Section 47. External Bus Interface (EBI)

HIGHLIGHTS

This section of the manual contains the following major topics:

47.1 Introduction ............................................................................................................... 47-2
47.2 Control Registers ....................................................................................................... 47-4
47.3 Interfacing to Various Devices ................................................................................... 47-14
47.4 Bus Configuration ..................................................................................................... 47-16
47.5 Device Configuration .................................................................................................. 47-17
47.6 Timing Diagrams ........................................................................................................ 47-21
47.7 Effects Of Reset ......................................................................................................... 47-24
47.8 Operation in Power-Saving Modes .......................................................................... 47-24
47.9 Related Application Notes ....................................................................................... 47-25
47.10 Revision History ...................................................................................................... 47-26
47.1 INTRODUCTION

The External Bus Interface (EBI) module provides a convenient, high-speed way to interface external parallel memory devices to the PIC32 family device.

With the EBI module, it is possible to connect asynchronous SRAM and NOR Flash devices, as well as non-memory devices, such as camera sensors. The EBI module also supports Low-Cost Controllerless (LCC) Graphics devices.

The features of the EBI module depend on the particular PIC32 device and the pin count, as shown in Table 47-1.

Table 47-1: EBI Module Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>100</th>
<th>124</th>
<th>144</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous SRAM</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Asynchronous NOR Flash</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Available address lines</td>
<td>20</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>8-bit data bus support</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>16-bit data bus support</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Available Chip Selects</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Timing mode sets</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8-bit R/W from 16-bit bus</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Performance (MHz)</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Non-memory device</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Figure 47-1: EBI System Block Diagram
Figure 47-2: EBI Module Pinout and Connections to External Devices

Note 1: No EBIA address pins are available on 64-pin devices and the EBIA<23:20> address pins are only available on 144-pin devices.
47.2  CONTROL REGISTERS

The EBI module for PIC32 devices contains the following Special Function Registers (SFRs):

- **EBICSx**: External Bus Interface Chip Select Register (x = 0-3)
  This register contains the base address in physical memory for the selected external Device.

- **EBIMSKx**: External Bus Interface Address Mask Register (x = 0-3)
  This register enables selection of the timing register set, as well as the Chip Select memory type and memory size.

- **EBISMTx**: External Bus Interface Static Memory Timing Register (x = 0-2)
  This register can be used to configure the static memory timing.

- **EBIFTRPD**: External Bus Interface Flash Timing Register
  This register defines the number of clock cycles to hold the external Flash memory in reset.

- **EBISMCON**: External Bus Interface Static Memory Control Register
  This register can be used to define the static memory width for register sets 0-2, and to select Flash Reset/Power-down mode during a device Reset.

- **CFGEBIA**: External Bus Interface Address Pin Configuration Register
  This register can be used to configure the address pins for the EBI module.

- **CFGEBIC**: External Bus Interface Control Pin Configuration Register
  This register can be used to configure the control pins for the EBI module.
Table 47-2 and Table 47-3 provide a brief summary of the related EBI registers. Corresponding registers appear after the summary, followed by a detailed description of each bit.

### Table 47-2: EBI SFR Summary

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Bit Range</th>
<th>Bit 31/15</th>
<th>Bit 30/14</th>
<th>Bit 29/13</th>
<th>Bit 28/12</th>
<th>Bit 27/11</th>
<th>Bit 26/10</th>
<th>Bit 25/9</th>
<th>Bit 24/8</th>
<th>Bit 23/7</th>
<th>Bit 22/6</th>
<th>Bit 21/5</th>
<th>Bit 20/4</th>
<th>Bit 19/3</th>
<th>Bit 18/2</th>
<th>Bit 17/1</th>
<th>Bit 16/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBICSx</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>
<td>—</td>
</tr>
<tr>
<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>
<td></td>
<td></td>
</tr>
<tr>
<td>EBIMSKx</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>
<td>—</td>
</tr>
<tr>
<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>
<td></td>
<td></td>
</tr>
<tr>
<td>EBISMTx</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>
<td>—</td>
</tr>
<tr>
<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>
<td></td>
<td></td>
</tr>
<tr>
<td>EBIFTRPD</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>
<td>—</td>
</tr>
<tr>
<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>
<td></td>
<td></td>
</tr>
<tr>
<td>EBISMCON</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>
<td>—</td>
</tr>
<tr>
<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>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:** — = unimplemented, read as ‘0’.

### Table 47-3: EBI Configuration Register Summary

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Bit Range</th>
<th>Bit 31/15</th>
<th>Bit 30/14</th>
<th>Bit 29/13</th>
<th>Bit 28/12</th>
<th>Bit 27/11</th>
<th>Bit 26/10</th>
<th>Bit 25/9</th>
<th>Bit 24/8</th>
<th>Bit 23/7</th>
<th>Bit 22/6</th>
<th>Bit 21/5</th>
<th>Bit 20/4</th>
<th>Bit 19/3</th>
<th>Bit 18/2</th>
<th>Bit 17/1</th>
<th>Bit 16/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFGEBIA</td>
<td>31:16</td>
<td>EBIPEIEN</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>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>
<td></td>
<td></td>
</tr>
<tr>
<td>CFGEBIC</td>
<td>31:16</td>
<td>—</td>
<td>EBRDYNV3</td>
<td>EBI</td>
<td>EBRDYNV2</td>
<td>EBI</td>
<td>EBRDYNV1</td>
<td>EBI</td>
<td>EBRDYN3</td>
<td>EBI</td>
<td>EBRDYN2</td>
<td>EBI</td>
<td>EBRDYN1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<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>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:** — = unimplemented, read as ‘0’.
Register 47-1: EBICSx: External Bus Interface Chip Select Register (x = 0-3)

<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>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>23:16</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-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>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>
</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 31-16 **CSADDR<15:0>**: Base Address for Device bits
Address in physical memory, which will select the external device.

bit 15-0 **Unimplemented**: Read as ‘0’
## Register 47-2: EBIMSKx: External Bus Interface Address Mask Register (x = 0-3)

<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>U-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>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>7:0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
</tbody>
</table>

**MEMTYPE<2:0>:** Select Memory Type for Chip Select ‘x’ bits

- 111 = Reserved
- 011 = Reserved
- 010 = NOR-Flash
- 001 = SRAM
- 000 = Reserved

**MEMSIZE<4:0>:** Select Memory Size for Chip Select ‘x’ bits

- 11111 = Reserved
- 01010 = Reserved
- 01001 = 16 MB
- 01000 = 8 MB
- 00111 = 4 MB
- 00110 = 2 MB
- 00101 = 1 MB
- 00100 = 512 KB
- 00011 = 256 KB
- 00010 = 128 KB
- 00001 = 64 KB (smaller memories alias within this range)
- 00000 = Chip Select is not used

### 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 31-11** Unimplemented: Read as ‘0’

**bit 10-8** **REGSEL<2:0>:** Timing Register Set for Chip Select ‘x’ bits

- 111 = Reserved
- 011 = Reserved
- 010 = Use EBITMGR2
- 001 = Use EBITMGR1
- 000 = Use EBITMGR0

**bit 7-5** **MEMTYPE<2:0>:** Select Memory Type for Chip Select ‘x’ bits

- 111 = Reserved
- 011 = Reserved
- 010 = NOR-Flash
- 001 = SRAM
- 000 = Reserved

**bit 4-0** **MEMSIZE<4:0>:** Select Memory Size for Chip Select ‘x’ bits

- 11111 = Reserved
- 01010 = Reserved
- 01001 = 16 MB
- 01000 = 8 MB
- 00111 = 4 MB
- 00110 = 2 MB
- 00101 = 1 MB
- 00100 = 512 KB
- 00011 = 256 KB
- 00010 = 128 KB
- 00001 = 64 KB (smaller memories alias within this range)
- 00000 = Chip Select is not used
Register 47-3: EBISMTx: External Bus Interface Static Memory Timing Register (x = 0-2)

<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>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>23:16</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>15:8</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>7:0</td>
<td>R/W-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-1</td>
<td>R/W-0</td>
<td>R/W-0</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: Read as '0'
bit 26 RDYMODE: Data Ready Device Select bit
The device associated with register set 'x' is a data-ready device, and will use the READY pin.
1 = Ready input is used
0 = Ready input is not used
bit 25-24 PAGESIZE<1:0>: Page Size for Page Mode Device bits
11 = 32-word page
10 = 16-word page
01 = 8-word page
00 = 4-word page
bit 23 PAGEMODE: Memory Device Page Mode Support bit
1 = Device supports Page mode
0 = Device does not support Page mode
bit 22-19 TPRC<3:0>: Page Mode Read Cycle Time bits
Read cycle time is TPRC + 1 clock cycle.
bit 18-16 TBTA<2:0>: Data Bus Turnaround Time bits
Clock cycles (0-7) for static memory between read-to-write, write-to-read, and read-to-read when Chip Select changes.
bit 15-10 TWP<5:0>: Write Pulse Width bits
Write pulse width is TWP + 1 clock cycle.
bit 9-8 TWR<1:0>: Write Address/Data Hold Time bits
Number of clock cycles to hold address or data on the bus.
bit 7-6 TAS<1:0>: Write Address Setup Time bits
Clock cycles for address setup time. A value of '0' is only valid in the case of SSRAM.
bit 5-0 TRC<5:0>: Read Cycle Time bits
Read cycle time is TRC + 1 clock cycle.
### Register 47-4: EBIFTRPD: External Bus Interface Flash Timing 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>U-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>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>7:0</td>
<td>R/W-x</td>
<td>R/W-x</td>
<td>R/W-x</td>
<td>R/W-x</td>
<td>R/W-x</td>
<td>R/W-x</td>
<td>R/W-x</td>
<td>R/W-x</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

### Note 1:
Please refer to the specific device data sheet for the actual reset values for these bits.

These bits define the number of clock cycles to wait after resetting the external Flash memory before starting any read/write accesses.
Register 47-5: EBISMCON: External Bus Interface Static Memory 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>U-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>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>7:0</td>
<td>R/W-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>RW-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

**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

**Unimplemented:**

bit 31-16: **Unimplemented:** Read as ‘0’

bit 15-13: **SMDWIDTH2<2:0>:** Static Memory Width for Register Set 2 bits

| 111 | Reserved |
| 110 | Reserved |
| 101 | Reserved |
| 100 | 8 bits   |
| 011 | Reserved |
| 010 | Reserved |
| 001 | Reserved |
| 000 | 16 bits  |

bit 12-10: **SMDWIDTH1<2:0>:** Static Memory Width for Register Set 1 bits

| 111 | Reserved |
| 110 | Reserved |
| 101 | Reserved |
| 100 | 8 bits   |
| 011 | Reserved |
| 010 | Reserved |
| 001 | Reserved |
| 000 | 16 bits  |

bit 9-7: **SMDWIDTH0<2:0>:** Static Memory Width for Register Set 0 bits

| 111 | Reserved |
| 110 | Reserved |
| 101 | Reserved |
| 100 | 8 bits   |
| 011 | Reserved |
| 010 | Reserved |
| 001 | Reserved |
| 000 | 16 bits  |

bit 6-1: **Unimplemented:** Read as ‘0’

**SMRP:** Flash Reset/Power-down mode Select bit

After a Reset, the controller internally performs a power-down for Flash, and then sets this bit to ‘1’.

| 1 | Flash is taken out of Power-down mode |
| 0 | Flash is forced into Power-down mode |
## Section 47. External Bus Interface (EBI)

### Register 47-6: CFGEBIA: External Bus Interface Address Pin Configuration 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>R/W-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></td>
<td>EBIPINEN</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>23:16</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td></td>
<td>EBIA23EN</td>
<td>EBIA22EN</td>
<td>EBIA21EN</td>
<td>EBIA20EN</td>
<td>EBIA19EN</td>
<td>EBIA18EN</td>
<td>EBIA17EN</td>
<td>EBIA16EN</td>
</tr>
<tr>
<td>15:8</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td></td>
<td>EBIA15EN</td>
<td>EBIA14EN</td>
<td>EBIA13EN</td>
<td>EBIA12EN</td>
<td>EBIA11EN</td>
<td>EBIA10EN</td>
<td>EBIA9EN</td>
<td>EBIA8EN</td>
</tr>
<tr>
<td>7:0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td></td>
<td>EBIA7EN</td>
<td>EBIA6EN</td>
<td>EBIA5EN</td>
<td>EBIA4EN</td>
<td>EBIA3EN</td>
<td>EBIA2EN</td>
<td>EBIA1EN</td>
<td>EBIA0EN</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:** EBIPINEN: EBI Pin Enable bit
- 1 = EBI controls access of pins shared with PMP
- 0 = Pins shared with EBI are available for general use

**bit 30-24:** Unimplemented: Read as ‘0’

**bit 23-0:** EBIA23EN:EBIA0EN: EBI Address Pin Enable bits
- 1 = The EBIAx pin is enabled for use by EBI
- 0 = The EBIAx pin has is available for general use

**Note:** When EBIMD = 1, the bits in this register are ignored and the pins are available for general use.
### Register 47-7: CFGEBIC: External Bus Interface Control Pin Configuration Register

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Bit 31/23/15/7</th>
<th>Bit 29/21/13/5</th>
<th>Bit 27/19/11/3</th>
<th>Bit 25/17/9/1</th>
<th>Bit 24/16/8/0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td>31:24</td>
<td>—</td>
<td>EBI</td>
<td>EBI</td>
<td>—</td>
<td>EBI</td>
</tr>
<tr>
<td></td>
<td>RDYINV3</td>
<td>RDYINV2</td>
<td>RDYINV1</td>
<td>—</td>
<td>EBI</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>R/W-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>R/W-0</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>—</td>
<td>RDYEN3</td>
<td>RDYEN2</td>
<td>RDYEN1</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EBIRDYINV3</td>
</tr>
<tr>
<td>15:8</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EBIRDYINV2</td>
</tr>
<tr>
<td></td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>EBIRDYINV1</td>
</tr>
<tr>
<td>7:0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>R/W-0</td>
<td>U-0</td>
<td>R/W-0</td>
</tr>
<tr>
<td></td>
<td>EBICSEN3</td>
<td>EBICSEN2</td>
<td>EBICSEN1</td>
<td>EBICSEN0</td>
<td>—</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** Unimplemented: Read as ‘0’
- **bit 30** **EBIRDYINV3**: EBIRDY3 Inversion Control bit
  - **1** = Invert EBIRDY3 pin before use
  - **0** = Do not invert EBIRDY3 pin before use
- **bit 29** **EBIRDYINV2**: EBIRDY2 Inversion Control bit
  - **1** = Invert EBIRDY2 pin before use
  - **0** = Do not invert EBIRDY2 pin before use
- **bit 28** **EBIRDYINV1**: EBIRDY1 Inversion Control bit
  - **1** = Invert EBIRDY1 pin before use
  - **0** = Do not invert EBIRDY1 pin before use
- **bit 27** Unimplemented: Read as ‘0’
- **bit 26** **EBIRDYEN3**: EBIRDY3 Pin Enable bit
  - **1** = EBIRDY3 pin is enabled for use by the EBI module
  - **0** = EBIRDY3 pin is available for general use
- **bit 25** **EBIRDYEN2**: EBIRDY2 Pin Enable bit
  - **1** = EBIRDY2 pin is enabled for use by the EBI module
  - **0** = EBIRDY2 pin is available for general use
- **bit 24** **EBIRDYEN1**: EBIRDY1 Pin Enable bit
  - **1** = EBIRDY1 pin is enabled for use by the EBI module
  - **0** = EBIRDY1 pin is available for general use
- **bit 23-18** Unimplemented: Read as ‘0’
- **bit 17** **EBIRDYLV3**: EBIRDYx Pin Sensitivity Control bit
  - **1** = Use level detect for EBIRDYx pins
  - **0** = Use edge detect for EBIRDYx pins
- **bit 16** **EBIRPEN**: EBIRP Pin Sensitivity Control bit
  - **1** = EBIRP pin is enabled for use by the EBI module
  - **0** = EBIRP pin is available for general use
- **bit 15-14** Unimplemented: Read as ‘0’
- **bit 13** **EBIWEEN**: EBIWE Pin Enable bit
  - **1** = EBIWE pin is enabled for use by the EBI module
  - **0** = EBIWE pin is available for general use

**Note:** When EBIMD = 1, the bits in this register are ignored and the pins are available for general use.
Register 47-7: CFGEBIC: External Bus Interface Control Pin Configuration Register (Continued)

bit 12  **EBIOEN:** EBIOE Pin Enable bit
  1 = EBIOE pin is enabled for use by the EBI module
  0 = EBIOE pin is available for general use

bit 11-10  **Unimplemented:** Read as ‘0’

bit 9  **EBIBSEN1:** EBIBS1 Pin Enable bit
  1 = EBIBS1 pin is enabled for use by the EBI module
  0 = EBIBS1 pin is available for general use

bit 8  **EBIBSEN0:** EBIBS0 Pin Enable bit
  1 = EBIBS0 pin is enabled for use by the EBI module
  0 = EBIBS0 pin is available for general use

bit 7  **EBICSSEN3:** EBICS3 Pin Enable bit
  1 = EBICS3 pin is enabled for use by the EBI module
  0 = EBICS3 pin is available for general use

bit 6  **EBICSSEN2:** EBICS2 Pin Enable bit
  1 = EBICS2 pin is enabled for use by the EBI module
  0 = EBICS2 pin is available for general use

bit 5  **EBICSSEN1:** EBICS1 Pin Enable bit
  1 = EBICS1 pin is enabled for use by the EBI module
  0 = EBICS1 pin is available for general use

bit 4  **EBICSSEN0:** EBICS0 Pin Enable bit
  1 = EBICS0 pin is enabled for use by the EBI module
  0 = EBICS0 pin is available for general use

bit 3-2  **Unimplemented:** Read as ‘0’

bit 1  **EBIDEN1:** EBI Data Upper Byte Pin Enable bit
  1 = EBID<15:8> pins are enabled for use by the EBI module
  0 = EBID<15:8> pins have reverted to general use

bit 0  **EBIDENO:** EBI Data Upper Byte Pin Enable bit
  1 = EBID<7:0> pins are enabled for use by the EBI module
  0 = EBID<7:0> pins have reverted to general use

**Note:** When EBIMD = 1, the bits in this register are ignored and the pins are available for general use.
47.3 INTERFACING TO VARIOUS DEVICES

To provide support for a wide range of external devices, the EBI module can be configured to understand such things as the type, size, and bus width of each attached device. Since this configuration is determined on a Chip Select basis, when mixing devices on the EBI, similar devices should be on the same Chip Select line.

47.3.1 Interfacing to NOR Flash Memory

Figure 47-3 shows an example of connecting the EBI bus to an asynchronous NOR Flash device.

Figure 47-3: Interface to an Asynchronous Flash Device

Note that the Write Protect (WP) pin on the Flash device is connected to a General Purpose I/O pin (RB6). This pin would not be under EBI control, hence it would be up to the user application to enable Flash writes prior to using the EBI to write the data, and disabling Flash writes when all writes are complete.

47.3.2 Interfacing to SRAM Memory

Figure 47-2 shows an example connecting the EBI bus to an Asynchronous SRAM memory device.

Figure 47-4: Interface to an Asynchronous SRAM Device
Section 47. External Bus Interface (EBI)

In the case of 8-bit memory devices, the Byte-Select (EBIBS) lines are not necessary and only EBID<7:0> would be connected. Two 8-bit memory devices can share the address and control lines, and only have separate data lines.

47.3.3 Interfacing to Non-Memory Devices

When non-memory devices are connected to the EBI bus, these devices provide a Ready line to indicate when valid data is on the data bus. This Ready line is connected to the EBIRDYx pin to connect to the EBI bus.

Figure 47-5 shows an example of connecting a non-Memory camera device to the EBI bus.

Figure 47-5: Interface to a Non-memory Device

In this case, when the Chip Select line is asserted, the EBI bus would wait to read the data lines until the camera asserts the EBIRDY1 line.

47.3.4 Combining Devices on the EBI Bus

Figure 47-6 shows an example of sharing some elements of the EBI bus to achieve certain design purposes.

Figure 47-6: Sharing a Connection Between SRAM and a LCD

In this case, the SRAM serves as a memory buffer for the LCD. As the EBI goes through the data in the buffer, it is clocked into the LCD, permitting it to display the contents in memory.
47.4 BUS CONFIGURATION

47.4.1 Configuring Address Lines

The CFGEBIA register controls the number of EBI lines in use and whether, overall, the EBI controls the address, data, and control lines, or if the lines are available for general use.

Each address line has an enable control in CFGEBIA. By setting the corresponding bit, the EBI controls that address line. Address bits in use should be contiguous.

The EBIPINEN bit (CFGEBIA<31>) determines control of the overall EBI lines. Setting the bit allows the EBI to control the corresponding pins. Clearing the bit allows the lines to be used for other purposes.

47.4.2 Configuring Control and Data Lines

The CFGEBIC register determines the settings for the EBICSx, EBIRDYx, EBIBSx, EBIWE, EBIOE, EBIRP, and EBIDx lines.

The EBICSEN0, EBICSEN1, EBICSEN2, and EBICSEN3 bits in the CFGEBIC register determine which EBICS lines are enabled. Setting a bit enables the corresponding EBICSx pin for EBI use. Clearing a bit disables the pin and allows it for general purpose use.

Three bits control the EBIRDYx lines, EBIRDYEN1, EBIRDYEN2, and EBIRDYEN3. When a bit is set, the corresponding EBIRDYx pin is enabled for use by the EBI module. Clearing a bit allows the EBIRDYx pin to be used for general use.

The EBIRDYINV1, EBIRDYINV2, and EBIRDYINV3 bits control the inversion of the EBIRDYx line prior to using it. When a bit is set, the EBIRDYx level will be inverted. When cleared, the signal is not inverted.

The final bit, EBIRDYLVL, determines whether the EBI module uses a level detect or an edge detect to determine when the EBIRDYx line is being asserted. Devices that have mixed assertion logic can thus be combined on the EBI bus by having separate EBIRDYx lines.

The EBIBSEN0, EBIBSEN1, EBIBWEEN, EBIOEEN, and EBIRPEN bits control whether the EBIBSx, EBIWE, EBIOE, and EBIRP pins are enabled for use by the EBI module. The pins are enabled when the corresponding control bit is set, or available for general use when the bit is cleared.

The EBIDEN0 and EBIDEN1 bits control which 8-bit portion of the EBIDx bus is enabled. Setting EBIDEN1 enables EBID<15:8> for use by the EBI bus, and setting EBIDEN0 enables the EBID<7:0> lines. Clearing the bits disables the corresponding side of the bus and allows the pins to be used for general use. Note that only enabling EBID<15:8> does not guarantee that the bus will only use 8-bit transfers. If the design requires only 8-bit data transfers, use EBID<7:0>.
47.5 DEVICE CONFIGURATION

The basic configuration for each device or group of devices that will be using a particular Chip Select line includes:

• Configuring the base address of the device
• The type of device
• The size of the device
• The bus timing for the device

47.5.1 Base Address

Configuring the base address of a memory device is done through the EBICSx register. This register sets the beginning address in the PIC32 physical memory space where the device will appear. It is a 16-bit value, which allows the minimum device size (64 KB) to have contiguous locations in memory. In addition, it is not required to have larger memory devices appear lower in memory than smaller devices. For example, it is possible to have a 64 KB device at the start of EBI memory (0x20000000), adjacent to a 16 MB device, which would start at 0x20010000.

Example code for configuring the EBICSx registers to handle four memory devices is provided in Example 47-1. This example maps a 1 MB device, followed by a 64 KB device, followed by another 1 MB device, and finally a 16 MB device.

Example 47-1:

```c
/* Device 1: 1 MB Flash going from 0x20000000 to 0x200FFFFF */
EBICS0 = 0x20000000;
/* Device 2: 64 KB SRAM going from 0x20100000 to 0x2010FFFF */
EBICS1 = 0x20100000;
/* Device 3: 1 MB Flash going from 0x20110000 to 0x2020FFFF */
EBICS2 = 0x20110000;
/* Device 4: 16 MB SRAM going from 0x20210000 to 0x2120FFFF */
EBICS3 = 0x20210000;
```

47.5.2 Device Type

The EBI bus needs to know what type of device is attached to a particular chip select line. This is done through the MEMTYPE field in the EBIMSKx register of each Chip Select. There are two options available, NOR Flash (MEMTYPE = 0b010) and SRAM (MEMTYPE = 0b001). Non-memory devices would not require this to be set.

Example code for configuring the memory types for the devices previously listed in Example 47-1 is shown in Example 47-2.

Example 47-2:

```c
EBIMSK0bits.MEMTYPE = 0b010; /* Device 1: NOR Flash */
EBIMSK1bits.MEMTYPE = 0b001; /* Device 2: SRAM */
EBIMSK2bits.MEMTYPE = 0b010; /* Device 3: NOR Flash */
EBIMSK3bits.MEMTYPE = 0b001; /* Device 4: SRAM */
```

47.5.3 Device Size

In addition to the type of memory device, the EBI bus needs to know the size of the attached device. The device size is configured by the MEMSIZE<4:0> (EBIMSKx<4:0>) bits.

The smallest memory size that can be accommodated by the EBI in a contiguous manner is 64 KB. Smaller devices would alias to that size, which results in gaps in the memory map.

Example code for configuring the device size for the devices listed in Example 47-2 is shown in Example 47-3.
Example 47-3:

<table>
<thead>
<tr>
<th>EBIMSK0 bits.MEMSIZE = 0b00101; /* Device 1: 1 MB */</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBIMSK1 bits.MEMSIZE = 0b00001; /* Device 2: 64 KB */</td>
</tr>
<tr>
<td>EBIMSK2 bits.MEMSIZE = 0b00101; /* Device 3: 1 MB */</td>
</tr>
<tr>
<td>EBIMSK3 bits.MEMSIZE = 0b01001; /* Device 4: 16 MB */</td>
</tr>
</tbody>
</table>

Example 47-4:

<table>
<thead>
<tr>
<th>EBIMSK0 bits.REGSEL = 0b000; /* Device 1: EBISMT0 */</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBIMSK1 bits.REGSEL = 0b001; /* Device 2: EBISMT1 */</td>
</tr>
<tr>
<td>EBIMSK2 bits.REGSEL = 0b000; /* Device 3: EBISMT0 */</td>
</tr>
<tr>
<td>EBIMSK3 bits.REGSEL = 0b010; /* Device 4: EBISMT2 */</td>
</tr>
</tbody>
</table>

47.5.4 Bus Timing

The EBI bus can handle a variety of memory timing requirements. There are three registers that configure the timing parameters, EBISMT0, EBISMT1, and EBISMT2. Identical memory devices should use the same EBISMTx register for timing purposes.

The REGSEL<2:0> bits (EBIMSKx<10:8>) are used to set which EBISMTx register will be used for each Chip Select line. Example code for setting up the bus timing is shown in Example 47-4.
### 47.5.5 Configuring Bus Timing

Since devices have different timing considerations, it is necessary to configure the EBI bus to automatically handle those timing differences. By configuring the EBISMTx register, the bus will adjust read and write timings as needed.

If a device has a Ready pin, the RDYMODE bit (EBISMTx<26>) is set.

For devices with Page mode, the PAGEMODE bit (EBISMTx<23>) enables support for the device in the EBI module. Then, the PAGESIZE<1:0> bits (EBISMTx<25:24>) configure the Page Size so that the EBI knows how many words to write for each page.

The remaining bits in the EBISMTx register control how the EBI bus timing works. Refer to the diagrams in **47.6 “Timing Diagrams”** to see visual representations of the bus cycles, and where the timing parameters are needed. Table 47-4 lists the individual timing parameters, and what they affect.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>EBISMTx Register Bit Name</th>
<th>Description</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>tPRC</td>
<td>TPRC&lt;3:0&gt;</td>
<td>Page mode read cycle time.</td>
<td>Read cycle time is TPRC+1 clock cycles.</td>
</tr>
<tr>
<td>tBTA</td>
<td>TBTA&lt;2:0&gt;</td>
<td>Bus turnaround time.</td>
<td>Clock cycles (0-7) for read-to-write, write-to-read, and read-to-read (different CS) transitions.</td>
</tr>
<tr>
<td>tWP</td>
<td>TWP&lt;5:0&gt;</td>
<td>Write Pulse Width.</td>
<td>Write pulse width is TWP+1 clock cycles.</td>
</tr>
<tr>
<td>tWR</td>
<td>TWR&lt;1:0&gt;</td>
<td>Write address/data hold time.</td>
<td>Number of clock cycles to hold address or data on the bus.</td>
</tr>
<tr>
<td>tAS</td>
<td>TAS&lt;1:0&gt;</td>
<td>Write address setup time.</td>
<td>Clock cycles for address setup time. A value of ‘0’ is only valid in case of SRAM.</td>
</tr>
<tr>
<td>tRC</td>
<td>TRC&lt;5:0&gt;</td>
<td>Read cycle time (non-Page mode memory).</td>
<td>Read cycle time is TRC + 1 clock cycles.</td>
</tr>
<tr>
<td>tRPD</td>
<td>TRPD&lt;11:0&gt;</td>
<td>Flash memory reset time.</td>
<td>Clock cycles after Flash reset before a read/write access.</td>
</tr>
</tbody>
</table>

**Note:** Clock cycles refers to system clock cycles, and is dependent on the speed of the SYSCLK when the system is running.
47.5.6 Reading and Writing to the EBI Module

Example 47-5 shows code that uses both reads and writes to SRAM attached to the EBI module. This example assumes a 200 MHz System Clock and that the TLB and MMU are set up correctly.

Example 47-5:

```c
// Global Defines
#define SRAM_ADDR_CS0 0xC0000000
#define RAM_SIZE 2*1024*1024

int main(void)
{
    uint32_t loop;
    uint32_t *addr;
    uint32_t val;
    // Note: ISSI SRAM (IS64WV102416BLL). All of the parameters of the EBI module are set up based on the timing of this RAM.
    // Enable address lines [0:17]
    // Controls access of pins shared with PMP
    CFGEBIA = 0x800FFFFF;
    // Enable write enable pin
    // Enable output enable pin
    // Enable byte select pin 0
    // Enable byte select pin 1
    // Enable Chip Select 0
    // Enable data pins [0:15]
    CFGEBIC = 0x00003313;
    // Connect CS0 to physical address
    EBICS0 = 0x20000000;
    // Memory size is set as 2 MB
    // Memory type is set as SRAM
    // Uses timing numbers in EBISMT0
    EBIMSK0 = 0x00000026;
    // Configure EBISMT0
    // ISSI device has read cycles time of 10 ns
    // ISSI device has address setup time of 0ns
    // ISSI device has address/data hold time of 2.5 ns
    // ISSI device has Write Cycle Time of 10 ns
    // Bus turnaround time is 0 ns
    // No page mode
    // No page size
    // No RDY pin
    EBISMT0 = 0x000029CA;
    // Keep default data width to 16-bits
    EBISMCON = 0x00000000;
    addr = (uint32_t *)SRAM_ADDR_CS0;
    // Write loop
    for (loop=0; loop < RAM_SIZE/4; loop++)
    {
        *addr++ = 0xAA55AA55;
    }
    // Read and verify loop
    addr = (uint32_t *)SRAM_ADDR_CS0; // reset address to beginning
    for (loop=0 ; loop < RAM_SIZE/4; loop++)
    {
        val = *addr++;
        if (val != 0xAA55AA55)
        {
            return (0); // Exit Failure
        }
    }
    return (1); // exit success
}
```


47.6 TIMING DIAGRAMS

47.6.1 Read/Write Access

Figure 47-7 shows the timing diagram of a read access. The EBI module checks the EBIRDYx pin after the tRC read access time. You need to ensure that the EBIRDYx signal is being driven with respect to the System Clock (SYSCLK). This avoids a possible race condition if EBIRDYx is driven by a different clock. When EBIRDYx is high, the EBI module latches the read data at the next rising clock edge.

Figure 47-7: Read Access of the Device with Ready Signal

![Timing Diagram](image)

Figure 47-8 shows the timing diagram of a write access. The EBI module checks the EBIRDYx pin after a time equal to tAS(address setup time) + tWP(write period). When EBIRDYx is high, the write is finished. You need to ensure that the EBIRDYx signal is being driven with respect to the SYSCLK. This avoids a possible race condition if EBIRDYx is driven by a different clock.

Figure 47-8: Write Access of the Device with Ready Signal

![Timing Diagram](image)
47.6.2 Static Memory

The static memory timing diagrams assume an internal delay of two System Bus clock cycles, which is the delay for a cycle to be active on the bus to the clock cycle where the corresponding memory command is seen on the memory bus.

Figure 47-9 shows the timing for a SRAM and Flash read operation, where tRC is the read cycle time.

Figure 47-9: SRAM and Flash Memory Read Timing

![SRAM and Flash Memory Read Timing Diagram]

Figure 47-10 shows the Flash page read operation, where tRC is the read cycle time and tPRC is the page mode read cycle time.

Figure 47-10: Flash Memory Page Read Timing

![Flash Memory Page Read Timing Diagram]
Section 47. External Bus Interface (EBI)

Figure 47-11 shows the SRAM and Flash timing for a write operation, where tAS is the address setup time, tWP is the write pulse period, and tWR is the write recovery time.

**Figure 47-11: SRAM and Flash Write Timing**

Figure 47-12 shows an example of inserting one idle clock for memory data bus turnaround time, where tBTA is the number of idle clock cycles.

**Figure 47-12: External Memory Data Bus Turnaround Timing**
47.7 EFFECTS OF RESET

47.7.1 On Reset
All EBI module registers are forced to their reset states on a device Reset. In addition, the CFGEBIA and CFGEBIC registers are forced to their Reset states.

47.7.2 After Reset
The EBI module is not active, and must be initialized prior to accessing memory in the EBI address space. In addition, the Translation Lookaside Buffer (TLB) of the CPU must be set up prior to accessing any external device.

47.8 OPERATION IN POWER-SAVING MODES

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

47.8.2 Idle Mode
When the device enters Idle mode, the EBI module continues to operate, and can execute transfers between internal and external memory.

47.8.3 Debug Mode
The behavior of the EBI module is unaltered in Debug mode.
47.9 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 External Bus Interface (EBI) 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) for additional application notes and code examples for the PIC32 family of devices.
47.10 REVISION HISTORY

Revision A (November 2013)

This is the initial released version of this 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, rpIC, 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, PICC, 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-Strike 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.

© 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  
Web Address:  
www.microchip.com

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

**Austin, TX**  
Tel: 512-257-3370

**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: 216-447-0464  
Fax: 216-447-0643

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

**Detroit**  
Novi, MI  
Tel: 248-848-4000

**Houston, TX**  
Tel: 281-894-5983

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

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

**New York, NY**  
Tel: 631-435-6000

**San Jose, CA**  
Tel: 408-735-9110

**Canada - Toronto**  
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 - Chengdu**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**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: 852-2943-5100  
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: 86-755-8864-2200  
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: 91-20-3019-1500

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

**Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

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

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 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: 886-7-213-7830

**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: 45-4450-2829

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-53-60-79

**Germany - Dusseldorf**  
Tel: 49-2129-3766400

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

**Germany - Pforzheim**  
Tel: 49-7231-424750

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

**Italy - Venice**  
Tel: 39-049-7625286

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

**Poland - Warsaw**  
Tel: 48-22-3325737

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

**Sweden - Stockholm**  
Tel: 46-8-5090-4654

**UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820

© 2013 Microchip Technology Inc.