HIGHLIGHTS

This section of the manual contains the following major topics:

34.1 Introduction ............................................................................................................... 34-2
34.2 Control Registers ......................................................................................................... 34-2
34.3 Data EEPROM Operation .......................................................................................... 34-6
34.4 Data EEPROM Interrupts ........................................................................................... 34-13
34.5 Data EEPROM Topics ............................................................................................... 34-14
34.6 Register Maps ............................................................................................................. 34-16
34.7 Related Application Notes ......................................................................................... 34-17
34.8 Revision History .......................................................................................................... 34-18
34.1 INTRODUCTION

This family reference manual section describes the data EEPROM module in the dsPIC33E/PIC24E family of devices.

Data EEPROM word storage is accessed via a register-based, non-memory mapped interface. The amount of word storage may vary by device. Refer to the “Data EEPROM” chapter in the specific device data sheet for more information.

34.2 CONTROL REGISTERS

Data EEPROM read and write operations are controlled using the following EEPROM control/status registers:

- **EECON: Data EEPROM Control Register**
  The EECON register is the primary register for all data EEPROM access. This register provides both command and status functionality.

- **EEKEY: Data EEPROM Key Register**
  The EEKEY register is monitored for specific write values. It is not a physical register and will return zeros when read.

- **EEADDR: Data EEPROM Address Register**
  The address of each data EEPROM command is selected by writing to the EEADDR register.

- **EEDATA: Data EEPROM DATA Register**
  The EEDATA register will contain the data being read from the data EEPROM after a read cycle. Prior to a data EEPROM write cycle, data is written to the EEDATA register.
Section 34. Data EEPROM

Register 34-1: EECON: Data EEPROM Control Register

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R-0, HS, HC</th>
<th>R/W-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
<th>U-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEON</td>
<td>EERDY</td>
<td>EEPIDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bit 15  EEON: Data EEPROM Module Power Down Control bit
1 = Data EEPROM power applied and enabled
0 = Disables module and removes power from the data EEPROM

bit 14  EERDY: Data EEPROM Ready status bit
1 = Data EEPROM Flash array is ready for access
0 = Data EEPROM Flash array is not ready for access
EERDY is cleared by hardware whenever a POR or BOR event occurs. It is set by hardware when EEON = 1 and the power-up timer has expired.

bit 13  EEPIDL: Data EEPROM Power Down in Idle Enable bit
1 = Remove power from the data EEPROM when device enters IDLE mode
0 = Keep data EEPROM powered in standby mode when device enters IDLE mode

bit 12-8 Unimplemented: Read as ‘0’

bit 7  EERW: Read/Write Control bit
If EEWREN = 0:
1 = No Operation or the read sequence has completed. If EEWREN = 0 and EERW is cleared by software, an ongoing read cycle will be aborted. Data returned may not be valid.

If EEWREN = 1:
1 = Starts memory word read command.
0 = No Operation or the write sequence has completed.

EEWREN: Data EEPROM Write Enable bit
1 = Enables write or erase operations
0 = Inhibits write or erase operations. Enables read operations.

bit 5-4 EEERR<1:0>: Data EEPROM Sequence Error Status bits
11 = BOR event has occurred
10 = Attempted execution of a read or write operation with an invalid sequence
01 = Word Program or Forced Word Erase verify error has occurred
00 = No error condition. A program, erase or read sequence is:
   - Underway (EEERR<1:0> = 00 and EERW = 1)
   - Is completed or has yet to occur (EEERR<1:0> = 00 and EERW = 0)

The EEERR<1:0> bits can be cleared by software, or as the result of the successful execution of a subsequent program or erase command, or when EEON = 0. The EEERR<1:0> bits may also be written by software (when EERW = 0) to support context restoration.
Register 34-1: EECON: Data EEPROM Control Register (Continued)

bit 3  
**EEIPE**: Imminent Page Erase Status bit  
1 = Indicates that the next word write to this EEPROM address will take more time  
0 = Indicates that the next word write to this EEPROM address will not take extra time.  
The EEIPE bit is set by hardware after a word write cycle, and can be cleared by software, or as the result of a write to the EEADDR register. Refer to 34.3.3.1 “Word Write Cycle Timing Variations” for more information.

bit 2  
**Unimplemented**: Read as '0'

bit 1-0  
**EECMD<1:0>**: Data EEPROM command Select bits  
11 = Reserved  
10 = Forced word erase (command not recommend for regular use)  
01 = Program word when EERW is set  
00 = Read word  
Address-based, word command. The EECMD<1:0> bits cannot be modified while EERW = 1.
### Register 34-2: EEEKY: Data EEPROM Key Register

<table>
<thead>
<tr>
<th></th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEKEY&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
<th>W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEKEY&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
</table>

**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**  
EEKEY<15:0>: Data EEPROM Key bits

The EEEKY bits are monitored for specific write values. This is not a physical register.

### Register 34-3: EEADDR: Data EEPROM Address Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEADDR&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEADDR&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
</table>

**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**  
EEADDR<15:0>: Data EEPROM Address bits

### Register 34-4: EEDATA: Data EEPROM DATA Register

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDATA&lt;15:8&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 15</th>
<th>bit 8</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDATA&lt;7:0&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 0</th>
</tr>
</thead>
</table>

**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**  
EEDATA<15:0>: Data EEPROM Data bits
34.3 DATA EEPROM OPERATION

On device power-up, the EEON bit (EECON<15>) will be clear. Power is disconnected from the data EEPROM and the data EEPROM is in its lowest power mode.

When the EEON bit is set, power is applied to the data EEPROM. After a power-up timer delay has expired, the EERDY bit (EECON<14>) is set by hardware, and the data EEPROM is ready to be accessed.

Once ready for access (EERDY = 1), all read and write operations are may be executed using the data EEPROM registers. In all cases, the data EEPROM is ready to read/write data only when EERDY = 1 and there is no ongoing command (EERW bit (EECON<7>) = 0).

Access to the data EEPROM may be disabled by its associated Peripheral Module Disable bit (PMD) or by clearing the EEON bit.

34.3.1 Data EEPROM Initialization

34.3.1.1 DATA EEPROM INITIALIZATION AFTER A POR EVENT

After a POR event, the EEON bit is clear, and the data EEPROM is powered down. To enable the data EEPROM, set the EEON bit (EEON = 1) and wait for the EERDY bit to be set by hardware.

34.3.1.2 DATA EEPROM INITIALIZATION AFTER A BOR EVENT

After a BOR event, the data EEPROM remains powered up (EEON = 1) if it was powered up prior to the BOR event. The EEERR<1:0> bits (EECON<5:4>) will contain 0x3, which indicates a BOR event has occurred. The EERDY bit will be clear. Wait for the EERDY bit to be set by hardware before accessing the data EEPROM.

34.3.1.3 DATA EEPROM INITIALIZATION AFTER NON-POR, NON-BOR EVENTS

After a non-POR/non-BOR event, the data EEPROM will remain powered-up (EEON = 1) if it was powered-up prior to the event. EERDY will remain set if it was previously set. For non-POR/non-BOR events, the EEPROM will complete its last command. User software need only wait for the completion of any ongoing command before proceeding.
Example 34-1: Data EEPROM Initialization Code Example

```c
#define BOR_EVENT_ERR 3

void data_EEPROM_init(void)
{
    if (EECONbits.EEON==1)  // Only a POR clears EEON. If EEON is set, something other than a POR event has occurred.
    {
        //Wait till EEPROM is ready
        while (EECONbits.EERDY==0);

        if (EECONbits.EEERR==BOR_EVENT_ERR)  //If a BOR event causes the reset
            // Reset the error
            EECONbits.EEERR=0;

            // If a BOR event occurred, the last command may not have completed.
            // Before continuing, the application software may want to verify the last write/erase location
        }
        else                                // If the EEON was set after a non-POR/non-BOR reset, the last command will be completed, and the data EEPROM is ok.
        {
            // If an EEPROM operation is currently underway, wait for it to complete before continuing.
            while (EECONbits.EERW==1);

            if (EECONbits.EEERR>0)  //If we had an error, do something about it
                // and then clear the error
                EECONbits.EEERR = 0;
        }
    }
    else                                //Basic EEPROM enable and power-up
        EECONbits.EEON=1; //Turn on the EEPROM

        //Wait until EEPROM is ready (~125us)
        while (EECONbits.EERDY==0);
}
```
34.3.2 Data EEPROM Read Command

Prior to reading the data EEPROM, the data EEPROM must be enabled and the EERDY status bit must be set by hardware. There can be no ongoing command (EERW = 0) when a new command is issued to the data EEPROM. To execute a data EEPROM read command:

1. Load the data EEPROM address to be read (i.e., EEADDR = 0x100)
2. Load the read command into EECMD<1:0> (i.e., EECONbits.EECMD = 0x0130)
3. Clear the EEWREN bit to enable for read access (i.e., EECONbits.EEWREN = 0)
4. Set the EERW to begin the read (i.e., EECONbits.EERW = 1)
5. Wait till the read cycle is complete and hardware clears EERW (i.e., while (EECONbits.EERW = 1))
6. Read the data from the EEDATA register (i.e., variable = EEDATA)

Example 34-2: Data EEPROM Read Command Code Example

```c
#define EEPROM_NOT_READY 8

/*
 * Function returns EEERR<1:0> bits or EEPROM_NOT_READY.
 * Requested data is stored in ee_data
 * Returns zero = no error occurred during the read cycle
 * - non-zero = error occurred during the read cycle
 * - or data EEPROM is not ready
 */

int data_EEPROM_read(unsigned int addr, int *ee_data)
{
    if (EECONbits.EERDY==1) // If data EEPROM to be ready
    {
        if (EECONbits.EERW==0) //If no operation underway
        {
            //Execute Read Command
            EEADDR = ee_addr;
            EECONbits.EECMD = 0; // load EECMD<1:0> with
                // data EEPROM read command
            EECONbits.EEWREN = 0; // Access for read
            EECONbits.EERW = 1; // Start the operation

            //If FOSCSEL.DFMSTALL is set, in the configuration bits,
            // this while() loop is unnecessary
            while (EECONbits.EERW==1); // Wait till read complete
            *ee_data = EEDATA; //Read the data
            return EECONbits.EEERR;
        }
    }
    return EEPROM_NOT_READY;
}
```

Software can clear the EERW bit while it is set for a read command. This will abort a read command.

- Data in the EEDATA register may be invalid after the read command is aborted
- There will be no error reported as a result of aborting a read command
- No read complete interrupt will occur as a result of aborting a read command

While EERW is set, writes to the EECON and EEADDR registers (other than EERW) will be ignored.

The only error that can be returned by a read command is "invalid command" (EEERR<1:0> = ‘b10).
34.3.3 Data EEPROM Write Command

Prior to writing to the data EEPROM, the data EEPROM must be enabled and the EERDY status bit must be set by hardware. There can be no ongoing command (EERW = 0) when a new command is issued to the data EEPROM. To execute a data EEPROM write command:

1. Load the data EEPROM address to be write (i.e., EEADDR = 0x100).
2. Load the program command into EECMD<1:0> (i.e., EECONbits.EECMD = 0x01).
3. Set the EEWREN bit to enable for write access (i.e., EECONbits.EEWREN = 1).
4. Load the data to be written into the EEDATA register (i.e., EEDATA = 0x1248).
5. Execute the data EEPROM unlock sequence (see Example 34-3).
6. Set the EERW to begin the word write (i.e., EECONbits.EERW = 1).
7. Wait until the read cycle is complete and hardware clears EERW (i.e., while (EECONbits.EERW = 1)).

Example 34-3: Data EEPROM Write Code Example

```c
#define EEPROM_NOT_READY 8

/*
 * Function returns EEERR<1:0> bits
 * Returns zero = no error occurred during the write cycle
 * non-zero = error occurred during the write cycle
 * or data EEPROM is not ready
 */
int data_EEPROM_write(unsigned int addr, int ee_data)
{
  if (EECONbits.EERDY==1) // If data EEPROM to be ready
  { // Execute Write Command
    if (EECONbits.EERW==0) // If no operation underway
    { //load EECMD<1:0> with write command
      EEADDR = ee_addr;
      EECONbits.EECMD = 1;
      EECONbits.EEWREN = 1;
      EEDATA = ee_data;
      asm("mov.w #0xED87, w0"); // Load key
      asm("mov.w #0x1248, w1");
      asm("disi 3"); // Disable interrupts before EEKEY write
      asm("mov.w w0, EEKEY"); // Write unlock sequence
      asm("mov.w w1, EEKEY"); // Start the write cycle
      asm("bset #7, EECON"); // EECONbits.EEWR = 1;
      while (EECONbits.EERW==1); // Wait for write cycle to complete
      EEERR = EECONbits.EEWR;
    }
  }
  return EEPROM_NOT_READY;
}
```

Unlike the read command, the write command cannot be aborted by software. During a write cycle, any software attempt to write to the EECON will be ignored.

**Note:** It is the user's responsibility to ensure that no attempt is made to concurrently write/erase user program Flash (associated with the NVM registers) and the data EEPROM. Failure to observe this limitation may result in a write failure of one or both target.
34.3.3.1 WORD WRITE CYCLE TIMING VARIATIONS

When data is written to the data EEPROM while executing time-critical applications, the user should consider data EEPROM write command timing variations.

The first 16 writes to a single EEPROM address take approximately 20 µs each. The 17th write cycle, and every 16th write cycles thereafter, will require a longer write cycle, taking approximately 20 ms.

The entire 20 µs or 20 ms period is considered a write cycle period. During this period, as with all write cycle periods, the EEWR bit remains high and no new data EEPROM commands may be attempted until the EEWR bit is cleared by hardware. During this period, the CPU may continue executing code which does not access the data EEPROM.

34.3.3.2 MANAGING THE LONG WRITE CYCLE

The EEPROM logic requires a longer duration to erase the memory cell before writing the new data on the 17th write cycle and every 16th write cycle thereafter. This period is significantly longer than the majority of write cycles. We will refer to these as a “long write cycle.”

Data EEPROM logic provides assistance in managing when this long write cycle occurs. The Imminent Page Erase Status bit, EEIPE (EECON)<3>, will be set following a write cycle when the next write to the same EEPROM address will require a long write cycle.

The user can choose to ignore the EEIPE bit, and the 20 ms period will occur on the next write cycle to that specific EEPROM address.

For time-critical applications, using the data EEPROM interrupts with corresponding Interrupt Service Routines (ISRs) may have some advantages over waiting for completion of a write command in a loop.

Alternatively, user software can examine the EEIPE bit after each word write to the data EEPROM and choose to incur this 20 ms period at a time more convenient for the user application. By rewriting the same data (or new data) to the same address, the “long write cycle” occurs when the user software chooses rather than at a random point in the future.

In the example, the function writes data to the EEPROM. If the EEIPE bit is set after the first write, a second write initiates the long write cycle.

Example 34-4: Data EEPROM Recommended Procedure to Hasten a Long Write Cycle

```c
/*
 * Function returns EEERR bits
 * Called to write data.
 * The first write cycle is always a short write cycle.
 * If EEIPE is set after the first write,
 * a second write is initiated to force the long write cycle.
 * Returns zero = no error occurred during the write cycle
 * non-zero = error occurred during the write cycle
 * or data EEPROM is not ready
 */

int data_EEPROM_write_managed_long_cycle(unsigned int ee_addr,
                                         int ee_data)
{
    int ret_err;

    ret_err = data_EEPROM_write(ee_addr, ee_data); // short write cycle

    if ((EECONbits.EEIPE == 1) && (ret_err == 0))
    {
        // Rewrite the data to the same location if EEIPE==1
        // This forces the long write cycle.
        // It will take ~20 ms to complete
        ret_err = data_EEPROM_write(ee_addr, ee_data);
    }

    return ret_err;
}
```
34.3.4 Data EEPROM Forced Word Erase

Under normal conditions, there is no need to attempt a Forced Word Erase. The data EEPROM has internal logic which automatically manages all read/erase/write command sequences. Software execution of the Forced Word Erase command should be used in response to write verification errors (EEERR<1:0> = 'b01 after a write).

In the event a verification error occurs during a write to the data EEPROM, the user can attempt a Forced Word Erase command.

The Forced Word Erase command is similar to the write command except EECMD<1:0> (EECON<1:0>) = 'b10 and no data is written to EEDATA.

Example 34-5: Data EEPROM Forced Word Erase Code Example

```c
#define EEPROM_NOT_READY 8

/*
 * Function returns EEERR<1:0> bits
 * Returns zero = no error occurred, word erased properly
 * non-zero = error occurred, word may have fixed bits
 * or data EEPROM is not ready
 */
int data_EEPROM_forced_word_erase(unsigned int addr)
{
    if (EECONbits.EERDY==1) // If data EEPROM to be ready
    {
        if (EECONbits.EERW==0) //If no operation underway
        {
            //Execute Erase Command
            EADDR = ee_addr;
            EECONbits.EECMD = 2; //load EECMD<1:0> with erase command
            EECONbits.EEWREN = 1; //Access for write or erase
            asm("mov.w #0xED87, w0"); //Load key
            asm("mov.w #0x1248, w1");
            asm("disi 3"); //Disable interrupts before EEKEY write
            asm("mov.w w0, EEKEY"); //Write Key to EEKEY
            asm("mov.w w1, EEKEY");
            //Start the erase cycle
            asm("bset #7, EECON"); //EECONbits.EEWR = 1;
            while (EECONbits.EERW==1);
            //Wait for erase cycle to complete
            // Return EEERR<1:0> bits. Zero if no error
            return EECONbits.EEERR;
        }
    }
    return EEPROM_NOT_READY;
}
```

Unlike the read command, the forced word erase command cannot be aborted by software. During an erase cycle, any software attempt to write to the EECON will be ignored.

**Note:** It is the user’s responsibility to ensure that no attempt is made to concurrently write/erase user program Flash (associated with the NVM registers) and the data EEPROM. Failure to observe this limitation may result in a write failure of one or both target.
34.3.5 Data EEPROM Error Handling

If an error occurs during operation of the data EEPROM, the data EEPROM internal logic can be reinitialized by clearing the EEERR<1:0> bits.

34.3.5.1 BOR ERROR (EEERR<1:0> = ’b11)

In the event of a BOR event, any active command is aborted. If a write or forced word erase command was ongoing, the user should verify the data at the last write/erase address.

An error interrupt is not initiated. EERDY is cleared.

Wait for EERDY to be set by hardware then clear the error by clearing the EEERR<1:0> bits.

Refer to 34.3.1.2 “Data EEPROM Initialization after a BOR event” for details.

34.3.5.2 INVALID COMMAND (EEERR<1:0> = ’b10)

In the event the user tries to initiate a command improperly or set EECMD<1:0> with a reserved command, the data EEPROM will return the ‘Invalid Command’ error.

Examples of improperly initiated commands include:
- Initiating a data EEPROM read command (EEWREN = 0 and EERW = 1) when the command is not a read command (EECMD<1:0> ≠ ’b00)
- Initiating a data EEPROM read command (EECMD<1:0> = ’b00 and EERW = 1) when write is enabled (EEWREN = 1)
- Initiating a data EEPROM write command (EWREN = 1 and EERW = 1) when the command is not a write command (EECMD<1:0> ≠ ’b01)
- Initiating a data EEPROM write command (EECMD<1:0> = ’b01 and EERW = 1) when write is disabled (EEWREN = 0)

Clear the error by clearing the EEERR<1:0> bits.

34.3.5.3 VERIFY ERROR (EEERR<1:0> = ’b01)

When a data EEPROM write or erase command is executed, internal EEPROM logic will verify the data written to the data EEPROM or that the location has been erased. If there is an error in writing or erasing the data, this verification error will result.

The verification error is not expected until a given data EEPROM address has exceeded its anticipated maximum write/erase cycles.

In the event of a verification error, software can attempt to recover the data EEPROM word storage location by executing the forced word erase command at the address where the error occurred. Alternatively, software may select a different data EEPROM address to store the data.
**34.4 DATA EEPROM INTERRUPTS**

See the device family reference manual section on Interrupts for how to enable/disable and set priority for all interrupts.

There are three interrupts associated with the data EEPROM:

- The data EEPROM Error Interrupt (EEERRIF)
- The data EEPROM Read Complete Interrupt (EERDIF)
- The data EEPROM Write Complete Interrupt (EEWRIF)

### Table 34-1: EEPROM Interrupt Conditions and Actions

<table>
<thead>
<tr>
<th>Description</th>
<th>Condition</th>
<th>EEPROM Interrupt</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort of read command</td>
<td>While EEWREN = 0, EERW = 1, software clears EERW.</td>
<td>None</td>
<td>Next data EEPROM command.</td>
</tr>
<tr>
<td>Read command complete</td>
<td>Software starts a read command. When EERW is cleared by hardware, the read cycle is complete.</td>
<td>EERDIF</td>
<td>Read data from EEDATA.</td>
</tr>
<tr>
<td>Write command complete</td>
<td>Starts a write command. When EERW is cleared by hardware, the write cycle is complete.</td>
<td>EEWRIF</td>
<td>Next data EEPROM command.</td>
</tr>
<tr>
<td>Write command with error</td>
<td>EEERR&lt;1:0&gt; = 01</td>
<td>EEERRIF</td>
<td>Refer to 34.3.5.3 “Verify Error (EEERR&lt;1:0&gt; = ‘b01’).”</td>
</tr>
<tr>
<td>Data EEPROM command error occurs</td>
<td>EEERR&lt;1:0&gt; = 10</td>
<td>EEERRIF</td>
<td>Clear EEERR&lt;1:0&gt; (EEERR&lt;1:0&gt; = 00) to reset data EEPROM logic. Verify last write,* if error caused by write.</td>
</tr>
<tr>
<td>Data EEPROM BOR Event</td>
<td>EEERR&lt;1:0&gt; = 11</td>
<td>None</td>
<td>Check for BOR event on initialization of data EEPROM. If last command was a write, verify last written data.</td>
</tr>
<tr>
<td>POR event</td>
<td>EEON = 0</td>
<td>None</td>
<td>Initialize data EEPROM.</td>
</tr>
<tr>
<td>BOR event</td>
<td>EEON = 1, EERDY = 0, EECMD&lt;1:0&gt; = 11</td>
<td>None</td>
<td>Wait for EERDY = 1 before accessing data EEPROM.</td>
</tr>
</tbody>
</table>

#### 34.4.1 Data EEPROM Error Interrupt (EEERRIF)

If the data EEPROM error interrupt is enabled, it should be of higher priority than all other interrupts. This will ensure that data EEPROM errors are serviced prior to any other ISRs which may access the data EEPROM.

Errors that occur while accessing the data EEPROM within an ISR can be quickly addressed.

#### 34.4.2 Data EEPROM Read Complete (EERDIF) and Write Complete (EEWRIF)

If the read complete interrupt and/or write complete interrupt is enabled, it should be of higher priority than that of any other interrupt except the data EEPROM Error Interrupt. This will ensure interrupts that occur when the background command completes will be serviced prior to using the data EEPROM again within another ISR.
### 34.5 DATA EEPROM TOPICS

#### 34.5.1 Data EEPROM Unlock Sequence – Necessary Before a Write or Forced Word Erase Commands

To protect the data stored in the data EEPROM, each time a write or forced word erase command is issued, it must include the following unlock sequence prior to setting EERW.

With EEWREN (EECON<6>) set, EERW cannot be set until the unlock sequence has been executed.

The unlock sequence is not necessary for a read command. Interrupts must be disabled when writing the unlock sequence to the EEKEY register.

**Example 34-6: Data EEPROM Unlock-for-Write Code Example**

```
; The data EEPROM Unlock sequence must be executed prior to attempting a 
; write or erase cycle
mov #0xED87, wo
mov #0x1248, w1
disi 2 ;prevent interrupts during write to EEKEY
mov wo, EEKEY
mov w1, EEKEY
```

#### 34.5.2 Configuration Bits

There are device Configuration Bits to be aware of when setting up the data EEPROM.

#### 34.5.2.1 FPOR CONFIGURATION REGISTER SETTINGs (FPOR<2:0>)

Data Flash Memory Cycle Count is the number of Tcy cycles used in a data flash memory access. Refer to the “Data EEPROM” chapter in the specific device data sheet for more information.

Select the value based on the application operating speed (Fcy). For the dsPIC33E/PIC24E device family, the setting should be 2, 1, or 0 dependent upon the device speed.

**Table 34-2: Data EEPROM Read Access Configuration**

<table>
<thead>
<tr>
<th>Max System Clock (MHz)</th>
<th>DFMCY&lt;2:0&gt; Setting</th>
<th>Flash Access (Cycles/Read)</th>
<th>Read Access (Cycles)</th>
<th>Read Access (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;200</td>
<td>7</td>
<td>8</td>
<td>48</td>
<td>240</td>
</tr>
<tr>
<td>&lt;175</td>
<td>6</td>
<td>7</td>
<td>43</td>
<td>246</td>
</tr>
<tr>
<td>&lt;150</td>
<td>5</td>
<td>6</td>
<td>38</td>
<td>253</td>
</tr>
<tr>
<td>&lt;125</td>
<td>4</td>
<td>5</td>
<td>33</td>
<td>264</td>
</tr>
<tr>
<td>&lt;100</td>
<td>3</td>
<td>4</td>
<td>28</td>
<td>280</td>
</tr>
<tr>
<td>&lt;75</td>
<td>2</td>
<td>3</td>
<td>23</td>
<td>306</td>
</tr>
<tr>
<td>&lt;50</td>
<td>1</td>
<td>2</td>
<td>18</td>
<td>360</td>
</tr>
<tr>
<td>&lt;25</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>520</td>
</tr>
</tbody>
</table>

Setting the value to something higher than recommended will slow device reads.

#### 34.5.2.2 FOSCSEL CONFIGURATION REGISTER SETTING (FOSCSEL<5>)

The user may elect to stall the device CPU until the completion of a data EEPROM read cycle. Stalling the CPU during an EEPROM read will prevent execution of application software while a read command is being executed. Refer to the “Data EEPROM” chapter in the specific device data sheet for more information.
34.5.3 Peripheral Module Disable (PMD)

When the Peripheral Module Disable (PMD) bit associated with the data EEPROM is set:
1. The data EEPROM will be immediately disabled.
2. Any ongoing command is immediately aborted.
3. Module clocks are disabled; power is disconnected from the module.

It is recommended that the user disable the data EEPROM write interrupt (if enabled) and wait till EERW is clear before setting the module’s PMD bit. This will avoid potentially incomplete write cycles.

Example 34-7: Data EEPROM Recommended Procedure for Setting the PMD bit

```c
void data_EEPROM_set_PMD( void )
{
    _EEWRIE = 0; //Disable data EEPROM write interrupt, if it
                  //was enabled
    while (EECONbits.EERW); //wait till last command is complete
    PMD4bits.EEMD = 1; //then disable the module
}
```
### 34.6 REGISTER MAPS

A summary of the registers associated with the dsPIC33E/PIC24E family Data EEPROM module is provided in Table 34-3.

**Table 34-3: Data EEPROM Register Map**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Bit 15</th>
<th>Bit 14</th>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>All Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>EECON</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EERW, EEWREN, EEERR&lt;1:0&gt;, EEIPE, —, ECCMD&lt;1:0&gt;</td>
</tr>
<tr>
<td>EEKEY</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EEKEY&lt;15:0&gt;</td>
</tr>
<tr>
<td>EEADDR</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EEADDR&lt;15:0&gt;</td>
</tr>
<tr>
<td>EEDATA</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EEDATA&lt;15:0&gt;</td>
</tr>
</tbody>
</table>

**Legend:**
- — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

**Note 1:** All bits are cleared by a POR event. After a BOR or other reset, they may not be clear. Refer to 34.3.1.2 “Data EEPROM Initialization after a BOR event” after a BOR event and 34.3.1.3 “Data EEPROM Initialization after non-POR, non-BOR events” after non-BOR events.

**Note 2:** All bits are cleared by a POR event. After non-BOR/non-POR events, they may not be clear. Refer to 34.3.1.3 “Data EEPROM Initialization after non-POR, non-BOR events” after non-BOR events.
### 34.7 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC33E/PIC24E product family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Data EEPROM module are:

<table>
<thead>
<tr>
<th>Title</th>
<th>Application Note #</th>
</tr>
</thead>
<tbody>
<tr>
<td>No related application notes at this time.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Note:** Please visit the Microchip web site ([www.microchip.com](http://www.microchip.com)) for additional Application Notes and code examples for the dsPIC33E/PIC24E family of devices.
34.8 REVISION HISTORY

Revision A (August 2012)

This is the initial released version of the document.
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.
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://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

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

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

Cleveland
Independence, OH
Tel: 216-447-0946
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7243
Fax: 972-818-2924

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

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

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

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

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

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

Australia - Sydney
Tel: 61-2-9888-6733
Fax: 61-2-9888-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

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

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

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

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC
India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

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

Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

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

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

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

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

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

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8890

Taiwan - Hsin Chu
Tel: 886-3-5770-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

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: 43-7242-2244-39
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

11/29/11