MICROCHIP PIC18F2420/2520/4420/4520

PIC18F2420/2520/4420/4520 Data Sheet Errata

Clarifications/Corrections to the Data Sheet:

In the Device Data Sheet (DS39631**A**), the following clarifications and corrections should be noted. Any silicon issues related to the PIC18F2420/2520/4420/4520 will be reported in a separate silicon errata. Please check the Microchip web site for any existing issues.

1. Module: Section 6.0 Flash Program Memory

The second paragraph in this section has been corrected to read as follows. Changes to the text are shown in **bold**:

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of **32** bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation may not be issued from user code.

2. Module: Figure 6.2: Table Write Operation

Note 1 in Figure 6.2 has been corrected to read as follows. Changes to the text are shown in **bold**:

Note 1: Table Pointer actually points to one of **32** holding registers, the address of which is determined by TBLPTRL<**4**:0>. The process for physically writing data to the program memory array is discussed in **Section 6.5 "Writing to Flash Program Memory"**.

3. Module: Section 6.2.4 Table Pointer Boundaries

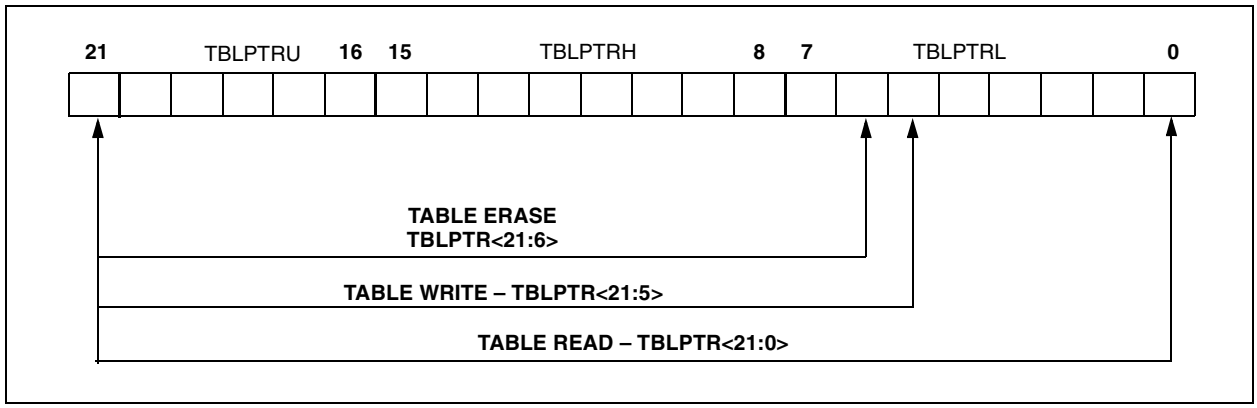
The third paragraph in this section has been corrected to read as follows. Changes to the text are shown in **bold**:

When a `TBLWT` is executed, the five LSbs of the Table Pointer register (TBLPTR<**4**:0>) determine which of the **32** program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 16 MSbs of the TBLPTR (TBLPTR<21:6>) determine which program memory block of **32** bytes is written to. For more detail, see **Section 6.5 "Writing to Flash Program Memory"**.

4. Module: Figure 6-3: Table Pointer Boundaries Based on Operation

Figure 6-3 has been replaced with the following figure:

FIGURE 6-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



5. Module: Section 6.5 Writing to Flash Program Memory

This section has been corrected to read as follows. Changes to the text are shown in **bold**:

The minimum programming block is **16** words or **32** bytes. Word or byte programming is not supported. Table writes are used internally to load the holding registers needed to program the Flash memory. There are **32** holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the `TBLWT` instruction may need to be executed **32** times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the **32** holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

The long write is necessary for programming the internal Flash. Instruction execution is halted while in a write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

Note:	The default value of the holding registers on device Resets and after write operations is FFh. A write of FFh to a holding register does not modify that byte. This means individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '0' to a '1'. When modifying individual bytes, it is not necessary to load all **32** holding registers before executing a write operation.

6. Module: Section 6.5.1 Flash Program Memory Write Sequence

This section has been corrected to read as follows. Changes to the text are shown in **bold**:

The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the row erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the **32** bytes into the holding registers with auto-increment.
7. Set the EECON1 register for the write operation:
   - set EEPGD bit to point to program memory;
   - clear the CFGS bit to access program memory;
   - set WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Verify the memory (table read).

This procedure will require about 6 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 6-3.

Note:	Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the **32** bytes in the holding register.

7. Module: Example 6-3: Writing to Flash Program Memory

In the WRITE_BUFFER_BACK section of the code in Example 6-3, the MOVLW instruction has been changed. The change to the text is shown in **bold**:

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY

```
WRITE_BUFFER_BACK
                    MOVLW     D'32                    ; number of bytes in holding register
                    MOVWF     COUNTER
```

8. Module: Configuration Word Register 2L

In Register 23-2, CONFIG2L, the description for bits 4-3 should show '11' as the *minimum* setting and '00' for the *maximum* setting. The changes to the text are shown in bold:

REGISTER 23-2: 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(1)	BORV0(1)	BOREN1(2)	BOREN0(2)	$\overline{\text{PWRTEN}}$(2)
bit 7							bit 0

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

bit 4-3 **BORV1:BORV0:** Brown-out Reset Voltage bits(1)

11 = **Minimum** setting

.

.

.

00 = **Maximum** setting

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

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 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit(2)

1 = PWRT disabled

0 = PWRT enabled

Note 1: See **Section 26.1 "DC Characteristics: Supply Voltage"** for specifications.

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

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

9. Module: Device ID Bit Values

The Device ID bit values in Register 23-12 (bit 7-5) and Register 23-13 (bit 7-0) are incorrectly stated in the Device Data Sheet. The correct values are shown in **bold** as follows:

REGISTER 23-12: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F2420/2520/4420/4520

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

bit 7-5 **DEV2:DEV0:** Device ID bits

110 = **PIC18F4420**
100 = **PIC18F4520**
010 = **PIC18F2420**
000 = **PIC18F2520**

bit 4-0 **REV4:REV0:** Revision ID bits

These bits are used to indicate the device revision.

Legend:		
R = Read-only bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

REGISTER 23-13: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F2420/2520/4420/4520

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

bit 7-0 **DEV10:DEV3:** Device ID bits

These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.

0001 0001 = **PIC18F2420/2520** devices
0001 0000 = **PIC18F4420/4520** devices

Note: These values for DEV10:DEV3 may be shared with other devices. The specific device is always identified by using the entire DEV10:DEV0 bit sequence.

Legend:		
R = Read-only bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

10. Module: Timer1

Additional text has been added to the existing Timer1 chapter to explain in more detail the proper method of updating the TMR1 registers in Asynchronous mode. The Real-Time Clock application example, cited in **Section 12.6 "Using Timer1 as a Real-Time Clock"**, has also been amended to reflect this method.

Section 12.7 "Considerations in Asynchronous Counter Mode", which follows, is added immediately following the existing Section 12.6. Example 12-1, shown below, replaces the existing Example 12-1 shown in the data sheet.

12.7 Considerations in Asynchronous Counter Mode

Following a Timer1 interrupt and an update to the TMR1 registers, the Timer1 module uses a falling edge on its clock source to trigger the next register update on the rising edge. If the update is completed after the clock input has fallen, the next rising edge will not be counted.

If the application can reliably update TMR1 before the timer input goes low, no additional action is needed. Otherwise, an adjusted update can be performed following a later Timer1 increment. This can be done by monitoring TMR1L within the interrupt routine until it increments, and then updating the TMR1H:TMR1L register pair while the clock is low, or one-half of the period of the clock source. Assuming that Timer1 is being used as a Real-Time Clock, the clock source is a 32.768 kHz crystal oscillator; in this case, one half period of the clock is 15.25 µs.

The Real-Time Clock application code in Example 12-1 shows a typical ISR for Timer1, as well as the optional code required if the update cannot be done reliably within the required interval.

EXAMPLE 12-1: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```
RTCinit
            MOVLW   80h             ; Preload TMR1 register pair
            MOVWF   TMR1H           ; for 1 second overflow
            CLRF    TMR1L
            MOVLW   b'00001111'     ; Configure for external clock,
            MOVWF   T1CON           ; Asynchronous operation, external oscillator
            CLRF    secs            ; Initialize timekeeping registers
            CLRF    mins            ;
            MOVLW   .12
            MOVWF   hours
            BSF     PIE1, TMR1IE    ; Enable Timer1 interrupt
            RETURN
RTCisr
                                    ; Insert the next 4 lines of code when TMR1
                                    ; can not be reliably updated before clock pulse goes low
            BTFSC   TMR1L,0         ; wait for TMR1L to become clear
            BRA     $-2             ; (may already be clear)
            BTFSS   TMR1L,0         ; wait for TMR1L to become set
            BRA     $-2             ; TMR1 has just incremented
                                    ; If TMR1 update can be completed before clock pulse goes low
                                    ; Start ISR here
            BSF     TMR1H, 7        ; Preload for 1 sec overflow
            BCF     PIR1, TMR1IF    ; Clear interrupt flag
            INCF    secs, F         ; Increment seconds
            MOVLW   .59             ; 60 seconds elapsed?
            CPFSGT  secs
            RETURN                  ; No, done
            CLRF    secs            ; Clear seconds
            INCF    mins, F         ; Increment minutes
            MOVLW   .59             ; 60 minutes elapsed?
            CPFSGT  mins
            RETURN                  ; No, done
            CLRF    mins            ; clear minutes
            INCF    hours, F        ; Increment hours
            MOVLW   .23             ; 24 hours elapsed?
            CPFSGT  hours
            RETURN                  ; No, done
            CLRF    hours           ; Reset hours
            RETURN                  ; Done
```

REVISION HISTORY

Rev A Document (2/2005)
Original version of this document. Includes Data Sheet Clarification issues 1 (Section 6.0 Flash Program Memory), 2 (Figure 6.2: Table Write Operation), 3 (Section 6.2.4 Table Pointer Boundaries), 4 (Figure 6-3: Table Pointer Boundaries Based on Operation), 5 (Section 6.5 Writing to Flash Program Memory), 6 (Section 6.5.1 Flash Program Memory Write Sequence), 7 (Example 6-3: Writing to Flash Program Memory) and 8 (Configuration Word Register 2L).

Rev B Document (7/2005)
Added Data Sheet Clarification issue 9 (Device ID Bit Values).

Rev C Document (6/2006)
Added issue 10 (Timer1).

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Habour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-3910
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

06/08/06