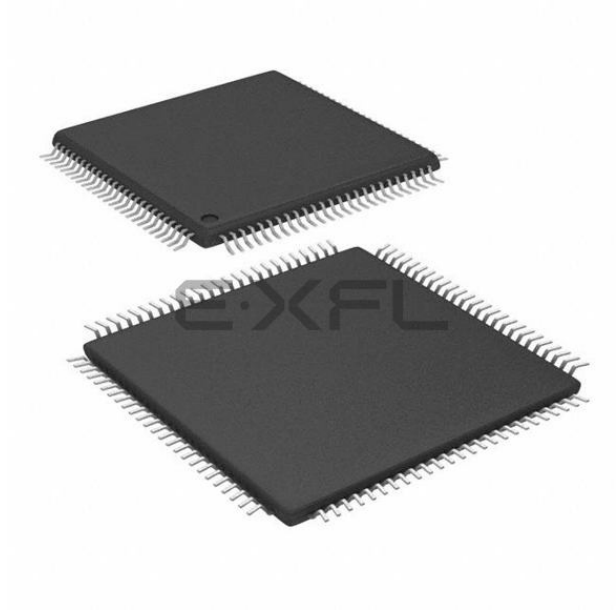
Microchip Technology - PIC24FJ128DA110-I/PT Datasheet



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, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, GFX, LVD, POR, PWM, WDT
Number of I/O	84
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128da110-i-pt

High-Performance CPU

- Modified Harvard Architecture
- Up to 16 MIPS Operation at 32 MHz
- 8 MHz Internal Oscillator
- 17-Bit x 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture with Flexible Addressing modes
- Linear Program Memory Addressing, up to 12 Mbytes
- Data Memory Addressing, up to 16 Mbytes:
  - 2K SFR space
  - 30K linear data memory
  - 66K extended data memory
  - Remaining (from 16 Mbytes) memory (external) can be accessed using extended data Memory (EDS) and EPMP (EDS is divided into 32-Kbyte pages)
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

Power Management:

- On-Chip Voltage Regulator of 1.8V
- Switch between Clock Sources in Real Time
- Idle, Sleep and Doze modes with Fast Wake-up and Two-Speed Start-up
- Run Mode: 800 μA/MIPS, 3.3V Typical
- Sleep mode Current Down to 20 μA, 3.3V Typical
- Standby Current with 32 kHz Oscillator: 22 μA, 3.3V Typical

Analog Features:

- 10-Bit, up to 24-Channel Analog-to-Digital (A/D) Converter at 500 ksps:
  - Operation is possible in Sleep mode
  - Band gap reference input feature
- Three Analog Comparators with Programmable Input/Output Configuration
- Charge Time Measurement Unit (CTMU):
  - Supports capacitive touch sensing for touch screens and capacitive switches
  - Minimum time measurement setting at 100 ps
- Available LVD Interrupt V_{LVD} Level

Special Microcontroller Features:

- Operating Voltage Range of 2.2V to 3.6V
- 5.5V Tolerant Input (digital pins only)
- Configurable Open-Drain Outputs on Digital I/O Ports
- High-Current Sink/Source (18 mA/18 mA) on all I/O Ports
- Selectable Power Management modes:
  - Sleep, Idle and Doze modes with fast wake-up
- Fail-Safe Clock Monitor (FSCM) Operation:
  - Detects clock failure and switches to on-chip, FRC oscillator
- On-Chip LDO Regulator
- Power-on Reset (POR) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR)
- Flexible Watchdog Timer (WDT) with On-Chip Low-Power RC Oscillator for Reliable Operation
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins
- JTAG Boundary Scan Support
- Flash Program Memory:
  - 10,000 erase/write cycle endurance (minimum)
  - 20-year data retention minimum
  - Selectable write protection boundary
  - Self-reprogrammable under software control
  - Write protection option for Configuration Words

TABLE 4-6: INTERRUPT CONTROLLER REGISTER MAP (CONTINUED)

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
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0	—	MI2C3IP2	MI2C3IP1	MI2C3IP0	—	SI2C3IP2	SI2C3IP1	SI2C3IP0	4444
IPC22	00D0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC23	00D2	—	—	—	—	—	—	—	—	—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0	0044
IPC25	00D6	—	—	—	—	—	—	—	—	—	—	—	—	—	GFX1IP2	GFX1IP1	GFX1IP0	0004
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'.
2: The Reset value in 64-pin devices are '0004'.

TABLE 4-7: TIMER 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
TMR1	0100	Timer1 Register																0000
PR1	0102	Timer1 Period Register																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																0000
TMR3	010A	Timer3 Register																0000
PR2	010C	Timer2 Period Register																FFFF
PR3	010E	Timer3 Period Register																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 Register																0000
TMR5HLD	0116	Timer5 Holding Register (for 32-bit operations only)																0000
TMR5	0118	Timer5 Register																0000
PR4	011A	Timer4 Period Register																FFFF
PR5	011C	Timer5 Period Register																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

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

TABLE 4-8: INPUT CAPTURE 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
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC1BUF	0144	Input Capture 1 Buffer Register																0000
IC1TMR	0146	Input Capture 1 Timer Value Register																xxxx
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C	Input Capture 2 Buffer Register																0000
IC2TMR	014E	Input Capture 2 Timer Value Register																xxxx
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	Input Capture 3 Buffer Register																0000
IC3TMR	0156	Input Capture 3 Timer Value Register																xxxx
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4CON2	015A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC4BUF	015C	Input Capture 4 Buffer Register																0000
IC4TMR	015E	Input Capture 4 Timer Value Register																xxxx
IC5CON1	0160	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC5CON2	0162	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC5BUF	0164	Input Capture 5 Buffer Register																0000
IC5TMR	0166	Input Capture 5 Timer Value Register																xxxx
IC6CON1	0168	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC6CON2	016A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC6BUF	016C	Input Capture 6 Buffer Register																0000
IC6TMR	016E	Input Capture 6 Timer Value Register																xxxx
IC7CON1	0170	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC7CON2	0172	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC7BUF	0174	Input Capture 7 Buffer Register																0000
IC7TMR	0176	Input Capture 7 Timer Value Register																xxxx
IC8CON1	0178	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC8CON2	017A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC8BUF	017C	Input Capture 8 Buffer Register																0000
IC8TMR	017E	Input Capture 8 Timer Value Register																xxxx
IC9CON1	0180	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC9CON2	0182	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC9BUF	0184	Input Capture 9 Buffer Register																0000
IC9TMR	0186	Input Capture 9 Timer Value Register																xxxx

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

TABLE 4-30: PERIPHERAL PIN SELECT REGISTER MAP (CONTINUED)

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
RPOR0	06C0	—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000
RPOR1	06C2	—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000
RPOR2	06C4	—	—	RP5R5(1)	RP5R4(1)	RP5R3(1)	RP5R2(1)	RP5R1(1)	RP5R0(1)	—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0	0000
RPOR3	06C6	—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000
RPOR4	06C8	—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000
RPOR5	06CA	—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000
RPOR6	06CC	—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0	—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0	0000
RPOR7	06CE	—	—	RP15R5(1)	RP15R4(1)	RP15R3(1)	RP15R2(1)	RP15R1(1)	RP15R0(1)	—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000
RPOR8	06D0	—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000
RPOR9	06D2	—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000
RPOR10	06D4	—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR11	06D6	—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000
RPOR12	06D8	—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000
RPOR13	06DA	—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0	—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0	0000
RPOR14	06DC	—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0	—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0	0000
RPOR15(1)	06DE	—	—	RP31R5(1)	RP31R4(1)	RP31R3(1)	RP31R2(1)	RP31R1(1)	RP31R0(1)	—	—	RP30R5(1)	RP30R4(1)	RP30R3(1)	RP30R2(1)	RP30R1(1)	RP30R0(1)	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

EXAMPLE 5-2: ERASING A PROGRAM MEMORY BLOCK ('C' LANGUAGE CODE)

```
// C example using MPLAB C30
    unsigned long progAddr = 0xXXXXXX;          // Address of row to write
    unsigned int offset;
//Set up pointer to the first memory location to be written
    TBLPAG = progAddr>>16;                      // Initialize PM Page Boundary SFR
    offset = progAddr & 0xFFFF;                 // Initialize lower word of address
    __builtin_tblwtl(offset, 0x0000);           // Set base address of erase block
                                                // with dummy latch write
    NVMCON = 0x4042;                            // Initialize NVMCON
    asm("DISI #5");                             // Block all interrupts with priority <7
                                                // for next 5 instructions
    __builtin_write_NVM();                      // check function to perform unlock
                                                // sequence and set WR
```

EXAMPLE 5-3: LOADING THE WRITE BUFFERS

```
; Set up NVMCON for row programming operations
       MOV     #0x4001, W0                      ;
       MOV     W0, NVMCON                       ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0                      ;
       MOV     W0, TBLPAG                       ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0                      ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0, W2                  ;
       MOV     #HIGH_BYTE_0, W3                 ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                       ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1, W2                  ;
       MOV     #HIGH_BYTE_1, W3                 ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                       ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2, W2                  ;
       MOV     #HIGH_BYTE_2, W3                 ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                       ; Write PM high byte into program latch
       •
       •
       •
; 63rd_program_word
       MOV     #LOW_WORD_63, W2                 ;
       MOV     #HIGH_BYTE_63, W3                ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0]                         ; Write PM high byte into program latch
```

EXAMPLE 5-4: INITIATING A PROGRAMMING SEQUENCE

```
       DISI    #5                       ; Block all interrupts with priority <7
                                        ; for next 5 instructions
       MOV.B   #0x55, W0
       MOV     W0, NVMKEY               ; Write the 0x55 key
       MOV.B   #0xAA, W1                ;
       MOV     W1, NVMKEY               ; Write the 0xAA key
       BSET    NVMCON, #WR              ; Start the programming sequence
       NOP                              ; Required delays
       NOP
       BTSC    NVMCON, #15              ; and wait for it to be
       BRA     $-2                      ; completed
```

TABLE 7-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU Event	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
External Interrupt 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
External Interrupt 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
Graphics Controller	100	0000DCh	0001DCh	IFS6<4>	IEC6<4>	IPC25<2:0>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
I2C3 Master Event	85	0000BEh	0001BEh	IFS5<5>	IEC5<5>	IPC21<6:4>
I2C3 Slave Event	84	0000BCh	0001BCh	IFS5<4>	IEC5<4>	IPC21<2:0>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Capture 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
Input Capture 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
Input Capture 8	23	000042h	000142h	IFS1<7>	IEC1<7>	IPC5<14:12>
Input Capture 9	93	0000CEh	0001CEh	IFS5<13>	IEC5<13>	IPC23<6:4>
Input Change Notification (ICN)	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
Low-Voltage Detect (LVD)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Output Compare 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
Output Compare 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
Output Compare 8	44	00006Ch	00016Ch	IFS2<12>	IEC2<12>	IPC11<2:0>
Output Compare 9	92	0000CCh	0001CCh	IFS5<12>	IEC5<12>	IPC23<2:0>
Enhanced Parallel Master Port (EPMP)(1)	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
Real-Time Clock and Calendar (RTCC)	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 Error	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 Event	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 Error	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
SPI3 Error	90	0000C8h	0001C8h	IFS5<10>	IEC5<10>	IPC22<10:8>
SPI3 Event	91	0000CAh	0001CAh	IFS5<11>	IEC5<11>	IPC22<14:12>

Note 1: Not available in 64-pin devices (PIC24FJXXXDAX06).

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(1)
- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Fast RC/16 Oscillator
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Selection Lock Enabled bit

If FSCM is enabled (FCKSM1 = 1):
- 1 = Clock and PLL selections are locked
- 0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit

If FSCM is disabled (FCKSM1 = 0):
Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.

bit 6 **IOLOCK:** I/O Lock Enable bit(2)
- 1 = I/O lock is active
- 0 = I/O lock is not active

bit 5 **LOCK:** PLL Lock Status bit(3)
- 1 = PLL module is in lock or PLL module start-up timer is satisfied
- 0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled

bit 4 **Unimplemented:** Read as '0'

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 = Enable the Secondary Oscillator
- 0 = Disable the Secondary Oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit
- 1 = Initiate an oscillator switch to the 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 FNOSC 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: Also resets to '0' during any valid clock switch or whenever a non PLL Clock mode is selected.

REGISTER 10-1: ANSA: PORTA ANALOG FUNCTION SELECTION REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	U-0
—	—	—	—	—	ANSA10	ANSA9	—
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSA7	ANSA6	—	—	—	—	—	—
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 10-9 **ANSA<10:9>:** 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 8 **Unimplemented:** Read as '0'

bit 7-6 **ANSA<7:6>:** 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'

Note 1: This register is not available on 64-pin devices (PIC24FJXXXDAX06).

10.4.6 PERIPHERAL PIN SELECT REGISTERS

The PIC24FJ256DA210 family of devices implements a total of 37 registers for remappable peripheral configuration:

- Input Remappable Peripheral Registers (21)
- Output Remappable Peripheral Registers (16)

Note: Input and output register values can only be changed if IOLOCK (OSCCON<6>) = 0. See **Section 10.4.4.1 "Control Register Lock"** for a specific command sequence.

REGISTER 10-8: RPINR0: PERIPHERAL PIN SELECT INPUT REGISTER 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **INT1R<5:0>:** Assign External Interrupt 1 (INT1) to Corresponding RPn or RPIn Pin bits

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

REGISTER 10-9: RPINR1: PERIPHERAL PIN SELECT INPUT REGISTER 1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
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
—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
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 **INT3R<5:0>:** Assign External Interrupt 3 (INT3) to Corresponding RPn or RPIn Pin bits

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

bit 5-0 **INT2R<5:0>:** Assign External Interrupt 2 (INT2) to Corresponding RPn or RPIn Pin bits

REGISTER 10-16: RPINR10: PERIPHERAL PIN SELECT INPUT REGISTER 10

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0
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
—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0
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 **IC8R<5:0>:** Assign Input Capture 8 (IC8) to Corresponding RPn or RPIn Pin bits

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

bit 5-0 **IC7R<5:0>:** Assign Input Capture 7 (IC7) to Corresponding RPn or RPIn Pin bits

REGISTER 10-17: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
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
—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
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 **OCFBR<5:0>:** Assign Output Compare Fault B (OCFB) to Corresponding RPn or RPIn Pin bits

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

bit 5-0 **OCFAR<5:0>:** Assign Output Compare Fault A (OCFA) to Corresponding RPn or RPIn Pin bits

16.0 INTER-INTEGRATED CIRCUIT™ (I^2C™)

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 *"PIC24F Family Reference Manual"*, **Section 24. "Inter-Integrated Circuit™ (I^2C™)"** (DS39702). The information in this data sheet supersedes the information in the FRM.

The Inter-Integrated Circuit™ (I^2C™) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

The I^2C module supports these features:

- Independent master and slave logic
- 7-bit and 10-bit device addresses
- General call address, as defined in the I^2C protocol
- Clock stretching to provide delays for the processor to respond to a slave data request
- Both 100 kHz and 400 kHz bus specifications
- Configurable address masking
- Multi-Master modes to prevent loss of messages in arbitration
- Bus Repeater mode, allowing the acceptance of all messages as a slave regardless of the address
- Automatic SCL

A block diagram of the module is shown in Figure 16-1.

16.1 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communications protocol for the device being communicated with. Typically, the sequence of events is as follows:

1. Assert a Start condition on SDAx and SCLx.
2. Send the I^2C device address byte to the slave with a write indication.
3. Wait for and verify an Acknowledge from the slave.
4. Send the first data byte (sometimes known as the command) to the slave.
5. Wait for and verify an Acknowledge from the slave.
6. Send the serial memory address low byte to the slave.
7. Repeat steps 4 and 5 until all data bytes are sent.
8. Assert a Repeated Start condition on SDAx and SCLx.
9. Send the device address byte to the slave with a read indication.
10. Wait for and verify an Acknowledge from the slave.
11. Enable master reception to receive serial memory data.
12. Generate an ACK or NACK condition at the end of a received byte of data.
13. Generate a Stop condition on SDAx and SCLx.

NOTES:

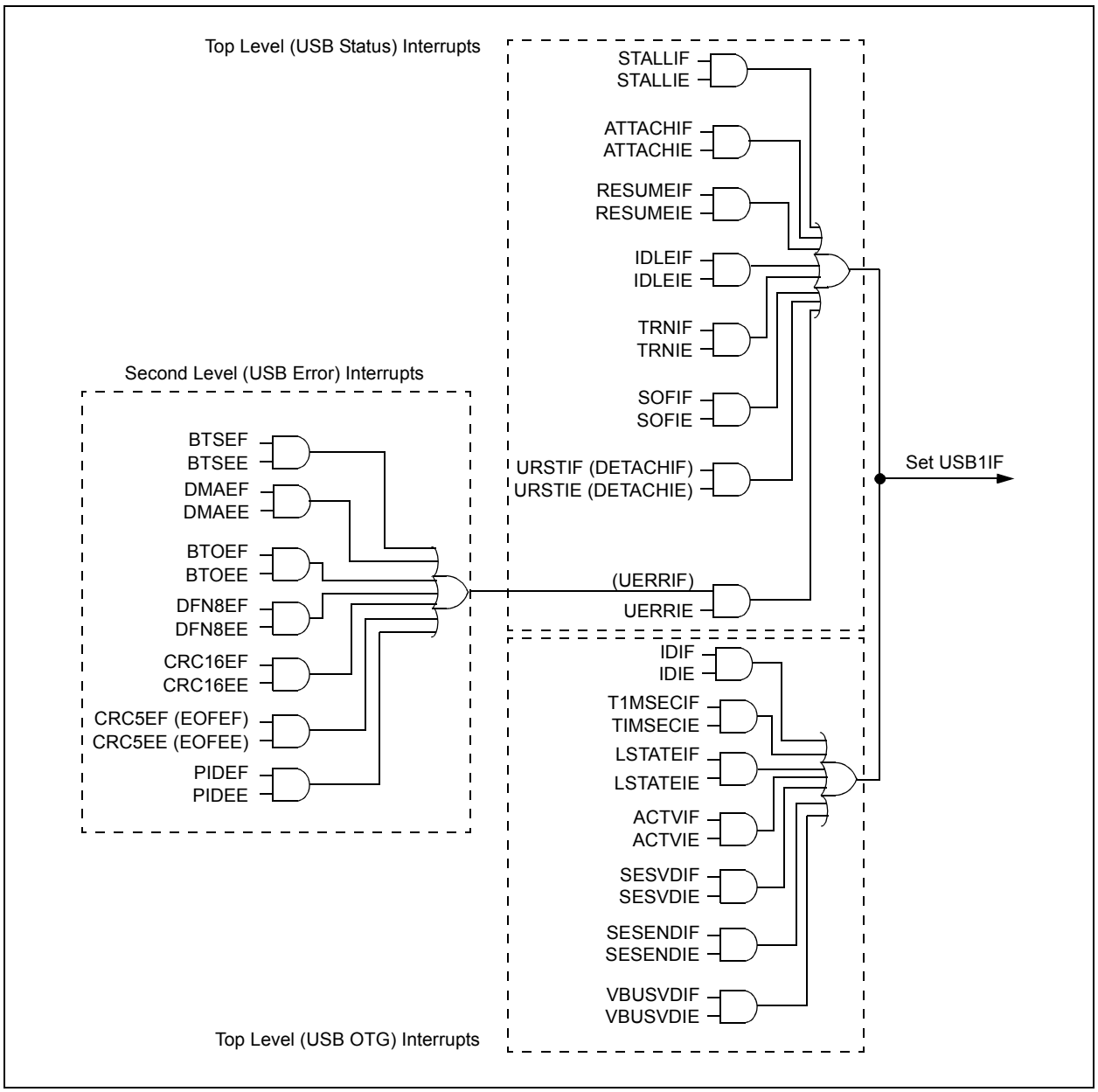
18.3 USB Interrupts

The USB OTG module has many conditions that can be configured to cause an interrupt. All interrupt sources use the same interrupt vector.

Figure 18-9 shows the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the U1IE and U1IR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the U1EIR and U1EIE registers. An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level.

Interrupts may be used to trap routine events in a USB transaction. Figure 18-10 provides some common events within a USB frame and their corresponding interrupts.

FIGURE 18-9: USB OTG INTERRUPT FUNNEL



18.5.2 COMPLETE A CONTROL TRANSACTION TO A CONNECTED DEVICE

1. Follow the procedure described in **Section 18.5.1 "Enable Host Mode and Discover a Connected Device"** to discover a device.
2. Set up the Endpoint Control register for bidirectional control transfers by writing 0Dh to U1EP0 (this sets the EPCONDIS, EPTXEN and EPHSHK bits).
3. Place a copy of the device framework setup command in a memory buffer. See *"Chapter 9 of the USB 2.0 Specification"* for information on the device framework command set.
4. Initialize the Buffer Descriptor (BD) for the current (even or odd) TX EP0 to transfer the eight bytes of command data for a device framework command (i.e., `GET DEVICE DESCRIPTOR`):
   a) Set the BD data buffer address (BD0ADR) to the starting address of the 8-byte memory buffer containing the command.
   b) Write 8008h to BD0STAT (this sets the UOWN bit and sets a byte count of 8).
5. Set the USB device address of the target device in the address register (U1ADDR<6:0>). After a USB bus Reset, the device USB address will be zero. After enumeration, it will be set to another value between 1 and 127.
6. Write D0h to U1TOK; this is a SETUP token to Endpoint 0, the target device's default control pipe. This initiates a SETUP token on the bus, followed by a data packet. The device handshake is returned in the PID field of BD0STAT after the packets are complete. When the USB module updates BD0STAT, a transfer done interrupt is asserted (the TRNIF flag is set). This completes the setup phase of the setup transaction as referenced in *"Chapter 9 of the USB Specification"*.
7. To initiate the data phase of the setup transaction (i.e., get the data for the `GET DEVICE DESCRIPTOR` command), set up a buffer in memory to store the received data.
8. Initialize the current (even or odd) RX or TX (RX for IN, TX for OUT) EP0 BD to transfer the data.
   a) Write C040h to BD0STAT. This sets the UOWN, configures Data Toggle (DTS) to DATA1 and sets the byte count to the length of the data buffer (64 or 40h in this case).
   b) Set BD0ADR to the starting address of the data buffer.
9. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., write 90h to U1TOK for an IN token for a `GET DEVICE DESCRIPTOR` command). This initiates an IN token on the bus followed by a data packet from the device to the host. When the data packet completes, the BD0STAT is written and a transfer done interrupt is asserted (the TRNIF flag is set). For control transfers with a single packet data phase, this completes the data phase of the setup transaction as referenced in *"Chapter 9 of the USB Specification"*. If more data needs to be transferred, return to step 8.
10. To initiate the status phase of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current (even or odd) TX EP0 BD to transfer the status data:
    a) Set the BDT buffer address field to the start address of the data buffer.
    b) Write 8000h to BD0STAT (set UOWN bit, configure DTS to DATA0 and set byte count to 0).
12. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., write 01h to U1TOK for an OUT token for a `GET DEVICE DESCRIPTOR` command). This initiates an OUT token on the bus followed by a zero length data packet from the host to the device. When the data packet completes, the BD is updated with the handshake from the device and a transfer done interrupt is asserted (the TRNIF flag is set). This completes the status phase of the setup transaction as described in *"Chapter 9 of the USB Specification"*.

Note: Only one control transaction can be performed per frame.

19.0 ENHANCED PARALLEL MASTER PORT (EPMP)

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 "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** (DS39730). The information in this data sheet supersedes the information in the FRM.

The Enhanced Parallel Master Port (EPMP) module is present in PIC24FJXXXDAX10 devices and not in PIC24FJXXXDAX06 devices. The EPMP provides a parallel 4-bit (Master mode only), 8-bit (Master and Slave modes) or 16-bit (Master mode only) data bus interface to communicate with off-chip modules, such as memories, FIFOs, LCD controllers and other microcontrollers. This module can serve as either the master or the slave on the communication bus. For EPMP Master modes, all external addresses are mapped into the internal Extended Data Space (EDS). This is done by allocating a region of the EDS for each chip select, and then assigning each chip select to a particular external resource, such as a memory or external controller. This region should not be assigned to another device resource, such as RAM or SFRs. To perform a write or read on an external resource, the CPU should simply perform a write or read within the address range assigned for EPMP.

Note: The EPMP module is not present in 64-pin devices (PIC24FJXXXDAX06).

The EPMP has an alternative master feature. The graphics controller module can control the EPMP directly in Alternate Master mode to access an external graphics buffer.

Key features of the EPMP module are:

- Extended Data Space (EDS) interface allows Direct Access from the CPU
- Up to 23 Programmable Address Lines
- Up to 2 Chip Select Lines
- Up to 2 Acknowledgement Lines (one per chip select)
- 4-bit, 8-bit or 16-bit wide Data Bus
- Programmable Strobe Options (per chip select)
  - Individual Read and Write Strobes or;
  - Read/$\overline{\text{Write}}$ Strobe with Enable Strobe
- Programmable Address/Data Multiplexing
- Programmable Address Wait States
- Programmable Data Wait States (per chip select)
- Programmable Polarity on Control Signals (per chip select)
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support
  - Address Support
  - 4-Byte Deep Auto-Incrementing Buffer
- Alternate Master feature

19.1 $\overline{\text{ALTPMP}}$ Setting

Many of the lower order EPMP address pins are shared with ADC inputs. This is an untenable situation for users that need both the ADC channels and the EPMP bus. If the user does not need to use all the address bits, then by clearing the $\overline{\text{ALTPMP}}$ (CW3<12>) Configuration bit, the lower order address bits can be mapped to higher address pins, which frees the ADC channels.

TABLE 19-1: ALTERNATE EPMP PINS

Pin	$\overline{\text{ALTPMP}}$ = 0	$\overline{\text{ALTPMP}}$ = 1
RA14	PMCS2	PMA22
RC4	PMA22	PMCS2
RF12	PMA5	PMA18
RG6	PMA18	PMA5
RG7	PMA20	PMA4
RA3	PMA4	PMA20
RG8	PMA21	PMA3
RA4	PMA3	PMA21

REGISTER 25-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CVREFP	CVREFM1	CVREFM0
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
CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
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 **CVREFP:** Voltage Reference Select bit (valid only when CREF is '1')
1 = VREF+ is used as a reference voltage to the comparators
0 = The CVR (4-bit DAC) within this module provides the the reference voltage to the comparators

bit 9-8 **CVREFM<1:0>:** Band Gap Reference Source Select bits (valid only when CCH<1:0> = 11)
00 = Band gap voltage is provided as an input to the comparators
01 = Band gap voltage divided-by-two is provided as an input to the comparators
10 = Band gap voltage divided-by-six is provided as an input to the comparators
11 = VREF+ pin is provided as an input the comparators

bit 7 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit is powered on
0 = CVREF circuit is powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit
1 = CVREF voltage level is output on the CVREF pin
0 = CVREF voltage level is disconnected from the CVREF pin

bit 5 **CVRR:** Comparator VREF Range Selection bit
1 = CVRSRC range should be 0 to 0.625 CVRSRC with CVRSRC/24 step size
0 = CVRSRC range should be 0.25 to 0.719 CVRSRC with CVRSRC/32 step size

bit 4 **CVRSS:** Comparator VREF Source Selection bit
1 = Comparator reference source CVRSRC = VREF+ – VREF-
0 = Comparator reference source CVRSRC = AVDD – AVSS

bit 3-0 **CVR<3:0>:** Comparator VREF Value Selection 0 ≤ CVR<3:0> ≤ 15 bits
When CVRR = 1:
CVREF = (CVR<3:0>/ 24) • (CVRSRC)
When CVRR = 0:
CVREF = 1/4 • (CVRSRC) + (CVR<3:0>/32) • (CVRSRC)

26.0 CHARGE TIME MEASUREMENT UNIT (CTMU)

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 associated "*PIC24F Family Reference Manual*", **Section 11. "Charge Time Measurement Unit (CTMU)"** (DS39724). The information in this data sheet supersedes the information in the FRM.

The Charge Time Measurement Unit (CTMU) is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. Its key features include:

- Four edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edges
- Time measurement resolution of 1 nanosecond
- Accurate current source suitable for capacitive measurement

Together with other on-chip analog modules, the CTMU can be used to precisely measure time, measure capacitance, measure relative changes in capacitance or generate output pulses that are independent of the system clock. The CTMU module is ideal for interfacing with capacitive-based sensors.

The CTMU is controlled through two registers: CTMUCON and CTMUICON. CTMUCON enables the module, and controls edge source selection, edge source polarity selection, and edge sequencing. The CTMUICON register controls the selection and trim of the current source.

26.1 Measuring Capacitance

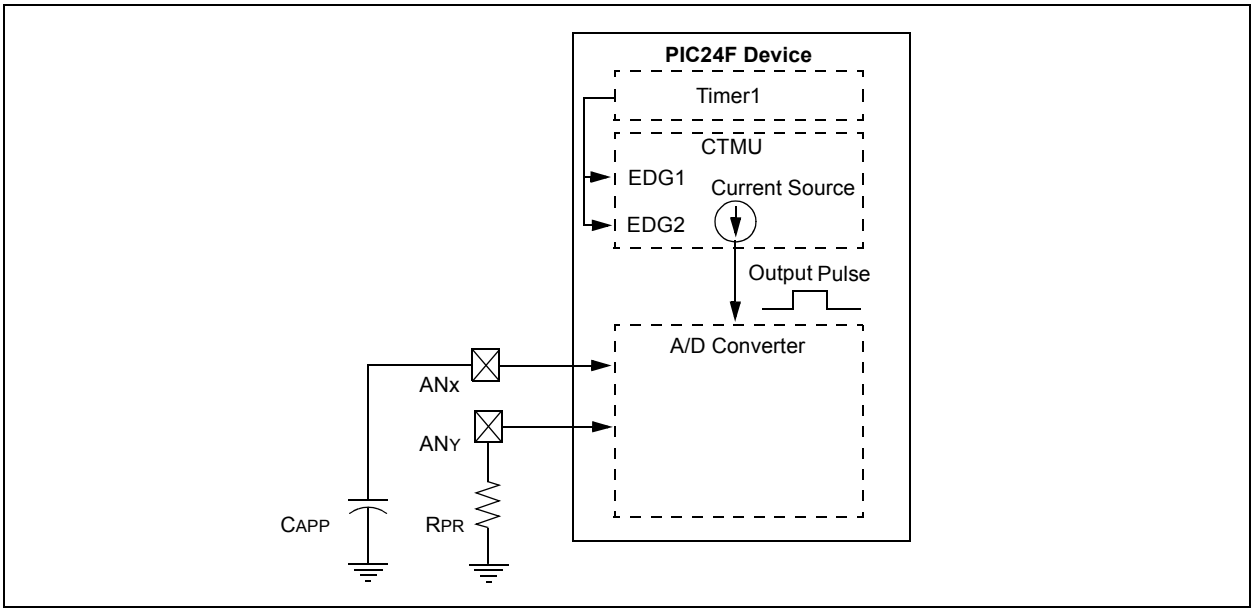
The CTMU module measures capacitance by generating an output pulse with a width equal to the time between edge events on two separate input channels. The pulse edge events to both input channels can be selected from four sources: two internal peripheral modules (OC1 and Timer1) and two external pins (CTEDG1 and CTEDG2). This pulse is used with the module's precision current source to calculate capacitance according to the relationship:

$$C = I \cdot \frac{dV}{dT}$$

For capacitance measurements, the A/D Converter samples an external capacitor (C_{APP}) on one of its input channels after the CTMU output's pulse. A precision resistor (R_{PR}) provides current source calibration on a second A/D channel. After the pulse ends, the converter determines the voltage on the capacitor. The actual calculation of capacitance is performed in software by the application.

Figure 26-1 shows the external connections used for capacitance measurements, and how the CTMU and A/D modules are related in this application. This example also shows the edge events coming from Timer1, but other configurations using external edge sources are possible. A detailed discussion on measuring capacitance and time with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

FIGURE 26-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



REGISTER 27-6: DEVREV: DEVICE REVISION REGISTER

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

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

U-0	U-0	U-0	U-0	R	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend: R = Readable bit	U = Unimplemented bit

bit 23-4 **Unimplemented:** Read as '0'

bit 3-0 **REV<3:0>:** Device revision identifier bits

27.2 On-Chip Voltage Regulator

All PIC24FJ256DA210 family devices power their core digital logic at a nominal 1.8V. This may create an issue for designs that are required to operate at a higher typical voltage, such as 3.3V. To simplify system design, all devices in the PIC24FJ256DA210 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is controlled by the ENVREG pin. Tying VDD to the pin enables the regulator, which in turn, provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR capacitor (such as ceramic) must be connected to the VCAP pin (Figure 27-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor (CEFC) is provided in **Section 30.1 "DC Characteristics"**.

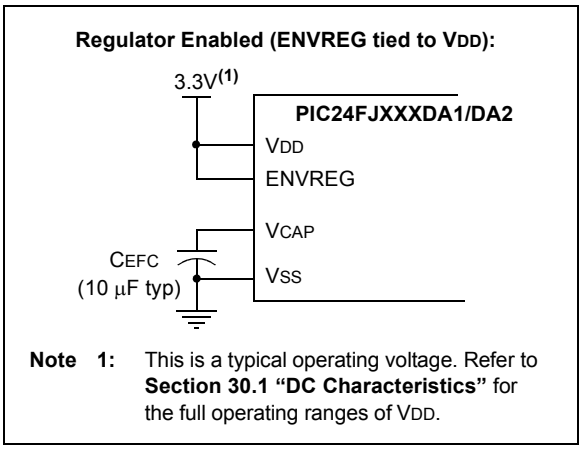
27.2.1 VOLTAGE REGULATOR LOW-VOLTAGE DETECTION

When the on-chip regulator is enabled, it provides a constant voltage of 1.8V nominal to the digital core logic.

The regulator can provide this level from a VDD of about 2.1V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels. In order to prevent "brown-out" conditions when the voltage drops too low for the regulator, the Brown-out Reset occurs. Then the regulator output follows VDD with a typical voltage drop of 300 mV.

To provide information about when the regulator voltage starts reducing, the on-chip regulator includes a simple Low-Voltage Detect circuit, which sets the Low-Voltage Detect Interrupt Flag, LVDIF (IFS4<8>). This can be used to generate an interrupt to trigger an orderly shutdown.

FIGURE 27-1: CONNECTIONS FOR THE ON-CHIP REGULATOR



Note 1: This is a typical operating voltage. Refer to **Section 30.1 "DC Characteristics"** for the full operating ranges of VDD.

27.2.2 ON-CHIP REGULATOR AND POR

When the voltage regulator is enabled, it takes approximately 10 μs for it to generate output. During this time, designated as TVREG, code execution is disabled. TVREG is applied every time the device resumes operation after any power-down, including Sleep mode. TVREG is determined by the status of the VREGS bit (RCON<8>) and the WUTSEL Configuration bits (CW3<11:10>). Refer to **Section 30.0 "Electrical Characteristics"** for more information on TVREG.

TABLE 30-15: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial					
Param No.	Characteristic	Min	Typ	Max	Units	Conditions	
F20	**FRC Accuracy @ 8 MHz(1,2)**	-1	±0.15	1	%	-40°C ≤ TA ≤ +85°C	2.2V ≤ VDD ≤ 3.6V
F21	**LPRC @ 31 kHz**	-20	—	20	%	-40°C ≤ TA ≤ +85°C	VCAP (on-chip regulator output voltage) = 1.8V

Note 1: Frequency calibrated at 25°C and 3.3V. OSCTUN bits can be used to compensate for temperature drift.

2: To achieve this accuracy, physical stress applied to the microcontroller package (ex., by flexing the PCB) must be kept to a minimum.

TABLE 30-16: RC OSCILLATOR START-UP TIME

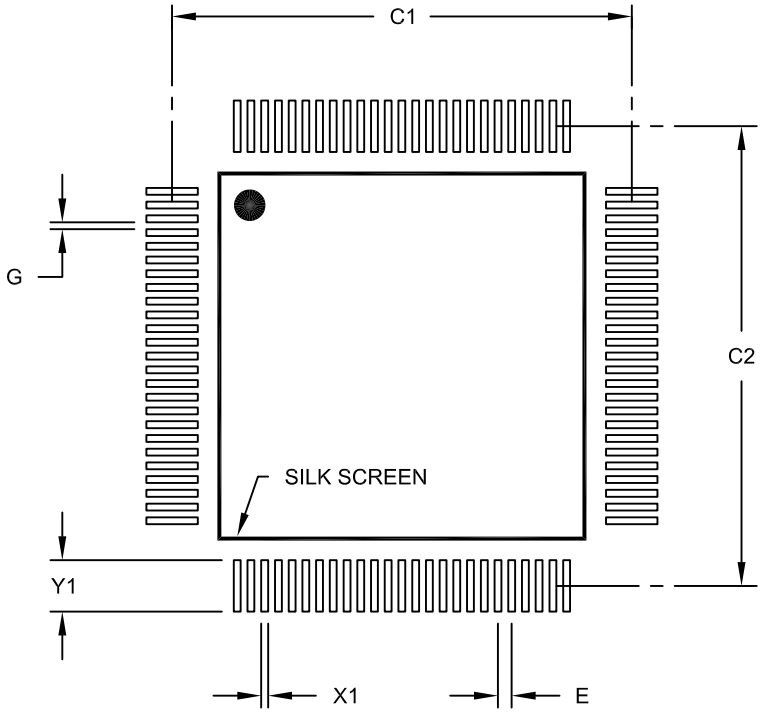
AC CHARACTERISTICS		Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Characteristic	Min	Typ	Max	Units	Conditions
	TFRC	—	15	—	μs	
	TLPRC	—	50	—	μs	

TABLE 30-17: RESET AND BROWN-OUT RESET REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
SY10	TMCL	$\overline{\text{MCLR}}$ Pulse width (Low)	2	—	—	μs	
SY12	TPOR	Power-on Reset Delay	—	2	—	μs	
SY13	TIOZ	I/O High-Impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	100	ns	
SY25	TBOR	Brown-out Reset Pulse Width	1	—	—	μs	VDD ≤ VBOR
	TRST	Internal State Reset Time	—	50	—	μs	

100-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm [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		0.40 BSC	
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X100)	X1			0.20
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2100A