

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I ² C, IrDA, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, GFX, LVD, POR, PWM, WDT
Number of I/O	84
Program Memory Size	256КВ (85.5К х 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96К х 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/pic24fj256da210t-i-pt

Email: info@E-XFL.COM

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

	Vector	Ιντ	AIVT	Interrupt Bit Locations			
	Number	Address	Address	Flag	Enable	Priority	
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>	
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>	
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>	
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>	
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>	
UART1 Error	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>	
UART1 Receiver	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>	
UART1 Transmitter	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>	
UART2 Error	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>	
UART2 Receiver	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>	
UART2 Transmitter	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>	
UART3 Error	81	0000B6h	0001B6h	IFS5<1>	IEC5<1>	IPC20<6:4>	
UART3 Receiver	82	0000B8h	0001B8h	IFS5<2>	IEC5<2>	IPC20<10:8>	
UART3 Transmitter	83	0000BAh	0001BAh	IFS5<3>	IEC5<3>	IPC20<14:12>	
UART4 Error	87	0000C2h	0001C2h	IFS5<7>	IEC5<7>	IPC21<14:12>	
UART4 Receiver	88	0000C4h	0001C4h	IFS5<8>	IEC5<8>	IPC22<2:0>	
UART4 Transmitter	89	0000C6h	0001C6h	IFS5<9>	IEC5<9>	IPC22<6:4>	
USB Interrupt	86	0000C0h	0001C0h	IFS5<6>	IEC5<6>	IPC21<10:8>	

TABLE 7-2: IMPLEMENTED INTERRUPT VECTORS (CONTINUED)

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

7.3 Interrupt Control and Status Registers

The PIC24FJ256DA210 family of devices implements a total of 40 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS6
- IEC0 through IEC6
- IPC0 through IPC25 (except IPC14, IPC17 and IPC24)
- INTTREG

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

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

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals. The IPCx registers are used to set the interrupt priority level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The INTTREG register contains the associated interrupt vector number and the new CPU interrupt priority level, which are latched into the Vector Number (VECNUM<6:0>) and the Interrupt Level (ILR<3:0>) bit fields in the INTTREG register. The new interrupt priority level is the priority of the pending interrupt.

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

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

7.4 Interrupt Setup Procedures

7.4.1 INITIALIZATION

To configure an interrupt source:

- 1. Set the NSTDIS (INTCON1<15>) control bit if nested interrupts are not desired.
- 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.

7.4.2 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.

7.4.3 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.

7.4.4 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 (Level 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.

x = Bit is unknown

REGISTER 10-28: RPINR29: PERIPHERAL PIN SELECT INPUT REGISTER 29

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-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
bit 7	•						bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplen	nented bit, read	d as '0'	

'0' = Bit is cleared

bit 15-6 Unimplemented: Read as '0'

'1' = Bit is set

-n = Value at POR

bit 5-0 SS3R<5:0>: Assign SPI3 Slave Select Input (SS31IN) to Corresponding RPn or RPIn Pin bits

13.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own 16-bit timer. To increase resolution, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, Modules 1 and 2 are paired, as are Modules 3 and 4, and so on.) The odd numbered module (ICx) provides the Least Significant 16 bits of the 32-bit register pairs and the even module (ICy) provides the Most Significant 16 bits. Wrap-arounds of the ICx registers cause an increment of their corresponding ICy registers.

Cascaded operation is configured in hardware by setting the IC32 bits (ICxCON2<8>) for both modules.

13.2 Capture Operations

The input capture module can be configured to capture timer values and generate interrupts on rising edges on ICx or all transitions on ICx. Captures can be configured to occur on all rising edges or just some (every 4th or 16th). Interrupts can be independently configured to generate on each event or a subset of events.

To set up the module for capture operations:

- 1. Configure the ICx input for one of the available Peripheral Pin Select pins.
- 2. If Synchronous mode is to be used, disable the sync source before proceeding.
- 3. Make sure that any previous data has been removed from the FIFO by reading ICxBUF until the ICBNE bit (ICxCON1<3>) is cleared.
- 4. Set the SYNCSEL bits (ICxCON2<4:0>) to the desired sync/trigger source.
- 5. Set the ICTSEL bits (ICxCON1<12:10>) for the desired clock source.
- 6. Set the ICI bits (ICxCON1<6:5>) to the desired interrupt frequency
- 7. Select Synchronous or Trigger mode operation:
 - a) Check that the SYNCSEL bits are not set to '00000'.
 - b) For Synchronous mode, clear the ICTRIG bit (ICxCON2<7>).
 - c) For Trigger mode, set ICTRIG, and clear the TRIGSTAT bit (ICxCON2<6>).
- 8. Set the ICM bits (ICxCON1<2:0>) to the desired operational mode.
- 9. Enable the selected sync/trigger source.

For 32-bit cascaded operations, the setup procedure is slightly different:

- 1. Set the IC32 bits for both modules (ICyCON2<8>) and (ICxCON2<8>), enabling the even numbered module first. This ensures the modules will start functioning in unison.
- 2. Set the ICTSEL and SYNCSEL bits for both modules to select the same sync/trigger and time base source. Set the even module first, then the odd module. Both modules must use the same ICTSEL and SYNCSEL settings.
- Clear the ICTRIG bit of the even module (ICyCON2<7>). This forces the module to run in Synchronous mode with the odd module, regardless of its trigger setting.
- 4. Use the odd module's ICI bits (ICxCON1<6:5>) to set the desired interrupt frequency.
- Use the ICTRIG bit of the odd module (ICxCON2<7>) to configure Trigger or Synchronous mode operation.
- Note: For Synchronous mode operation, enable the sync source as the last step. Both input capture modules are held in Reset until the sync source is enabled.
- Use the ICM bits of the odd module (ICxCON1<2:0>) to set the desired capture mode.

The module is ready to capture events when the time base and the sync/trigger source are enabled. When the ICBNE bit (ICxCON1<3>) becomes set, at least one capture value is available in the FIFO. Read input capture values from the FIFO until the ICBNE clears to '0'.

For 32-bit operation, read both the ICxBUF and ICyBUF for the full 32-bit timer value (ICxBUF for the Isw, ICyBUF for the msw). At least one capture value is available in the FIFO buffer when the odd module's ICBNE bit (ICxCON1<3>) becomes set. Continue to read the buffer registers until ICBNE is cleared (performed automatically by hardware).

14.0 OUTPUT COMPARE WITH DEDICATED TIMERS

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 35. "Output Compare with Dedicated Timer" (DS39723). The information in this data sheet supersedes the information in the FRM.

Devices in the PIC24FJ256DA210 family feature all of the 9 independent output compare modules. Each of these modules offers a wide range of configuration and operating options for generating pulse trains on internal device events, and can produce pulse-width modulated waveforms for driving power applications.

Key features of the output compare module include:

- Hardware configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 31 user-selectable trigger/sync sources available
- Two separate period registers (a main register, OCxR, and a secondary register, OCxRS) for greater flexibility in generating pulses of varying widths
- Configurable for single pulse or continuous pulse generation on an output event, or continuous PWM waveform generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

14.1 General Operating Modes

14.1.1 SYNCHRONOUS AND TRIGGER MODES

When the output compare module operates in a free-running mode, the internal 16-bit counter, OCxTMR, runs counts up continuously, wrapping around from 0xFFFF to 0x0000 on each overflow, with its period synchronized to the selected external clock source. Compare or PWM events are generated each time a match between the internal counter and one of the period registers occurs.

In Synchronous mode, the module begins performing its compare or PWM operation as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the module's internal counter is reset. In Trigger mode, the module waits for a sync event from another internal module to occur before allowing the counter to run.

Free-running mode is selected by default or any time that the SYNCSEL bits (OCxCON2<4:0>) are set to '00000'. Synchronous or Trigger modes are selected any time the SYNCSEL bits are set to any value except '00000'. The OCTRIG bit (OCxCON2<7>) selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

14.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own set of 16-bit timer and duty cycle registers. To increase resolution, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, Modules 1 and 2 are paired, as are Modules 3 and 4, and so on.) The odd numbered module (OCx) provides the Least Significant 16 bits of the 32-bit register pairs and the even module (OCy) provides the Most Significant 16 bits. Wrap-arounds of the OCx registers cause an increment of their corresponding OCy registers.

Cascaded operation is configured in hardware by setting the OC32 bit (OCxCON2<8>) for both modules. For more details on cascading, refer to the "*PIC24F Family Reference Manual*", **Section 35.** "**Output Compare with Dedicated Timer**".

To set up the SPI module for the Standard Master mode of operation:

- 1. If using interrupts:
 - a) Clear the SPIxIF bit in the respective IFS register.
 - b) Set the SPIxIE bit in the respective IEC register.
 - c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
- Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 1.
- 3. Clear the SPIROV bit (SPIxSTAT<6>).
- 4. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
- Write the data to be transmitted to the SPIxBUF register. Transmission (and reception) will start as soon as data is written to the SPIxBUF register.

To set up the SPI module for the Standard Slave mode of operation:

- 1. Clear the SPIxBUF register.
- 2. If using interrupts:
 - a) Clear the SPIxIF bit in the respective IFS register.
 - b) Set the SPIxIE bit in the respective IEC register.
 - c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
- Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 0.
- 4. Clear the SMP bit.
- If the CKE bit (SPIxCON1<8>) is set, then the SSEN bit (SPIxCON1<7>) must be set to enable the SSx pin.
- 6. Clear the SPIROV bit (SPIxSTAT<6>).
- 7. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).





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

bit 1		SPITBF: SPIx Transmit Buffer Full Status bit							
		1 = Transmit not yet started, SPIxTXB is full							
		0 = Transmit started, SPIxTXB is empty							
		In Standard Buffer mode:							
		Automatically set in hardware when the CPU writes to the SPIxBUF location, loading SPIxTXB. Automatically cleared in hardware when the SPIx module transfers data from SPIxTXB to SPIxSR.							
		In Enhanced Buffer mode:							
		Automatically set in hardware when the CPU writes to the SPIxBUF location, loading the last available buffer location.							
		Automatically cleared in hardware when a buffer location is available for a CPU write.							
bit 0		SPIRBF: SPIx Receive Buffer Full Status bit							
		1 = Receive complete, SPIxRXB is full							
		0 = Receive is not complete, SPIxRXB is empty							
		In Standard Buffer mode:							
		Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB.							
		Automatically cleared in hardware when the core reads the SPIxBUF location, reading SPIxRXB.							
		In Enhanced Buffer mode:							
		Automatically set in hardware when SPIx transfers data from the SPIxSR to the buffer, filling the last unread buffer location.							
		Automatically cleared in hardware when a buffer location is available for a transfer from SPIxSR.							
Note	1:	If SPIEN = 1, these functions must be assigned to available RPn/RPIn pins before use. See Section 10.4 "Peripheral Pin Select (PPS)" for more information.							

© 2010 Microchip Technology Inc.

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	_	_	_	—	—
bit 15				-			bit 8
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	_	—	—	—	SPIFE	SPIBEN
bit 7							bit 0
Legend:							
R = Readabl	e bit	W = Writable	bit	U = Unimplen	nented bit, rea	d as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unk	nown
bit 15	FRMEN: Fran	ned SPIx Supp	ort bit				
	1 = Framed S	SPIx support is	enabled				
	0 = Framed S	SPIx support is	disabled				
bit 14	SPIFSD: Fran	me Sync Pulse	Direction Cont	rol on SSx Pin	bit		
	1 = Frame sy	/nc pulse input	(Slave) t (master)				
hit 13	SPIEPOL · Er	ame Sync Puls	e Polarity bit (F	Frame mode on			
bit 15	1 = Frame system	/nc nulse is act	ive-high	Tame mode of	iiy)		
	0 = Frame sy	/nc pulse is act	ive-low				
bit 12-2	Unimplemen	ted: Read as '	כי				
bit 1	SPIFE: Frame	e Sync Pulse E	dge Select bit				
	1 = Frame sy	/nc pulse coinc	ides with the fi	rst bit clock			
	0 = Frame sy	/nc pulse prece	des the first bi	t clock			
bit 0	SPIBEN: Enh	nanced Buffer E	nable bit				
	1 = Enhance	d buffer is enab	oled				
	0 = Enhance	d buffer is disal	bled (Legacy n	node)			

REGISTER 15-3: SPIxCON2: SPIx CONTROL REGISTER 2

BDs have a fixed relationship to a particular endpoint, depending on the buffering configuration. Table 18-2 provides the mapping of BDs to endpoints. This relationship also means that gaps may occur in the BDT if endpoints are not enabled contiguously. This, theoretically, means that the BDs for disabled endpoints could be used as buffer space. In practice, users should avoid using such spaces in the BDT unless a method of validating BD addresses is implemented.

18.2.1 BUFFER OWNERSHIP

Because the buffers and their BDs are shared between the CPU and the USB module, a simple semaphore mechanism is used to distinguish which is allowed to update the BD and associated buffers in memory. This is done by using the UOWN bit as a semaphore to distinguish which is allowed to update the BD and associated buffers in memory. UOWN is the only bit that is shared between the two configurations of BDnSTAT.

When UOWN is clear, the BD entry is "owned" by the microcontroller core. When the UOWN bit is set, the BD entry and the buffer memory are "owned" by the USB peripheral. The core should not modify the BD or its

corresponding data buffer during this time. Note that the microcontroller core can still read BDnSTAT while the SIE owns the buffer and vice versa.

The Buffer Descriptors have a different meaning based on the source of the register update. Register 18-1 and Register 18-2 show the differences in BDnSTAT depending on its current "ownership".

When UOWN is set, the user can no longer depend on the values that were written to the BDs. From this point, the USB module updates the BDs as necessary, overwriting the original BD values. The BDnSTAT register is updated by the SIE with the token PID and the transfer count is updated.

18.2.2 DMA INTERFACE

The USB OTG module uses a dedicated DMA to access both the BDT and the endpoint data buffers. Since part of the address space of the DMA is dedicated to the Buffer Descriptors, a portion of the memory connected to the DMA must comprise a contiguous address space properly mapped for the access by the module.

TABLE 18-2:	ASSIGNMENT OF BUFFER DESCRIPTORS FOR THE DIFFERENT
	BUFFERING MODES

	BDs Assigned to Endpoint											
Endpoint	Mode 0 (No Ping-Pong)		Mode 1 (Ping-Pong on EP0 OUT)		Mode 2 (Ping-Pong on all EPs)		Mode 3 (Ping-Pong on all other EPs, except EP0)					
	Out	In	Out	In	Out	In	Out	In				
0	0	1	0 (E), 1 (O)	2	0 (E), 1 (O)	2 (E), 3 (O)	0	1				
1	2	3	3	4	4 (E), 5 (O)	6 (E), 7 (O)	2 (E), 3 (O)	4 (E), 5 (O)				
2	4	5	5	6	8 (E), 9 (O)	10 (E), 11 (O)	6 (E), 7 (O)	8 (E), 9 (O)				
3	6	7	7	8	12 (E), 13 (O)	14 (E), 15 (O)	10 (E), 11 (O)	12 (E), 13 (O)				
4	8	9	9	10	16 (E), 17 (O)	18 (E), 19 (O)	14 (E), 15 (O)	16 (E), 17 (O)				
5	10	11	11	12	20 (E), 21 (O)	22 (E), 23 (O)	18 (E), 19 (O)	20 (E), 21 (O)				
6	12	13	13	14	24 (E), 25 (O)	26 (E), 27 (O)	22 (E), 23 (O)	24 (E), 25 (O)				
7	14	15	15	16	28 (E), 29 (O)	30 (E), 31 (O)	26 (E), 27 (O)	28 (E), 29 (O)				
8	16	17	17	18	32 (E), 33 (O)	34 (E), 35 (O)	30 (E), 31 (O)	32 (E), 33 (O)				
9	18	19	19	20	36 (E), 37 (O)	38 (E), 39 (O)	34 (E), 35 (O)	36 (E), 37 (O)				
10	20	21	21	22	40 (E), 41 (O)	42 (E), 43 (O)	38 (E), 39 (O)	40 (E), 41 (O)				
11	22	23	23	24	44 (E), 45 (O)	46 (E), 47 (O)	42 (E), 43 (O)	44 (E), 45 (O)				
12	24	25	25	26	48 (E), 49 (O)	50 (E), 51 (O)	46 (E), 47 (O)	48 (E), 49 (O)				
13	26	27	27	28	52 (E), 53 (O)	54 (E), 55 (O)	50 (E), 51 (O)	52 (E), 53 (O)				
14	28	29	29	30	56 (E), 57 (O)	58 (E), 59 (O)	54 (E), 55 (O)	56 (E), 57 (O)				
15	30	31	31	32	60 (E), 61 (O)	62 (E), 63 (O)	58 (E), 59 (O)	60 (E), 61 (O)				

Legend: (E) = Even transaction buffer, (O) = Odd transaction buffer

REGISTER 18-20: U1EIE: USB ERROR INTERRUPT ENABLE REGISTER									
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0		
_	<u> </u>	<u> </u>	<u> </u>	<u> </u>	<u> </u>		_		
bit 15							bit 8		
Г									
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	PIDEE		
hit 7						EOFEE	bit 0		
DIC 7							bit 0		
Legend:									
R = Readable	e bit	W = Writable	bit	U = Unimplem	nented bit, read	1 as '0'			
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkn	own		
bit 15-8	Unimplemen	ted: Read as 'd)'						
bit 7	BTSEE: Bit S	tuff Error Interr	upt Enable bit						
	1 = Interrupt	is enabled							
hit 6		ted: Read as '	۰ ،						
DIL O		Leu: Reau as () t Enchlo hit						
DIL D	1 = Interrunt	is enabled							
	0 = Interrupt	is disabled							
bit 4	BTOEE: Bus	Turnaround Tir	ne-out Error In	terrupt Enable	bit				
	1 = Interrupt	is enabled							
	0 = Interrupt	is disabled							
bit 3	DFN8EE: Dat	ta Field Size Er	ror Interrupt Er	hable bit					
	1 = Interrupt 0 = Interrupt	is enabled							
bit 2	CRC16EE: C	RC16 Failure Ir	nterrupt Enable	e bit					
	1 = Interrupt	is enabled	·						
	0 = Interrupt	is disabled							
bit 1	For Device me CRC5EE: CR	<u>ode:</u> C5 Host Error	Interrupt Enabl	e bit					
	1 = Interrupt	is enabled							
	0 = Interrupt	is disabled							
	EOFEE: Fnd-	<u>ie:</u> .of-Frame Error	interrupt Enak	ole bit					
	1 = Interrupt	is enabled							
	0 = Interrupt	is disabled							
bit 0	PIDEE: PID C	Check Failure Ir	nterrupt Enable	bit					
	1 = Interrupt	is enabled							
		IS UISADIEU							

Pin Name	Туре	Description
PMA<22:16>	0	Address bus bits<22-16>
	0	Address bus bit<15>
PMA<15> PMCS2	0	Chip Select 2 (alternate location)
1 WA 132, 1 W032	I/O	Data bus bit<15> when port size is 16 bits and address is multiplexed
	0	Address bus bit<14>
PMA<14> PMCS1	0	Chip Select 1 (alternate location)
	I/O	Data bus bit 14 when port size is 16-bit and address is multiplexed
	0	Address bus bit< 13-8>
PMA<13:8>	I/O	Data bus bits<13-8> when port size is 16 bits and address is multiplexed
PMA<7:3>	0	Address bus bit< 7-3>
PMA<2>, PMALU	0	Address bus bit<2>
	0	Address latch upper strobe for multiplexed address
	I/O	Address bus bit<1>
	0	Address latch high strobe for multiplexed address
	I/O	Address bus bit<0>
FIVIASU2, FIVIALL	0	Address latch low strobe for multiplexed address
PMD<15:8>	I/O	Data bus bits<15-8> when address is not multiplexed
	I/O	Data bus bits<7-4>
PMD<7:4>	0	Address bus bits<7-4> when port size is 4 bits and address is multiplexed with 1 address phase
PMD<3:0>	I/O	Data bus bits<3-0>
PMCS1	I/O	Chip Select 1
PMCS2	0	Chip Select 2
PMWR, PMENB	I/O	Write strobe or Enable signal depending on Strobe mode
PMRD, PMRD/PMWR	I/O	Read strobe or Read/Write signal depending on Strobe mode
PMBE1	0	Byte indicator
PMBE0	0	Nibble or byte indicator
PMACK1		Acknowledgment 1
PMACK2		Acknowledgment 2

TABLE 19-2: PARALLEL MASTER PORT PIN DESCRIPTION

REGISTER 19-6:	PMCSxBS: CH	IP SELECT x BASE	ADDRESS REGISTER

R/W ⁽¹⁾	R/W ⁽¹⁾	R/W ⁽¹⁾	R/W ⁽¹⁾				
BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16
bit 15							bit 8
R/W ⁽¹⁾	U-0	U-0	U-0	R/W ⁽¹⁾	U-0	U-0	U-0
BASE15		—	—	BASE11	—	—	—
bit 7							bit 0
Legend:							
R = Readabl	e bit	W = Writable	bit	U = Unimplemented bit, read as '0'			
-n = Value at	POR	'1' = Bit is set	t	'0' = Bit is cle	ared	x = Bit is unkno	wn

bit 15-7 BASE<23:15>: Chip Select x Base Address bits⁽²⁾

bit 6-4 Unimplemented: Read as '0'

bit 3 BASE<11>: Chip Select x Base Address bits⁽²⁾

bit 2-0 Unimplemented: Read as '0'

Note 1: Value at POR is 0x0200 for PMCS1BS and 0x0600 for PMCS2BS.

2: If the whole PMCS2BS register is written together as 0x0000, then the last EDS address for the Chip Select 1 will be 0xFFFFFF. In this case, the Chip Select 2 should not be used. PMCS1BS has no such feature.

NOTES:

ALWDHR: ALARM WEEKDAY AND HOURS VALUE REGISTER⁽¹⁾ REGISTER 20-9:

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8
U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0
Legend:							
R = Readable bit W = Writable bit			bit	U = Unimplem	nented bit, read	l as '0'	

-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
R = Readable bit	vv = vvntable bit	O = Onimplemented bit, read	

bit 15-11	Unimplemented: Read as '0'
bit 10-8	WDAY<2:0>: Binary Coded Decimal Value of Weekday Digit bits
	Contains a value from 0 to 6.
bit 7-6	Unimplemented: Read as '0'
bit 5-4	HRTEN<1:0>: Binary Coded Decimal Value of Hour's Tens Digit bits
	Contains a value from 0 to 2.
bit 3-0	HRONE<3:0>: Binary Coded Decimal Value of Hour's Ones Digit bits
	Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 20-10: ALMINSEC: ALARM MINUTES AND SECONDS VALUE REGISTER

U-0	R/W-x						
_	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8
U-0	R/W-x						
_	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

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

bit 15	Unimplemented: Read as '0'
bit 14-12	MINTEN<2:0>: Binary Coded Decimal Value of Minute's Tens Digit bits
	Contains a value from 0 to 5.
bit 11-8	MINONE<3:0>: Binary Coded Decimal Value of Minute's Ones Digit bits
	Contains a value from 0 to 9.
bit 7	Unimplemented: Read as '0'
bit 6-4	SECTEN<2:0>: Binary Coded Decimal Value of Second's Tens Digit bits
	Contains a value from 0 to 5.
bit 3-0	SECONE<3:0>: Binary Coded Decimal Value of Second's Ones Digit bits
	Contains a value from 0 to 9.

bit 0

NOTES:

REGISTER 22-32: G1DBEN: DATA I/O PAD ENABLE REGISTER

		-					
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GDBEN15	GDBEN14	GDBEN13	GDBEN12	GDBEN11	GDBEN10	GDBEN9	GDBEN8
bit 15							bit 8

| R/W-0 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| GDBEN7 | GDBEN6 | GDBEN5 | GDBEN4 | GDBEN3 | GDBEN2 | GDBEN1 | GDBEN0 |
| 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 15-0 **GDBEN<15:0>:** Display Data Pads Output Enable bits

1 = Corresponding display data (GD<x>) pin is enabled

0 = Corresponding display data (GD<x>) pin is disabled

GDBEN<15:0> can be used to disable or enable specific data signals while the DPPINOE bit (G1CON3<9>) is set.

DPPINOE	GDBENx (where x = 0 to 15)	
1	1	Display data signal (GD) associated with GDBENx is enabled.
1	0	Display data signal (GD) associated with GDBENx is disabled.
0	x	Display data signal (GD) associated with GDBENx is disabled.

27.4 Program Verification and Code Protection

PIC24FJ256DA210 family devices provide two complimentary methods to protect application code from overwrites and erasures. These also help to protect the device from inadvertent configuration changes during run time.

27.4.1 GENERAL SEGMENT PROTECTION

For all devices in the PIC24FJ256DA210 family, the on-chip program memory space is treated as a single block, known as the General Segment (GS). Code protection for this block is controlled by one Configuration bit, GCP. This bit inhibits external reads and writes to the program memory space. It has no direct effect in normal execution mode.

Write protection is controlled by the GWRP bit in the Configuration Word. When GWRP is programmed to '0', internal write and erase operations to program memory are blocked.

27.4.2 CODE SEGMENT PROTECTION

In addition to global General Segment protection, a separate subrange of the program memory space can be individually protected against writes and erases. This area can be used for many purposes where a separate block of write and erase-protected code is needed, such as bootloader applications. Unlike common boot block implementations, the specially protected segment in the PIC24FJ256DA210 family devices can be located by the user anywhere in the program space and configured in a wide range of sizes.

Code segment protection provides an added level of protection to a designated area of program memory by disabling the NVM safety interlock whenever a write or erase address falls within a specified range. It does not override General Segment protection controlled by the GCP or GWRP bits. For example, if GCP and GWRP are enabled, enabling segmented code protection for the bottom half of program memory does not undo General Segment protection for the top half. The size and type of protection for the segmented code range are configured by the WPFPx, WPEND, WPCFG and WPDIS bits in Configuration Word 3. Code segment protection is enabled by programming the WPDIS bit (= 0). The WPFP bits specify the size of the segment to be protected, by specifying the 512-word code page that is the start or end of the protected segment. The specified region is inclusive, therefore, this page will also be protected.

The WPEND bit determines if the protected segment uses the top or bottom of the program space as a boundary. Programming WPEND (= 0) sets the bottom of program memory (000000h) as the lower boundary of the protected segment. Leaving WPEND unprogrammed (= 1) protects the specified page through the last page of implemented program memory, including the Configuration Word locations.

A separate bit, WPCFG, is used to protect the last page of program space, including the Flash Configuration Words. Programming WPCFG (= 0) protects the last page in addition to the pages selected by the WPEND and WPFP<7:0> bits setting. This is useful in circumstances where write protection is needed for both the code segment in the bottom of the memory and the Flash Configuration Words.

The various options for segment code protection are shown in Table 27-2.

30.1 DC Characteristics





TABLE 30-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Тур	Max	Unit
PIC24FJ256DA210 family:					
Operating Junction Temperature Range	TJ	-40	—	+125	°C
Operating Ambient Temperature Range	TA	-40	—	+85	°C
Power Dissipation (with ENVREG = 1): Internal Chip Power Dissipation: $PINT = VDD x (IDD - \Sigma IOH)$ I/O Pin Power Dissipation: $PI/O = \Sigma (\{VDD - VOH\} x IOH) + \Sigma (VOL x IOL)$	PD	I	W		
Maximum Allowed Power Dissipation	PDMAX	(Τјмах – Τα)/θја			W

TABLE 30-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Тур	Max	Unit	Note
Package Thermal Resistance, 12x12x1 mm TQFP	θJA	69.4		°C/W	(Note 1)
Package Thermal Resistance, 10x10x1 mm TQFP	θJA	76.6	—	°C/W	(Note 1)
Package Thermal Resistance, 9x9x0.9 mm QFN	θJA	28.0	_	°C/W	(Note 1)
Package Thermal Resistance, 10x10x1.1 mm BGA	θJA	40.2	_	°C/W	(Note 1)

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

DC CHARACTERISTICS		Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated)					
	1	1	Operating tempe	$-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial			
Param No.	Symbo I	Characteristic	Min	Min Typ ⁽¹⁾		Units	Conditions
	VIL	Input Low Voltage ⁽³⁾					
DI10		I/O Pins with ST Buffer	Vss	—	0.2 Vdd	V	
DI11		I/O Pins with TTL Buffer	Vss	—	0.15 Vdd	V	
DI15		MCLR	Vss	—	0.2 VDD	V	
DI16		OSCI (XT mode)	Vss	—	0.2 VDD	V	
DI17		OSCI (HS mode)	Vss	—	0.2 VDD	V	
DI18		I/O Pins with I ² C™ Buffer:	Vss	—	0.3 Vdd	V	
DI19		I/O Pins with SMBus Buffer:	Vss	—	0.8	V	SMBus enabled
	Vih	Input High Voltage ⁽³⁾					
DI20		I/O Pins with ST Buffer: with Analog Functions, Digital Only	0.8 Vdd 0.8 Vdd	_	Vdd 5.5	V V	
DI21		I/O Pins with TTL Buffer: with Analog Functions, Digital Only	0.25 VDD + 0.8 0.25 VDD + 0.8	_	Vdd 5.5	V V	
DI25		MCLR	0.8 VDD	—	Vdd	V	
DI26		OSCI (XT mode)	0.7 Vdd	—	Vdd	V	
DI27		OSCI (HS mode)	0.7 Vdd	—	Vdd	V	
DI28		I/O Pins with I ² C™ Buffer: with Analog Functions, Digital Only	0.7 Vdd 0.7 Vdd	_	Vdd 5.5	V V	
DI29		I/O Pins with SMBus Buffer: with Analog Functions, Digital Only	2.1 2.1		Vdd 5.5	V V	$2.5V \leq V\text{PIN} \leq V\text{DD}$
DI30	ICNPU	CNxx Pull-up Current	150	350	550	μA	VDD = 3.3V, VPIN = VSS
DI30A	ICNPD	CNxx Pull-down Current	15	70	150	μA	VDD = 3.3V, VPIN = VDD
	lı∟	Input Leakage Current ⁽²⁾					
DI50		I/O Ports	_	_	<u>+</u> 1	μΑ	$Vss \le VPIN \le VDD$, pin at high-impedance
DI51		Analog Input Pins	_	—	<u>+</u> 1	μA	$\label{eq:VSS} \begin{split} &VSS \leq V\text{PIN} \leq V\text{DD}, \\ &\text{pin at high-impedance} \end{split}$
DI55		MCLR	—	—	<u>+</u> 1	μA	$VSS \leq VPIN \leq VDD$
DI56		OSCI/CLKI	—	_	<u>+</u> 1	μA	$\label{eq:VSS} \begin{array}{l} VSS \leq VPIN \leq VDD, \\ EC, XT \text{and} HS \text{modes} \end{array}$

TABLE 30-7: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

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

2: Negative current is defined as current sourced by the pin.

3: Refer to Table 1-1 for I/O pins buffer types.





TABLE 30-18: CLKO AND I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial					
Param No.	Symbol	Characteristic	cteristic Min Typ ⁽¹⁾ Max			Units	Conditions	
DO31	TIOR	Port Output Rise Time	_	10	25	ns		
DO32	TIOF	Port Output Fall Time	_	10	25	ns		
DI35	TINP	INTx Pin High or Low Time (input)	20	—	—	ns		
DI40	Trbp	CNx High or Low Time (input)	2	—	—	Тсү		

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