



Welcome to [E-XFL.COM](https://www.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	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	80KB (40K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2682t-i-so

3.4.3 RC_IDLE MODE

In RC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator block using the INTOSC multiplexer. This mode allows for controllable power conservation during Idle periods.

From RC_RUN, this mode is entered by setting the IDLEN bit and executing a `SLEEP` instruction. If the device is in another Run mode, first set IDLEN, then set the SCS1 bit and execute `SLEEP`. Although its value is ignored, it is recommended that SCS0 also be cleared; this is to maintain software compatibility with future devices. The INTOSC multiplexer may be used to select a higher clock frequency, by modifying the IRCF bits, before executing the `SLEEP` instruction. When the clock source is switched to the INTOSC multiplexer, the primary oscillator is shut down and the OSTS bit is cleared.

If the IRCF bits are set to any non-zero value or the INTSRC bit is set, the INTOSC output is enabled. The IOFS bit becomes set, after the INTOSC output becomes stable, after an interval of TIOBST (parameter 39, Table 27-10). Clocks to the peripherals continue while the INTOSC source stabilizes. If the IRCF bits were previously at a non-zero value, or INTSRC was set before the `SLEEP` instruction was executed and the INTOSC source was already stable, the IOFS bit will remain set. If the IRCF bits and INTSRC are all clear, the INTOSC output will not be enabled, the IOFS bit will remain clear and there will be no indication of the current clock source.

When a wake event occurs, the peripherals continue to be clocked from the INTOSC multiplexer. After a delay of TCSD following the wake event, the CPU begins executing code being clocked by the INTOSC multiplexer. The IDLEN and SCS bits are not affected by the wake-up. The INTRC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

3.5 Exiting Idle and Sleep Modes

An exit from Sleep mode or any of the Idle modes is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power-managed modes. The clocking subsystem actions are discussed in each of the power-managed modes (see **Section 3.2 “Run Modes”**, **Section 3.3 “Sleep Mode”** and **Section 3.4 “Idle Modes”**).

3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode or the Sleep mode to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

On all exits from Idle or Sleep modes by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 9.0 “Interrupts”**).

A fixed delay of interval TCSD following the wake event is required when leaving the Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

3.5.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power-managed mode (see **Section 3.2 “Run Modes”** and **Section 3.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 24.2 “Watchdog Timer (WDT)”**).

The WDT timer and postscaler are cleared by executing a `SLEEP` or `CLRWDT` instruction, the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifying the IRCF bits in the OSCCON register if the internal oscillator block is the device clock source.

3.5.3 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock becomes ready. At that time, the OSTS bit is set and the device begins executing code. If the internal oscillator block is the new clock source, the IOFS bit is set instead.

The exit delay time from Reset to the start of code execution depends on both the clock sources before and after the wake-up and the type of oscillator if the new clock source is the primary clock. Exit delays are summarized in Table 3-2.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see **Section 24.3 “Two-Speed Start-up”**) or Fail-Safe Clock Monitor (see **Section 24.4 “Fail-Safe Clock Monitor”**) is enabled, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Execution is clocked by the internal oscillator block until either the primary clock becomes ready or a power-managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

PIC18F2682/2685/4682/4685

**TABLE 5-1: SPECIAL FUNCTION REGISTER MAP FOR
PIC18F2682/2685/4682/4685 DEVICES (CONTINUED)**

Address	Name	Address	Name	Address	Name	Address	Name
DFFh	—	DDFh	—	DBFh	—	D9Fh	—
DFEh	—	DDEh	—	DBEh	—	D9Eh	—
DFDh	—	DDDh	—	DBDh	—	D9Dh	—
DFCh	TXBIE	DDCh	—	DBCh	—	D9Ch	—
DFBh	—	DDbh	—	DBbh	—	D9Bh	—
DFAh	BIE0	DDAh	—	DBAh	—	D9Ah	—
DF9h	—	DD9h	—	DB9h	—	D99h	—
DF8h	BSEL0	DD8h	SDFLC	DB8h	—	D98h	—
DF7h	—	DD7h	—	DB7h	—	D97h	—
DF6h	—	DD6h	—	DB6h	—	D96h	—
DF5h	—	DD5h	RXFCON1	DB5h	—	D95h	—
DF4h	—	DD4h	RXFCON0	DB4h	—	D94h	—
DF3h	MSEL3	DD3h	—	DB3h	—	D93h	RXF15EIDL
DF2h	MSEL2	DD2h	—	DB2h	—	D92h	RXF15EIDH
DF1h	MSEL1	DD1h	—	DB1h	—	D91h	RXF15SIDL
DF0h	MSEL0	DD0h	—	DB0h	—	D90h	RXF15SIDH
DEFh	—	DCFh	—	DAFh	—	D8Fh	—
DEEh	—	DCEh	—	DAEh	—	D8Eh	—
DEDh	—	DCDh	—	DADh	—	D8Dh	—
DECh	—	DCCh	—	DACH	—	D8Ch	—
DEBh	—	DCBh	—	DABh	—	D8Bh	RXF14EIDL
DEAh	—	DCAh	—	DAAh	—	D8Ah	RXF14EIDH
DE9h	—	DC9h	—	DA9h	—	D89h	RXF14SIDL
DE8h	—	DC8h	—	DA8h	—	D88h	RXF14SIDH
DE7h	RXFBCON7	DC7h	—	DA7h	—	D87h	RXF13EIDL
DE6h	RXFBCON6	DC6h	—	DA6h	—	D86h	RXF13EIDH
DE5h	RXFBCON5	DC5h	—	DA5h	—	D85h	RXF13SIDL
DE4h	RXFBCON4	DC4h	—	DA4h	—	D84h	RXF13SIDH
DE3h	RXFBCON3	DC3h	—	DA3h	—	D83h	RXF12EIDL
DE2h	RXFBCON2	DC2h	—	DA2h	—	D82h	RXF12EIDH
DE1h	RXFBCON1	DC1h	—	DA1h	—	D81h	RXF12SIDL
DE0h	RXFBCON0	DC0h	—	DA0h	—	D80h	RXF12SIDH

- Note** 1: Registers available only on PIC18F4X8X devices; otherwise, the registers read as '0'.
2: When any TX_ENn bit in RX_TX_SELn is set, then the corresponding bit in this register has transmit properties.
3: This is not a physical register.

PIC18F2682/2685/4682/4685

REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
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 7	INT2IP: INT2 External Interrupt Priority bit 1 = High priority 0 = Low priority
bit 6	INT1IP: INT1 External Interrupt Priority bit 1 = High priority 0 = Low priority
bit 5	Unimplemented: Read as '0'
bit 4	INT2IE: INT2 External Interrupt Enable bit 1 = Enables the INT2 external interrupt 0 = Disables the INT2 external interrupt
bit 3	INT1IE: INT1 External Interrupt Enable bit 1 = Enables the INT1 external interrupt 0 = Disables the INT1 external interrupt
bit 2	Unimplemented: Read as '0'
bit 1	INT2IF: INT2 External Interrupt Flag bit 1 = The INT2 external interrupt occurred (must be cleared in software) 0 = The INT2 external interrupt did not occur
bit 0	INT1IF: INT1 External Interrupt Flag bit 1 = The INT1 external interrupt occurred (must be cleared in software) 0 = The INT1 external interrupt did not occur

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

PIC18F2682/2685/4682/4685

16.4.7.1 Auto-Shutdown and Auto-Restart

The auto-shutdown feature can be configured to allow automatic restarts of the module following a shutdown event. This is enabled by setting the PRSEN bit of the ECCP1DEL register (ECCP1DEL<7>).

In Shutdown mode with PRSEN = 1 (Figure 16-10), the ECCPASE bit will remain set for as long as the cause of the shutdown continues. When the shutdown condition clears, the ECCPASE bit is cleared. If PRSEN = 0 (Figure 16-11), once a shutdown condition occurs, the ECCPASE bit will remain set until it is cleared by firmware. Once ECCPASE is cleared, the Enhanced PWM will resume at the beginning of the next PWM period.

Note: Writing to the ECCPASE bit is disabled while a shutdown condition is active.

Independent of the PRSEN bit setting, if the auto-shutdown source is one of the comparators, the shutdown condition is a level. The ECCPASE bit cannot be cleared as long as the cause of the shutdown persists.

The Auto-Shutdown mode can be forced by writing a '1' to the ECCPASE bit.

16.4.8 START-UP CONSIDERATIONS

When the ECCP1 module is used in the PWM mode, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins. When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the OFF state until the microcontroller drives the I/O pins with the proper signal levels, or activates the PWM output(s).

The ECCP1M1:ECCP1M0 bits (ECCP1CON<1:0>) allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (P1A/P1C and P1B/P1D). The PWM output polarities must be selected before the PWM pins are configured as outputs. Changing the polarity configuration while the PWM pins are configured as outputs is not recommended, since it may result in damage to the application circuits.

The P1A, P1B, P1C and P1D output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pins for output at the same time as the ECCP1 module may cause damage to the application circuit. The ECCP1 module must be enabled in the proper output mode and complete a full PWM cycle before configuring the PWM pins as outputs. The completion of a full PWM cycle is indicated by the TMR2IF bit being set as the second PWM period begins.

FIGURE 16-10: PWM AUTO-SHUTDOWN (PRSEN = 1, AUTO-RESTART ENABLED)

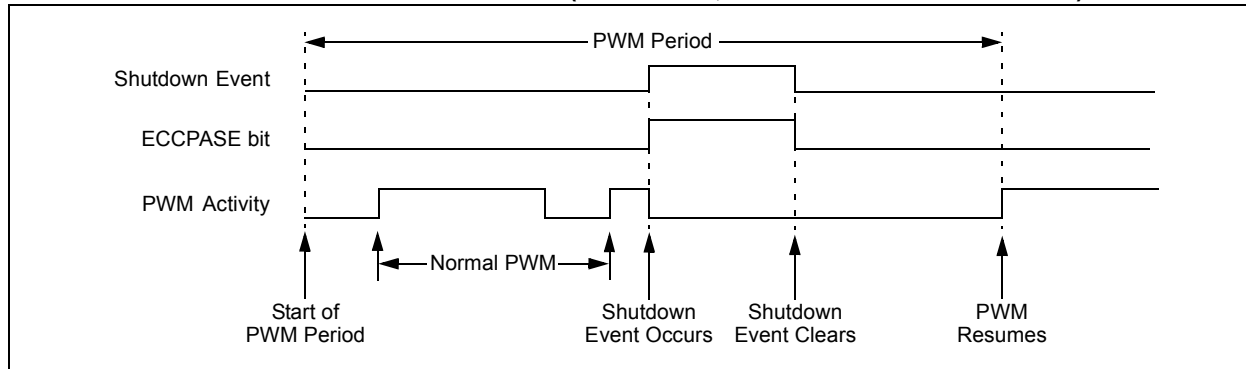


FIGURE 16-11: PWM AUTO-SHUTDOWN (PRSEN = 0, AUTO-RESTART DISABLED)

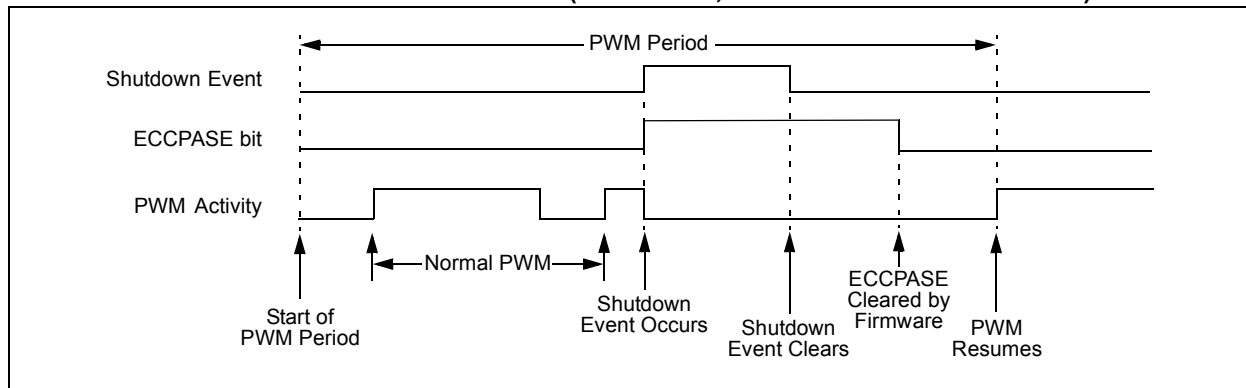


FIGURE 18-4: ASYNCHRONOUS TRANSMISSION

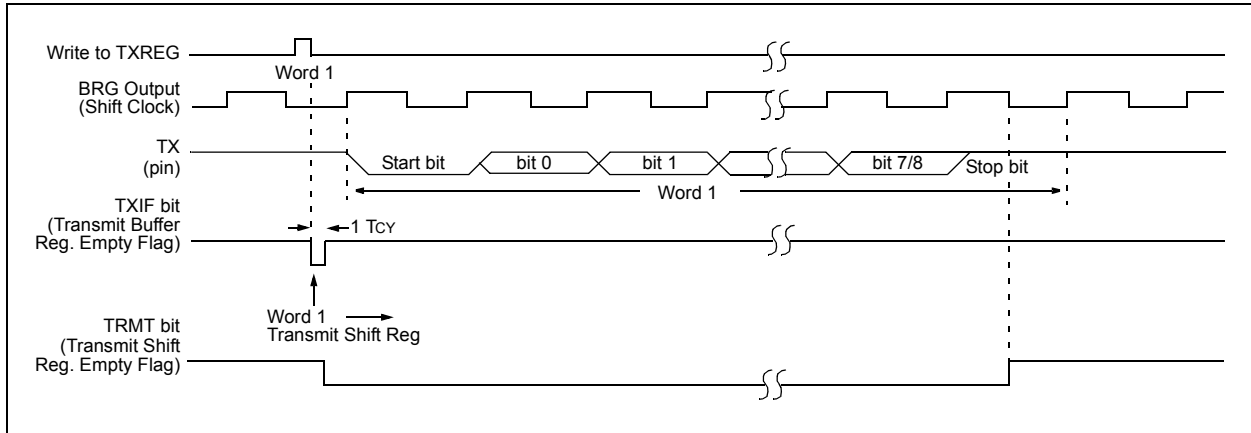


FIGURE 18-5: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)

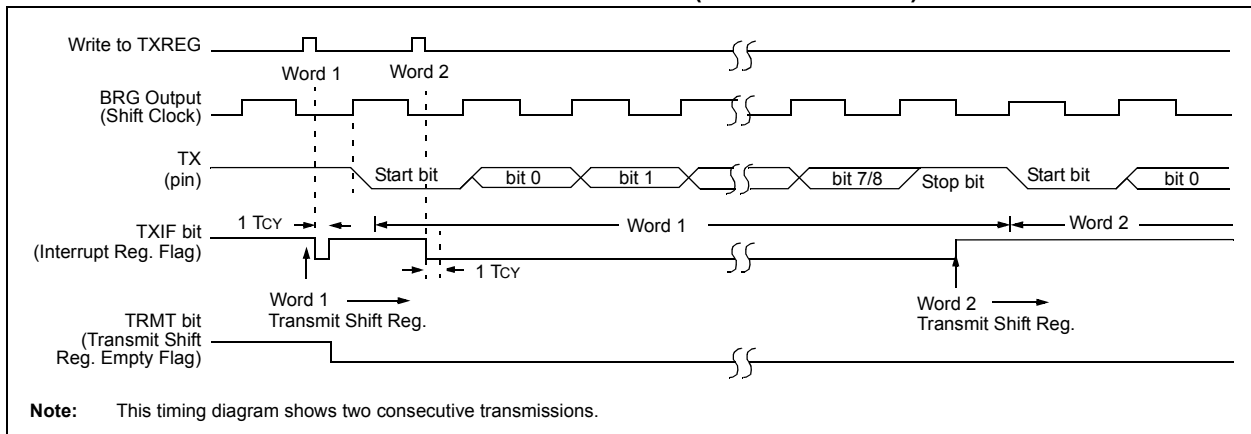


TABLE 18-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	53
TXREG	EUSART Transmit Register								53
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	53
SPBRGH	EUSART Baud Rate Generator Register High Byte								53
SPBRG	EUSART Baud Rate Generator Register Low Byte								53

Legend: — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Reserved in PIC18F2682/2685 devices; always maintain these bits clear.

PIC18F2682/2685/4682/4685

22.6 Operation During Sleep

When enabled, the HLVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the HLVDIF bit will be set and the device will wake-up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

22.7 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the HLVD module to be turned off.

TABLE 22-1: REGISTERS ASSOCIATED WITH HIGH/LOW-VOLTAGE DETECT MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
HLVDCON	VDIRMAG	—	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	52
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
PIR2	OSCFIF	CMIF ⁽¹⁾	—	EEIF	BCLIF	HLVDIF	TMR3IF	ECCP1IF ⁽¹⁾	54
PIE2	OSCFIE	CMIE ⁽¹⁾	—	EEIE	BCLIE	HLVDIE	TMR3IE	ECCP1IE ⁽¹⁾	54
IPR2	OSCFIP	CMIP ⁽¹⁾	—	EEIP	BCLIP	HLVDIP	TMR3IP	ECCP1IP ⁽¹⁾	53

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the HLVD module.

Note 1: These bits are available in PIC18F4682/4685 devices and reserved in PIC18F2682/2685 devices.

PIC18F2682/2685/4682/4685

EXAMPLE 23-3: TRANSMITTING A CAN MESSAGE USING BANKED METHOD

```
; Need to transmit Standard Identifier message 123h using TXB0 buffer.
; To successfully transmit, CAN module must be either in Normal or Loopback mode.
; TXB0 buffer is not in access bank. And since we want banked method, we need to make sure
; that correct bank is selected.
BANKSEL TXB0CON                ; One BANKSEL in beginning will make sure that we are
                                ; in correct bank for rest of the buffer access.

; Now load transmit data into TXB0 buffer.
MOVLW MY_DATA_BYTE1            ; Load first data byte into buffer
MOVWF TXB0D0                    ; Compiler will automatically set "BANKED" bit
; Load rest of data bytes - up to 8 bytes into TXB0 buffer.
...
; Load message identifier
MOVLW 60H                      ; Load SID2:SID0, EXIDE = 0
MOVWF TXB0SIDL
MOVLW 24H                      ; Load SID10:SID3
MOVWF TXB0SIDH
; No need to load TXB0EIDL:TXB0EIDH, as we are transmitting Standard Identifier Message only.

; Now that all data bytes are loaded, mark it for transmission.
MOVLW B'00001000'              ; Normal priority; Request transmission
MOVWF TXB0CON

; If required, wait for message to get transmitted
BTFSC TXB0CON, TXREQ            ; Is it transmitted?
BRA $-2                        ; No. Continue to wait...

; Message is transmitted.
```

PIC18F2682/2685/4682/4685

REGISTER 23-35: BnDLC: TX/RX BUFFER n DATA LENGTH CODE REGISTERS IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL<n>) = 1]⁽¹⁾

U-0	R/W-x	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0
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 7 **Unimplemented:** Read as '0'
- bit 6 **TXRTR:** Transmitter Remote Transmission Request bit
1 = Transmitted message will have RTR bit set
0 = Transmitted message will have RTR bit cleared
- bit 5-4 **Unimplemented:** Read as '0'
- bit 3-0 **DLC3:DLC0:** Data Length Code bits
1111-1001 = Reserved
1000 = Data length = 8 bytes
0111 = Data length = 7 bytes
0110 = Data length = 6 bytes
0101 = Data length = 5 bytes
0100 = Data length = 4 bytes
0011 = Data length = 3 bytes
0010 = Data length = 2 bytes
0001 = Data length = 1 bytes
0000 = Data length = 0 bytes

Note 1: These registers are available in Mode 1 and 2 only.

REGISTER 23-36: BSEL0: BUFFER SELECT REGISTER 0⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
B5TXEN	B4TXEN	B3TXEN	B2TXEN	B1TXEN	B0TXEN	—	—
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 7-2 **B5TXEN:B0TXEN:** Buffer 5 to Buffer 0 Transmit Enable bit
1 = Buffer is configured in Transmit mode
0 = Buffer is configured in Receive mode
- bit 1-0 **Unimplemented:** Read as '0'

Note 1: These registers are available in Mode 1 and 2 only.

PIC18F2682/2685/4682/4685

REGISTER 23-51: MSEL3: MASK SELECT REGISTER 3⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FIL15_1	FIL15_0	FIL14_1	FIL14_0	FIL13_1	FIL13_0	FIL12_1	FIL12_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 7-6 **FIL15_1:FIL15_0:** Filter 15 Select bits 1 and 0
11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0
- bit 5-4 **FIL14_1:FIL14_0:** Filter 14 Select bits 1 and 0
11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0
- bit 3-2 **FIL13_1:FIL13_0:** Filter 13 Select bits 1 and 0
11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0
- bit 1-0 **FIL12_1:FIL12_0:** Filter 12 Select bits 1 and 0
11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

Note 1: This register is available in Mode 1 and 2 only.

PIC18F2682/2685/4682/4685

REGISTER 23-54: BRGCON3: BAUD RATE CONTROL REGISTER 3

R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKDIS	WAKFIL	—	—	—	SEG2PH2 ⁽¹⁾	SEG2PH1 ⁽¹⁾	SEG2PH0 ⁽¹⁾
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 7 **WAKDIS:** Wake-up Disable bit

1 = Disable CAN bus activity wake-up feature

0 = Enable CAN bus activity wake-up feature

bit 6 **WAKFIL:** Selects CAN bus Line Filter for Wake-up bit

1 = Use CAN bus line filter for wake-up

0 = CAN bus line filter is not used for wake-up

bit 5-3 **Unimplemented:** Read as '0'

bit 2-0 **SEG2PH2:SEG2PH0:** Phase Segment 2 Time Select bits⁽¹⁾

111 = Phase Segment 2 time = 8 x T_Q

110 = Phase Segment 2 time = 7 x T_Q

101 = Phase Segment 2 time = 6 x T_Q

100 = Phase Segment 2 time = 5 x T_Q

011 = Phase Segment 2 time = 4 x T_Q

010 = Phase Segment 2 time = 3 x T_Q

001 = Phase Segment 2 time = 2 x T_Q

000 = Phase Segment 2 time = 1 x T_Q

Note 1: Ignored if SEG2PHTS bit (BRGCON2<7>) is '0'.

PIC18F2682/2685/4682/4685

TABLE 23-1: CAN CONTROLLER REGISTER MAP (CONTINUED)

Address ⁽¹⁾	Name	Address	Name	Address	Name	Address	Name
E7Fh	CANCON_RO4 ⁽²⁾	E5Fh	CANCON_RO6 ⁽²⁾	E3Fh	CANCON_RO8 ⁽²⁾	E1Fh	— ⁽⁴⁾
E7Eh	CANSTAT_RO4 ⁽²⁾	E5Eh	CANSTAT_RO6 ⁽²⁾	E3Eh	CANSTAT_RO8 ⁽²⁾	E1Eh	— ⁽⁴⁾
E7Dh	B5D7	E5Dh	B3D7	E3Dh	B1D7	E1Dh	— ⁽⁴⁾
E7Ch	B5D6	E5Ch	B3D6	E3Ch	B1D6	E1Ch	— ⁽⁴⁾
E7Bh	B5D5	E5Bh	B3D5	E3Bh	B1D5	E1Bh	— ⁽⁴⁾
E7Ah	B5D4	E5Ah	B3D4	E3Ah	B1D4	E1Ah	— ⁽⁴⁾
E79h	B5D3	E59h	B3D3	E39h	B1D3	E19h	— ⁽⁴⁾
E78h	B5D2	E58h	B3D2	E38h	B1D2	E18h	— ⁽⁴⁾
E77h	B5D1	E57h	B3D1	E37h	B1D1	E17h	— ⁽⁴⁾
E76h	B5D0	E56h	B3D0	E36h	B1D0	E16h	— ⁽⁴⁾
E75h	B5DLC	E55h	B3DLC	E35h	B1DLC	E15h	— ⁽⁴⁾
E74h	B5EIDL	E54h	B3EIDL	E34h	B1EIDL	E14h	— ⁽⁴⁾
E73h	B5EIDH	E53h	B3EIDH	E33h	B1EIDH	E13h	— ⁽⁴⁾
E72h	B5SIDL	E52h	B3SIDL	E32h	B1SIDL	E12h	— ⁽⁴⁾
E71h	B5SIDH	E51h	B3SIDH	E31h	B1SIDH	E11h	— ⁽⁴⁾
E70h	B5CON	E50h	B3CON	E30h	B1CON	E10h	— ⁽⁴⁾
E6Fh	CANCON_RO5	E4Fh	CANCON_RO7	E2Fh	CANCON_RO9	E0Fh	— ⁽⁴⁾
E6Eh	CANSTAT_RO5	E4Eh	CANSTAT_RO7	E2Eh	CANSTAT_RO9	E0Eh	— ⁽⁴⁾
E6Dh	B4D7	E4Dh	B2D7	E2Dh	B0D7	E0Dh	— ⁽⁴⁾
E6Ch	B4D6	E4Ch	B2D6	E2Ch	B0D6	E0Ch	— ⁽⁴⁾
E6Bh	B4D5	E4Bh	B2D5	E2Bh	B0D5	E0Bh	— ⁽⁴⁾
E6Ah	B4D4	E4Ah	B2D4	E2Ah	B0D4	E0Ah	— ⁽⁴⁾
E69h	B4D3	E49h	B2D3	E29h	B0D3	E09h	— ⁽⁴⁾
E68h	B4D2	E48h	B2D2	E28h	B0D2	E08h	— ⁽⁴⁾
E67h	B4D1	E47h	B2D1	E27h	B0D1	E07h	— ⁽⁴⁾
E66h	B4D0	E46h	B2D0	E26h	B0D0	E06h	— ⁽⁴⁾
E65h	B4DLC	E45h	B2DLC	E25h	B0DLC	E05h	— ⁽⁴⁾
E64h	B4EIDL	E44h	B2EIDL	E24h	B0EIDL	E04h	— ⁽⁴⁾
E63h	B4EIDH	E43h	B2EIDH	E23h	B0EIDH	E03h	— ⁽⁴⁾
E62h	B4SIDL	E42h	B2SIDL	E22h	B0SIDL	E02h	— ⁽⁴⁾
E61h	B4SIDH	E41h	B2SIDH	E21h	B0SIDH	E01h	— ⁽⁴⁾
E60h	B4CON	E40h	B2CON	E20h	B0CON	E00h	— ⁽⁴⁾

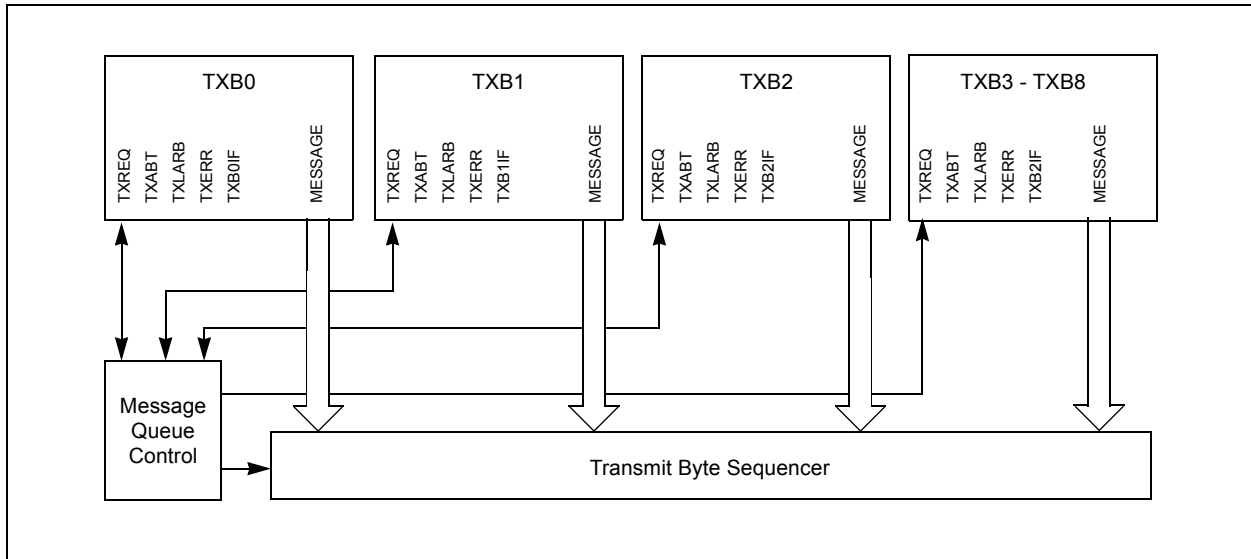
- Note 1:** Shaded registers are available in Access Bank low area, while the rest are available in Bank 15.
- Note 2:** CANSTAT register is repeated in these locations to simplify application firmware. Unique names are given for each instance of the controller register due to the Microchip header file requirement.
- Note 3:** These registers are not CAN registers.
- Note 4:** Unimplemented registers are read as '0'.

23.6.3 TRANSMIT PRIORITY

Transmit priority is a prioritization within the PIC18F2682/2685/4682/4685 devices of the pending transmittable messages. This is independent from and not related to any prioritization implicit in the message arbitration scheme built into the CAN protocol. Prior to sending the SOF, the priority of all buffers that are queued for transmission is compared. The transmit

buffer with the highest priority will be sent first. If two buffers have the same priority setting, the buffer with the highest buffer number will be sent first. There are four levels of transmit priority. If TXP bits for a particular message buffer are set to '11', that buffer has the highest possible priority. If TXP bits for a particular message buffer are set to '00', that buffer has the lowest possible priority.

FIGURE 23-2: TRANSMIT BUFFERS



PIC18F2682/2685/4682/4685

REGISTER 24-2: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	BORV1	BORV0	BOREN1 ⁽¹⁾	BOREN0 ⁽¹⁾	PWRTEN ⁽¹⁾
bit 7							bit 0

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

u = Unchanged from programmed state

bit 7-5 **Unimplemented:** Read as '0'

bit 4-3 **BORV1:BORV0:** Brown-out Reset Voltage bits

11 = Minimum setting

.

.

.

00 = Maximum setting

bit 2-1 **BOREN1:BOREN0:** Brown-out Reset Enable bits⁽¹⁾

11 = Brown-out Reset enabled in hardware only (SBOREN is disabled)

10 = Brown-out Reset enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)

01 = Brown-out Reset enabled and controlled by software (SBOREN is enabled)

00 = Brown-out Reset disabled in hardware and software

bit 0 **PWRTEN:** Power-up Timer Enable bit⁽¹⁾

1 = PWRT disabled

0 = PWRT enabled

Note 1: The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

PIC18F2682/2685/4682/4685

IORLW Inclusive OR Literal with W

Syntax: IORLW k

Operands: $0 \leq k \leq 255$

Operation: $(W) .OR. k \rightarrow W$

Status Affected: N, Z

Encoding:

0000	1001	kkkk	kkkk
------	------	------	------

Description: The contents of W are ORed with the eight-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: IORLW 35h

Before Instruction
W = 9Ah

After Instruction
W = BFh

IORWF Inclusive OR W with f

Syntax: IORWF f {,d {,a}}

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

Operation: $(W) .OR. (f) \rightarrow \text{dest}$

Status Affected: N, Z

Encoding:

0001	00da	ffff	ffff
------	------	------	------

Description: Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: IORWF RESULT, 0, 1

Before Instruction
RESULT = 13h
W = 91h

After Instruction
RESULT = 13h
W = 93h

PIC18F2682/2685/4682/4685

MOVFF Move f to f

Syntax: MOVFF f_s, f_d

Operands: $0 \leq f_s \leq 4095$
 $0 \leq f_d \leq 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1100	ffff	ffff	ffff _s
1111	ffff	ffff	ffff _d

1st word (source)

2nd word (destin.)

Description: The contents of source register ' f_s ' are moved to destination register ' f_d '. Location of source ' f_s ' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination ' f_d ' can also be anywhere from 000h to FFFh. Either source or destination can be W (a useful special situation). MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVFF REG1, REG2

Before Instruction

REG1 = 33h
 REG2 = 11h

After Instruction

REG1 = 33h
 REG2 = 33h

MOVLB Move Literal to Low Nibble in BSR

Syntax: MOVLW k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow \text{BSR}$

Status Affected: None

Encoding:

0000	0001	kkkk	kkkk
------	------	------	------

Description: The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0', regardless of the value of $k_7:k_4$.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR

Example: MOVLB 5

Before Instruction

BSR Register = 02h

After Instruction

BSR Register = 05h

26.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

26.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, pre-processor, and one-step driver, and can run on multiple platforms.

26.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

26.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility