Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

System ACE CompactFlash Solution

DS080 (v2.0) October 1, 2008 **Product Specification**

Features

- System-Level Features:
  - High-capacity pre-engineered configuration solution for FPGAs[1]
  - System ACE™ CF Controller XCCACE-TQG144I device
  - Maximum CompactFlash (CF) partition capacity of 2 GB
  - Non-volatile system storage solution
  - Flexible configuration interfaces
  - System configuration rates of up to 30 Mb/s
  - Board space requirement as low as 25 cm^2
- System ACE CF Controller:
  - CompactFlash interface supports most standard third-party CompactFlash (Type I or Type II) cards (up to 8 GB), and Hitachi Microdrives (up to 6 GB)
  - Configuration of a target FPGA chain through IEEE 1149.1 JTAG with a throughput up to 16.7 Mb/s
  - Interfaces include CompactFlash, JTAG, and MPU
  - MPU interface is compatible with various microprocessor and microcontroller bus interfaces, including the Xilinx FPGA-based PowerPC ® and MicroBlaze™ processors
  - IEEE 1149.1 Boundary-Scan Standard Compliant (JTAG)
  - Supports FAT12 and FAT16 file systems
  - Compact 144-pin TQFP package
  - Low power

General Description

Xilinx developed the System Advanced Configuration Environment (System ACE) to address the need for a space-efficient, pre-engineered, high-density configuration solution for systems with multiple FPGAs. System ACE technology is a ground-breaking in-system programmable configuration solution that provides substantial savings in development effort and cost per bit over traditional PROM and embedded solutions for high-capacity FPGA systems.

The System ACE CF solution combines Xilinx expertise in configuration control with industry expertise in commodity memories.

As shown in Figure 1, the System ACE CompactFlash solution is a chipset, consisting of a controller device (System ACE CF controller) and a commercially available CompactFlash storage device.

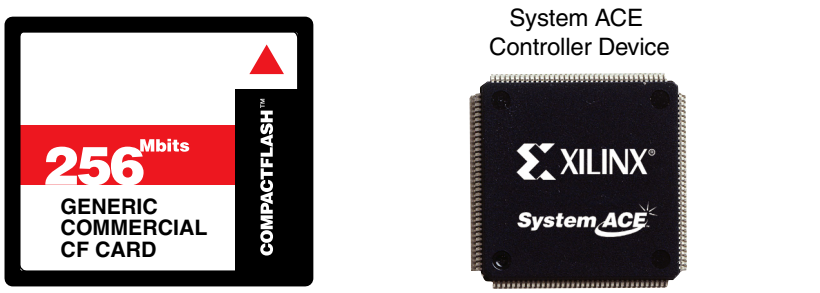

Figure 1: **System ACE CompactFlash Solution**

1. System ACE CF does not support configuration of Xilinx CPLD or PROM devices.

Figure 2 shows that the System ACE CF controller contains multiple interfaces, including CompactFlash, MPU, and JTAG, to allow for a highly flexible configuration solution. For added flexibility, a CompactFlash or Hitachi Microdrive storage device can be used to store multiple bitstreams. The combination of the System ACE CF controller and a standard CompactFlash or Hitachi Microdrive storage device delivers a powerful configuration solution for high-density FPGA systems.

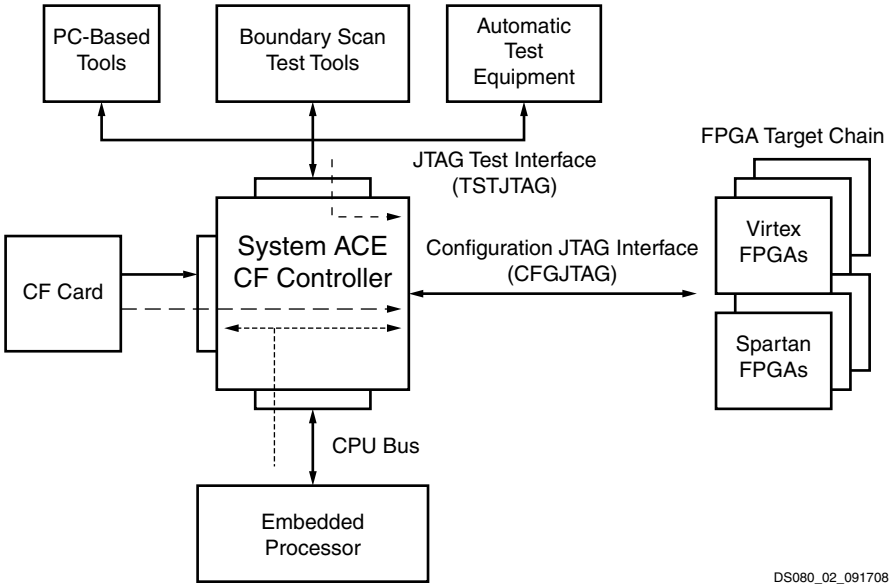

Figure 2: **System ACE CF Controller Interfaces**

System ACE CF Controller

The System ACE CF controller manages FPGA configuration data. The controller provides an intelligent interface between an FPGA target chain and various supported configuration sources; it can target multiple FPGA devices using JTAG at a selectable throughput of up to 16.7 Mbits/sec. As shown in Figure 3, three interfaces are available for configuring a target FPGA chain through the Configuration JTAG Port. These interfaces are: CompactFlash, Microprocessor (MPU), and Test JTAG.

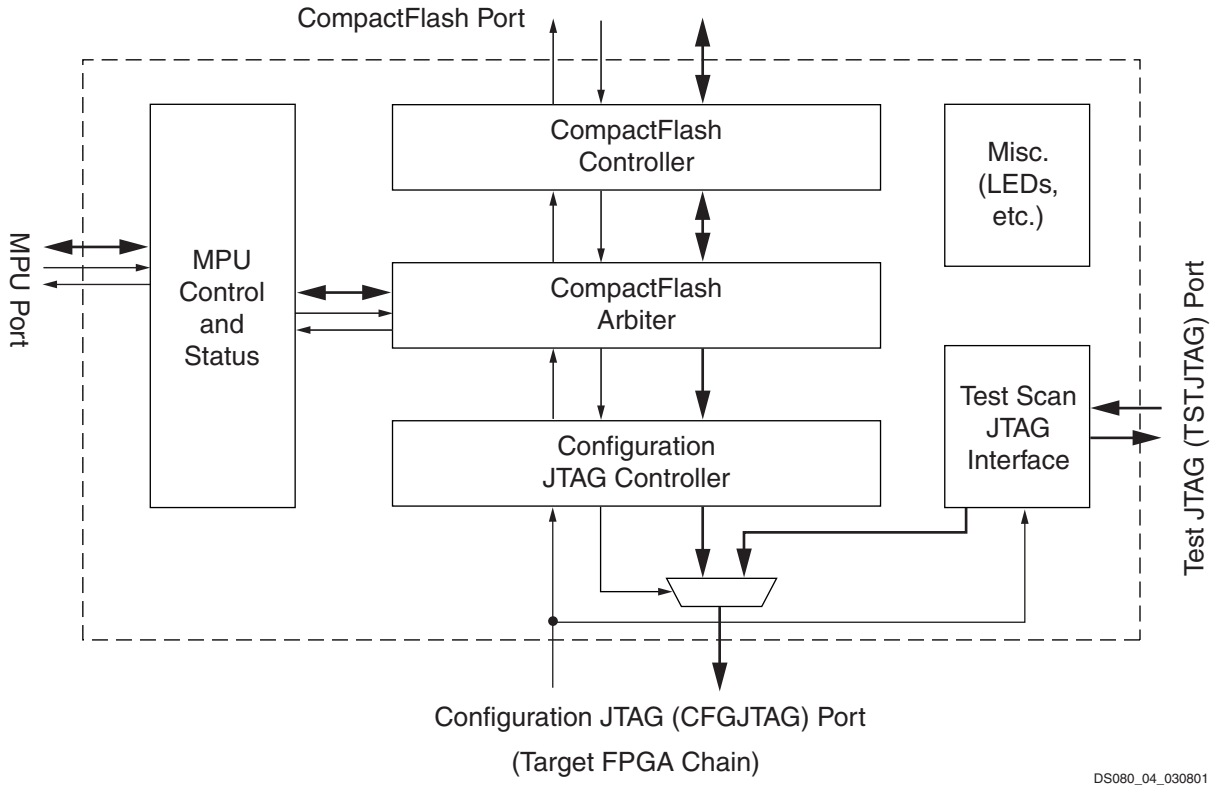

Figure 3: **System ACE CF Controller Block Diagram**

The directory structure used by the System ACE CF controller enables it to support both CompactFlash and Hitachi Microdrive devices through the CompactFlash port.

The MPU interface has access to the CompactFlash port, the Configuration JTAG port, and local control/status features. The Test JTAG port is used when doing Boundary-Scan testing of the target FPGA chain or the System ACE CF controller. Details about each interface are discussed below.

The System ACE CF controller has two main power supplies: the core power supply (V_{CCL}) and a CompactFlash/Test JTAG interface power supply (V_{CCH}). The V_{CCH} power source supplies the Test JTAG and CompactFlash port levels. These two interfaces must be powered at 3.3V. The V_{CCL} core power source supplies the MPU and Configuration JTAG ports, which can be run at 3.3V or 2.5V. It is important to note that the MPU and Configuration JTAG interfaces are always powered at the same voltage. Considerations for the interface voltage are discussed in **Typical Configuration Modes**, page 37. See Figure 4, page 4.

- Shaded output buffers drive $V_{OH} = V_{CCL}$ = 2.5V or 3.3V
- Shaded input buffers sense $V_{IH} = V_{CCL}$ = 2.5V or 3.3V
- All non-shaded output buffers drive $V_{OH} = V_{CCH}$ = 3.3V
- All non-shaded input buffers sense $V_{IH} = V_{CCH}$ = 3.3V
- "LS" denotes level-shifter
- Core voltage level = V_{CCL} = 2.5V or 3.3V

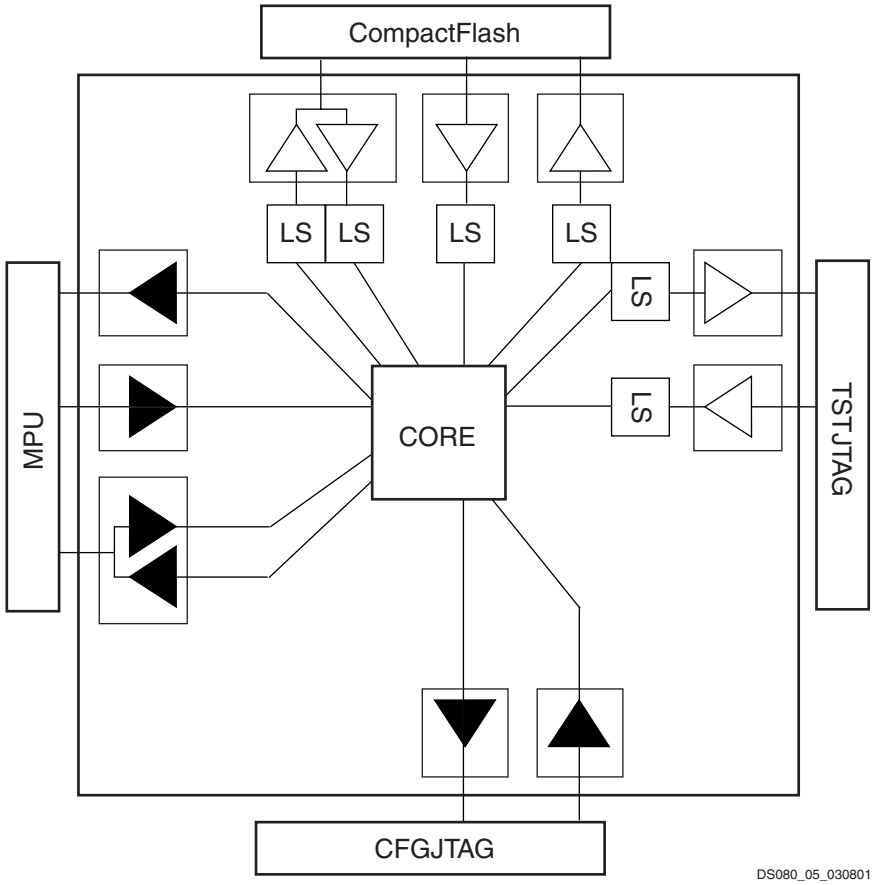

Figure 4: **System ACE CF Controller I/O Requirements**

Status Indicators

The System ACE CF controller has indicator pins (Table 1) to help monitor device status during operation.

Table 1: **System ACE CF Controller Status Indicators**

Name	Pin	Description
$\overline{\text{STATLED}}$	95	• When on, the Status LED indicates that configuration is DONE. • When blinking, this LED indicates that configuration is still in progress. • When off this LED indicates that configuration is in an IDLE state.
$\overline{\text{ERRLED}}$	96	• When on, the ERROR LED indicates that an error occurred. • When blinking, this LED indicates that no CompactFlash device was found when the CompactFlash for the Configuration JTAG interface was enabled. • When off, this LED indicates that no errors are detected.

Resetting the System ACE CF Controller

There are three types of reset of the System ACE CF controller:

1. Power-on-reset (POR)
2. Device reset
3. Configuration controller reset

Power-on-Reset (POR)

The POR circuit is used to reset the entire System ACE CF controller device upon device power up. The built-in POR circuit can be bypassed in order to use an external POR circuit. To bypass the built-in POR circuit, the POR_BYPASS pin should be set to '1' and the POR_RESET pin is used to reset the device (see Table 2).

Note: If the V_{CCL} rail reaches the threshold voltage before the V_{CCH} rail reaches its threshold voltage, then consider using an external POR circuit or $\overline{\text{RESET}}$ pin to hold the device into reset until the V_{CCH} rail reaches the threshold voltage.

Table 2: **POR Functionality**

POR_BYPASS[1]	POR_RESET	Description
'0'	Don't care	Built-in POR circuit is used to reset the device.
'1'	'0'	External POR circuit is selected but the device is not being reset.
'1'	'1' [2]	External POR circuit is selected and the device is being reset.

1. The POR_BYPASS pin should be held at a static '0' or '1' while the System ACE CF controller is receiving power.
2. Hold at '1' for at least one microsecond.

Device-Level Reset

The entire System ACE CF controller device can be reset by asserting the $\overline{\text{RESET}}$ pin of the System ACE CF Controller. The timing associated with this operation is shown in Figure 5, page 6.

CompactFlash Card Reset

The CompactFlash card can be issued a soft reset command by issuing a ResetMemCard command through the CMD[2:0] bits in the **SECCNTCMDREG Register (BYTE address 014h-15h, WORD address 0Ah)**, page 29.

Configuration Controller Reset

The configuration controller portion of the System ACE CF device can be reset by asserting CFGRESET = '1' in the CONTROLREG MPU register (CFGRESET is bit 7). Asserting CFGRESET = '1' will reset the portion of the System ACE CF device that controls the reading of ACE file data from the CF card and configuration of the devices connected to the CFGJTAG port. The CFGRESET register is used in conjunction with the CFGMODE and CFGSTART pins/registers to control this configuration process.

Note: It is important to assert CFGRESET='1' while accessing CompactFlash card sector data via the MPU port, otherwise a CFGERROR condition could result.

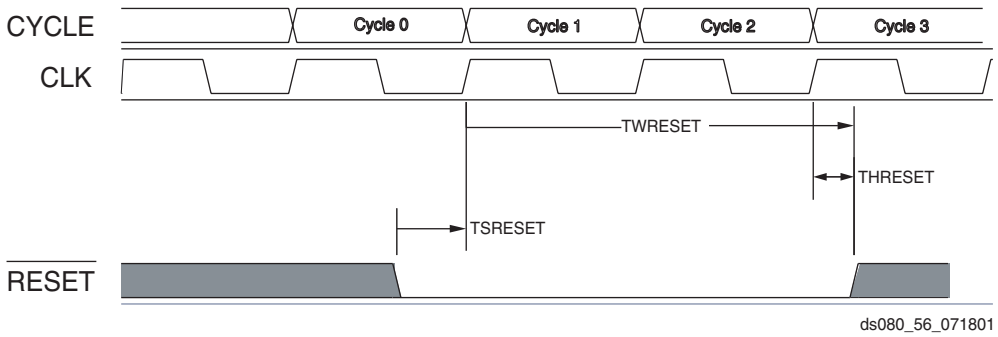

Figure 5: **System ACE RESET Function Timing Diagram**

Table 3: **System ACE RESET**

Symbol	Parameter	Min	Max	Units
TW(RESET)	System ACE CF controller Reset pulse width	3(1)		rising edges
TH(RESET)	Reset hold time after rising edge of CLK	4		ns
TS(RESET)	System ACE CF controller Reset setup up time before rising edge of CLK	7(1)		ns

Notes:
1. When using the System ACE CF controller RESET, TSRESET + TWRESET of three rising edges of CLK is required.

Interfaces Overview

This section discusses the details of each supported System ACE CF controller interface.

CompactFlash Interface (CF)

The CompactFlash interface is the key System ACE CF controller interface for high-capacity systems. The CompactFlash port can accommodate any standard CompactFlash module (up to 8 GB) or Hitachi Microdrives (up to 6 GB), all with the same form factor and board space requirements.

The use of standard CompactFlash devices gives system designers access to high-density Flash memory in a very efficient footprint that does not change with density. CompactFlash is a removable medium which simplifies making changes to the memory contents or upgrading the memory density.

The CompactFlash interface is comprised of two sub-components: a CompactFlash Controller and a CompactFlash Arbiter. The CompactFlash Controller detects the presence and maintains the status of the CompactFlash device. This Controller also handles all CompactFlash device access bus cycles, and abstracts and implements CompactFlash commands such as soft reset, identify drive, and read/write sector(s). The CompactFlash Arbiter controls the interface between the MPU and the Configuration JTAG Controller for access to the CompactFlash data buffer.

When using the CompactFlash card as the configuration source, the CFGTCK output for the System ACE CF controller device is derived from the CLK input to the System ACE CF controller. The operating frequency of the CFGTCK is the same as CLK:

- The minimum clock operating frequency is 0 MHz.
- The maximum clock operating frequency is either 33 MHz *or* the maximum JTAG TCK clock speed dictated by the devices in the JTAG chain and/or the board design. The lowest of these values should be used.

CompactFlash devices are compliant with multiple read and write modes. The System ACE CF controller only supports ATA Common Memory Read and Write functions. Figure 6 and Figure 7, page 8 provide detailed timing information on these functions.

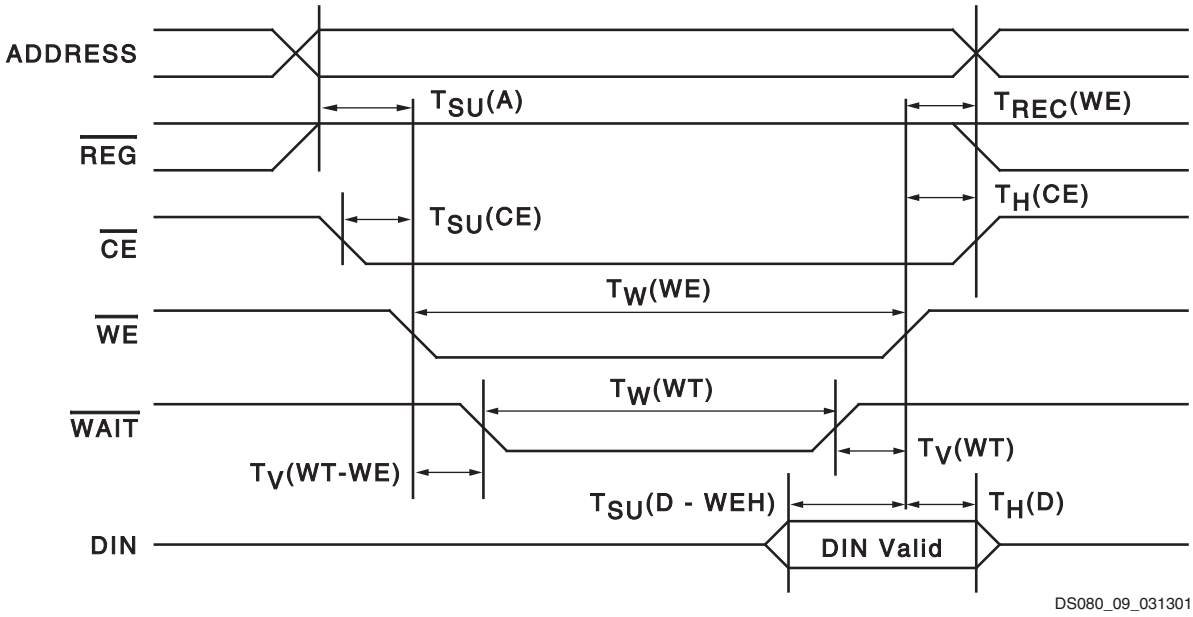

Figure 6: **CompactFlash Common Memory Write Timing Diagram**

Table 4: **Common Memory Write Timing**

Item	Symbol	IEEE Symbol	Min (ns)	Max (ns)
Data Setup before WE	T_{SU}(D-WEH)	tDVWH	80	
Data Hold following WE	T_H(D)	tlWMDX	30	
WE Pulse Width	T_W(WE)	tWLWH	150	
Address Setup Time	T_{SU}(A)	tAVWL	30	
CE Setup before WE	T_{SU}(CE)	tELWL	0	
Write Recovery Time	T_{REC}(WE)	tWMAX	30	

Table 4: **Common Memory Write Timing *(Continued)***

Item	Symbol	IEEE Symbol	Min (ns)	Max (ns)
CE Hold following WE	T_H(CE)	tGHEH	20	
Wait Delay Falling from WE	T_V(WT-WE)	tWLWTV		35
WE HIGH from Wait Release	T_V(WT)	tWTHWH	0	
Wait Width Time (Default Speed)	T_W(WT)	tWTLWTH		350

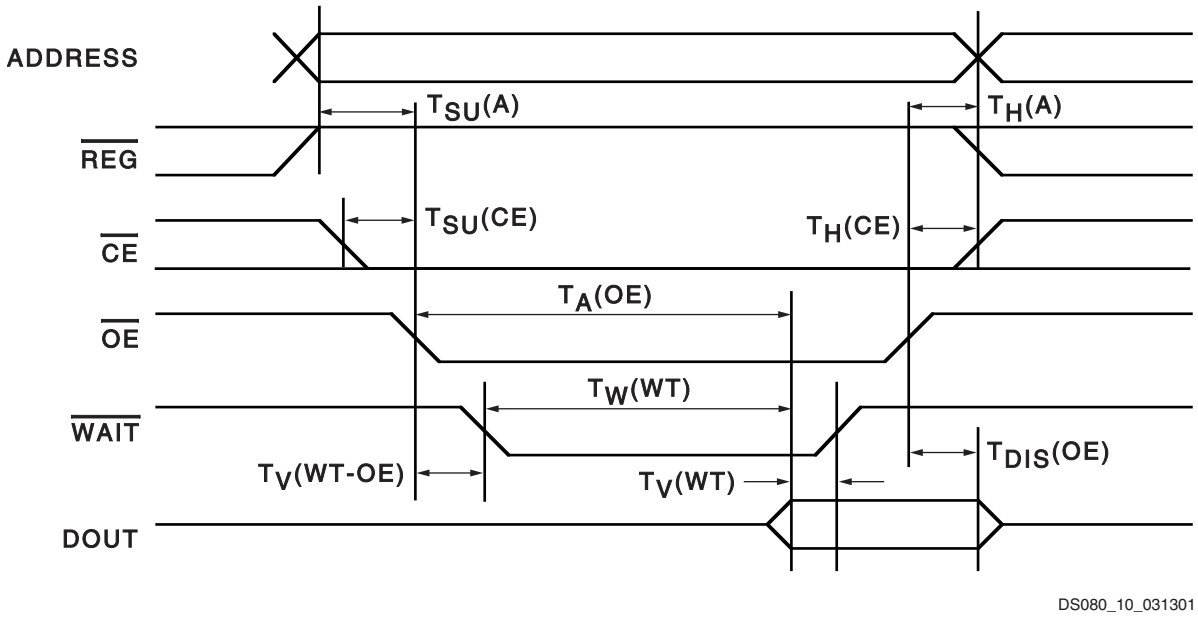

Figure 7: **CompactFlash Common Memory Read Timing Diagram**

Table 5: **Common Memory Read Timing**

Item	Symbol	IEEE Symbol	Min (ns)	Max (ns)
Output Enable Access Time	T_A(OE)	tGLQV		125
Output Disable Time from OE	T_{DIS}(OE)	tGHQZ		100
Address Setup Time	T_{SU}(A)	tAVGL	30	
Address Hold Time	T_H(A)	tGHAX	20	
CE Setup before OE	T_{SU}(CE)	tELGL	0	
CE Hold following OE	T_H(CE)	tGHEH	20	
Wait Delay Falling from OE	T_V(WT-OE)	tGLWTV		35
Data Setup for Wait Release	T_V(WT)	tQVWTH		0
Wait Width Time (Default Speed)	T_W(WT)	tWTLWTH		350

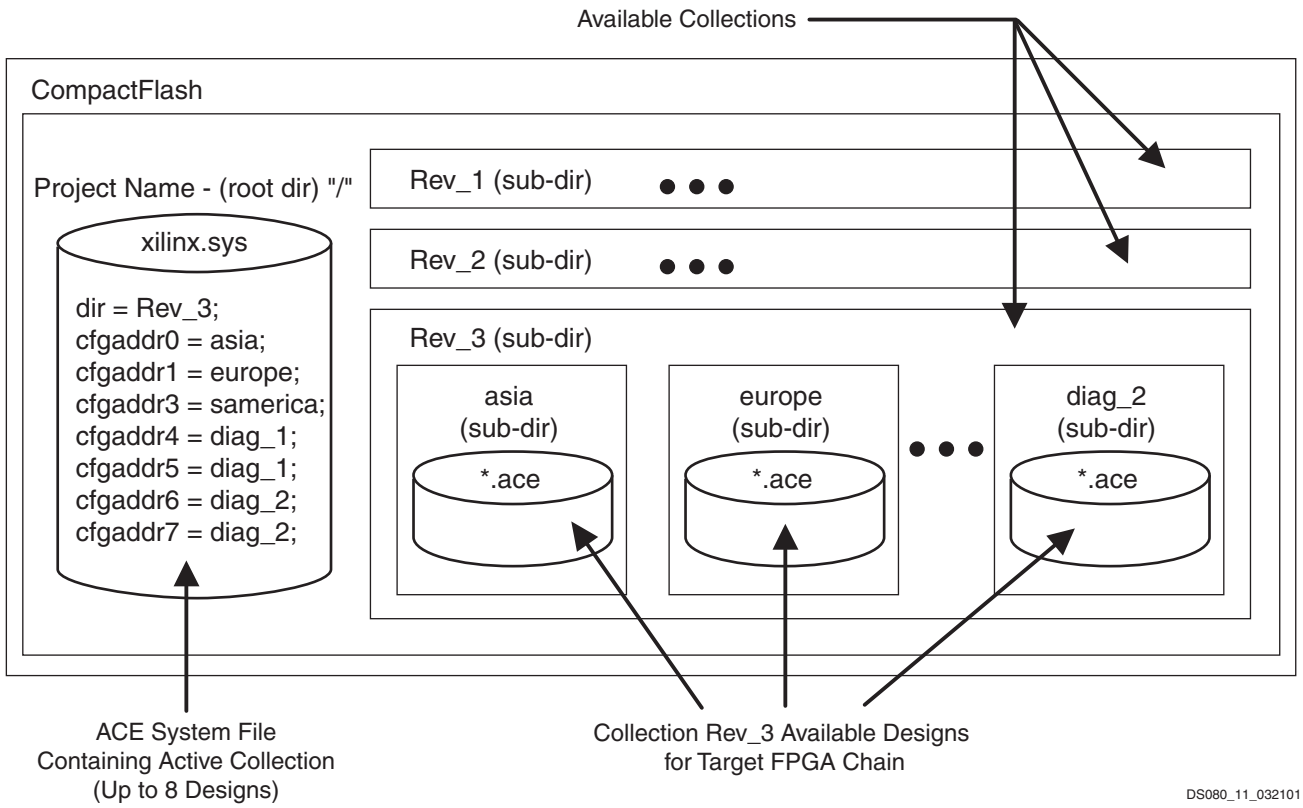

Figure 8: **System ACE Directory Structure**

System ACE CF Directory Structure

A basic understanding of the typical System ACE CF file and directory structure (shown in Figure 8) is useful when programming an FPGA target system with a CompactFlash device in the System ACE solution.

The ACE file is at the lowest level of the directory structure. The Xilinx iMPACT software converts a revision of a design (bitstream) into an ACE file. An ACE file represents a single set of bitstreams for a particular chain of devices.

The next level up in the file structure is a collection. The collection consists of eight ACE files grouped together. All of the ACE files in a collection (directory) can be addressed when in the System ACE CF environment. There can be several collections stored on a CompactFlash device, but only one collection can be active at any given time.

The `xilinx.sys` file determines the collection from which designs can be read.

The hierarchical design of the System ACE CF directory structure provides the ability to maintain multiple revisions or collections of different designs in a single CompactFlash device. Each collection directory can contain one or more designs that reside in different subdirectories. Each design subdirectory should contain a single ACE file that represents a single set of bitstreams for a particular chain of devices. In addition to FPGA configuration information, the collection and design subdirectories can contain other information pertaining to the system design such as system software, documentation, etc.

The `xilinx.sys` file in the root directory of the CompactFlash device is used to control which of the designs within the active collection is to be used to configure the chain of target devices. Only one collection, containing up to eight designs, can be active at one time.

The System ACE CF controller parses the `xilinx.sys` file to determine the active collection designs and uses the three configuration address pins or MPU register bits (CFGADDR) to select the desired design. If no `xilinx.sys` file exists in the root directory of the CompactFlash device, a single ACE file in the root directory is used by System ACE as the active design.

System ACE CF File Structure Requirements

- The System ACE CF file structure must be on the first partition of the CompactFlash device.
- The System ACE CF partition must be formatted as DOS FAT12 or FAT16.
- The `xilinx.sys` file or single ACE file must be in the root directory. The ACE file used only if `xilinx.sys` is not found. The `xilinx.sys` file describes one collection directory with up to eight subdirectories.
- The `xilinx.sys` file must contain the line dir=<*dir_name*>; where <*dir_name*> is the name of the collection directory
- The subsequent 8 lines of the `xilinx.sys` file must consist of the lines cfgaddr<*n*>=<*subdir_name*>; where <*n*> is 1 through 8 and <*subdir_name*> is the name of a design sub-directory in the collection. In the case of fewer than 8 designs in the collection, always start with cfgaddr0 and only use contiguous cfgaddr locations.
-
- Only one ACE file should exist in the ROOT directory and/or in each `\<dir>\<cfgaddr>` folder pointed to by the `xilinx.sys` file.
- When sourcing from the MPU, the total length of the ACE file must be a multiple of 32 bytes. Otherwise, additional dummy bytes (1s or 0s) should be sent to DATABUFREG to flush the last data buffer, allowing the controller to correctly load the final commands in the ACE file.
- All directories accessed by the System ACE CF controller must be formatted in a valid FAT 8.3 file name format.
- ACE file names can be up to eight characters long and must include the `.ace` file extension.
- All directories and ACE file names cannot contain these reserved characters:

left angle bracket	<
right angle bracket	>
colon	:
quote mark	"
forward slash	/
back slash	\
pipe	|

- The Partition Boot Record (PBR) for the first CompactFlash partition that is used by the System ACE CF controller must specify only one reserved sector.
- The CompactFlash card must be formatted with a sector-per-cluster size greater than 1.
- Other files and directories can coexist with System ACE files and directories.
- 2 GB is the maximum capacity partition that the System ACE CF controller can access using the FAT16 file system:

  (65,535 clusters max) X (32 KB per cluster max) = 2,147,123,200 bytes → 2 GB
- 16 MB is the maximum capacity partition that the System ACE CF controller can access using the FAT12 file system:

  (4,086 clusters max) X (4 KB per cluster max) = 16,736,256 bytes → 16 MB

System ACE CF Formatting Requirements

Three potential problem areas arise when formatting the CF card for use with the System ACE CF controller:

1. **Sectors-per-Cluster Size**

   A CF card formatted with only one sector (512 bytes) per cluster can cause problems for the System ACE CF controller.

   When the Windows OS formats the CF card, it uses a formula to determine what it believes to be an optimal sectors-per-cluster value, based on the size of the CF partition and other factors. This can lead some Windows OS versions to specify one sector (512 bytes) per cluster in some CF configurations. For example, this situation is known to occur when formatting 32 MB CF cards with Windows 2000 and Windows XP. Disk formatting utilities (such as mkdosfs, available from http://www1.mager.org/mkdosfs) can be used to avoid this situation.

2. **FAT12 or FAT16 Format**

   The System ACE CF controller does not recognize the FAT32 file system. It was designed to recognize only the FAT12 and FAT16 formats.

3. **Reserved Sectors**

   Reserved sectors are the sectors in the reserved region of the volume starting at the first sector of the volume. The System ACE CF controller can only read a CF card that is formatted with *one reserved sector* in the Partition Boot Record.

Specifying Sectors-per-Cluster and FAT Version

To correct the first two of these formatting issues, the CF card should always be formatted with a sectors-per-cluster size greater than 1 (UnitSize greater than 512), and the FAT file system version should be specified. This can be done using the **`format`** command with the **`/fs:`** and **`/a:`** options in this syntax:

```
format <volume> [/fs:<FileSystem>]
[/a:<UnitSize>]
```

For example:

```
format D: /FS:FAT /A:1024
```

Controlling the Number of Reserved Sectors

Windows 2000, Windows NT, and Windows 98 default to *one reserved sector* when formatting. Therefore, formatting the CF card using these Windows operating systems is not problematical in this regard.

In Windows XP, however, the DOS `format` command automatically formats the CF card with *from two to eight* reserved sectors, depending on the density of the CF card.

Because the DOS format command does not allow specification of the number of reserved sectors, an alternate disk formatting utility (such as mkdosfs, available from http://www1.mager.org/mkdosfs) must be used. When the CF card is correctly formatted, Windows XP can be used to perform normal file access (read/write) operations without causing any additional problems.

Microprocessor Interface (MPU)

The MPU Interface provides a useful means of monitoring the status of and controlling the System ACE CF controller, as well as CompactFlash card READ / WRITE data. The MPU is not required for normal operation, but when used, it provides numerous capabilities. This interface enables communication between an MPU device and a CompactFlash module and the FPGA target system.

The MPU interface is composed of a set of registers that provide a means for communicating with CompactFlash control logic, configuration control logic, and other resources in the System ACE CF controller. Specifically, this interface can be used to read the identity of a CompactFlash device and read/write sectors from or to a CompactFlash device.

The MPU interface can also be used to control configuration flow. The MPU interface enables monitoring of System ACE CF controller configuration status and error conditions. The MPU interface can be used to delay configuration, start configuration, determine the source of configuration (CompactFlash or MPU), control the bitstream version, reset the device, etc.

Two important issues should be understood when using the microprocessor port:

- For the System ACE CF controller to be properly synchronized, the device driving the MPU interface must be synchronized to the CLK signal
- The MPU must comply with System ACE timing requirements

This general-purpose microprocessor interface can update the CompactFlash, read the ACE status, or obtain direct access to the JTAG configuration ports using the ACE Microprocessor commands. This interface supports either 8-bit (default) or 16-bit data transfers. The bus width can be configured dynamically.

All communications between the System ACE CF controller and a host microprocessor involve transfer of data to or from ACE registers. There are 128 addressable registers in 8-bit mode and 64 addressable registers in 16-bit mode. For easy selection of a new configuration from CompactFlash data, the MPU interface allows for easy reconfiguration of an FPGA chain or capability.

When using the MPU interface as the configuration source, the CFGTCK output for the System ACE CF controller device is derived from the CLK input to the System ACE CF controller (supplied by the MPU), and the operating frequency of the CFGTCK is the same as CLK.

- The minimum clock operating frequency is 0 MHz.
- The maximum clock operating frequency is either 33 MHz *or* the maximum JTAG TCK clock speed dictated by the devices in the JTAG chain and/or the board design. The lowest of these values should be used.

The following sections describe supported operations when using the MPU interface.

MPU Port Signal Description

MPU interface port signals are described in Table 6.

Table 6: **MPU Interface Port Signal Description**

Name	Width	Direction	Active	Description
MPA	7	In	N/A	Synchronous address inputs. The internal address register is loaded by MPA by a combination of the rising edge of CLK and $\overline{\text{MPCE}}$ LOW.
MPD	16	In/Out	N/A	Synchronous data input/output pins. Both the data input and output path are registered and triggered by the rising edge of CLK.
$\overline{\text{MPCE}}$	1	In	LOW	Synchronous active LOW chip enable. $\overline{\text{MPCE}}$ LOW is used to enable the MPU interface. $\overline{\text{MPCE}}$ LOW is also used in conjunction with $\overline{\text{MPOE}}$ LOW to enable the MPD output.

Table 6: **MPU Interface Port Signal Description *(Continued)***

Name	Width	Direction	Active	Description
$\overline{\text{MPWE}}$	1	In	LOW	Synchronous active LOW write enable. A high-to-low-to-high transition must occur on $\overline{\text{MPWE}}$ in three consecutive clock cycles in order for the write to take place.During a valid write cycle, $\overline{\text{MPCE}}$ must be LOW and MPD must be valid during the clock cycle that $\overline{\text{MPWE}}$.
$\overline{\text{MPOE}}$	1	In	LOW	Asynchronous active LOW output enable. Both $\overline{\text{MPOE}}$ and $\overline{\text{MPCE}}$ must be LOW to read from the MPU interface. When either $\overline{\text{MPOE}}$ or $\overline{\text{MPCE}}$ is HIGH, the MPD pins of the System ACE CF controller are in a high-impedance state.
MPBRDY	1	Out	HIGH	Synchronous active HIGH buffer ready output. During data buffer read mode MPBRDY is HIGH when the data in the DATABUF buffer is valid. During data buffer write mode MPBRDY is HIGH when data can be written to the DATABUF buffer.
MPIRQ	1	Out	HIGH	Synchronous active HIGH interrupt request output. MPIRQ HIGH indicates that an interrupt condition has occurred in the MPU interface. All interrupt conditions must be manually cleared before MPIRQ will go LOW. MPIRQ is always LOW when interrupts are disabled.

MPU Timing Description

This section contains timing diagrams for the MPU interface. Parameters used in the timing diagrams are described in Table 7.

Table 7: **MPU Interface Timing Parameters**

Symbol	Parameter	Min	Max	Units
tSA	Address setup time	4	--	ns
tSCE	Chip enable setup time	4	--	ns
tSWE	Write enable setup time	12	--	ns
tSOE	Output enable setup time	12	--	ns
tSD	Data setup time	4	--	ns
tDD	Clock HIGH to valid data	--	22	ns
tDOE	Chip/Output enable LOW to valid data	--	13	ns
tDBRDY	Clock HIGH to buffer ready valid	--	22	ns
tH	Hold time	4	--	ns

Single Register Read Cycle

The single register read cycle is shown in Figure 9, page 13. A single register read is accomplished by asserting a valid address (MPA), asserting the chip enable ($\overline{\text{MPCE}}$ = LOW) and de-asserting the write enable ($\overline{\text{MPWE}}$ = HIGH) during the first clock cycle (Cycle 0). These signals should hold these values at least until the rising edge of the fourth clock cycle (Cycle 3).

The output enable signal should be asserted ($\overline{\text{MPOE}}$ = LOW) during the third clock cycle (Cycle 2). Register data associated with the specified address appears on the MPD bus two clock cycles after the falling edge of $\overline{\text{MPCE}}$ during the assertion of $\overline{\text{MPCE}}$. The register read cycle is then completed by de-asserting the output enable during the fourth clock cycle (Cycle 3).

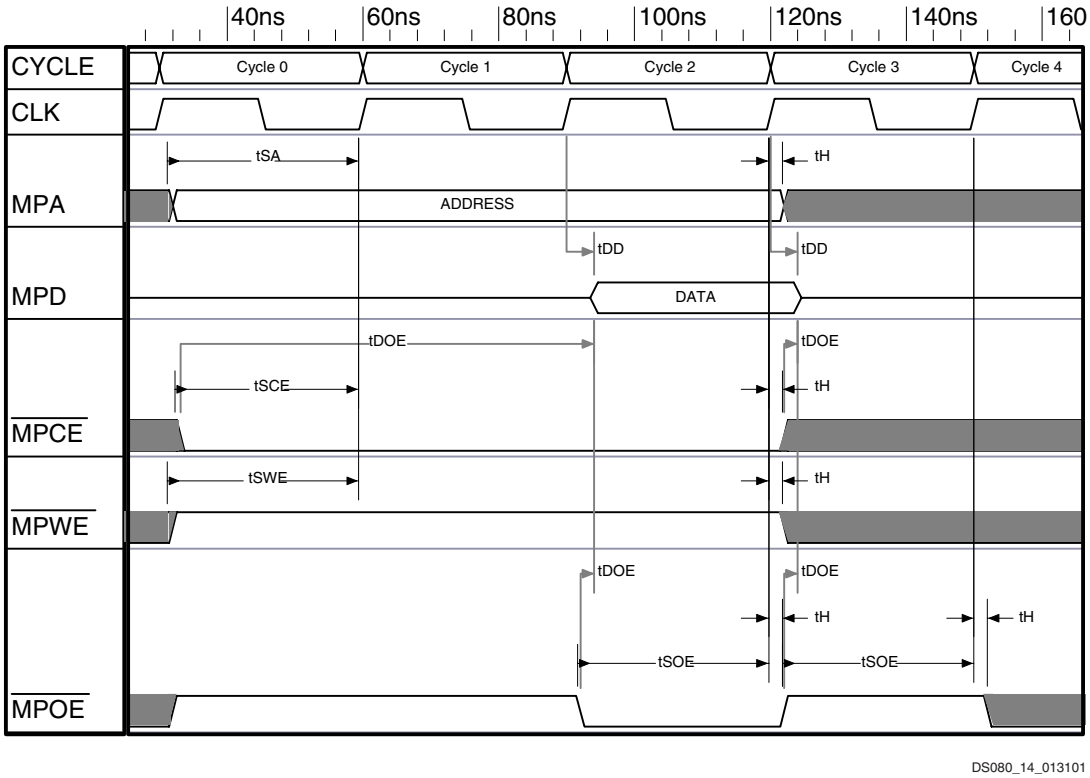

Figure 9: **Single Read From an ACE Register**

Single Register Write Cycle

The single register write cycle is shown in Figure 10. A single register write is accomplished by asserting a valid address (MPA), asserting the chip enable ($\overline{\text{MPCE}}$ = LOW) and de-asserting the output enable ($\overline{\text{MPOE}}$ = HIGH) during the first clock cycle (Cycle 0). These signals should hold these values at least until the rising edge of the third clock cycle (Cycle 2).

The write enable signal should be asserted ($\overline{\text{MPWE}}$ = LOW) during the second clock cycle (Cycle 1). Data (MPD) to be written to the specified address should be asserted during the same clock cycle that the write enable is asserted (Cycle 1). The register write cycle is then completed by de-asserting the write enable during the third clock cycle (Cycle 2).

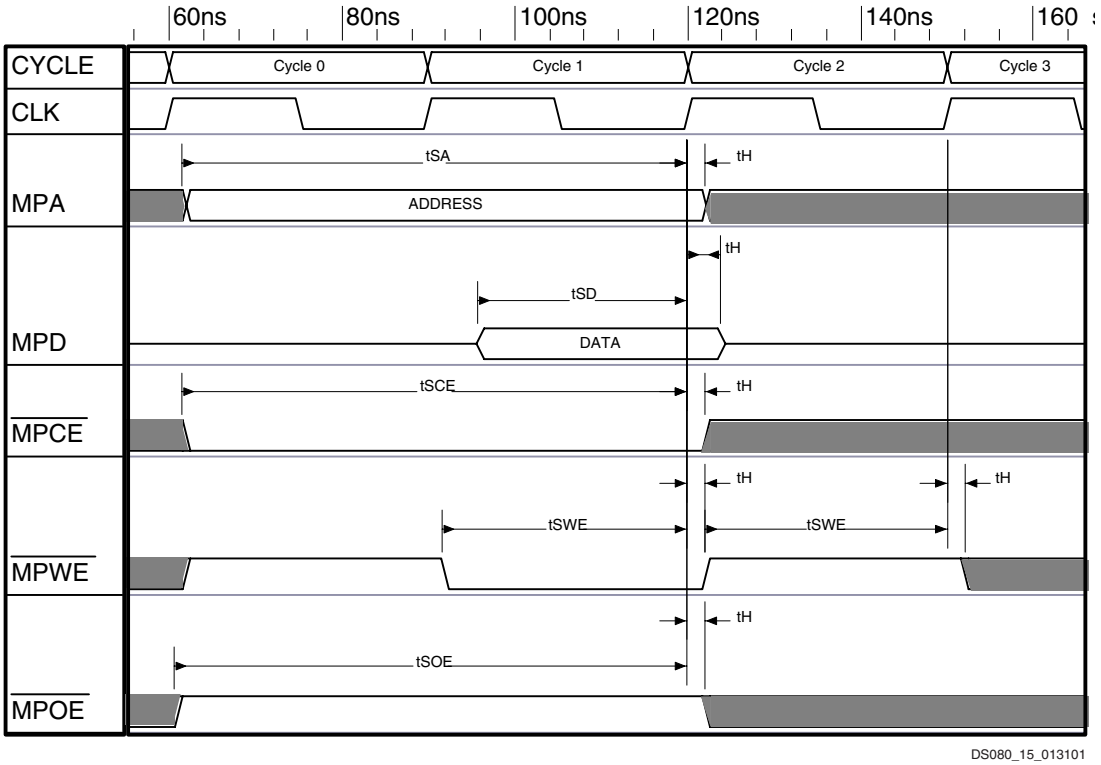

Figure 10: **Single WORD Write to an ACE Register**

Multiple Register Read Timing

The minimum timing requirements for sequential register read cycles are shown in Figure 11. Sequential read cycles are identical to single read cycles, except that the chip enable ($\overline{\text{MPCE}}$) and write enable ($\overline{\text{MPWE}}$) signals do not need to be de-asserted between read cycles.

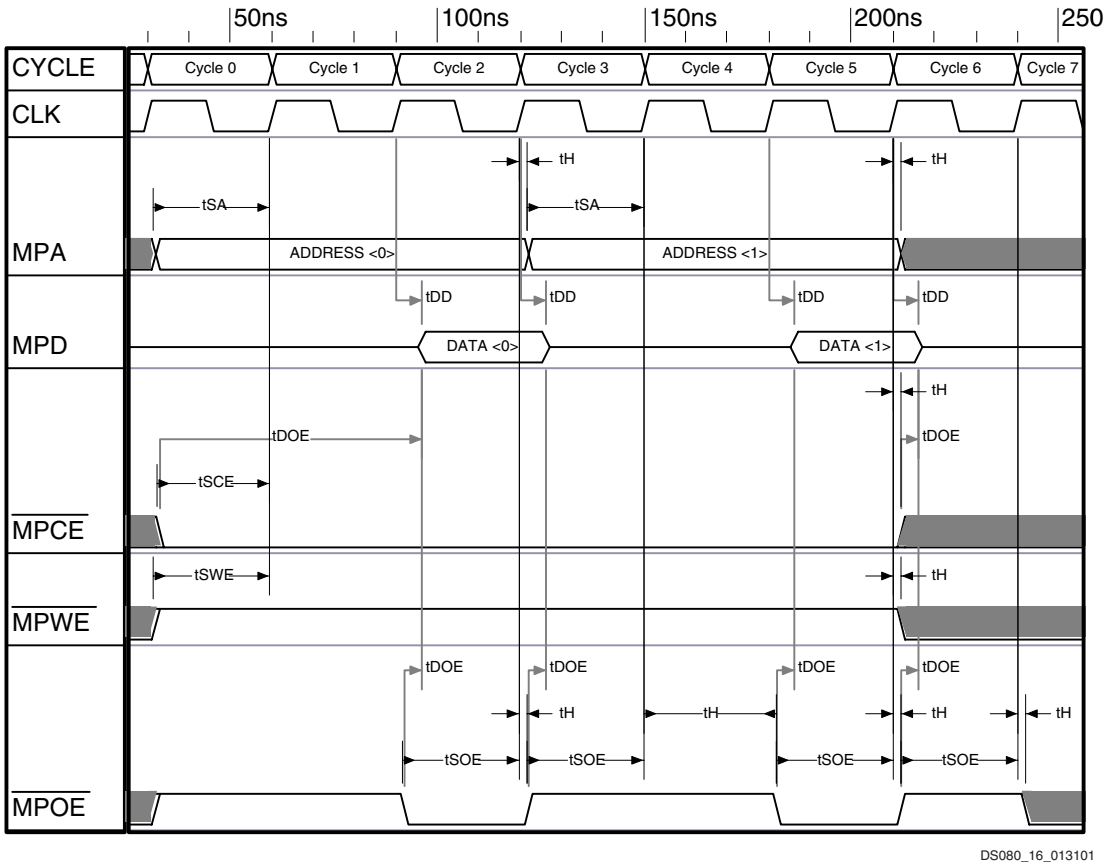

Figure 11: **Multiple WORD Reads From ACE Register(s)**

Multiple Register Write Timing

The minimum timing requirements for sequential write cycles are shown in Figure 12. Sequential write cycles are identical to single write cycles except that the chip enable ($\overline{\text{MPCE}}$) and output enable ($\overline{\text{MPOE}}$) signals do not need to be de-asserted between write cycles.

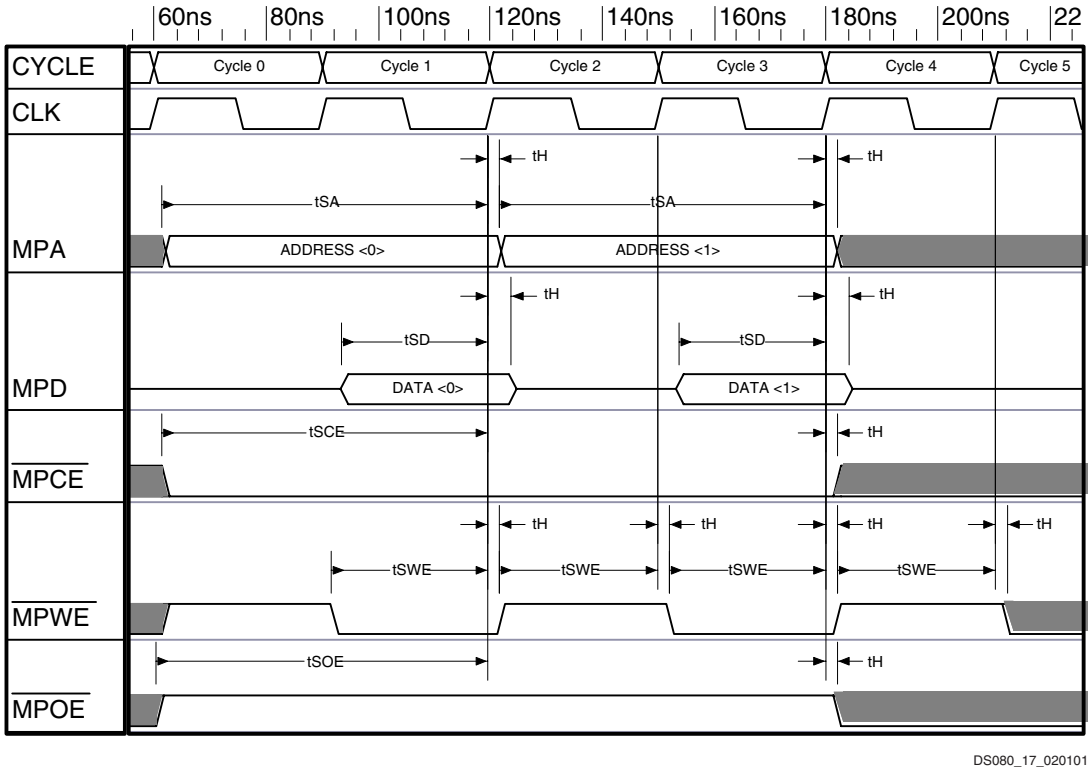

Figure 12: **Multiple WORD Writes to ACE Register(s)**

Data Buffer Ready Timing

The data buffer ready (MPBRDY) signal indicates whether the data buffer is ready to accept new data during a write cycle or whether the data buffer contains valid data to be read during a read cycle. The data buffer itself is sixteen words deep, where each word is 16 bits wide.

The data buffer mode transfer direction is identified by the state of the DATABUFMODE bit in the STATUSREG register:

- DATABUFMODE = 0 indicates data buffer read mode
- DATABUFMODE = 1 indicates data buffer write mode

The data buffer mode depends on the type of command that was issued to the System ACE CF controller. If an IdentifyMemCard or ReadMemCard command was issued, then the data buffer remains in read mode until the command is finished executing (i.e., all sector data has been read from the buffer). If a WriteMemCard command was issued, then the data buffer remains in write mode until the command is finished executing (i.e., all sector data has been written to the buffer).

Data Buffer Read Cycle Ready Timing

When the data buffer is in read mode and the last data word is read from the buffer, the data buffer ready signal will go inactive (MPBRDY = LOW) two clock cycles following the last clock cycle that the output enable is active ($\overline{\text{MPOE}}$ = LOW). Any attempt to read data out of an "empty" data buffer ($\overline{\text{MPOE}}$ = LOW while MPBRDY = LOW) results in invalid data. Valid and invalid data buffer reads are shown in Figure 13.

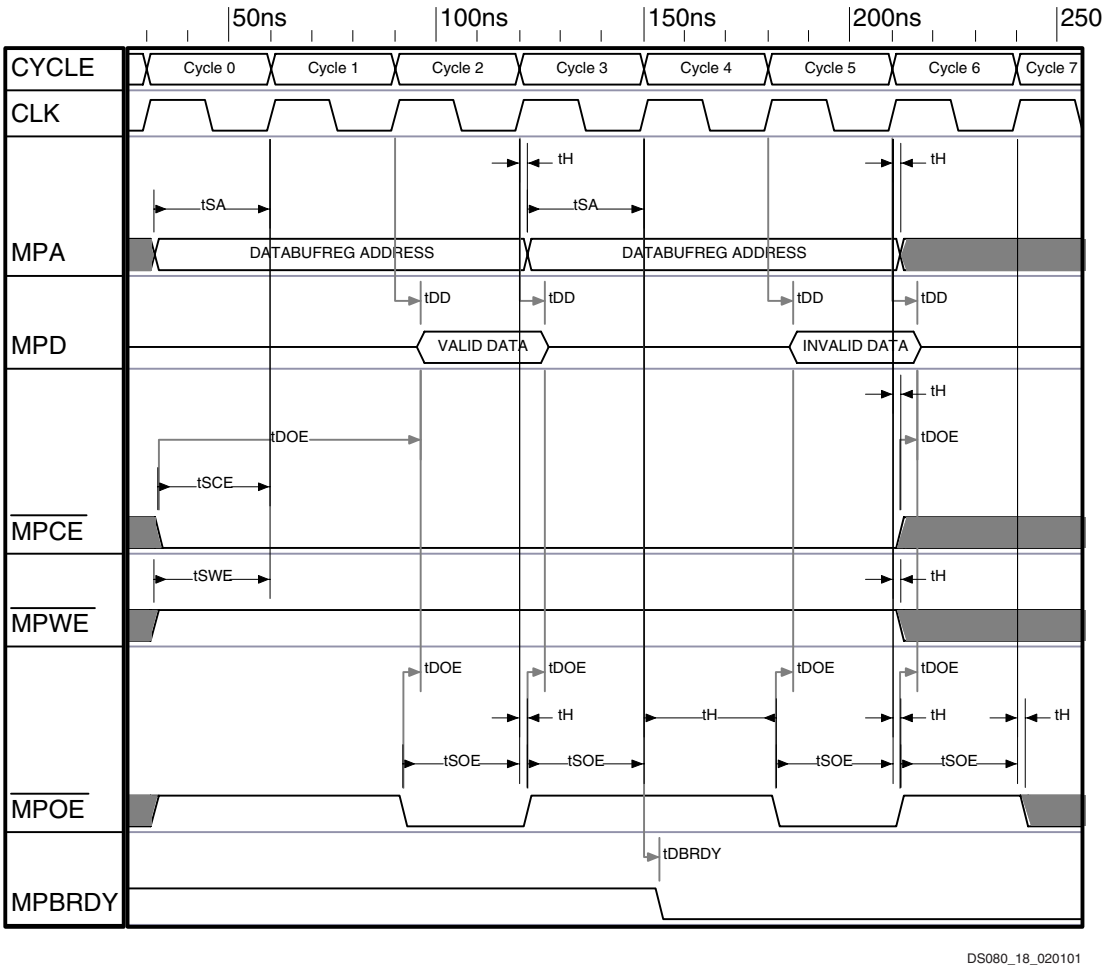

Figure 13: **Valid and Invalid Reads From DATABUFREG Data Buffer**

Data Buffer Write Cycle Ready Timing

When the data buffer is in write mode and the last available space for a data word has been filled, the data buffer ready signal will go inactive (MPBRDY = LOW) two clock cycles following the last clock cycle that the write enable is active ($\overline{\text{MPWE}}$ = LOW). Any attempt to write data to a "full" data buffer ($\overline{\text{MPWE}}$ = LOW while MPBRDY = LOW) does not result in a successful write to the buffer. Valid and invalid data buffer writes are shown in Figure 14.

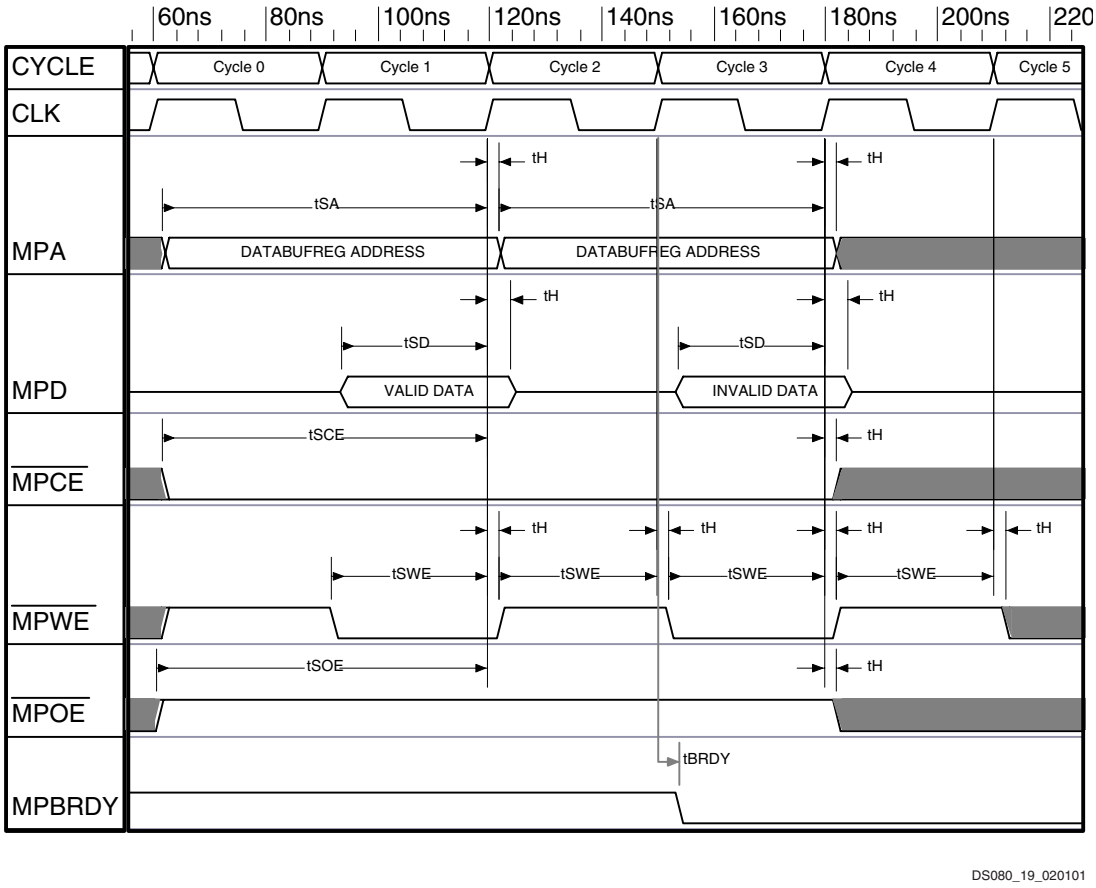

Figure 14: **Valid and Invalid Writes to DATABUFREG Data Buffer**

Interrupt Timing

The interrupt request and clearing cycles are shown in Figure 15. In Figure 15, the interrupt request (MPIRQ = HIGH) occurs sometime before Cycle 0. The interrupt request is cleared by performing a single MPU write cycle that sets RESETIRQ = 1 (bit number 11) in the CONTROLREG(15:0) register (BYTE address 0x19 or WORD address 0x0C).

The MPU interrupt request line (MPIRQ) remains active HIGH until the RESETIRQ bit is set. The MPIRQ line becomes inactive LOW two cycles after the completion of the RESETIRQ write cycle (Cycle 4). For subsequent MPU interrupt requests to be enabled, the RESETIRQ bit must be reset and one of the three IRQ enable bits (DATABUFRDYIRQ, ERRORIRQ, and/or CFGDONEIRQ) in the CONTROLREG register should be set.

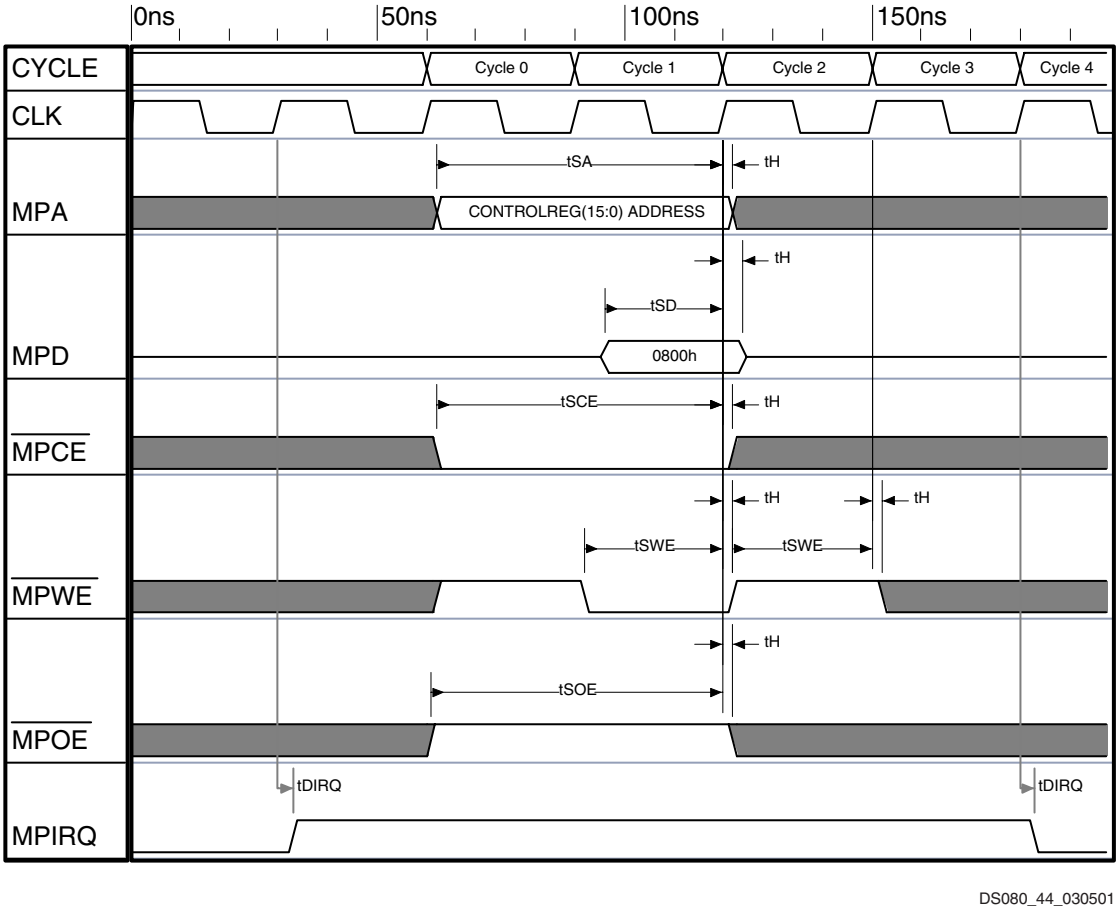

Figure 15: **Interrupt Request Timing**

Register Specification

The BYTE-mode register space of the MPU interface is shown in Table 8.

Table 8: **Register Address Map (BYTE Mode Addresses)**

BYTE Address (MPA [6:0])	Register Name	Width	Mode	Description
0x00	BUSMODEREG	1	RW	Used to control the data bus access mode (8-bit BYTE mode or 16-bit WORD mode)
0x01	BUSMODEREG	1	RW	
0x02	--	--	--	Reserved
0x03	--	--	--	Reserved
0x04	STATUSREG(7:0)	8	R	Used to monitor System ACE CF controller status
0x05	STATUSREG(15:8)	8	R	
0x06	STATUSREG(23:16)	8	R	
0x07	STATUSREG(31:24)	8	R	
0x08	ERRORREG(7:0)	8	R	Used to indicate any existing error condition
0x09	ERRORREG(15:8)	8	R	
0x0A	ERRORREG(23:16)	8	R	
0x0B	ERRORREG(31:24)	8	R	
0x0C	CFGLBAREG(7:0)	8	R	Logical block address used by the Configuration Controller during CompactFlash data transfers
0x0D	CFGLBAREG(15:8)	8	R	
0x0E	CFGLBAREG(23:16)	8	R	
0x0F	CFGLBAREG(27:24)	4	R	
0x10	MPULBAREG(7:0)	8	RW	Logical block address used by the MPU interface during CompactFlash data transfers
0x11	MPULBAREG(15:8)	8	RW	
0x12	MPULBAREG(23:16)	8	RW	
0x13	MPULBAREG(27:24)	4	RW	
0x14	SECCNTCMDREG(7:0)	8	RW	Sector count and CompactFlash command register
0x15	SECCNTCMDREG(15:8)	8	RW	
0x16	VERSIONREG(7:0)	8	R	Version register
0x17	VERSIONREG(15:8)	8	R	
0x18	CONTROLREG(7:0)	8	RW	Used to control System ACE CF controller operations
0x19	CONTROLREG(15:8)	8	RW	
0x1A	CONTROLREG(23:16)	8	RW	
0x1B	CONTROLREG(31:24)	8	RW	
0x1C	FATSTATREG(7:0)	8	R	Contains information about the FAT table of the first valid partition found in the CompactFlash device.
0x1D	FATSTATREG(15:8)	8	R	
0x1E through 0x3F	--	--	--	Reserved
Even Values 0x40 through 0x5E	DATABUFREG(7:0)	8	RW	Address range that provides read and write access to the data buffer.
Odd Values 0x41 through 0x5F	DATABUFREG(15:8)	8	RW	

The 16-bit WORD mode register space of the MPU interface is shown in Table 9.

Table 9: **Register Address Map (WORD Mode Addresses)**

WORD Address (MPA [6:1])	Register Name	Width	Mode	Description
0x00	BUSMODEREG	1	RW	Used to control the data bus access mode (8-bit BYTE mode or 16-bit WORD mode)
0x01	--	--	--	Reserved
0x02	STATUSREG(15:0)	16	R	Used to monitor System ACE CF controller status
0x03	STATUSREG(31:16)	16	R	
0x04	ERRORREG(15:0)	16	R	Used to indicate any existing error condition
0x05	ERRORREG(31:16)	16	R	
0x06	CFGLBAREG(15:0)	16	R	Logical block address used by the Configuration Controller during CompactFlash data transfers
0x07	CFGLBAREG(27:16)	12	R	
0x08	MPULBAREG(15:0)	16	RW	Logical block address used by the MPU interface during CompactFlash data transfers
0x09	MPULBAREG(27:16)	12	RW	
0x0A	SECCNTCMDREG(15:0)	16	RW	Sector count and CompactFlash command register
0x0B	VERSIONREG(15:0)	16	R	Version register
0x0C	CONTROLREG(15:0)	16	RW	Used to control System ACE CF controller operations
0x0D	CONTROLREG(31:16)	16	RW	
0x0E	FATSTATREG(15:0)	16	R	Contains information about the FAT table of the first valid partition found in the CompactFlash device.
0x0F through 0x1F	--	--	--	Reserved
0x20 through 0x2F	DATABUFREG(15:0)	16	RW	Address range that provides read and write access to the data buffer.

BUSMODEREG Register (BYTE address 00h-01h, WORD address 00h)

The BUSMODEREG register is used to control the mode of the MPU address and data bus. The single-bit BUSMODEREG register is aliased across two BYTE addresses (0x00-0x01) and one 16-bit WORD address (0x0). This register aliasing ensures that the MPU bus mode can be set regardless of the mode of the microprocessor that is communicating with the System ACE CF controller. Table 10 provides a description of the BUSMODEREG register bits.

Table 10: **BUSMODEREG Register Bit Descriptions**

Bit	Name	Description
0	BUSMODE0	The BUSMODE bits are used to select the width of the data bus portion of the Microprocessor bus (default is 0): • When 0, the MPU interface is in BYTE mode (all MPU address bits are used, but only MPU data bits 7:0 are used). • When 1, the MPU interface is in WORD mode (all MPU data bits are used, but only MPU address bits 6:1 are used).
1	--	Reserved
2	--	Reserved
3	--	Reserved
4	--	Reserved
5	--	Reserved
6	--	Reserved
7	--	Reserved

STATUSREG Register (BYTE address 04h-07h, WORD address 02h-03h)

The STATUSREG register allows a microprocessor to monitor important System ACE CF controller operating modes. This is also the register that is read upon receiving an IRQ request in order to identify an interrupt source. Table 11 provides a description of the STATUSREG register bits.

Table 11: **STATUSREG Register Bit Descriptions**

Bit	Name	Description
0	CFGLOCK	Configuration controller lock status: • 0 means that the configuration controller does not currently have a lock on the CompactFlash controller resource • 1 means that the configuration controller has successfully locked the CompactFlash controller resource
1	MPULOCK	MPU interface lock status: • 0 means that the MPU interface does not currently have a lock on the CompactFlash controller resource • 1 means that the MPU interface has successfully locked the CompactFlash controller resource
2	CFGERROR	Configuration Controller error status: • 0 means that no Configuration Controller error condition exists • 1 means that an error has occurred in the Configuration Controller (check the ERRORREG register for more information)
3	CFCERROR	CompactFlash Controller error status: • 0 means that no CompactFlash Controller error condition exists • 1 means that an error has occurred in the CompactFlash controller (check the ERRORREG register for more information)

Table 11: **STATUSREG Register Bit Descriptions *(Continued)***

Bit	Name	Description
4	CFDETECT	CompactFlash detect flag: • 0 means that no CompactFlash device is connected to the System ACE CF controller • 1 means that a CompactFlash is connected to the System ACE CF controller
5	DATABUFRDY	Data buffer ready status: • 0 means that the data buffer is not ready for data transfer • 1 means that the data buffer is ready for data to be transferred out of the buffer when reading from the CompactFlash controller or into the buffer when writing to the CompactFlash or Configuration controller
6	DATABUFMODE	Data buffer mode status: • 0 means read-only mode • 1 means write-only mode
7	CFGDONE	Configuration DONE status: • 0 means that the configuration process has not completed • 1 means that the entire System ACE CF controller configuration file has been executed and configuration of all devices in the target Boundary-Scan chain is complete
8	RDYFORCFCMD	Ready for CompactFlash controller command: • 0 means not ready for command • 1 means ready for command
9	CFGMODEPIN	Configuration mode pin (note that this can be overridden by the CFGMODE bit in the CONTROLREG register): • 1 means automatically start the configuration process immediately after System ACE CF controller Reset • 0 means wait for CFGSTART bit in CONTROLREG before starting the configuration process
10	--	Reserved
11	--	Reserved
12	--	Reserved
13	CFGADDRPIN0	Configuration address pins that are used as an offset into the system configuration file in the CompactFlash device used to locate the System ACE CF controller configuration data file (note that these pins can be overridden by the contents of the CFGADDRBIT[2:0] of the CONTROLREG register)
14	CFGADDRPIN1	
15	CFGADDRPIN2	
16	--	Reserved
17	CFBSY	CompactFlash BUSY bit (reflects the state of the BSY bit in the status register of the CompactFlash device): • 0 means that the CompactFlash device is not busy • 1 means that the CompactFlash command register and data buffer cannot be accessed; Bits 18-23 of the STATUSREG register are not valid when this bit is set to 1
18	CFRDY	CompactFlash ready for operation bit (reflects the state of the RDY bit in the status register of the CompactFlash device): • 0 means the CompactFlash device is NOT ready to accept commands • 1 means CompactFlash device is ready to accept commands
19	CFDWF	CompactFlash data write fault bit (reflects the state of the DWF bit in the status register of the CompactFlash device): • 0 means that a write fault has NOT occurred • 1 means that a write fault has occurred

Table 11: **STATUSREG Register Bit Descriptions *(Continued)***

Bit	Name	Description
20	CFDSC	CompactFlash ready bit (reflects the state of the DSC bit in the status register of the CompactFlash device): • 0 means that the CompactFlash device is NOT ready • 1 means that the CompactFlash device is ready
21	CFDRQ	CompactFlash data request bit (reflects the state of the DRQ bit in the status register of the CompactFlash device): • 0 means that no data is ready to be transferred to/from the data buffer of the CompactFlash device • 1 means that information be transferred to/from the data buffer of the CompactFlash device
22	CFCORR	CompactFlash correctable error bit (reflects the state of the CORR bit in the status register of the CompactFlash device): • 0 means that a correctable data error was NOT encountered • 1 means that a correctable data error was encountered (check the ERRORREG register for more information)
23	CFERR	CompactFlash ERROR bit (reflects the state of the ERR bit in the status register of the CompactFlash device): • 0 means that no error has occurred during the execution of the previous command • 1 means that the previous command has ended in some type of error (check the ERRORREG register for more information)
24	--	Reserved
25	--	Reserved
26	--	Reserved
27	--	Reserved
28	--	Reserved
29	--	Reserved
30	--	Reserved
31	--	Reserved

ERRORREG Register (BYTE address 08h-0Bh, WORD address 04h-05h)

The ERRORREG register identifies specific information on any error conditions that might exist in the System ACE CF controller. Table 12 provides a description of the ERRORREG register bits.

Table 12: **ERRORREG Register Bit Descriptions**

Bit	Name	Description
0	CARDRESETERR	CompactFlash card reset error: • 0 means no error • 1 means that the CompactFlash card has failed to reset properly before a time-out condition occurred
1	CARDRDYERR	CompactFlash card ready error: • 0 means no error • 1 means that the CompactFlash card has failed to become properly ready for commands before a time-out condition occurred
2	CARDREADERR	CompactFlash card read error: • 0 means no error • 1 means that a CompactFlash data read command (either ReadMemCardData or IdentifyMemCard) has failed

Table 12: **ERRORREG Register Bit Descriptions *(Continued)***

Bit	Name	Description
3	CARDWRITEERR	CompactFlash card write error: • 0 means no error • 1 means that a CompactFlash data write command (WriteMemCardData) has failed
4	SECTORRDYERR	CompactFlash sector ready: • 0 means no error • 1 means that a sector has failed to become properly valid during a CompactFlash read or write command before a time-out condition occurred
5	CFGADDRERR	CFGADDR error: • 0 means no error • 1 means that the CFGADDR (i.e., the CFGADDR(15:0) register or CFGADDR(1:0) pins, depending on the state of the FORCECFGADDR bit in the CONTROLREG register) does not correspond to a valid location in the CompactFlash
6	CFGFAILED	Configuration failure error: • 0 means no error • 1 means that configuration of one or more devices in the target Boundary-Scan chain has failed
7	CFGREADERR	Configuration read error: • 0 means no error • 1 means that an error occurred while reading configuration information from CompactFlash
8	CFGINSTRERR	Configuration instruction error: • 0 means no error • 1 means that an invalid instruction was encountered during configuration
9	CFGINITERR	Configuration INIT monitor error: • 0 means no error • 1 means that the $\overline{\text{CFGINIT}}$ pin did not go HIGH within 500 ms of the start of configuration
10	--	Reserved
11	CFBBK	CompactFlash bad block error (reflects the state of the BBK bit in the error register of the CompactFlash device): • 0 means no error • 1 means that a bad block has been detected
12	CFUNC	CompactFlash uncorrectable error (reflects the state of the UNC bit in the error register of the CompactFlash device): • 0 means no error • 1 means that an uncorrectable error has been encountered
13	CFIDNF	CompactFlash ID not found error (reflects the state of the IDNF bit in the error register of the CompactFlash device): • 0 means no error • 1 means that the requested sector ID is in error or cannot be found
14	CFABORT	CompactFlash command abort error (reflects the state of the ABRT bit in the error register of the CompactFlash device): • 0 means no error • 1 means that the command has been aborted because of a CompactFlash status condition (i.e., Not Ready, Write Fault) or when an invalid command has been issued