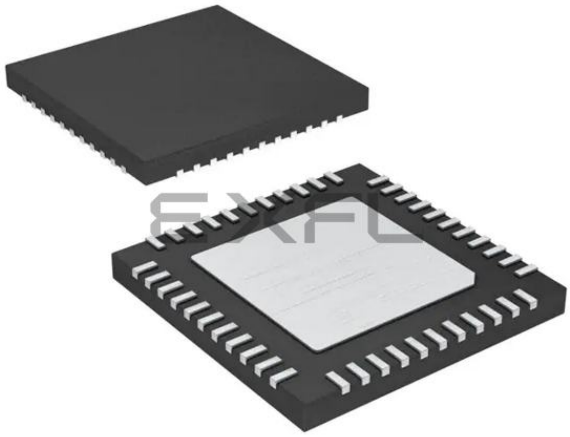


Welcome to E-XFL.COM

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, LINbus, PMP/PSP, SmartCard, SPI, UART/USART
Peripherals	AES, Brown-out Detect/Reset, DMA, I²S, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 13x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga204t-i-ml

TABLE 4-4: ICN REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CNPD1	0056	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE(1)	CN9PDE(1)	CN8PDE(1)	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	—	CN30PDE	CN29PDE	CN28PDE(1)	CN27PDE	CN26PDE(1)	CN25PDE(1)	CN24PDE	CN23PDE	CN22PDE	CN21PDE	CN20PDE(1)	CN19PDE(1)	CN18PDE(1)	CN17PDE(1)	CN16PDE	0000
CNPD3	005A	—	—	—	—	—	—	—	—	—	—	—	CN36PDE(1)	CN35PDE(1)	CN34PDE(1)	CN33PDE(1)	—	0000
CNEN1	0062	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE(1)	CN9IE(1)	CN8IE(1)	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	—	CN30IE	CN29IE	CN28IE(1)	CN27IE	CN26IE(1)	CN25IE(1)	CN24IE	CN23IE	CN22IE	CN21IE	CN20IE(1)	CN19IE(1)	CN18IE(1)	CN17IE(1)	CN16IE	0000
CNEN3	0066	—	—	—	—	—	—	—	—	—	—	—	CN36IE(1)	CN35IE(1)	CN34IE(1)	CN33IE(1)	—	0000
CNPU1	006E	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE(1)	CN9PUE(1)	CN8PUE(1)	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	—	CN30PUE	CN29PUE	CN28PUE(1)	CN27PUE	CN26PUE(1)	CN25PUE(1)	CN24PUE	CN23PUE	CN22PUE	CN21PUE	CN20PUE(1)	CN19PUE(1)	CN18PUE(1)	CN17PUE(1)	CN16PUE	0000
CNPU3	0072	—	—	—	—	—	—	—	—	—	—	—	CN36PUE(1)	CN35PUE(1)	CN34PUE(1)	CN33PUE(1)	—	0000

Legend: — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

Note 1: These bits are unimplemented in 28-pin devices, read as ‘0’.

TABLE 4-23: REAL-TIME CLOCK AND CALENDAR (RTCC) REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ALRMVAL	011E	Alarm Value Register Window Based on ALRMPTR<1:0>																xxxx
ALCFGRPT	0120	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0122	RTCC Value Register Window Based on RTCPTR<1:0>																xxxx
RCFGCAL	0124	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	Note 1
RTCPWC	0126	PWCEN	PWCPOL	PWCPRE	PWSPRE	RTCLK1	RTCLK0	RTCOUT1	RTCOUT0	—	—	—	—	—	—	—	—	Note 1

Legend: — = unimplemented, read as ‘0’; x = unknown value on Reset. Reset values are shown in hexadecimal.

Note 1: The status of the RCFGCAL and RTCPWC registers on POR is ‘0000’ and on other Resets, it is unchanged.

TABLE 4-24: DATA SIGNAL MODULATOR (DSM) REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
MDCON	02FA	MDEN	—	MDSIDL	—	—	—	—	—	—	MDOE	MDSLR	MDOPOL	—	—	—	MDBIT	0020
MDSRC	02FC	—	—	—	—	—	—	—	—	SODIS	—	—	—	MS3	MS2	MS1	MS0	0000
MDCAR	02FE	CHODIS	CHPOL	CHSYNC	—	CH3	CH2	CH1	CH0	CLODIS	CLPOL	CLSYNC	—	CL3	CL2	CL1	CL0	0000

Legend: — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

TABLE 4-25: COMPARATOR REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CMSTAT	0242	CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT	—	—	—	—	—	C3OUT	C2OUT	C1OUT	0000
CVRCON	0244	—	—	—	—	—	CVREFP	CVREFM1	CVREFM0	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0246	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM2CON	0248	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	024A	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000

Legend: — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

4.3.2 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

The TBLRDL and TBLWTL instructions offer a direct method of reading or writing the lower word of any address within the program space without going through Data Space. The TBLRDH and TBLWTH instructions are the only method to read or write the upper 8 bits of a program space word as data.

The PC is incremented by two for each successive 24-bit program word. This allows program memory addresses to directly map to Data Space addresses. Program memory can thus be regarded as two, 16-bit word-wide address spaces, residing side by side, each with the same address range. TBLRDL and TBLWTL access the space which contains the least significant data word, and TBLRDH and TBLWTH access the space which contains the upper data byte.

Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

1. TBLRDL (Table Read Low): In Word mode, it maps the lower word of the program space location (P<15:0>) to a data address (D<15:0>).

   In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when byte select is '1'; the lower byte is selected when it is '0'.

2. TBLRDH (Table Read High): In Word mode, it maps the entire upper word of a program address (P<23:16>) to a data address. Note that D<15:8>, the 'phantom' byte, will always be '0'.

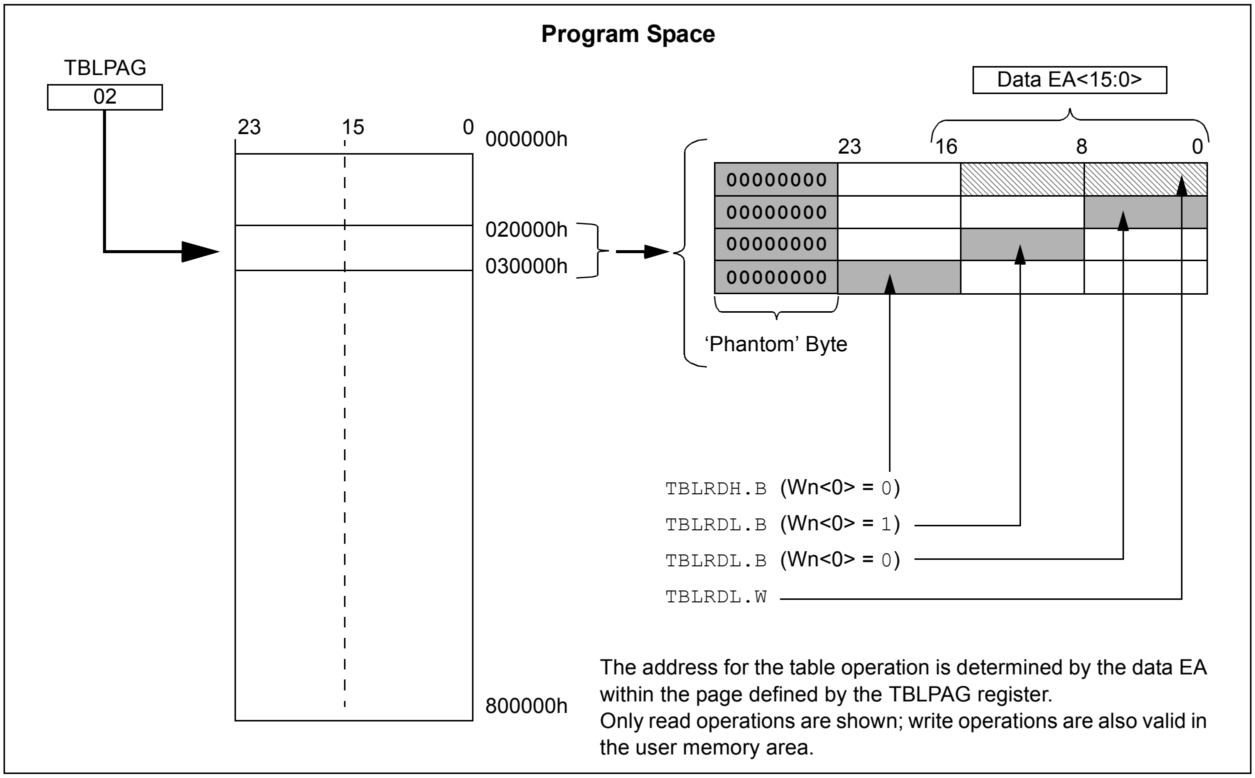
   In Byte mode, it maps the upper or lower byte of the program word to D<7:0> of the data address, as above. Note that the data will always be '0' when the upper 'phantom' byte is selected (Byte Select = 1).

In a similar fashion, two table instructions, TBLWTH and TBLWTL, are used to write individual bytes or words to a program space address. The details of their operation are described in **Section 6.0 "Flash Program Memory"**.

For all table operations, the area of program memory space to be accessed is determined by the Table Memory Page Address register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user and configuration spaces. When TBLPAG<7> = 0, the table page is located in the user memory space. When TBLPAG<7> = 1, the page is located in configuration space.

Note:	Only Table Read operations will execute in the configuration memory space where Device IDs are located; Table Write operations are not allowed.

FIGURE 4-9: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



The address for the table operation is determined by the data EA within the page defined by the TBLPAG register.
Only read operations are shown; write operations are also valid in the user memory area.

5.1.6 CHANNEL PRIORITY

Each DMA channel functions independently of the others, but also competes with the others for access to the data and DMA buses. When access collisions occur, the DMA Controller arbitrates between the channels using a user-selectable priority scheme. Two schemes are available:

- Round Robin: When two or more channels collide, the lower numbered channel receives priority on the first collision. On subsequent collisions, the higher numbered channels each receive priority based on their channel number.
- Fixed Priority: When two or more channels collide, the lowest numbered channel always receives priority, regardless of past history.

5.2 Typical Setup

To set up a DMA channel for a basic data transfer:

1. Enable the DMA Controller (DMAEN = `1`) and select an appropriate channel priority scheme by setting or clearing PRSSEL.
2. Program DMAH and DMAL with appropriate upper and lower address boundaries for data RAM operations.
3. Select the DMA channel to be used and disable its operation (CHEN = `0`).
4. Program the appropriate source and destination addresses for the transaction into the channel's DMASRCn and DMADSTn registers. For PIA Addressing mode, use the base address value.
5. Program the DMACNTn register for the number of triggers per transfer (One-Shot or Continuous modes) or the number of words (bytes) to be transferred (Repeated modes).
6. Set or clear the SIZE bit to select the data size.
7. Program the TRMODE<1:0> bits to select the Data Transfer mode.
8. Program the SAMODE<1:0> and DAMODE<1:0> bits to select the addressing mode.
9. Enable the DMA channel by setting CHEN.
10. Enable the trigger source interrupt.

5.3 Peripheral Module Disable

Unlike other peripheral modules, the channels of the DMA Controller cannot be individually powered down using the Peripheral Module Disable x (PMDx) registers. Instead, the channels are controlled as two groups. The DMA0MD bit (PMD7<4>) selectively controls DMACH0 through DMACH3. The DMA1MD bit (PMD7<5>) controls DMACH4 and DMACH5. Setting both bits effectively disables the DMA Controller.

5.4 Registers

The DMA Controller uses a number of registers to control its operation. The number of registers depends on the number of channels implemented for a particular device.

There are always four module-level registers (one control and three buffer/address):

- DMACON: DMA Control Register (Register 5-1)
- DMAH and DMAL: DMA High and Low Address Limit Registers
- DMABUF: DMA Transfer Data Buffer

Each of the DMA channels implements five registers (two control and three buffer/address):

- DMACHn: DMA Channel n Control Register (Register 5-2)
- DMAINTn: DMA Channel n Interrupt Control Register (Register 5-3)
- DMASRCn: DMA Data Source Address Pointer for Channel n Register
- DMADSTn: DMA Data Destination Source for Channel n Register
- DMACNTn: DMA Transaction Counter for Channel n Register

For PIC24FJ128GA204 family devices, there are a total of 34 registers.

6.0 FLASH PROGRAM MEMORY

> **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, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Program Memory"** (DS39715). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ128GA204 family of devices contains internal Flash program memory for storing and executing application code. The program memory is readable, writable and erasable. The Flash memory can be programmed in four ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self-Programming (RTSP)
- JTAG
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24FJ128GA204 family device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (named PGECx and PGEDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear ($\overline{\text{MCLR}}$). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

RTSP is accomplished using `TBLRD` (Table Read) and `TBLWT` (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 64 instructions (192 bytes) at a time and erase program memory in blocks of 512 instructions (1536 bytes) at a time.

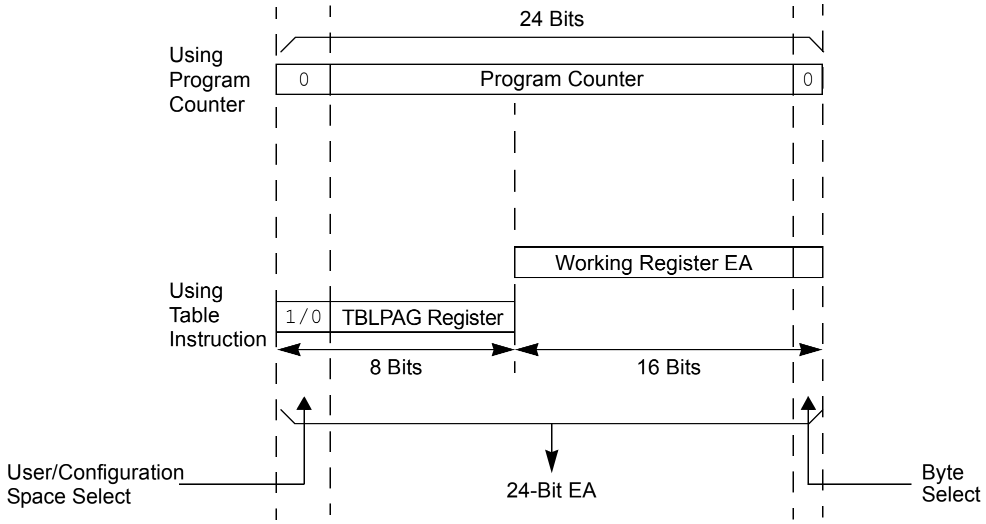
6.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 6-1.

The `TBLRDL` and the `TBLWTL` instructions are used to read or write to bits<15:0> of program memory. `TBLRDL` and `TBLWTL` can access program memory in both Word and Byte modes.

The `TBLRDH` and `TBLWTH` instructions are used to read or write to bits<23:16> of program memory. `TBLRDH` and `TBLWTH` can also access program memory in Word or Byte mode.

FIGURE 6-1: ADDRESSING FOR TABLE REGISTERS



REGISTER 8-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	DMA4IF	PMPIF	—	—	OC6IF	OC5IF	IC6IF
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
IC5IF	IC4IF	IC3IF	DMA3IF	CRYROLLIF	CRYFREEIF	SPI2TXIF	SPI2IF
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 **DMA4IF:** DMA Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12-11 **Unimplemented:** Read as ‘0’

bit 10 **OC6IF:** Output Compare Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC5IF:** Output Compare Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **IC6IF:** Input Capture Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **IC5IF:** Input Capture Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC4IF:** Input Capture Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC3IF:** Input Capture Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **DMA3IF:** DMA Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **CRYROLLIF:** Cryptographic Rollover Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **CRYFREEIF:** Cryptographic Buffer Free Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 8-11: IFS6: INTERRUPT FLAG STATUS REGISTER 6

U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0
—	—	—	—	—	FSTIF	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 15-11 **Unimplemented:** Read as ‘0’

bit 10 **FSTIF:** FRC Self-Tune Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9-0 **Unimplemented:** Read as ‘0’

REGISTER 8-12: IFS7: INTERRUPT FLAG STATUS REGISTER 7

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

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	JTAGIF	—	—	—	—	—
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 15-6 **Unimplemented:** Read as ‘0’

bit 5 **JTAGIF:** JTAG Controller Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4-0 **Unimplemented:** Read as ‘0’

REGISTER 8-16: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3 (CONTINUED)

bit 1	**SI2C2IE:** Slave I2C2 Event Interrupt Enable bit `1` = Interrupt request is enabled `0` = Interrupt request is not enabled
bit 0	**Unimplemented:** Read as ‘`0`’

Note 1: If an external interrupt is enabled, the interrupt input must also be configured to an available RPn or RPIn pin. For more information, see **Section 11.4 “Peripheral Pin Select (PPS)”**.

8.4 Interrupt Setup Procedures

8.4.1 INITIALIZATION

To configure an interrupt source:

1. Set the NSTDIS (INTCON1<15>) control bit if nested interrupts are not desired.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits for all enabled interrupt sources may be programmed to the same non-zero value.

Note:	At a device Reset, the IPCx registers are initialized, such that all user interrupt sources are assigned to Priority Level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

8.4.2 INTERRUPT SERVICE ROUTINE (ISR)

The method that is used to declare an Interrupt Service Routine (ISR) and initialize the IVT with the correct vector address will depend on the programming language (i.e., ‘C’ or assembler), and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles; otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

8.4.3 TRAP SERVICE ROUTINE (TSR)

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

8.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to Priority Level 7 by inclusive ORing the value, 0Eh, with SRL.

To enable user interrupts, the `POP` instruction may be used to restore the previous SR value.

Note that only user interrupts with a priority level of 7 or less can be disabled. Trap sources (Levels 8-15) cannot be disabled.

The `DISI` instruction provides a convenient way to disable interrupts of Priority Levels 1-6 for a fixed period of time. Level 7 interrupt sources are not disabled by the `DISI` instruction.

18.2 Transmitting in 8-Bit Data Mode

1. Set up the UARTx:
   a) Write appropriate values for data, parity and Stop bits.
   b) Write appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UARTx.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
4. Write a data byte to the lower byte of the UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with the next rising edge of the baud clock.
5. Alternatively, the data byte may be transferred while UTXEN = `0` and then, the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated as per interrupt control bits, UTXISEL<1:0>.

18.3 Transmitting in 9-Bit Data Mode

1. Set up the UARTx (as described in **Section 18.2 “Transmitting in 8-Bit Data Mode”**).
2. Enable the UARTx.
3. Set the UTXEN bit (causes a transmit interrupt).
4. Write UxTXREG as a 16-bit value only.
5. A word write to UxTXREG triggers the transfer of the 9-bit data to the TSR. The serial bit stream will start shifting out with the first rising edge of the baud clock.
6. A transmit interrupt will be generated as per the setting of control bits, UTXISELx.

18.4 Break and Sync Transmit Sequence

The following sequence will send a message frame header, made up of a Break, followed by an auto-baud Sync byte.

1. Configure the UARTx for the desired mode.
2. Set UTXEN and UTXBRK to set up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write ‘55h’ to UxTXREG; this loads the Sync character into the transmit FIFO.
5. After the Break has been sent, the UTXBRK bit is reset by hardware. The Sync character now transmits.

18.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UARTx (as described in **Section 18.2 “Transmitting in 8-Bit Data Mode”**).
2. Enable the UARTx.
3. Set the URXEN bit (UxSTA<12>).
4. A receive interrupt will be generated when one or more data characters have been received as per interrupt control bits, URXISEL<1:0>.
5. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
6. Read UxRXREG.

The act of reading the UxRXREG character will move the next character to the top of the receive FIFO, including a new set of PERR and FERR values.

18.6 Operation of $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Control Pins

UARTx Clear-to-Send ($\overline{\text{UxCTS}}$) and Request-to-Send ($\overline{\text{UxRTS}}$) are the two hardware controlled pins that are associated with the UARTx modules. These two pins allow the UARTx to operate in Simplex and Flow Control mode. They are implemented to control the transmission and reception between the Data Terminal Equipment (DTE). The UEN<1:0> bits in the UxMODE register configure these pins.

18.7 Infrared Support

The UARTx module provides two types of infrared UART support: one is the IrDA clock output to support an external IrDA encoder and decoder device (legacy module support), and the other is the full implementation of the IrDA encoder and decoder. Note that because the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is ‘`0`’.

18.7.1 IrDA CLOCK OUTPUT FOR EXTERNAL IrDA SUPPORT

To support external IrDA encoder and decoder devices, the BCLKx pin (same as the $\overline{\text{UxRTS}}$ pin) can be configured to generate the 16x baud clock. With UEN<1:0> = `11`, the BCLKx pin will output the 16x baud clock if the UARTx module is enabled. It can be used to support the IrDA codec chip.

21.2 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

21.2.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through corresponding Register Pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTR<1:0> bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 21-1).

By writing the RTCVALH byte, the RTCC Pointer value, the RTCPTR<1:0> bits decrement by one until they reach '00'. Once they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 21-1: RTCVAL REGISTER MAPPING

RTCPTR<1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR<1:0> bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 21-2).

By writing the ALRMVALH byte, the ALRMPTR<1:0> bits (the Alarm Pointer value) decrement by one until they reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 21-2: ALRMVAL REGISTER MAPPING

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes, the ALRMPTR<1:0> value will be decremented. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

> **Note:** This only applies to read operations and not write operations.

21.2.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RCFGCAL<13>) must be set (see Example 21-1).

> **Note:** To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 21-1.

21.2.3 SELECTING RTCC CLOCK SOURCE

The clock source for the RTCC module can be selected using the RTCLK<1:0> bits in the RTCPWC register. When the bits are set to '00', the Secondary Oscillator (SOSC) is used as the reference clock and when the bits are '01', LPRC is used as the reference clock. When RTCLK<1:0> = 10 and 11, the external power line (50 Hz and 60 Hz) is used as the clock source.

EXAMPLE 21-1: SETTING THE RTCWREN BIT

```
asm    volatile(“push w7”);
asm    volatile(“push w8”);
asm    volatile(“disi #5”);
asm    volatile(“mov #0x55, w7”);
asm    volatile(“mov w7, _NVMKEY”);
asm    volatile(“mov #0xAA, w8”);
asm    volatile(“mov w8, _NVMKEY”);
asm    volatile(“bset _RCFGCAL, #13”);   //set the RTCWREN bit
asm    volatile(“pop w8”);
asm    volatile(“pop w7”);
```

REGISTER 22-3: CRYSTAT: CRYPTOGRAPHIC STATUS REGISTER

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

R/HSC-x[1]	R/HSC-0[1]	R/C-0, HS[2]	R/C-0, HS[2]	U-0	R/HSC-0[1]	R/HSC-x[1]	R/HSC-x[1]
CRYBSY[4]	TXTABSY	CRYABRT[5]	ROLLOVR	—	MODFAIL[3]	KEYFAIL[3,4]	PGMFAIL[3,4]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
HS = Hardware Settable bit	C = Clearable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Reset State Conditional bit

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **CRYBSY:** Cryptographic OTP Array Busy Status bit[1,4]

1 = The cryptography module is performing a cryptographic operation or OTP operation
0 = The module is not currently performing any operation

bit 6 **TXTABSY:** CRYTXTA Busy Status bit[1]

1 = The CRYTXTA register is busy and may not be written to
0 = The CRYTXTA is free and may be written to

bit 5 **CRYABRT:** Cryptographic Operation Aborted Status bit[2,5]

1 = Last operation was aborted by clearing the CRYGO bit in software
0 = Last operation completed normally (CRYGO cleared in hardware)

bit 4 **ROLLOVR:** Counter Rollover Status bit[2]

1 = The CRYTXTB counter rolled over on the last CTR mode operation; once set, this bit must be cleared by software before the CRYGO bit can be set again
0 = No rollover event has occurred

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **MODFAIL:** Mode Configuration Fail Flag bit[1,3]

1 = Currently selected operating and Cipher mode configuration is invalid; the CRYWR bit cannot be set until a valid mode is selected (automatically cleared by hardware with any valid configuration)
0 = Currently selected operating and Cipher mode configurations are valid

bit 1 **KEYFAIL:** Key Configuration Fail Status bit[1,3,4]

See Table 22-1 and Table 22-2 for invalid key configurations.

1 = Currently selected key and mode configurations are invalid; the CRYWR bit cannot be set until a valid mode is selected (automatically cleared by hardware with any valid configuration)
0 = Currently selected configurations are valid

bit 0 **PGMFAIL:** Key Storage/Configuration Program Fail Flag bit[1,3,4]

1 = The page indicated by KEYPG<3:0> is reserved or locked; the CRYWR bit cannot be set and no programming operation can be started
0 = The page indicated by KEYPG<3:0> is available for programming

Note 1: These bits are reset on system Resets or whenever the CRYMD bit is set.

2: These bits are reset on system Resets when the CRYMD bit is set or when CRYGO is cleared.

3: These bits are functional even when the module is disabled (CRYON = 0); this allows mode configurations to be validated for compatibility before enabling the module.

4: These bits are automatically set during all OTP read operations, including the initial read at POR. Once the read is completed, the bit assumes the proper state that reflects the current configuration.

5: If this bit is set, a cryptographic operation cannot be performed.

REGISTER 24-2: AD1CON2: A/D CONTROL REGISTER 2 (CONTINUED)

bit 6-2 **SMPI<4:0>:** Interrupt Sample/DMA Increment Rate Select bits

When DMAEN = 1:
11111 = Increments the DMA address after completion of the 32nd sample/conversion operation
11110 = Increments the DMA address after completion of the 31st sample/conversion operation
•
•
•
00001 = Increments the DMA address after completion of the 2nd sample/conversion operation
00000 = Increments the DMA address after completion of each sample/conversion operation

When DMAEN = 0:
11111 = Interrupts at the completion of the conversion for each 32nd sample
11110 = Interrupts at the completion of the conversion for each 31st sample
•
•
•
00001 = Interrupts at the completion of the conversion for every other sample
00000 = Interrupts at the completion of the conversion for each sample

bit 1 **BUFM:** Buffer Fill Mode Select bit[1]

1 = Starts buffer filling at ADC1BUF0 on first interrupt and ADC1BUF8 on next interrupt
0 = Always starts filling buffer at ADC1BUF0

bit 0 **ALTS:** Alternate Input Sample Mode Select bit

1 = Uses channel input selects for Sample A on first sample and Sample B on next sample
0 = Always uses channel input selects for Sample A

Note 1: These bits are only applicable when the buffer is used in FIFO mode (BUFREGEN = 0). In addition, BUFS is only used when BUFM = 1.

REGISTER 25-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0, HS	R-0, HSC
CON	COE	CPOL	—	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1[(1)]	EVPOL0[(1)]	—	CREF	—	—	CCH1	CCH0
bit 7							bit 0

Legend:	HS = Hardware Settable bit	HSC = Hardware Settable/Clearable 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 **CON:** Comparator Enable bit

1 = Comparator is enabled
0 = Comparator is disabled

bit 14 **COE:** Comparator Output Enable bit

1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only

bit 13 **CPOL:** Comparator Output Polarity Select bit

1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 12-10 **Unimplemented:** Read as ‘0’

bit 9 **CEVT:** Comparator Event bit

1 = Comparator event that is defined by EVPOL<1:0> has occurred; subsequent triggers and interrupts are disabled until the bit is cleared
0 = Comparator event has not occurred

bit 8 **COUT:** Comparator Output bit

When CPOL = 0:
1 = VIN+ > VIN-
0 = VIN+ < VIN-

When CPOL = 1:
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 7-6 **EVPOL<1:0>:** Trigger/Event/Interrupt Polarity Select bits[(1)]

11 = Trigger/event/interrupt is generated on any change of the comparator output (while CEVT = 0)
10 = Trigger/event/interrupt is generated on the high-to-low transition of the comparator output
01 = Trigger/event/interrupt is generated on the low-to-high transition of the comparator output
00 = Trigger/event/interrupt generation is disabled

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **CREF:** Comparator Reference Select bit (non-inverting input)

1 = Non-inverting input connects to the internal CVREF voltage
0 = Non-inverting input connects to the CxINA pin

bit 3-2 **Unimplemented:** Read as ‘0’

Note 1: If the EVPOL<1:0> bits are set to a value other than ‘00’, the first interrupt generated will occur on any transition of COUT. Subsequent interrupts will occur based on the EVPOLx bits setting.

29.0 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, refer to the following sections of the *"dsPIC33/PIC24 Family Reference Manual"*. The information in this data sheet supersedes the information in the FRMs.
>
> - **"Watchdog Timer (WDT)"** (DS39697)
> - **"High-Level Device Integration"** (DS39719)
> - **"Programming and Diagnostics"** (DS39716)

PIC24FJ128GA204 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
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation (ICE)

29.1 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 detailed explanation of the various bit functions is provided in Register 29-1 through Register 29-6.

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

29.1.1 CONSIDERATIONS FOR CONFIGURING PIC24FJ128GA204 FAMILY DEVICES

In PIC24FJ128GA204 family devices, the Configuration bytes are implemented as volatile memory. This means that configuration data must be programmed each time the device is powered up. Configuration data is stored in the four words at the top of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 29-1. These are packed representations of the actual device Configuration bits, whose actual locations are distributed among several locations in configuration space. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration registers during device Resets.

> **Note:** Configuration data is reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The upper byte of all Flash Configuration Words in program memory should always be '`0000 0000`'. This makes them appear to be `NOP` instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing '`0`'s to these locations has no effect on device operation.

> **Note:** Performing a page erase operation on the last page of program memory clears the Flash Configuration Words, enabling code protection as a result. Therefore, users should avoid performing page erase operations on the last page of program memory.

TABLE 29-1: FLASH CONFIGURATION WORD LOCATIONS FOR THE PIC24FJ128GA204 FAMILY

Device	Configuration Word Addresses			
	1	2	3	4
PIC24FJ64GA2XX	ABFEh	ABFCh	ABFAh	ABF8h
PIC24FJ128GA2XX	157FEh	157FCh	157FAh	157F8h

NOTES:

TABLE 32-13: COMPARATOR DC SPECIFICATIONS

Operating Conditions: -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	VIOFF	Input Offset Voltage	—	20	±40	mV	**(Note 1)**
D301	VICM	Input Common-Mode Voltage	0	—	VDD	V	**(Note 1)**
D302	CMRR	Common-Mode Rejection Ratio	55	—	—	dB	**(Note 1)**
D306	IQCMP	AVDD Quiescent Current per Comparator	—	27	—	μs	Comparator enabled
D307	TRESP	Response Time	—	300	—	ns	**(Note 2)**
D308	TMC2OV	Comparator Mode Change to Valid Output	—	—	10	μs	

Note 1: Parameters are characterized but not tested.

2: Measured with one input at VDD/2 and the other transitioning from VSS to VDD, 40 mV step, 15 mV overdrive.

TABLE 32-14: COMPARATOR VOLTAGE REFERENCE DC SPECIFICATIONS

Operating Conditions: 2.0V < VDD < 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
VR310	TSET	Settling Time	—	—	10	μs	**(Note 1)**
VRD311	CVRAA	Absolute Accuracy	-100	—	100	mV	
VRD312	CVRUR	Unit Resistor Value (R)	—	4.5	—	kΩ	

Note 1: Measures the interval while CVR<4:0> transitions from ‘11111’ to ‘00000’.

TABLE 32-38: SPIx MODULE SLAVE MODE (CKE = 1) TIMING REQUIREMENTS

AC 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.	**Symbol**	**Characteristic[1]**	**Min**	**Typ[2]**	**Max**	**Units**	**Conditions**
SP70	TscL	SCKx Input Low Time	30	—	—	ns	
SP71	TscH	SCKx Input High Time	30	—	—	ns	
SP72	TscF	SCKx Input Fall Time[3]	—	10	25	ns	
SP73	TscR	SCKx Input Rise Time[3]	—	10	25	ns	
SP30	TdoF	SDOx Data Output Fall Time[3]	—	—	—	ns	See Parameter DO32
SP31	TdoR	SDOx Data Output Rise Time[3]	—	—	—	ns	See Parameter DO31
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid After SCKx Edge	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP50	TssL2scH, TssL2scL	$\overline{\text{SSx}}$ ↓ to SCKx ↓ or SCKx ↑ Input	120	—	—	ns	
SP51	TssH2doZ	$\overline{\text{SSx}}$ ↑ to SDOx Output High-Impedance[4]	10	—	50	ns	
SP52	TscH2ssH TscL2ssH	$\overline{\text{SSx}}$ ↑ After SCKx Edge	1.5 TCY + 40	—	—	ns	
SP60	TssL2doV	SDOx Data Output Valid After $\overline{\text{SSx}}$ Edge	—	—	50	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

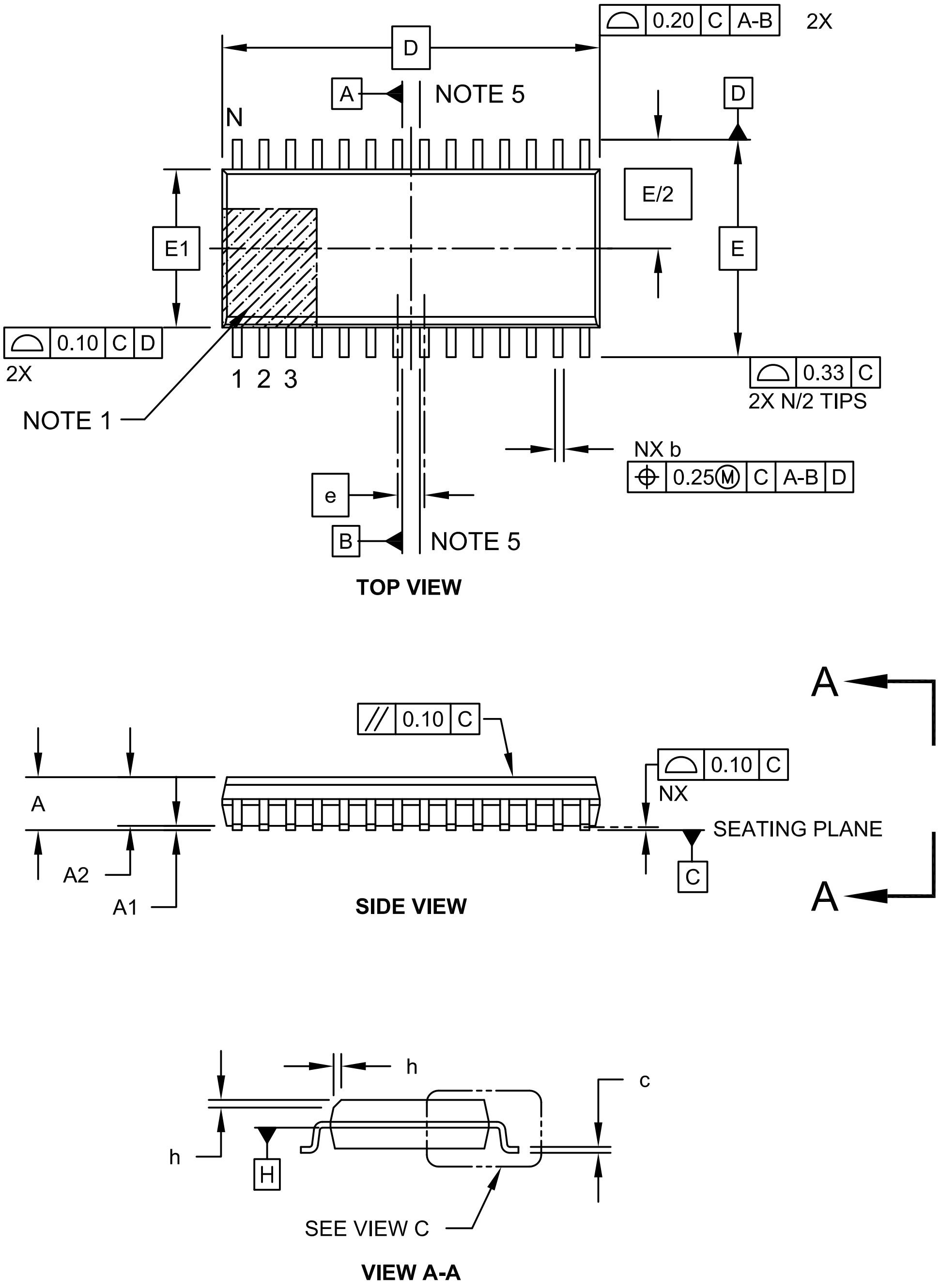
2: Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.

3: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPIx pins.

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



TOP VIEW

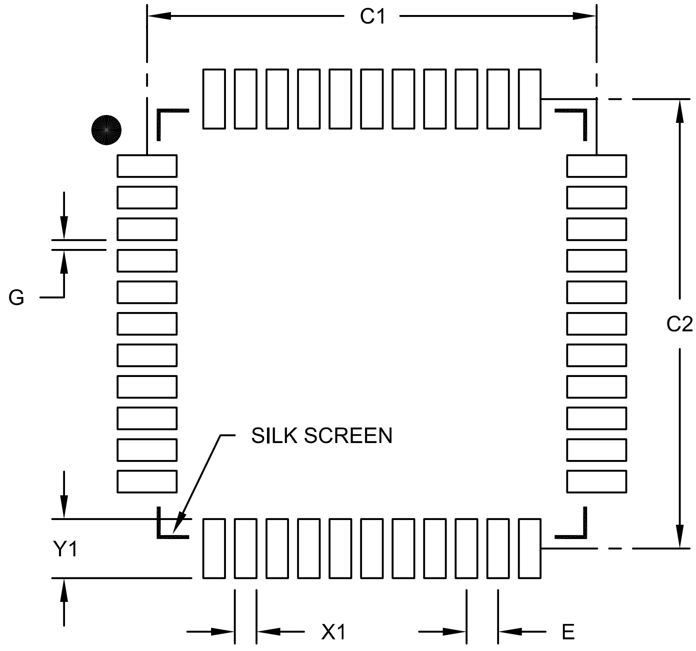
SIDE VIEW

VIEW A-A

Microchip Technology Drawing C04-052C Sheet 1 of 2

44-Lead Plastic Thin Quad Flatpack (PT) 10X10X1 mm Body, 2.00 mm Footprint [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B