Section 41. Prefetch Module for Devices with L1 CPU Cache

HIGHLIGHTS

This section of the manual contains the following major topics:

41.1 Introduction ............................................................................................................... 41-2
41.2 Prefetch Module Overview ....................................................................................... 41-3
41.3 Control Registers ...................................................................................................... 41-5
41.4 Prefetch Module Operation ....................................................................................... 41-8
41.5 Prefetch Module Configurations ............................................................................... 41-8
41.6 Prefetch Module Predictive Prefetch Behavior ......................................................... 41-8
41.7 Coherency Support .................................................................................................... 41-9
41.8 Effects of Reset ........................................................................................................ 41-9
41.9 Error Conditions ....................................................................................................... 41-10
41.10 Operation in Power-Saving Modes ......................................................................... 41-11
41.11 Related Application Notes ..................................................................................... 41-12
41.12 Revision History ..................................................................................................... 41-13
41.1 INTRODUCTION

This section describes the features and operation of the Prefetch module for PIC32 devices with L1 CPU Cache. Prefetch module features increase system performance for most applications.

41.1.1 Prefetch Module Features

The Prefetch module includes the following features:

- 4 x 16 byte fully associative lines
- One line for CPU instructions
- One line for CPU data
- Two lines for peripheral data
- 16 byte parallel memory fetch
- Configurable predictive prefetch
- Error detection and correction
41.2 PREFETCH MODULE OVERVIEW

The Prefetch module is a performance enhancing module included in PIC32 devices with L1 CPU caches. When running at high-clock rates, Wait states must be inserted into Program Flash Memory (PFM) read transactions to meet the access time of the PFM. Wait states can be hidden to the core by prefetching and storing instructions in a temporary holding area that the CPU can access quickly. Although the data path to the CPU is 32 bits wide, the data path to the PFM is 128 bits wide. This wide data path provides the same bandwidth to the CPU as a 32-bit path running at four times the frequency.

The Prefetch module holds a subset of PFM in temporary holding spaces known as lines. Each line contains a tag and data field. Normally, the lines hold a copy of what is currently in memory to make instructions or data available to the CPU without Wait states.

Data located in the PFM may be requested by the CPU or by a peripheral. If the requested data is not currently stored in a Prefetch module line, a read is performed to the PFM at the correct address, and the data is supplied to the Prefetch module and to the CPU or peripheral. If the requested data is stored in the Prefetch module and is valid, the data is supplied to the CPU or peripheral without Wait states.

Figure 41-1 shows a block diagram of the Prefetch module. Logically, the Prefetch module fits between the System Bus module and the PFM.
41.2.1 Prefetch Module Line Organization

The Prefetch module consists of two arrays, data and tag, each of which hold four lines. A data array consists of program instructions, program data, or peripheral data. Address matches are based on the physical address, not the virtual address.

Each line in the tag array contains the following information:
- Tag – Physical address of the data held in the data line
- Valid bit
- Type – CPU instruction, CPU data, or peripheral data
- Double-bit Error Detected (DED) bit

Each line in the data array contains 16 bytes of data. Depending on the line, the data can be CPU instructions, CPU data, or peripheral data.

Figure 41-2 and Figure 41-3 illustrate the organization of a line.

Figure 41-2: Tag Line

<table>
<thead>
<tr>
<th>31</th>
<th>4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTAG&lt;31:4&gt;</td>
<td>VALID</td>
</tr>
</tbody>
</table>

Figure 41-3: Data Line

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD 3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD 2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD 0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 41-4: Prefetch Module Arrays

<table>
<thead>
<tr>
<th>Line #</th>
<th>Tag Array(1)</th>
<th>Data Array(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TAG Valid 0 CPU-I</td>
<td>WORD 3 WORD 2 WORD 1 WORD 0</td>
</tr>
<tr>
<td>1</td>
<td>TAG Valid 0 CPU-D</td>
<td>WORD 3 WORD 2 WORD 1 WORD 0</td>
</tr>
<tr>
<td>2</td>
<td>TAG Valid 0 P-Data</td>
<td>WORD 3 WORD 2 WORD 1 WORD 0</td>
</tr>
<tr>
<td>3</td>
<td>TAG Valid 0 P-Data</td>
<td>WORD 3 WORD 2 WORD 1 WORD 0</td>
</tr>
</tbody>
</table>

Note 1: These arrays cannot be read or written by the user application.
### 41.3 CONTROL REGISTERS

The Prefetch module for PIC32 devices with L1 CPU cache contains the following Special Function Registers (SFRs):

- **PRECON: Prefetch Module Control Register**
  This register manages configuration of the Prefetch module and controls Wait states.

- **PRESTAT: Prefetch Module Status Register**
  This register contains status information for error correction and detection.

Table 41-1 provides a brief summary of the related Prefetch module registers. Corresponding registers appear after the summary, followed by a detailed description of each bit.

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit 31/15</th>
<th>Bit 30/14</th>
<th>Bit 29/13</th>
<th>Bit 28/12</th>
<th>Bit 27/21</th>
<th>Bit 26/20</th>
<th>Bit 25/19</th>
<th>Bit 24/18</th>
<th>Bit 23/17</th>
<th>Bit 22/16</th>
<th>Bit 21/15</th>
<th>Bit 20/14</th>
<th>Bit 19/13</th>
<th>Bit 18/12</th>
<th>Bit 17/11</th>
<th>Bit 16/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRECON</td>
<td>31:16</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>
</tr>
<tr>
<td></td>
<td>15:0</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>
</tr>
<tr>
<td>PRESTAT</td>
<td>31:16</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>
</tr>
<tr>
<td></td>
<td>15:0</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>
</tr>
</tbody>
</table>

**Legend:** — = unimplemented, read as '0'.

**Note 1:** These registers have associated Clear, Set and Invert registers at offsets of 0x4, 0x8, and 0xC bytes, respectively. The Clear, Set and Invert registers have the same name with CLR, SET, or INV appended to the register name (e.g., PRECONCLR). Writing a ‘1’ to any bit position in these registers will clear, set or invert valid bits in the associated register. Reads from these registers should be ignored.
### Register 41-1: PRECON: Prefetch Module Control Register

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Bit 31/23/15/7</th>
<th>Bit 30/22/14/6</th>
<th>Bit 29/21/13/5</th>
<th>Bit 28/20/12/4</th>
<th>Bit 27/19/11/3</th>
<th>Bit 26/18/10/2</th>
<th>Bit 25/17/9/1</th>
<th>Bit 24/16/8/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>23:16</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>—</td>
</tr>
<tr>
<td>15:8</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>—</td>
</tr>
<tr>
<td>7:0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>R/W-1</td>
<td>R/W-1</td>
</tr>
</tbody>
</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 31-27**  
**Unimplemented:** Write '0'; ignore read

**bit 26**  
**PFMSECEN:** Flash SEC Interrupt Enable bit
- **1** = Generate an interrupt when the PFMSEC bit (PRESTAT<26>) is set
- **0** = Do not generate an interrupt when the PFMSEC bit is set

**bit 25-6**  
**Unimplemented:** Write '0'; ignore read

**bit 5-4**  
**PREFEN<1:0>:** Predictive Prefetch Enable bits
- **11** = Enable predictive prefetch for any address
- **10** = Enable predictive prefetch for CPU instructions and CPU data
- **01** = Enable predictive prefetch for CPU instructions only
- **00** = Disable predictive prefetch

**bit 3**  
**Unimplemented:** Write '0'; ignore read

**bit 2-0**  
**PFMWS<2:0>:** PFM Access Time Defined in Terms of SYSCLK Wait States bits
- **111** = Seven Wait states
- **110** = Six Wait states
- **101** = Five Wait states
- **100** = Four Wait states
- **011** = Three Wait states
- **010** = Two Wait states
- **001** = One Wait state
- **000** = Zero Wait state
## Section 41. Prefetch Module for Devices with L1 CPU Cache

### Register 41-2: PRESTAT: Prefetch Module Status Register

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Bit 31/23/15/7</th>
<th>Bit 30/22/14/6</th>
<th>Bit 29/21/13/5</th>
<th>Bit 28/20/12/4</th>
<th>Bit 27/19/11/3</th>
<th>Bit 26/18/10/2</th>
<th>Bit 25/17/9/1</th>
<th>Bit 24/16/8/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>HS, R/C-0</td>
<td>HS, R/W-0</td>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>23:16</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>15:8</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
</tr>
<tr>
<td>7:0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
<td>HS, HC, R/W-0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PFMSDED</td>
<td>PFMSEC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **HS** = Set by hardware
- **HC** = Cleared by hardware
- **C** = Clearable bit
- **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 31-28
**Unimplemented:** Write ‘0’; ignore read

#### bit 27
**PFMDED:** Flash Double-bit Error Detected (DED) Status bit
- This bit is set in hardware and can only be cleared (i.e., set to ‘0’) in software.
- 1 = A DED error has occurred
- 0 = A DED error has not occurred

#### bit 26
**PFMSEC:** Flash Single-bit Error Corrected (SEC) Status bit
- 1 = A SEC error occurred when PFMSECCNT<7:0> was equal to zero
- 0 = A SEC error has not occurred

#### bit 25-8
**Unimplemented:** Write ‘0’; ignore read

#### bit 7-0
**PFMSECCNT<7:0>:** Flash SEC Count bits
- Decrements by 1 its count value each time an SEC error occurs. Holds at zero. When an SEC error occurs when PFMSECCNT<7:0> is zero, the PFMSEC status bit is set. If PFMSECCEN is also set, a Prefetch module interrupt event is generated.
41.4 PREFETCH MODULE OPERATION

The Prefetch module is designed to complement an L1 CPU cache rather than replace it. A single 128-bit (16-byte) line holds instructions or data from the PFM. The Prefetch module uses the Wait states value from the PFMWS<2:0> bits (PRECON<2:0>) to determine how long it must wait for a Flash access when it reads instructions or data from the PFM. If the instructions or data already reside in a Prefetch module line, the Prefetch module returns the instruction or data in zero Wait states. For CPU instructions, if prefetch is enabled and the code is 100% linear, the Prefetch module will provide instructions back to the CPU with Wait states only on the first instruction of the Prefetch module line.

One Prefetch module line is allocated to CPU data and two lines are allocated to peripheral data. Which of these lines are enabled is determined by the PREFEN<1:0> bits (PRECON<5:4>). Although the lines are enabled by type, the type is not used for matching. Therefore, the line allocated to CPU data and filled by CPU data can be read by a CPU instruction read or a non-CPU peripheral data read. A non-CPU peripheral could be DMA or any other peripheral that has read access to the PFM.

The Prefetch module does not support preloading, address masking, or line locking.

41.5 PREFETCH MODULE CONFIGURATIONS

The PRECON register controls the general configurations available for accelerating instruction and data accesses to the Flash memory system. The Prefetch module implements the following general options:

- The PFMWS<2:0> bits (PRECON<2:0>) control the number of system clock cycles required to access the PFM
- The PREFEN<1:0> bits (PRECON<5:4>) control which types of reads are predictively prefetched
- The PFMSECEN bit (PRECON<26>) controls whether the Prefetch module generates an interrupt event on a specific count of single bit errors corrected by the Flash Error Correction Code (ECC)

41.6 PREFETCH MODULE PREDICTIVE PREFETCH BEHAVIOR

When configured for predictive prefetch, the Prefetch module predicts the next line address, fetches the data, and then stores it in the prefetch buffer. If the requested instruction or data is not in a Prefetch module line, and the read address matches the predicted address, the contents of the prefetch buffer are loaded in the Prefetch module line while simultaneously returning the critical word to the read initiator.

If enabled, the prefetch function starts predicting based on the first address read to the PFM. When the first line is placed in the Prefetch module, the module simply increments the address to the next 16-byte aligned address and starts a PFM access.

Predictive prefetches, like all PFM read accesses, are never aborted. If a new address request does not match the predicted address, a new PFM access occurs after the current access finishes. The PREFEN<1:0> bits (PRECON<5:4>) control what types of requests can start a predictive prefetch. They can be CPU instruction only, CPU instruction and data, or CPU and peripheral reads. The use of CPU and peripheral data read prefetching is beneficial for reading large data structures in the PFM. One such case would be to verify the entire flash with a Hash or CRC value using DMA. For all other use models, it is best to only allow prediction on CPU instructions.

If the selected system clock speed is sufficiently low enough to access the Flash at zero Wait states, predictive prefetch is detrimental and should be disabled.
41.7 COHERENCY SUPPORT

When a PFM programming event occurs, the Prefetch module invalidates all lines and the contents of the prefetch buffer. If a transaction is in progress, the invalidation occurs after completion. When programming or erasing a Flash page, a read of that Flash page will cause the transaction to stall until the erase or program event completes.

41.8 EFFECTS OF RESET

41.8.1 On Reset

Upon a device Reset, the following occurs:
- All lines are invalidated
- All tag bits are cleared

41.8.2 After Reset

The module operates as per the values in the PRECON register (Register 41-1).
41.9  ERROR CONDITIONS

The Prefetch module handles and reports information about two error types: ECC Double-bit Error Detected (DED) and ECC Single-bit Error Corrected (SEC). The ECC Error detection logic is enabled and disabled using the configuration bits, FECCCON<1:0> (DEVCFG0<9:8>). Refer to the “Special Features” chapter in the specific device data sheet for information on the DEVCFG0 Configuration register.

The ECC logic increases the read access delay from the PFM. Depending on the frequency of the system clock, the wait states may be different between ECC enabled and ECC disabled. Please see the specific device data sheet for flash access timing specifications for a particular device.

41.9.1  ECC Double-bit Error Detected (DED)

A read from the Flash memory that results in a PFM ECC DED causes the Prefetch module to return a bus exception error to the initiator. If that initiator is the CPU, it recognizes the bus exception error, prevents the instruction from executing, or read data from loading, and generates an exception using the bus exception error vector.

When an ECC DED error occurs, the PFMDED bit (PRESTAT<27>) is set. The exception handling code can then check this bit to determine whether the exception was caused by a PFM ECC DED event. This bit must be cleared in software by the exception handler.

41.9.2  ECC Single Error Corrected (SEC)

A PFM ECC SEC event is not a critical error and as such is reported through an interrupt. The user has the option to enable or disable this interrupt through the PFMSECEN bit (PRECON<26>). The data in the Prefetch module is correct, and no further ECC events are generated for addresses that hit the data line as long as that data is in the Prefetch module.

Each read that returns from the PFM with an ECC SEC status causes the PFMSECCNT<7:0> bits (PRESTAT<7:0>) to decrement by one. If PFMSECCNT<7:0> is zero and a PFM ECC SEC event occurs, the PFMSEC bit (PRESTAT<26>) is set and an interrupt is generated. Therefore, the PFMSECCNT<7:0> bits should be set to the number of PFM ECC SEC events desired for an interrupt minus 1. For example, to generate an interrupt after five PFM ECC SEC events, PFMSECCNT<7:0> should be set to four ('00000100'). The Prefetch module does not reload the PFMSECCNT<7:0> bits when it reaches zero. Software must write the desired count each time it services the PFMSEC interrupt.

Software can generate an ECC SEC interrupt by setting the PFMSECEN bit and then setting the PFMSEC bit. If the PFMSEC bit is already set when PFMSECEN is set, the Prefetch module will also generate an ECC SEC interrupt. The ECC SEC interrupt persists as long as the PFMSECEN and PFMSEC bits remain set.

Note: ECC errors are captured for predictive prefetch reads of the PFM. However, those errors are not reported until, and unless, that data is used by the system.

Note: CPU instructions or data prefetched from the PFM will always be loaded into the Prefetch module, even if a DED error is generated. The Prefetch module line containing the DED data will be tagged as valid until the line is replaced.
41.10 OPERATION IN POWER-SAVING MODES

41.10.1 Sleep Mode
When the device enters Sleep mode, the Prefetch module is disabled and placed into a low-power state where no clocking occurs in the module.

41.10.2 Idle Mode
When the device enters Idle mode, the Prefetch module and its clock source remain functional and the CPU stops executing code. Any outstanding prefetch completes before the Prefetch module stops its clock through automatic clock gating.

41.10.3 Debug Mode
The behavior of the Prefetch module is unaltered in Debug mode.
## 41.11 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 PIC32 device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Prefetch Module for Devices with L1 CPU Cache 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 PIC32 family of devices.
41.12  REVISION HISTORY

Revision A (August 2012)
This is the initial released version of the document.

Revision B (September 2013)
This revision includes the following updates:
• All references to BMX and Bus Matrix were updated to System Bus
• Minor updates to text and formatting were incorporated throughout 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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, nPIC, SSI, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, P ICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2012-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.


QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, 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://www.microchip.com/support](http://www.microchip.com/support)  
Web Address:  
[www.microchip.com](http://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: 216-447-0034  
Fax: 216-447-0643

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
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: 852-2866-5511  
Fax: 852-2866-5799

**China - Beijing**  
Tel: 86-10-8589-7000  
Fax: 86-10-8528-2104

**China - Chengdu**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**China - Chongqing**  
Tel: 86-23-8990-9588  
Fax: 86-23-8990-9500

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

**China - Hong Kong SAR**  
Tel: 852-2940-0100  
Fax: 852-2401-3431

**China - Nanjing**  
Tel: 86-25-8473-2187  
Fax: 86-25-8473-2189

**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: 86-24-2334-2329  
Fax: 86-24-2334-2393

**China - Shenzhen**  
Tel: 86-755-8864-2200  
Fax: 86-755-8203-1760

**China - Wuhan**  
Tel: 86-27-5980-5381  
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: 91-20-3019-1500

**Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**Japan - Tokyo**  
Tel: 81-3-6800-3770  
Fax: 81-3-6800-3771

**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: 65-6334-8850

**Taiwan - Hsin Chu**  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**  
Tel: 886-7-213-7828  
Fax: 886-7-330-9005

**Taiwan - Taipei**  
Tel: 886-2-2508-8600  
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

08/20/13