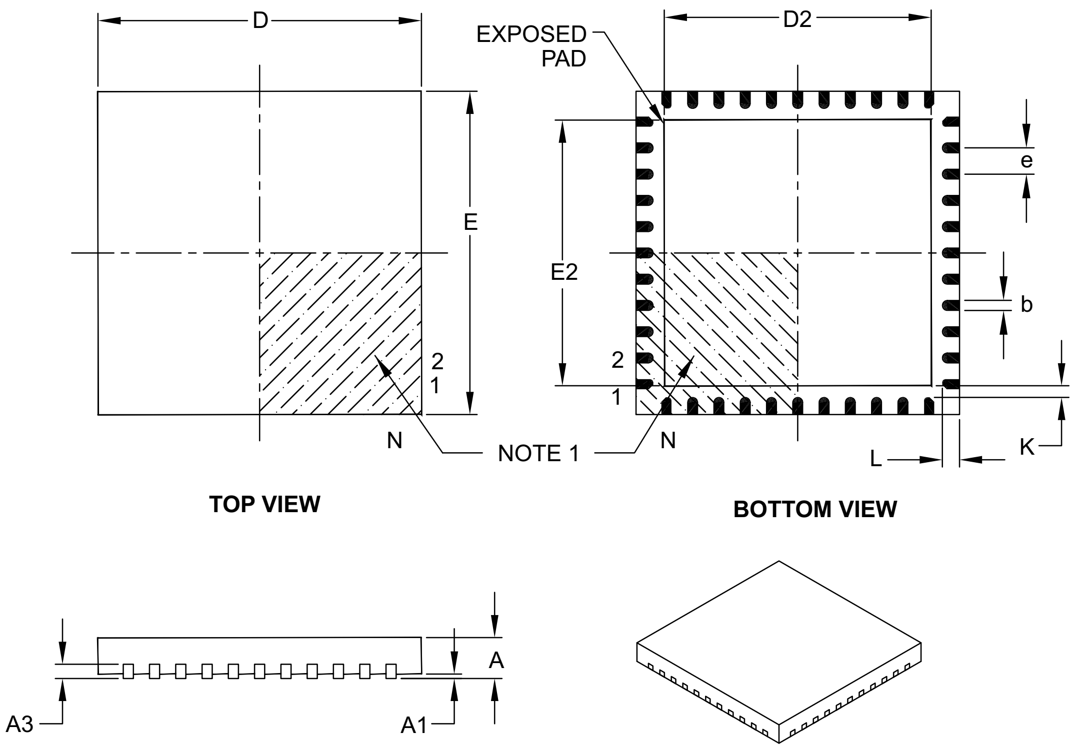
TABLE 28-10: DC CHARACTERISTICS: PROGRAM MEMORY

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Sym	Characteristic	Min	Typ(1)	Max	Units	Conditions
D130	EP	Cell Endurance	10,000	—	—	E/W	-40°C to +85°C
D131	VPR	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
	VPEW	Supply Voltage for Self-Timed Writes					
D132A		VDDCORE	2.25	—	3.6	V	
D132B		VDD	2.35	—	3.6	V	
D133A	TIW	Self-Timed Write Cycle Time	—	3	—	ms	
D133B	TIE	Self-Timed Page Erase Time	40	—	—	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	7	—	mA	

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

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		44	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		8.00 BSC	
Exposed Pad Width	E2	6.30	6.45	6.80
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.30	6.45	6.80
Contact Width	b	0.25	0.30	0.38
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

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

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B