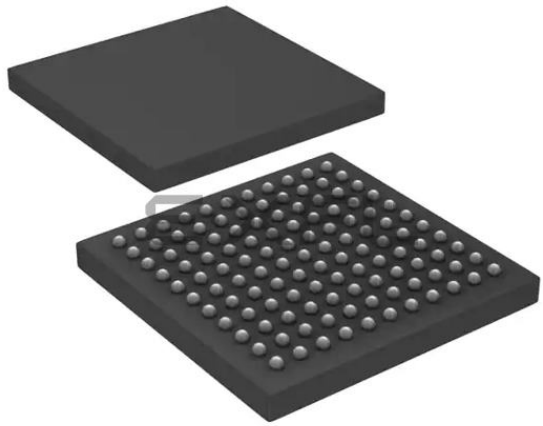
Microchip Technology - PIC24FJ1024GA610-I/BG Datasheet



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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	85
Program Memory Size	1MB (341.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 24x10/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	121-TFBGA
Supplier Device Package	121-TFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj1024ga610-i-bg

TABLE 6: COMPLETE PIN FUNCTION DESCRIPTIONS (PIC24FJXXXGA610 BGA)

Pin	Full Pin Name	Pin	Full Pin Name
A1	HLVDIN/CTED8/PMD4/RE4	E1	AN16/**RPI41**/OCM3C/PMCS2/RC4
A2	CTED9/PMD3/RE3	E2	**RPI40**/OCM2D/RC3
A3	OCM2F/CTED10/RG13	E3	AN17/C1IND/**RP21**/ICM1/OCM1A/PMA5/RG6
A4	PMD0/RE0	E4	**RPI39**/OCM2C/RC2
A5	PMD8/RG0	E5	N/C
A6	PMD10/RF1	E6	PMD9/RG1
A7	N/C	E7	N/C
A8	N/C	E8	**RPI35**/PMBE1/RA15
A9	**RPI42**/OCM3E/PMD12/RD12	E9	CLC4OUT/**RP2**/$\overline{\text{U6RTS}}$/U6BCLK/ICM5/RD8
A10	**RP23**/PMACK1/RD2	E10	**RP4**/PMACK2/RD9
A11	**RP24**/U5TX/ICM4/RD1	E11	**RPI36**/PMA22/RA14
B1	N/C	F1	$\overline{\text{MCLR}}$
B2	OCM1C/CTED3/RG15	F2	AN19/C2IND/**RP19**/ICM2/OCM2A/PMA3/RG8
B3	PMD2/RE2	F3	AN20/C1INC/C2INC/C3INC/**RP27**/OCM2B/PMA2/PMALU/RG9
B4	PMD1/RE1	F4	AN18/C1INC/**RP26**/OCM1B/PMA4/RG7
B5	AN22/OCM1F/PMA17/RA7	F5	VSS
B6	$\overline{\text{U5CTS}}$/OC6/PMD11/RF0	F6	N/C
B7	VCAP	F7	N/C
B8	**RP20**/PMRD/$\overline{\text{PMWR}}$/RD5	F8	VDD
B9	**RP22**/ICM7/PMBE0/RD3	F9	OSCI/CLKI/RC12
B10	VSS	F10	VSS
B11	SOSCO/C3INC/**RPI37**/PWRLCLK/RC14	F11	OSCO/CLKO/RC15
C1	SCL3/IC5/PMD6/RE6	G1	**RPI33**/PMCS1/RE8
C2	VDD	G2	AN21/**RPI34**/PMA19/RE9
C3	OCM2E/RG12	G3	TMS/OCM3D/RA0
C4	CTED11/PMA16/RG14	G4	N/C
C5	AN23/OCM1E/RA6	G5	VDD
C6	N/C	G6	VSS
C7	C3INA/$\overline{\text{U5RTS}}$/U5BCLK/OC5/PMD15/RD7	G7	VSS
C8	**RP25**/PMWR/PMENB/RD4	G8	N/C
C9	N/C	G9	TDO/RA5
C10	SOSCI/C3IND/RC13	G10	SDA2/PMA20/RA3
C11	**RP12**/PMA14/PMCS1/RD11	G11	TDI/PMA21/RA4
D1	**RPI38**/OCM1D/RC1	H1	PGEC3/AN5/C1INA/**RP18**/ICM3/OCM3A/RB5
D2	SDA3/IC6/PMD7/RE7	H2	PGED3/AN4/C1INB/**RP28**/OCM3B/RB4
D3	IC4/CTED4/PMD5/RE5	H3	N/C
D4	N/C	H4	N/C
D5	N/C	H5	N/C
D6	N/C	H6	VDD
D7	C3INB/U5RX/OC4/PMD14/RD6	H7	N/C
D8	OCM3F/PMD13/RD13	H8	RF7
D9	CLC3OUT/**RP11**/$\overline{\text{U6CTS}}$/ICM6/RD0	H9	INT0/RF6
D10	N/C	H10	SCL1/RG2
D11	**RP3**/PMA15/PMCS2/RD10	H11	PMPCS1/SCL2/RA2

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select (PPS) functions.

Note: Pinouts are subject to change.

TABLE 4-6: SFR MAP: 0200h BLOCK

File Name	Address	All Resets	File Name	Address	All Resets
INPUT CAPTURE			**OUTPUT CAPTURE (CONTINUED)**		
IC1CON1	0200	0000	OC4R	0254	xxxx
IC1CON2	0202	000D	OC4TMR	0256	xxxx
IC1BUF	0204	0000	OC5CON1	0258	0000
IC1TMR	0206	0000	OC5CON2	025A	000C
IC2CON1	0208	0000	OC5RS	025C	xxxx
IC2CON2	020A	000D	OC5R	025E	xxxx
IC2BUF	020C	0000	OC5TMR	0260	xxxx
IC2TMR	020E	0000	OC6CON1	0262	0000
IC3CON1	0210	0000	OC6CON2	0264	000C
IC3CON2	0212	000D	OC6RS	0266	xxxx
IC3BUF	0214	0000	OC6R	0268	xxxx
IC3TMR	0216	0000	OC6TMR	026A	xxxx
IC4CON1	0218	0000	**MULTIPLE OUTPUT CAPTURE/COMPARE/PWM**		
IC4CON2	021A	000D	CCP1CON1L	026C	0000
IC4BUF	021C	0000	CCP1CON1H	026E	0000
IC4TMR	021E	0000	CCP1CON2L	0270	0000
IC5CON1	0220	0000	CCP1CON2H	0272	0100
IC5CON2	0222	000D	CCP1CON3L	0274	0000
IC5BUF	0224	0000	CCP1CON3H	0276	0000
IC5TMR	0226	0000	CCP1STATL	0278	00x0
IC6CON1	0228	0000	CCP1STATH	027A	0000
IC6CON2	022A	000D	CCP1TMRL	027C	0000
IC6BUF	022C	0000	CCP1TMRH	027E	0000
IC6TMR	022E	0000	CCP1PRL	0280	FFFF
OUTPUT COMPARE			CCP1PRH	0282	FFFF
OC1CON1	0230	0000	CCP1RAL	0284	0000
OC1CON2	0232	000C	CCP1RAH	0286	0000
OC1RS	0234	xxxx	CCP1RBL	0288	0000
OC1R	0236	xxxx	CCP1RBH	028A	0000
OC1TMR	0238	xxxx	CCP1BUFL	028C	0000
OC2CON1	023A	0000	CCP1BUFH	028E	0000
OC2CON2	023C	000C	CCP2CON1L	0290	0000
OC2RS	023E	xxxx	CCP2CON1H	0292	0000
OC2R	0240	xxxx	CCP2CON2L	0294	0000
OC2TMR	0242	xxxx	CCP2CON2H	0296	0100
OC3CON1	0244	0000	CCP2CON3L	0298	0000
OC3CON2	0246	000C	CCP2CON3H	029A	0000
OC3RS	0248	xxxx	CCP2STATL	029C	00x0
OC3R	024A	xxxx	CCP2STATH	029E	0000
OC3TMR	024C	xxxx	CCP2TMRL	02A0	0000
OC4CON1	024E	0000	CCP2TMRH	02A2	0000
OC4CON2	0250	000C	CCP2PRL	02A4	FFFF
OC4RS	0252	xxxx	CCP2PRH	02A6	FFFF

Legend: — = unimplemented, read as '0'; x = undefined. Reset values are shown in hexadecimal.

TABLE 4-7: SFR MAP: 0300h BLOCK (CONTINUED)

File Name	Address	All Resets	File Name	Address	All Resets
UART (CONTINUED)			**UART (CONTINUED)**		
U3STA	03C6	0110	U5BRG	03E4	0000
U3TXREG	03C8	xxxx	U5ADMD	03E6	0000
U3RXREG	03CA	0000	U6MODE	03E8	0000
U3BRG	03CC	0000	U6STA	03EA	0110
U3ADMD	03CE	0000	U6TXREG	03EC	xxxx
U4MODE	03D0	0000	U6RXREG	03EE	0000
U4STA	03D2	0110	U6BRG	03F0	0000
U4TXREG	03D4	xxxx	U6ADMD	03F2	0000
U4RXREG	03D6	0000	**SPI**		
U4BRG	03D8	0000	SPI1CON1	03F4	0x00
U4ADMD	03DA	0000	SPI1CON2	03F6	0000
U5MODE	03DC	0000	SPI1CON3	03F8	0000
U5STA	03DE	0110	SPI1STATL	03FC	0028
U5TXREG	03E0	xxxx	SPI1STATH	03FE	0000
U5RXREG	03E2	0000			

Legend: — = unimplemented, read as '0'; x = undefined. Reset values are shown in hexadecimal.

TABLE 4-10: SFR MAP: 0600h BLOCK

File Name	Address	All Resets	File Name	Address	All Resets
I/O			**PORTD (CONTINUED)**		
PADCON	065E	0000	ANSD	06A6	FFFF
IOCSTAT	0660	0000	IOCPD	06A8	0000
PORTA(1)			IOCND	06AA	0000
TRISA	0662	FFFF	IOCFD	06AC	0000
PORTA	0664	0000	IOCPUD	06AE	0000
LATA	0666	0000	IOCPDD	06B0	0000
ODCA	0668	0000	**PORTE**		
ANSA	066A	FFFF	TRISE	06B2	FFFF
IOCPA	066C	0000	PORTE	06B4	0000
IOCNA	066E	0000	LATE	06B6	0000
IOCFA	0670	0000	ODCE	06B8	0000
IOCPUA	0672	0000	ANSE	06BA	FFFF
IOCPDA	0674	0000	IOCPE	06BC	0000
PORTB			IOCNE	06BE	0000
TRISB	0676	FFFF	IOCFE	06C0	0000
PORTB	0678	0000	IOCPUE	06C2	0000
LATB	067A	0000	IOCPDE	06C4	0000
ODCB	067C	0000	**PORTF**		
ANSB	067E	FFFF	TRISF	06C6	FFFF
IOCPB	0680	0000	PORTF	06C8	0000
IOCNB	0682	0000	LATF	06CA	0000
IOCFB	0684	0000	ODCF	06CC	0000
IOCPUB	0686	0000	IOCPF	06D0	0000
IOCPDB	0688	0000	IOCNF	06D2	0000
PORTC			IOCFF	06D4	0000
TRISC	068A	FFFF	IOCPUF	06D6	0000
PORTC	068C	0000	IOCPDF	06D8	0000
LATC	068E	0000	**PORTG**		
ODCC	0690	0000	TRISG	06DA	FFFF
ANSC	0692	FFFF	PORTG	06DC	0000
IOCPC	0694	0000	LATG	06DE	0000
IOCNC	0696	0000	ODCG	06E0	0000
IOCFC	0698	0000	ANSG	06E2	FFFF
IOCPUC	069A	0000	IOCPG	06E4	0000
IOCPDC	069C	0000	IOCNG	06E6	0000
PORTD			IOCFG	06E8	0000
TRISD	069E	FFFF	IOCPUG	06EA	0000
PORTD	06A0	0000	IOCPDG	06EC	0000
LATD	06A2	0000			
ODCD	06A4	0000			

Legend: — = unimplemented, read as '0'; x = undefined. Reset values are shown in hexadecimal.

Note 1: PORTA and all associated bits are unimplemented in 64-pin devices and read as '0'.

TABLE 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	FNOSC<2:0>	Notes
Oscillator with Frequency Division (OSCFDIV)	Internal/External	111	**1, 2, 3**
Digitally Controlled Oscillator (DCO)	Internal	110	**3**
Low-Power RC Oscillator (LPRC)	Internal	101	**3**
Secondary (Timer1) Oscillator (SOSC)	Secondary	100	**3**
Primary Oscillator (XT, HS or EC) with PLL Module	Primary	011	**4**
Primary Oscillator (XT, HS or EC)	Primary	010	**4**
Fast RC Oscillator with PLL Module (FRCPLL)	Internal	001	**3**
Fast RC Oscillator (FRC)	Internal	000	**3**

Note 1: The input oscillator to the OSCFDIV Clock mode is determined by the RCDIV<2:0> (CLKDIV<10:8) bits. At POR, the default value selects the FRC module.

2: This is the default oscillator mode for an unprogrammed (erased) device.

3: OSCO pin function is determined by the OSCIOFCN Configuration bit.

4: The POSCMD<1:0> Configuration bits select the oscillator driver mode (XT, HS or EC).

9.3 Control Registers

The operation of the oscillator is controlled by five Special Function Registers:

- OSCCON
- CLKDIV
- OSCTUN
- OSCDIV
- OSCFDIV

In addition, two registers are used to control the DCO:

- DCOCON
- DCOTUN

The OSCCON register (Register 9-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources. OSCCON is protected by a write lock to prevent inadvertent clock switches. See **Section 9.4 “Clock Switching Operation”** for more information.

The CLKDIV register (Register 9-2) controls the features associated with Doze mode, as well as the postscalers for the OSCFDIV Clock mode and the PLL module.

The OSCTUN register (Register 9-3) allows the user to fine-tune the FRC Oscillator over a range of approximately ±1.5%. It also controls the FRC self-tuning features described in **Section 9.5 “FRC Active Clock Tuning”**.

The OSCDIV and OSCFDIV registers provide control for the system Oscillator Frequency Divider.

9.3.1 DCO OVERVIEW

The DCO (Digitally Controlled Oscillator) is a low-power alternative to the FRC. It can generate a wider selection of operating frequencies and can be trimmed to correct process variations if an exact frequency is required. However, the DCO is not designed for use with USB applications and cannot meet USB timing restrictions.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 3 **CF:** Clock Fail Detect bit
1 = FSCM has detected a clock failure
0 = No clock failure has been detected

bit 2 **POSCEN:** Primary Oscillator Sleep Enable bit
1 = Primary Oscillator continues to operate during Sleep mode
0 = Primary Oscillator is disabled during Sleep mode

bit 1 **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit
1 = Enables Secondary Oscillator
0 = Disables Secondary Oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit
1 = Initiates an oscillator switch to a clock source specified by the NOSC<2:0> bits
0 = Oscillator switch is complete

Note 1: Reset values for these bits are determined by the FNOSCx Configuration bits.

2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1', once the IOLOCK bit is set, it cannot be cleared.

3: This bit also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

REGISTER 11-5: ANSE: PORTE ANALOG FUNCTION SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	U-0
—	—	—	—	—	—	ANSE9(1)	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	U-0	U-0	U-0	U-0
—	—	—	ANSE4	—	—	—	—
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 15-10 **Unimplemented:** Read as '0'

bit 9 **ANSE9:** PORTE Analog Function Selection bit(1)
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 8-5 **Unimplemented:** Read as '0'

bit 4 **ANSE4:** PORTE Analog Function Selection bit
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

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

Note 1: ANSE9 is not available on 64-pin devices.

REGISTER 11-6: ANSG: PORTG ANALOG FUNCTION SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1
—	—	—	—	—	—	ANSG<9:8>	
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSG<7:6>		—	—	—	—	—	—
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 15-10 **Unimplemented:** Read as '0'

bit 9-6 **ANSG<9:6>:** PORTG Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

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

REGISTER 11-26: RPINR18: PERIPHERAL PIN SELECT INPUT REGISTER 18

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **U1CTSR<5:0>:** Assign UART1 Clear-to-Send ($\overline{\text{U1CTS}}$) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **U1RXR<5:0>:** Assign UART1 Receive (U1RX) to Corresponding RPn or RPIn Pin bits

REGISTER 11-27: RPINR19: PERIPHERAL PIN SELECT INPUT REGISTER 19

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **U2CTSR<5:0>:** Assign UART2 Clear-to-Send ($\overline{\text{U2CTS}}$) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **U2RXR<5:0>:** Assign UART2 Receive (U2RX) to Corresponding RPn or RPIn Pin bits

REGISTER 11-38: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R5(1)	RP5R4(1)	RP5R3(1)	RP5R2(1)	RP5R1(1)	RP5R0(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP5R<5:0>:** RP5 Output Pin Mapping bits(1)
Peripheral Output Number n is assigned to pin, RP5 (see Table 11-4 for peripheral function numbers).

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

bit 5-0 **RP4R<5:0>:** RP4 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP4 (see Table 11-4 for peripheral function numbers).

Note 1: This pin is not available on 64-pin devices.

REGISTER 11-39: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
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

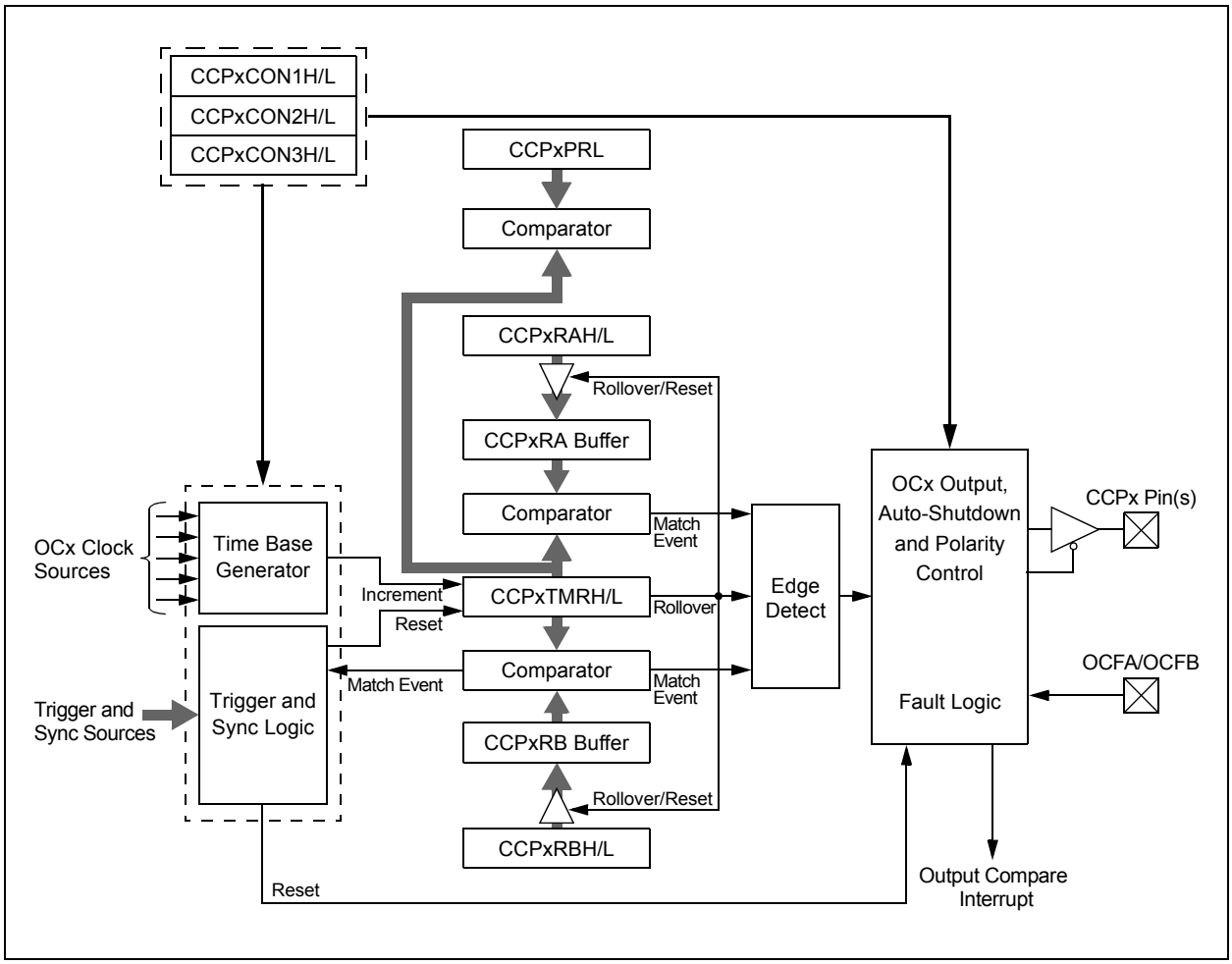
bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP7R<5:0>:** RP7 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP7 (see Table 11-4 for peripheral function numbers).

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

bit 5-0 **RP6R<5:0>:** RP6 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP6 (see Table 11-4 for peripheral function numbers).

FIGURE 16-5: OUTPUT COMPARE x BLOCK DIAGRAM



REGISTER 19-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

bit 7-6 **URXISEL<1:0>:** UARTx Receive Interrupt Mode Selection bits
11 = Interrupt is set on an RSR transfer, making the receive buffer full (i.e., has 4 data characters)
10 = Interrupt is set on an RSR transfer, making the receive buffer 3/4 full (i.e., has 3 data characters)
0x = Interrupt is set when any character is received and transferred from the RSR to the receive buffer; receive buffer has one or more characters

bit 5 **ADDEN:** Address Character Detect bit (bit 8 of received data = 1)
1 = Address Detect mode is enabled (if 9-bit mode is not selected, this does not take effect)
0 = Address Detect mode is disabled

bit 4 **RIDLE:** Receiver Idle bit (read-only)
1 = Receiver is Idle
0 = Receiver is active

bit 3 **PERR:** Parity Error Status bit (read-only)
1 = Parity error has been detected for the current character (the character at the top of the receive FIFO)
0 = Parity error has not been detected

bit 2 **FERR:** Framing Error Status bit (read-only)
1 = Framing error has been detected for the current character (the character at the top of the receive FIFO)
0 = Framing error has not been detected

bit 1 **OERR:** Receive Buffer Overrun Error Status bit (clear/read-only)
1 = Receive buffer has overflowed
0 = Receive buffer has not overflowed (clearing a previously set OERR bit, 1 → 0 transition); will reset the receive buffer and the RSR to the empty state

bit 0 **URXDA:** UARTx Receive Buffer Data Available bit (read-only)
1 = Receive buffer has data, at least one more character can be read
0 = Receive buffer is empty

Note 1: The value of this bit only affects the transmit properties of the module when the IrDA® encoder is enabled (IREN = 1).

2: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. For more information, see **Section 11.4 “Peripheral Pin Select (PPS)”**.

REGISTER 20-20: U1EIE: USB ERROR INTERRUPT ENABLE REGISTER

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

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE EOFEE	PIDEE
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 15-8 **Unimplemented:** Read as '0'

bit 7 **BTSEE:** Bit Stuff Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

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

bit 5 **DMAEE:** DMA Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 4 **BTOEE:** Bus Turnaround Time-out Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

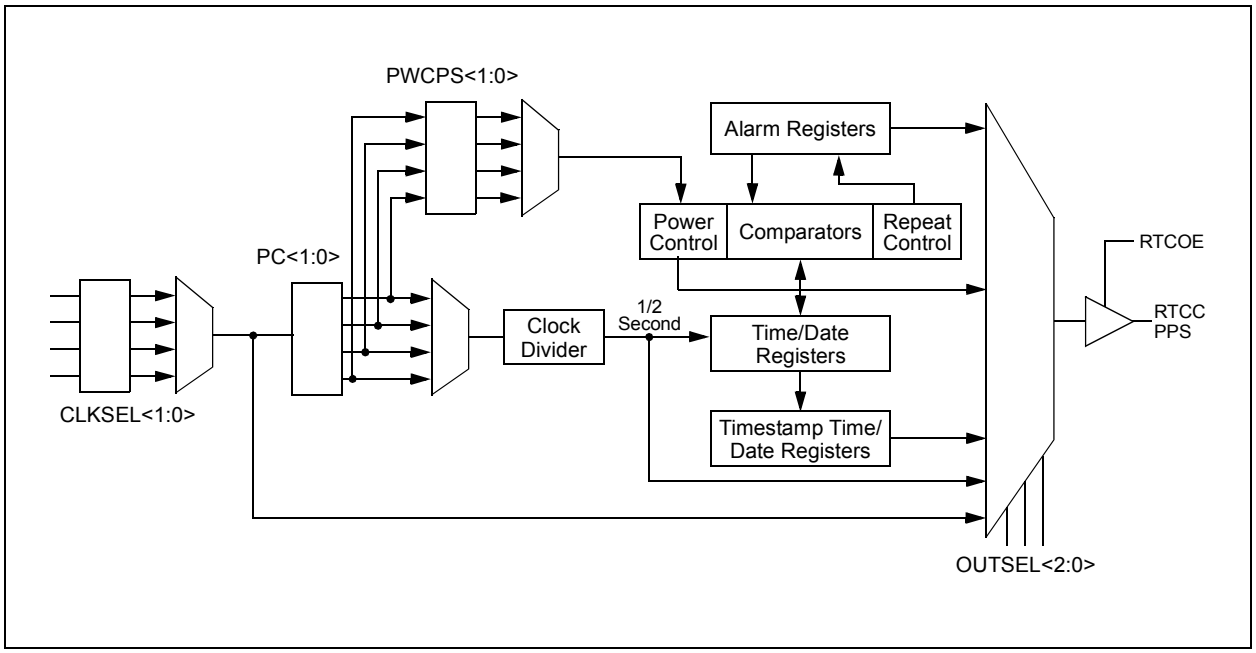
bit 3 **DFN8EE:** Data Field Size Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 2 **CRC16EE:** CRC16 Failure Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 1 For Device mode:
CRC5EE: CRC5 Host Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled
For Host mode:
EOFEE: End-of-Frame (EOF) Error interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 0 **PIDEE:** PID Check Failure Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

FIGURE 22-1: RTCC BLOCK DIAGRAM

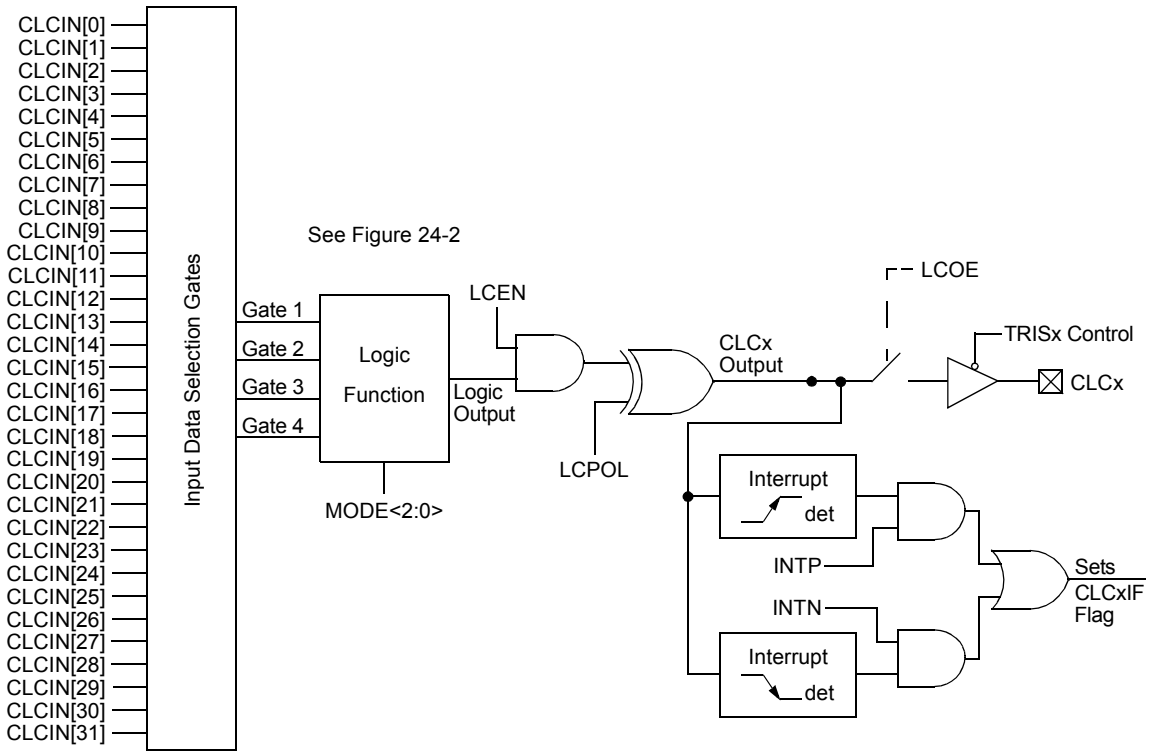


24.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLC) module allows the user to specify combinations of signals as inputs to a logic function and to use the logic output to control other peripherals or I/O pins. This provides greater flexibility and potential in embedded designs, since the CLC module can operate outside the limitations of software execution and supports a vast amount of output designs.

There are four input gates to the selected logic function. These four input gates select from a pool of up to 32 signals that are selected using four data source selection multiplexers. Figure 24-1 shows an overview of the module. Figure 24-3 shows the details of the data source multiplexers and logic input gate connections.

FIGURE 24-1: CLCx MODULE



Note: All register bits shown in this figure can be found in the CLCxCONL register.

25.4 Achieving Maximum A/D Converter (ADC) Performance

In order to get the shortest overall conversion time (called the "throughput") while maintaining accuracy, several factors must be considered. These are described in detail below.

- Dependence of AVDD – If the AVDD supply is < 2.7V, the Charge Pump Enable bit (PUMPEN, AD1CON3<13>) should be set to '1'. The input channel multiplexer has a varying resistance with AVDD (the lower AVDD, the higher the internal switch resistance). The charge pump provides a higher internal AVDD to keep the switch resistance as low as possible.
- Dependence on TAD – The ADC timing is driven by TAD, not TCYC. Selecting the TAD time correctly is critical to getting the best ADC throughput. It is important to note that the overall ADC throughput is not simply the 'Conversion Time' of the SAR; it is the combination of the Conversion Time, the Sample Time and additional TAD delays for internal synchronization logic.
- Relationship between TCYC and TAD – There is not a fixed 1:1 timing relationship between TCYC and TAD. The fastest possible throughput is fundamentally set by TAD (min), not by TCYC. The TAD time is set as a programmable integer multiple of TCYC by the ADCS<7:0> bits. Referring to Table 33-26, the TAD (min) time is greater than the 4 MHz period of the dedicated ADC RC clock generator. Therefore, TAD must be 2 TCYC in order to use the RC clock for fastest throughput. The TAD (min) is a multiple of 3.597 MHz as opposed to 4 MHz. To run as fast as possible, TCYC must be a multiple of TAD (min) because values of ADCSx are integers. For example, if a standard "color burst" crystal of 14.31818 MHz is used, TCYC is 279.4 ns, which is very close to TAD (min) and the ADC throughput is optimal. Running at 16 MHz will actually reduce the throughput, because TAD will have to be 500 ns as the TCYC of 250 ns violates TAD (min).
- Dependence on driving Source Resistance (RS) – Certain transducers have high output impedance (> 2.5 kΩ). Having a high RS will require longer sampling time to charge the S/H capacitor through the resistance path (see Figure 25-3). The worst case scenario is a full-range voltage step of AVSS to AVDD, with the sampling cap at AVSS. The capacitor time constant is (RS + RIC + RSS) (CHOLD) and the sample time needs to be 6 time constants minimum (8 preferred). Since the ADC logic timing is TAD-based, the sample time (in TAD) must be long enough, over all conditions, to charge/discharge CHOLD. Do not assume one TAD is sufficient sample time; longer times may be required to achieve the accuracy needed by the application. The value of CHOLD is 40 pF.

A small amount of charge is present at the ADC input pin when the sample switch is closed. If RS is high, this will generate a DC error exceeding 1 LSB. Keeping RS < 50Ω is recommenced for best results. The error can also be reduced by increasing sample time (a 2 kΩ value of RS requires a 3 µS sample time to eliminate the error).

- Calculating Throughput – The throughput of the ADC is based on TAD. The throughput is given by:

$$Throughput = \left(\frac{1}{Sample\ Time + SAR\ Conversion\ Time + Clock\ Sync\ Time}\right)$$

where:

Sample Time is the calculated TAD periods for the application.

SAR Conversion Time is 12 TAD for 10-bit and 14 TAD for 12-bit conversions.

Clock Sync Time is 2.5 TAD (worst case scenario).

For example, using an 8 MHz FRC means the TCYC = 250 ns. This requires: TAD = 2 TCYC = 500 ns. Therefore, the throughput is:

$$Throughput = \left(\frac{1}{500\ ns + 14 \cdot 500\ ns + 2.5 \cdot 500\ ns}\right) = 114.28\ KS/sec$$

Note that the clock sync delay could be as little as 1.5 TAD, which could produce 121 KS/sec, but that cannot be ensured as the timing relationship is asynchronous and not specified. The worst case timing of 2.5 TAD should be used to calculate throughput.

For example, if a certain transducer has a 20 kΩ output impedance, the maximum sample time is determined by:

$$Sample\ Time = 6 \cdot (RS + RIC + RSS) \cdot CHOLD$$
$$= 6 \cdot (20K + 250 + 350) \cdot 40\ pF$$
$$= 4.95\ \mu S$$

If TAD = 500 ns, this requires a Sample Time of 4.95 us/ 500 ns = 10 TAD (for a full-step voltage on the transducer output).

REGISTER 28-1: CTMUCON1L: CTMU CONTROL REGISTER 1 LOW

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **CTMUSIDL:** CTMU Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation and routes the current source to the comparator pin
0 = Disables edge delay generation and routes the current source to the selected A/D input pin

bit 11 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 10 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 9 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 8 **CTTRIG:** CTMU Trigger Control bit
1 = Trigger output is enabled
0 = Trigger output is disabled

bit 7-2 **ITRIM<5:0>:** Current Source Trim bits
011111 = Maximum positive change from nominal current
011110
•
•
•
000001 = Minimum positive change from nominal current
000000 = Nominal current output specified by IRNG<1:0>
111111 = Minimum negative change from nominal current
•
•
•
100010
100001 = Maximum negative change from nominal current

REGISTER 28-2: CTMUCON1H: CTMU CONTROL REGISTER 1 HIGH (CONTINUED)

bit 5-2 **EDG2SEL<3:0>:** Edge 2 Source Select bits
1111 = CMP C3OUT
1110 = CMP C2OUT
1101 = CMP C1OUT
1100 = Peripheral clock
1011 = IC3 interrupt
1010 = IC2 interrupt
1001 = IC1 interrupt
1000 = CTED13 pin
0111 = CTED12 pin
0110 = CTED11 pin(1)
0101 = CTED10 pin(1)
0100 = CTED9 pin
0011 = CTED1 pin
0010 = CTED2 pin
0001 = OC1
0000 = Timer1 match

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

bit 0 **IRNGH:** High-Current Range Select bit
1 = Uses the higher current ranges (550 µA-2.2 mA)
0 = Uses the lower current ranges (550 nA-50 µA)
Current output is set by the IRNG<1:0> bits in the CTMUCON1L register.

Note 1: CTED3, CTED7, CTED10 and CTED11 are not available on 64-pin packages.

REGISTER 30-3: FSEC CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
AIVTDIS	—	—	—	CSS2	CSS1	CSS0	CWRP
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
GSS1	GSS0	GWRP	—	BSEN	BSS1	BSS0	BWRP
bit 7							bit 0

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

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **AIVTDIS:** Alternate Interrupt Vector Table Disable bit
1 = Disables AIVT; INTCON2<8> (AIVTEN) bit is not available
0 = Enables AIVT; INTCON2<8> (AIVTEN) bit is available

bit 14-12 **Unimplemented:** Read as '1'

bit 11-9 **CSS<2:0>:** Configuration Segment Code Protection Level bits
111 = No protection (other than CWRP)
110 = Standard security
10x = Enhanced security
0xx = High security

bit 8 **CWRP:** Configuration Segment Program Write Protection bit
1 = Configuration Segment is not write-protected
0 = Configuration Segment is write-protected

bit 7-6 **GSS<1:0>:** General Segment Code Protection Level bits
11 = No protection (other than GWRP)
10 = Standard security
0x = High security

bit 5 **GWRP:** General Segment Program Write Protection bit
1 = General Segment is not write-protected
0 = General Segment is write-protected

bit 4 **Unimplemented:** Read as '1'

bit 3 **BSEN:** Boot Segment Control bit
1 = No Boot Segment is enabled
0 = Boot Segment size is determined by BSLIM<12:0>

bit 2-1 **BSS<1:0>**: Boot Segment Code Protection Level bits
11 = No protection (other than BWRP)
10 = Standard security
0x = High security

bit 0 **BWRP:** Boot Segment Program Write Protection bit
1 = Boot Segment can be written
0 = Boot Segment is write-protected

TABLE 30-2: DEVICE ID REGISTERS

Address	Name	Bit															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FF0000h	DEVID	FAMID<7:0>								DEV<7:0>							
FF0002h	DEVREV	—												REV<3:0>			

TABLE 30-3: DEVICE ID BIT FIELD DESCRIPTIONS

Bit Field	Register	Description
FAMID<7:0>	DEVID	Encodes the family ID of the device.
DEV<7:0>	DEVID	Encodes the individual ID of the device.
REV<3:0>	DEVREV	Encodes the sequential (numerical) revision identifier of the device.

TABLE 30-4: PIC24FJ1024GA610/GB610 FAMILY DEVICE IDs

Device	DEVID
PIC24FJ128GA606	6000h
PIC24FJ256GA606	6008h
PIC24FJ512GA606	6010h
PIC24FJ1024GA606	6018h
PIC24FJ128GA610	6001h
PIC24FJ256GA610	6009h
PIC24FJ512GA610	6011h
PIC24FJ1024GA610	6019h
PIC24FJ128GB606	6004h
PIC24FJ256GB606	600Ch
PIC24FJ512GB606	6014h
PIC24FJ1024GB606	601Ch
PIC24FJ128GB610	6005h
PIC24FJ256GB610	600Dh
PIC24FJ512GB610	6015h
PIC24FJ1024GB610	601Dh

30.2 Unique Device Identifier (UDID)

All PIC24FJ1024GA610/GB610 family devices are individually encoded during final manufacturing with a Unique Device Identifier, or UDID. The UDID cannot be erased by a bulk erase command or any other user-accessible means. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is a requirement. It may also be used by the application manufacturer for any number of things that may require unique identification, such as:

- Tracking the device
- Unique serial number
- Unique security key

The UDID comprises five 24-bit program words. When taken together, these fields form a unique 120-bit identifier.

The UDID is stored in five read-only locations, located between 801600h and 801608h in the device configuration space. Table 30-5 lists the addresses of the identifier words.

TABLE 30-5: UDID ADDRESSES

UDID	Address	Description
UDID1	801600	UDID Word 1
UDID2	801602	UDID Word 2
UDID3	801604	UDID Word 3
UDID4	801606	UDID Word 4
UDID5	801608	UDID Word 5

31.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

31.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker