Intel® ICH6 I/O Controller Hub 6 R (ICH6R) Advance Host Controller Interface (AHCI)

Programmer's Reference Manual (PRM)

June 2004

Document Number: 301444-001

Contents

Figures

Tables

Revision History

Revision Number	Description	Revision Date
1.0	Public release	June 2004

§

1 About This Document

This document was prepared to assist BIOS software providers and Operating System (OS) providers in supporting the Intel® I/O Controller Hub 6 (ICH6) SATA AHCI (Serial ATA Advanced Host Controller Interface) feature set and programming interface. This document also describes functions that the BIOS and the OS shall perform in order to ensure correct and reliable operation of the platform. This document will be supplemented from time to time with specification updates. The specification updates contain information relating to the latest programming changes. Check with your Intel representative for availability of specification updates.

1.1 General Requirements

It is assumed that the reader has a working knowledge of Serial ATA (SATA)/AHCI architecture. Also, the reader should have an understanding of BIOS (including ACPI) and device driver development for the target operating systems.

1.2 Conventions

1.2.1 Register Access

This document uses the following notation as related to register access: RegOffset.BitOffset.

Where:

- RegOffset specifies the name of the register to be accessed (either in I/O or PCI configuration space)
- BitOffset specifies the name of a bit contained within RegOffset that is to be accessed.

Example (Uses the Class Code register defined in the table below):

Assume the following standard PCI configuration register (Class Code) where CC.SCC refers to the Sub Class Code (SCC - bits 0:7) implemented within the Class Code (CC – offset 0Ah) register in the PCI Configuration space.

Offset 0Ah - Class Code Register (CC)

Bits	Type	Reset	Description
15:08	RO	01h	**Base Class Code (BCC):** Indicates that this is a mass storage device.
07:00	RO	01h	**Sub Class Code (SCC):** Indicates that this is an IDE device.

1.2.2 Keywords

- **Mandatory** – A keyword indicating items to be implemented as defined by this document.
- **System Software** – A keyword that refers to both BIOS and operating system software unless specifically stated otherwise.
- **Shall** – A keyword indicating a mandatory requirement. Equivalent to the term “must”.
- **Should** – A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “it is recommended”.
- **May** – A keyword indicating flexibility of choice with no implied preference.

§

2 Unsupported AHCI Features

The Intel ICH6 SATA implementation does not implement the following *optional* AHCI capabilities:

- Cold Presence Detect
- Port Selector
- Port Multiplier FIS-based switching
- Non-zero DMA Offsets
- Enclosure Management
- Gen-2 Speeds

§

3 Overview

The ICH6 SATA HBA (Host Bus Adapter) is a “hybrid” device. It is considered a hybrid device because it supports both a parallel ATA (P-ATA) programming interface as well as a new programming interface as described in the SATA AHCI (Advanced Host Controller Interface) specification. Conceptually, the ICH6 SATA HBA supports three configurations:

1. **Compatible** – intended for operating systems that do not comprehend SATA, AHCI or Native IDE mode. The Compatible configuration is described in section *4.2 Compatible Configuration.*
2. **Enhanced Non-AHCI** – intended for operating systems that do not comprehend SATA, AHCI but do comprehend Native IDE mode. The Enhanced Non-AHCI configuration is described in section *4.3 Enhanced Non-AHCI Configuration*
3. **Enhanced AHCI** – intended for operating systems that do comprehend SATA and AHCI. The Enhanced AHCI configuration is described in section *4.4 Enhanced AHCI Configuration.*

The SATA HBA can function independently of, or in conjunction with the parallel P-ATA host controller. The ICH6 can support a maximum of six ATA devices: two P-ATA devices plus four SATA devices. While ICH6 **does not** implement a **physical**, secondary P-ATA channel, the physical, primary channel can be viewed, by system software, as a logical secondary channel. This is applicable only when the SATA AHCI is in the Combined configuration. Combined configuration is discussed in subsequent sections of this document.

§

4 Theory of Operation

This section describes the proper usage and programming of the SATA HBA by BIOS and the OS when the host controller is operating in either Compatible or Enhanced configurations.

4.1 Modes

Depending on how it is configured (Compatible or Enhanced Non-AHCI), the ICH6 SATA HBA can also be programmed to operate in one of two sub-modes: Legacy or Native IDE.

4.1.1 Legacy Mode

A host controller (channel) configured for Legacy mode of operation has the following requirements:

- Shall have its Programming Interface (PCI Configuration offset 9h) register set for legacy mode
- Shall interrupt via IRQ14 (primary channel) and IRQ15 (secondary channel)
- Command and control block are accessed at fixed I/O locations:
  - ***Command Block Offset:*** 01F0h for Primary and 0170h for Secondary
  - ***Control Block Offset:***03F4h for Primary and 0374h for Secondary

Note: Legacy mode is only applicable when the SATA HBA is configured for Compatible configuration.

4.1.2 Native IDE Mode

A host controller (channel) configured for Native IDE mode of operation has the following requirements:

- Shall have its Programming Interface (PCI Configuration offset 9h) register set for native mode.
- Shall interrupt via the interrupt pin specified by the ***INTR*** register located in PCI configuration space offset 3Ch.
- Command and control blocks are accessed via I/O space specified by the following BARs located by the following PCI configuration offsets:
  - Offset 10h - Primary Command Block Base Address
  - Offset 14h – Primary Control Block Base Address
  - Offset 18h – Secondary Command Block Base Address
  - Offset 1Ch - Secondary Control Block Base Address

Note: Native IDE mode is only applicable when the SATA HBA is configured for the Enhanced Non-AHCI configuration (refer to section *4.3 Enhanced Non-AHCI Configuration*), one of the Compatible configurations (refer to sections *4.2.4 Compatible Configuration (Non-Combined) - Option* 1 and *4.2.5 Compatible Configuration (Non-Combined) - Option* 2) or the Enhanced AHCI configuration (refer to section *4.4 Enhanced AHCI Configuration*); Native IDE mode is not supported when the ICH6 is in the Compatible-Combined configuration (refer to section *4.2.6 Compatible Configuration - Option 3 (Combined)*).

4.2 Compatible Configuration

The Compatible configuration is for the express purpose of **maintaining backward compatibility** with those operating systems that:

- Lack support for Native IDE mode of operation

  Or

- Lack support for the AHCI programming interface.

Note: The Compatible configuration is not operating system selectable; this configuration shall only be selected by the system BIOS during POST**.**

4.2.1 Lack of Native IDE Mode Support

Since the ICH6 supports up to 6 ATA devices (2 P-ATA + 4 SATA), it is possible for an end user to populate a platform with a combination of P-ATA and SATA devices. The fact that the ICH6 supports up to 6 ATA devices is not necessarily an issue for older operating systems; it is the fact that both the P-ATA controller and SATA controller can be configured to consume legacy resources (see Section: *4.1.1 Legacy Mode*) that causes the issues (i.e., the primary and secondary channel interfaces will attempt to share the same resources).

The Compatible configuration solves this dilemma. The Compatible configuration allows certain device configurations to be accessed by system software without the issue of legacy resource conflicts and yet remain backward compatible with operating systems that don’t implement Native IDE mode functionality. It is important to note that the Compatible configuration is limited to supporting a maximum of 4 ATA devices (2 P-ATA + 2 SATA).

4.2.2 Lack of AHCI Programming Interface Support

Because the registers for the new features associated with the AHCI are located in a separate AHCI BAR (e.g., not implemented through the standard primary/secondary command/control or bus master BARs), existing operating system software may not be aware of this and as such will still need to access the SATA HBA through the traditional command/control/bus master registers.

The ICH6 supports three Compatible configuration options. These options are summarized as follows and are discussed in detail in subsequent sections:

- P-ATA devices only (maximum of 2) – Compatible Configuration Option 1
- SATA devices only (maximum of 4) – Compatible Configuration Option 2

- P-ATA (maximum of 2) and SATA devices (maximum of 2) – Compatible Configuration Option 3

Note: Proper support of these options requires that system BIOS provide a setup option (that is subsequently saved in non-volatile memory) that allows the end user to select an option that is appropriate for their particular hardware/operating system configuration. To insure backward compatibility with existing software, BIOS should exercise caution if it is capable of dynamically selecting a configuration.

4.2.3 Additional Register Support

Support of certain Compatible configuration options requires that the ICH6 implement an additional hardware register that is configurable via BIOS. This register is located in the SATA HBA's PCI configuration space at offset 90h and is defined below. The usage model for this register is described in subsequent sections.

4.2.3.1 Offset 90h: MAP – Port Mapping Register

This register is set by BIOS during POST. The exact value programmed into this register is based on a BIOS setup option or as a result of an intelligent device configuration detection algorithm. Modifications of this register by BIOS after POST (e.g., as a result of a docking event) is not recommended, as the underlying operating system software may not comprehend the new device topology and may result in undefined behavior.

Table 1. MAP – Mapping Register

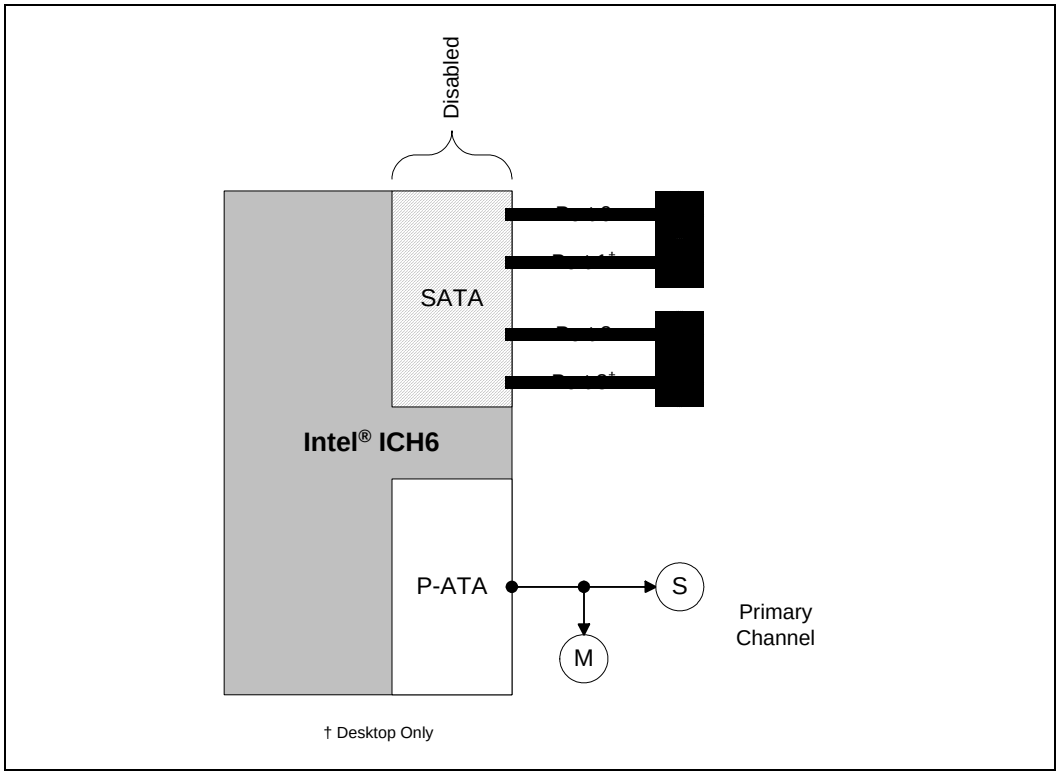
<table>
<tr><th>Bit</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>7</td><td>RW</td><td>0</td><td>Use SATA Class Code (USCC): When set, the SATA class code is reported in the CC.SCC field, and the SATA programming interface is reported in the PI register. When cleared, the value in these fields is determined by the SKU type (RAID/Non-RAID) and BIOS. Proper usage of this bit is described in subsequent sections.</td></tr>
<tr><td>06:02</td><td>RO</td><td>0</td><td>Reserved</td></tr>
<tr><td>01:00</td><td>RW</td><td>00</td><td>Map Value (MV): The value in the bits below indicate the address range the SATA ports responds to, and whether or not the PATA and SATA functions are combined. When in combined mode, the AHCI memory space is not available and AHCI may not be used.
<table>
<tr><th rowspan="2">Bits</th><th rowspan="2">Mode</th><th colspan="2">Primary</th><th colspan="2">Secondary</th></tr>
<tr><th>Master</th><th>Slave</th><th>Master</th><th>Slave</th></tr>
<tr><td>00</td><td>Non-combined</td><td>Port 0</td><td>Port 2</td><td>Port 1†</td><td>Port 3†</td></tr>
<tr><td>01‡</td><td>Combined</td><td colspan="2">PATA</td><td>Port 1</td><td>Port 3</td></tr>
<tr><td>10</td><td>Combined</td><td>Port 0</td><td>Port 2</td><td colspan="2">PATA</td></tr>
<tr><td>11</td><td colspan="5">Reserved</td></tr>
</table>
NOTES:
† Desktop only; ICH6-M does not implement SATA ports 1 and 3
‡ Desktop only; the bit setting of 01b is reserved in ICH6-M</td></tr>
</table>

Caution: Programming the ***MAP.MV*** register with values other than those specified above will result in undefined hardware behavior. Proper programming of the ***MV*** bits is discussed in subsequent sections.

4.2.4 Compatible Configuration (Non-Combined) - Option 1

This option is selected when one or more (maximum of 2) P-ATA devices are to be used. SATA device(s) **may or may not** be attached to the SATA ports, but will not be accessible to software. Figure 1 illustrates this configuration:

Figure 1. Compatible Configuration - Option 1



Note: In the figure above, devices represented by dotted lines may be attached, but are not accessible to software. The ICH6 **does not** implement a secondary P-ATA channel, regardless of its configuration.

Note: This configuration requires no additional programming of the ***Port Mapping*** register, as in this configuration it has no effect on the P-ATA or SATA functions.

To enable this configuration, system BIOS:

1. Shall insure that the SATA ports are not enabled. This is accomplished by writing '0' to bits 3:0 in the ***Port Control and Status (PCS)*** register. Refer to section *5.4.1 Port Enabling/Disabling* for additional details.
2. Shall not program the SATA (Device 31, Function 2) controller's base address registers (Offsets 10h – 24h in PCI configuration space).

3. Shall disable access to the SATA controller's I/O space by programming the command register (PCI configuration, offset 04h, bit 0) with a '0'.
4. Shall disable the SATA function by programming bit 2 (***Serial ATA Disable***) of the ***Function Disable*** register with a '1'. This will insure that the PCI configuration registers associated with the SATA function are not decoded and thus will insure that operating system configuration software does not enumerate and configure the SATA function.

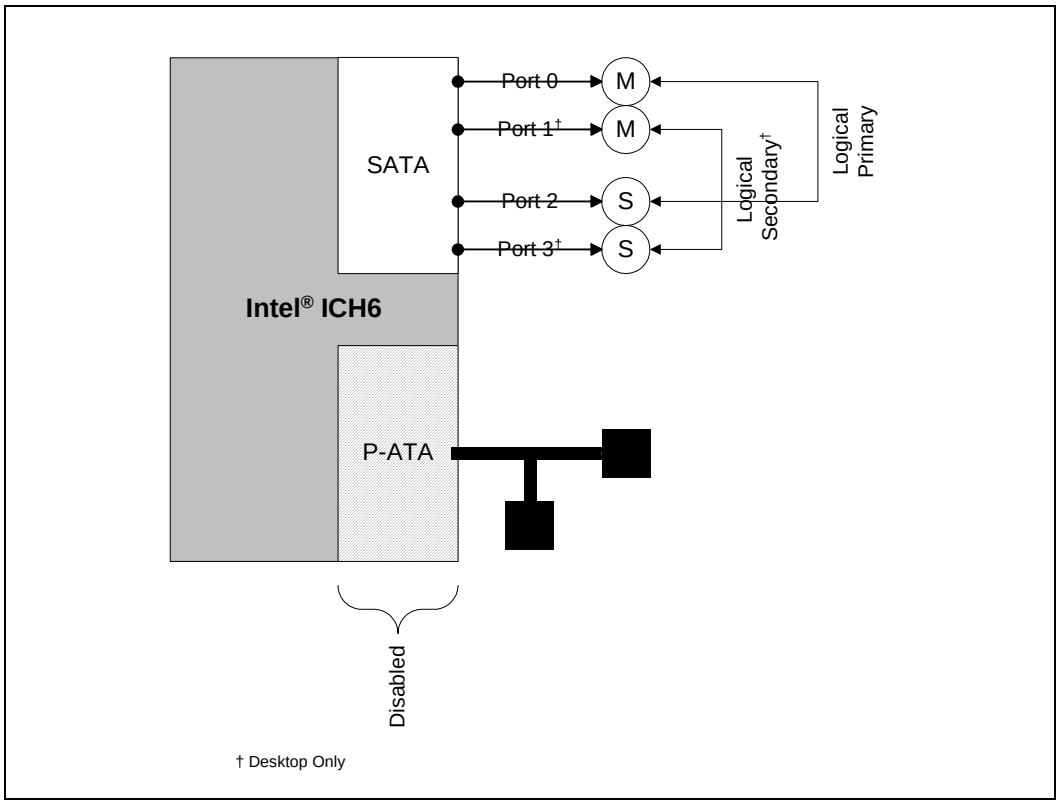
Note: The function disable register is found inside the RCRB (Root Complex Base Address), offset 3418h. The memory mapped base address of the RCRB is found by reading from PCI configuration offset F0h for device 31h, function 00h on PCI Bus 0.

5. Shall program the P-ATA registers appropriately (the exact details are beyond the scope of this document).

4.2.5 Compatible Configuration (Non-Combined) - Option 2

This option is selected when one or more (maximum of 4) SATA devices are to be used. P-ATA device(s) **may or may not** be attached to the P-ATA primary channel, but will not be accessible to software. Figure 2 illustrates this configuration:

Figure 2. Compatible Configuration - Option 2



Note: In the figure above, devices represented by dotted lines may be attached, but are not accessible to software.

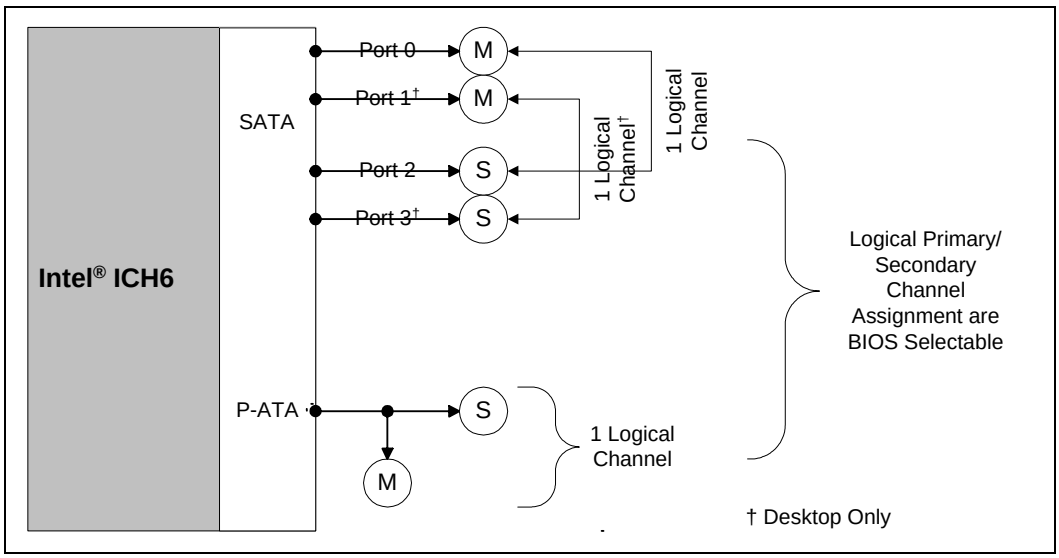
To enable this configuration, system BIOS:

1. Shall not program the P-ATA (Device 31, Function 1) controller's base address registers (Offsets 10h – 24h in PCI configuration space).
2. Shall disable access to the P-ATA controller's I/O space by programming the command register (PCI configuration, offset 04h, bit 0) with a '0'.
3. Shall disable the P-ATA function by programming bit 1 (***Parallel ATA Disable***) of the ***Function Disable*** register with a '1'. This will insure that the PCI configuration registers associated with the P-ATA function are not decoded and thus will insure that operating system configuration software does not enumerate and configure the P-ATA function.
4. Shall program the ***MAP.MV*** to a value of '00b'. This indicates that the ICH6 SATA HBA will operate in a traditional *master-slave* mode. In this mode SATA devices attached to ports 0 and 2 will act as master and slave devices (logical primary channel), respectively and SATA devices attached to ports 1 and 3 act as master and slave devices (logical secondary channel), respectively.
5. Shall insure that the ***Sub-Class Code (SCC)*** register is programmed to 01h (IDE). ***SCC*** is described in section *5.2 Class Code Register (CC).*
6. Insure that the Programming Interface (PI) register is programmed properly. This is described in section *5.3 Programming Interface Register (PI).*
7. Shall program the SATA registers appropriately. Because the programming interface is identical to that of P-ATA, BIOS should follow the same programming guidelines as used for the P-ATA (the exact details are beyond the scope of this document).

4.2.6 Compatible Configuration - Option 3 (Combined)

This option is selected when at least one SATA device (maximum of 2) and at least one P-ATA device (maximum of 2) are connected to both SATA and P-ATA host controllers and are both accessible by software. This configuration is referred to as the 'Combined' mode. In Combined mode of operation, devices can be supported in legacy mode only (i.e. Native IDE mode is not supported by the ICH6 when configured for Combined mode). It is important to note that in this configuration, the SATA and P-ATA host controllers share functionality but appear as a single PCI function. In this case, the actual P-ATA function is hidden from system software (i.e. cannot be enumerated or accessed directly) but P-ATA devices connected to the function may still be accessed. These devices are accessed via the standard P-ATA-compatible register set exposed by the SATA controller Figure 3 illustrates this configuration:

Figure 3. Compatible Configuration - Option 3

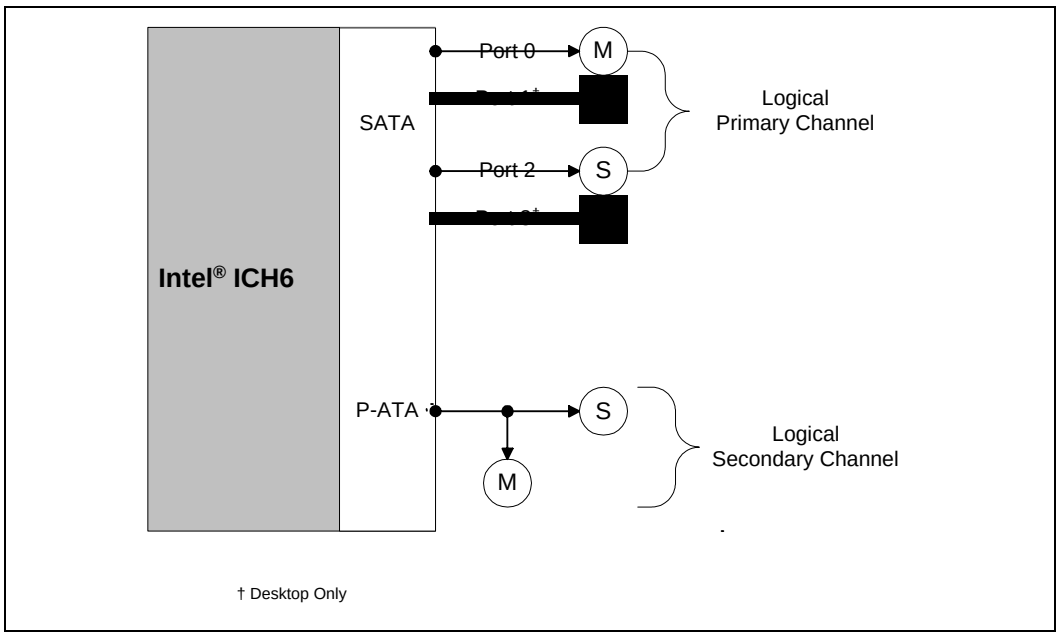


To enable this configuration, system BIOS:

1. Shall not program the P-ATA (Device 31, Function 1) controller's base address registers (Offsets 10h – 24h in PCI configuration space).
2. Shall disable access to the P-ATA controller's I/O space by programming the command register (PCI configuration, offset 04h, bit 0) with a '0'.
3. Shall disable the P-ATA function by programming bit 1 (***Parallel ATA Disable***) of the ***Function Disable*** register with a '1'. This will insure that the PCI configuration registers associated with the P-ATA function are not decoded and thus will insure that operating system configuration software does not enumerate and configure the P-ATA function.
4. Shall insure that the ***Sub-Class Code (SCC)*** register is programmed to 01h (IDE). ***SCC*** is described in section *5.2 Class Code Register (CC).*
5. Insure that the ***Programming Interface (PI)*** register is programmed properly. This is described in section *5.3 Programming Interface Register (PI).*
6. Shall program the ***MAP.MV*** register as follows:

   - If SATA is the logical ***secondary*** channel (port 1 is master and port 3 is slave) and P-ATA is the logical ***primary*** channel ***MAP.MV*** must be programmed with '01b'. Figure 4 illustrates this configuration:

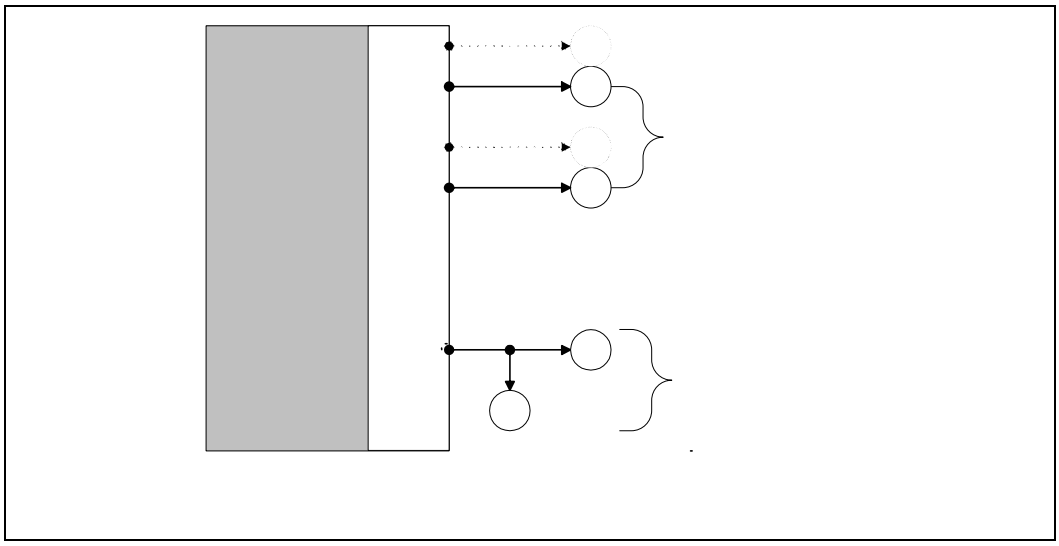
Figure 4. Compatible Configuration - Option 3a



Note: In the figure above, devices represented by dotted lines may be attached, but are not accessible to software.

- If SATA is the logical ***primary*** channel (port 0 is master and port 2 is slave) and P-ATA is the logical ***secondary*** channel then ***MAP.MV*** must be programmed with '10b'. Figure 5 illustrates this configuration:

Figure 5. Compatible Configuration - Option 3b (Desktop Only)



Note: In the figure above, devices represented by dotted lines may be attached, but are not accessible to software.

4.3 Enhanced Non-AHCI Configuration

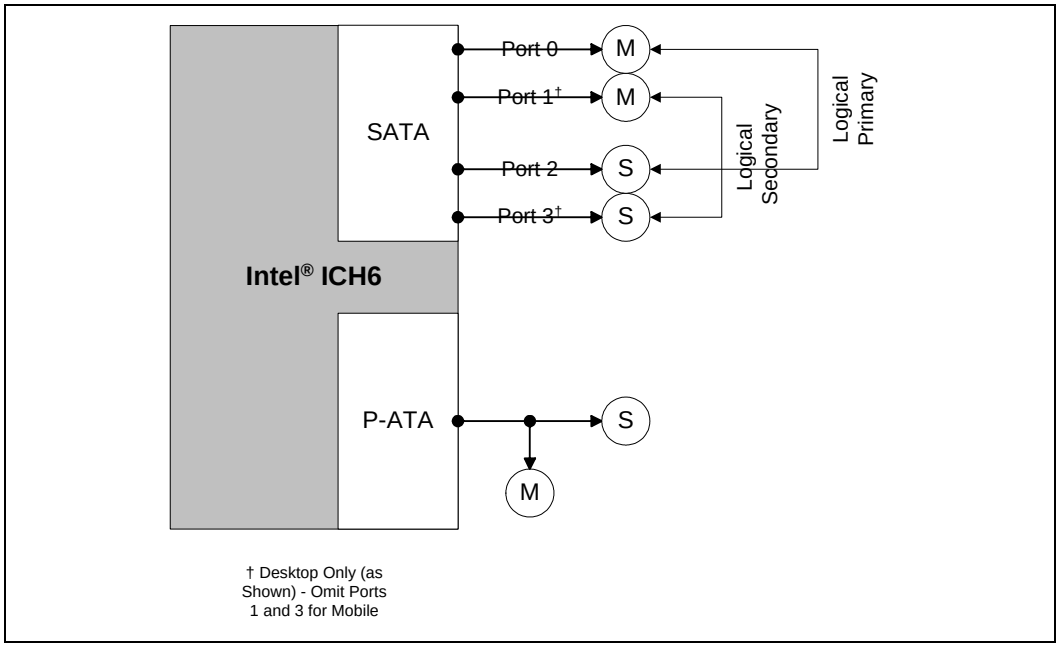
The Enhanced Non-AHCI configuration is intended for those operating systems (e.g., Microsoft Windows* 2000, Windows* XP) that comprehend both legacy and native modes of operation. It is the preferred hardware configuration for those operating systems that:

1. Comprehend and support multiple ATA HBAs present in a system.
2. Comprehend and support ATA HBAs configured for native mode.
3. Do not comprehend SATA HBAs configured for AHCI mode – Operating system support is limited to HBAs that implement a traditional, 'task-file' style programming interface

The Enhanced Non-AHCI configuration allows both the P-ATA and SATA host controllers to be used, providing a maximum of 6 ATA (2 P-ATA + 4 SATA) devices that can be used independently and simultaneously. In this configuration, the operating system can detect both the P-ATA HBA and the SATA HBA. The P-ATA HBA will consist of a single primary channel with up to 2 devices (master/slave) while the SATA HBA will consist of two logical channels (primary and secondary), with each channel supporting up to two devices in a master/slave configuration. Figure 6 illustrates this configuration:

Note: At least one ATA HBA (either the SATA or P-ATA) must be programmed for Native IDE when using this configuration (i.e., both HBA can be in Legacy mode).

Figure 6. Enhanced Non-AHCI Configuration



To enable this configuration, system BIOS shall:

1. Insure that ***AE*** is clear (‘0’). Note that this is the default value on initial power up. See section *5 Register Descriptions*
2. *AHCI* Enable (AE).
3. Insure that ***MAP.MV*** is programmed to ‘00b’. Note that this is the default value on initial power up.
4. Shall insure that the ***Sub-Class Code (SCC)*** register is programmed to 01h (IDE). ***SCC*** is described in section *5.2 Class Code Register (CC).*
5. Insure that the ***Programming Interface (PI)*** register is programmed properly. This is described in section *5.3 Programming Interface Register (PI).*

4.4 Enhanced AHCI Configuration

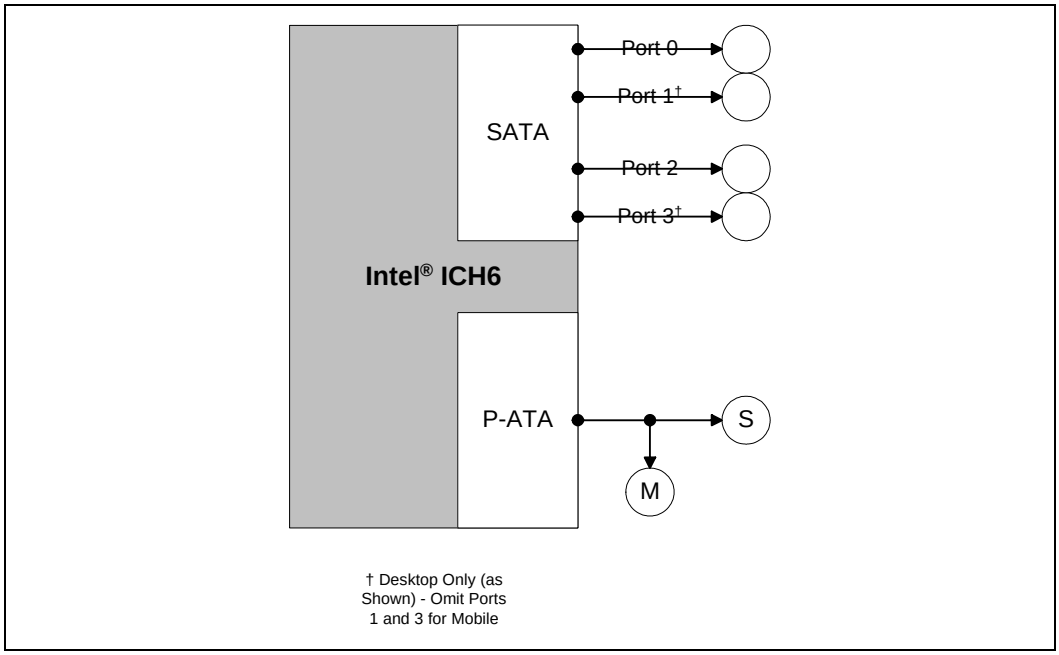
The Enhanced AHCI configuration is intended for those operating systems that:

1. Comprehend and support one or more ATA HBAs present in a system.
2. Comprehend and support SATA HBAs that are/can be configured for AHCI mode – operating system support is provided for mass storage HBAs that are SATA, AHCI 1.0 specification compliant.
3. Comprehend and support RAID.

In this configuration, the operating system can detect both the P-ATA HBA and the SATA HBA. The P-ATA HBA will consist of a single physical primary channel with up to 2 devices (master/slave) while the SATA HBA will consist of four ports where each port is capable of supporting a single SATA drive or a single Port Multiplier that hosts up to 15 SATA drives. Figure 7 illustrates this configuration.

Note: Refer to the *Serial ATA II Specification – Port Multiplier* for more information. This document is available from the SATA Working Group (www.serialata.org)

Figure 7. Enhanced AHCI Configuration



To enable this configuration, system BIOS shall:

1. Insure that the ***MAP.MV*** register is programmed to '00b'.
2. Insure that the **Sub-Class Code (SCC)** register is programmed to 06h (SATA). **SCC** is described in section *5.2 Class Code Register (CC).*

If the system software is designed to take advantage of AHCI functionality (i.e. will access the HBA via its memory mapped AHCI BAR) then it shall insure that ***AE*** is set ('1'). See section *5.1 AHCI Enable (AE)* for details.

Note: The AHCI BAR (Base Address Register) is defined as the value programmed at offset 24h in PCI Configuration space for the AHCI HBA. The memory mapped address space used by the HBA is accessed via this BAR. See the Serial ATA Advanced Host Controller Interface Specification for details.

Caution: Caution must be exercised for system BIOS' that are AHCI aware (system software is aware of the AHCI programming interface and programs to it), since booting to a non-AHCI aware operating system (system software is not aware of the AHCI programming interface) will result in undetermined behavior. Operating system software does not have this restriction as once the OS is loaded and running it is presumed to "own" the SATA HBA until the system is re-booted. Proper programming of the ***AE*** and the ***SCC*** registers must be observed. Table 2 illustrates the proper programming combinations:

Table 2. Sub-Class/AHCI Enable Combinations

AHCI Aware	Non-AHCI Aware	Sub-Class (SCC) Programming (Must be Programmed by BIOS)	AHCI Enable Flag (AE)	Result
N/A	BIOS + OS	IDE	0	Legal – Uses Legacy Interface
N/A	BIOS + OS	SATA	0	Illegal – SCC must indicate IDE
N/A	BIOS + OS	IDE	1	Illegal – AE Flag must be ‘0’
N/A	BIOS + OS	SATA	1	Illegal – AE Flag must be ‘0’ and. SCC must indicate IDE
BIOS + OS	N/A	IDE	1 – Set by BIOS	Illegal – AE flag must be ‘0’ when SCC is IDE
BIOS + OS	N/A	SATA	1 – Set by BIOS	Legal
BIOS + OS	N/A	SATA	0	Illegal – AE Flag Must be set to ‘1’
BIOS	OS	SATA	1 – Set by BIOS	Illegal – AE Flag must be ‘0’ and. SCC must indicate IDE
BIOS	OS	IDE	1 – Set by BIOS	Illegal – AE Flag must be ‘0’
BIOS	OS	SATA	0	Illegal – SCC must indicate IDE
OS	BIOS	SATA	1 – Set by OS	Legal
OS	BIOS	SATA or IDE	1 – Set by BIOS	Illegal – AE Flag Must Be ‘0’
OS	BIOS	IDE	0	Legal, but not optimal as an AHCI aware OS wouldn’t want to use the non-AHCI programming interface
OS	BIOS	SATA	0	Illegal – AE flag must be set to ‘1’

§

5 Register Descriptions

5.1 AHCI Enable (AE)

The ***AE*** flag is located in the ***Global HBA Control (GHC)*** register (AHCI BAR + 04h). This is a system software programmable flag that informs the ICH6 SATA HBA as to how system software will access the SATA HBA functionality:

- If ***AE*** is set to '1', then it is required that system software will only access the SATA HBA functionality via memory registers defined by the AHCI BAR. When ***AE*** is set to '1', the HBA is considered to be functioning in AHCI mode.
- If ***AE*** is set to '0', then it is required that system software will only access the SATA HBA functionality via legacy I/O addresses (see section *4.1.1 Legacy Mode*) or through I/O addresses defined by BARs 0-4 (see section *4.1.2 Native IDE* Mode. When ***AE*** is set to '0', the HBA is considered to be functioning in non-AHCI mode.

Note: System software can still access registers in AHCI memory space if ***AE*** is clear, however it shall not attempt to perform any transactions to an attached device using AHCI specific mechanism.

Note: The PCI configuration space for the SATA HBA is always accessible, regardless of how ***AE*** is programmed.

5.2 Class Code Register (CC)

The ***Class Code (CC)*** register is located in the SATA HBA's PCI Configuration space at offset 0Ah. The class code is used by system software to further identify a hardware component so as to insure that proper support is provided by the operating system.

The ICH6 AHCI HBA can be configured to identify itself as a RAID controller, an IDE controller or a SATA (AHCI compliant) controller. The default class code for the HBA is dependent on the ICH6 SKU. Table 3 describes the supported class types and register programmability:

Table 3. Class Code Register for SATA

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>15:08</td><td>RO</td><td>01h</td><td>Base Class Code (BCC): Indicates that this is a mass storage device.</td></tr>
<tr><td>`</td><td>See Desc</td><td>See Desc</td><td>Sub Class Code (SCC): This field takes on one of several values, depending upon the SKU (RAID or Non-RAID) and MAP.USCC, as shown in the following table:
<table>
<tr><th rowspan="2">SKU</th><th rowspan="2">MAP.USCC</th><th colspan="2">Register</th></tr>
<tr><th>Attribute</th><th>Value</th></tr>
<tr><td>Non-RAID</td><td>0</td><td>RO</td><td>01h (IDE)</td></tr>
<tr><td>Non-RAID</td><td>1</td><td>RO</td><td>06h (SATA)</td></tr>
<tr><td>RAID</td><td>N/A</td><td>RWO</td><td>04h (RAID)</td></tr>
</table>
</td></tr>
</table>

The ***Sub-Class Code (SCC)*** for the SATA HBA is only programmable by the system BIOS during POST. Any attempts at subsequent re-programming of the register, by system software, will be ignored. The value and programmability of this register is dependent on the platform SKU and the programming of the ***MAP.USCC*** bit.

Note: The exact value that gets programmed into this register will be based on what level of support the target operating system provides for native-AHCI functionality and RAID. Programming this register with a value that is inconsistent with the level of support provided by operating system will result in indeterminate behavior.

The following summarizes how the ***SCC*** and ***MAP.USCC*** registers shall be programmed:

Note: BIOS shall not re-configure the HBA (i.e. reprogram ***MAP.USCC*** from 1->0) once it has been configure for AHCI mode as it is read-write once. BIOS shall not program the ***MAP.USCC*** register when the RAID SKU is being used. System BIOS can determine if the RAID SKU is in use by an examination of the ***SCC*** register, as the RAID SKU power on default for this register is 04h.

- For non-RAID enabled platforms:
  - If the target operating system ***does not support*** AHCI then system BIOS shall leave ***MAP.USCC*** as ‘0’. This will result in a ***SCC*** value of 01h (RO)(IDE) and a ***PI*** value of 8Ah (RW). Note that for non-RAID platforms, these are the power on default values.
  - If the target operating system ***supports*** AHCI then system BIOS shall set ***MAP.USCC*** to ‘1’. This will result in a ***SCC*** value of 06h (RO)(SATA) and a ***PI*** value of 01h (RO).
- For RAID enabled platforms:
  - If the target operating system supports ***RAID*** then system BIOS shall program the ***SCC*** register with a value of 04h (RAID). No programming of the ***PI*** register is permitted as this register becomes RO ‘0’.
  - If the target operating system ***does not support*** AHCI or if RAID is not desired (i.e. want IDE mode) then system BIOS shall program the ***SCC*** register with a value of 01h (IDE). Proper programming of the ***PI*** register is described in section *5.3 Programming Interface Register (PI)*.

— If the target operating system ***supports*** AHCI, but RAID and IDE is not desired, then system BIOS shall program the ***SCC*** register with a value of 06h (AHCI). No programming of the ***PI*** register is required, as this register becomes RO '1'.

5.3 Programming Interface Register (PI)

The ***Programming Interface (PI)*** register is located in the SATA HBA's PCI Configuration space at offset 09h. The exact contents and meaning of this register is dependent on the value of the ***CC*** register (IDE [01h], SATA [06h] or RAID [04h]) and the value of the ***MAP.USCC*** bit.

5.3.1 IDE Mode Considerations

As stated previously, the P-ATA and SATA (non-AHCI mode) channels can be configured to operate in either Native mode or Legacy mode. This is controlled via the ***PI*** register. The ***PI*** register is found in both the SATA and P-ATA functions and can be modified by both BIOS (during POST) and operating system software. Table 4 illustrates the possible values for the ***PI*** register when the ***CC*** register indicates that the SATA HBA is configured as a Mass storage, IDE device and ***MAP.USCC*** is set to '0':

Table 4. PI – Programming Interface For Mass Storage, IDE Device

Bit	Type	Reset	Description
7	*RO*	*1*	*Indicates the SATA Controller supports bus master operation.*
6:4	*RO*	*0*	*Reserved*
3	RO	1/0	**Secondary Mode Native Capable (SNC):** Indicates that the secondary controller supports both legacy and native modes. When MAP.MV is any value other than 00, this bit reports as a '0'. When MAP.MV is '00', this bit reports as a '1'.
2	RW/ RO	0	**Secondary Mode Native Enable (SNE):** Determines the mode that the secondary channel is operating in. '0' corresponds to 'compatibility', '1' means PCI native. If this bit is set by SW, then the PNE bit must also be set by SW. While in theory these bits can be programmed separately, such a configuration is not supported by today's software and is not supported by this hardware. When MAP.MV is any value other than 00, this bit is read-only. When MAP.MV is '00', this bit is read/write.
1	RO	1/0	**Primary Mode Native Capable (PNC):** Indicates that the primary controller supports both legacy and native modes. When MAP.MV is any value other than 00, this bit reports as a '0'. When MAP.MV is '00', this bit reports as a '1'.
0	RW/ RO	0	**Primary Mode Native Enable (PNE):** Determines the mode that the primary channel is operating in. '0' corresponds to 'compatibility', '1' means PCI native. If this bit is set by SW, then the SNE bit must also be set by SW. While in theory these bits can be programmed separately, such a configuration is not supported by today's software and is not supported by this hardware. When MAP.MV is any value other than 00, this bit is read-only. When MAP.MV is '00', this bit is read/write.

Note: The ICH6 does permit the SATA and P-ATA host controllers to simultaneously operate in native mode if they are programmed to do so by the operating system. Due to potential OS incompatibilities, it is a requirement (when in Enhanced, non-AHCI mode) that the system BIOS programs the P-ATA and SATA host controllers exactly as described in Table 5. Improper programming *could* result in undefined behavior.

Some Microsoft operating systems have specific platform support requirements when operating on systems capable of native mode of operation. Refer to paper titled *BIOS Settings for Native-Mode-Capable ATA Controllers*, available from Microsoft Corporation at: http://www.microsoft.com/hwdev/tech/storage/Native-modeATA.asp, for additional OS related details.

Table 5 illustrates the valid values that system BIOS can use for the ***PI*** register when in the Enhanced Non-AHCI configuration:

Table 5. Valid BIOS Option for the Programming Interface Register (Non-Combined Mode)

SATA – Device 31, Function 2		P-ATA – Device 31, Function 1	
Primary Mode Native Enable (PNE)	Secondary Mode Native Enable (SNE)	Primary Mode Native Enable (PNE)	Secondary Mode Native Enable (SNE)
1	1	0	0

Note: The SATA and P-ATA host controllers do not support the programming of the primary and secondary channels differently (i.e., primary as native and secondary as legacy). Programming the controllers with values other than those specified above is illegal and will result in undefined hardware behavior.

Table 6 illustrates an illegal programming combination. System BIOS programming of all the channels (both the SATA and P-ATA controllers) for legacy mode or native will result in undefined behavior. Operating system software that supports legacy mode only requires that the SATA host controller is in the Compatible configuration.

Table 6. Illegal BIOS Options for the Programming Interface Register

SATA – Device 31, Function 2		P-ATA – Device 31, Function 1	
Primary Mode Native Enable (PNE)	Secondary Mode Native Enable (SNE)	Primary Mode Native Enable (PNE)	Secondary Mode Native Enable (SNE)
0	0	0	0
0	0	1	1
1	1	1	1

Note: Following a reset, the SATA and P-ATA controllers will both be configured for legacy mode. Therefore it is very important that system BIOS program these registers with legal values as defined in Table 5. Valid BIOS Option for the Programming Interface Register (Non-Combined Mode)

5.3.1.1 Combined Configuration Considerations

When configured for Combined mode (see section *4.2.6 Compatible Configuration - Option 3 (Combined)*), the ICH6 SATA HBA does not support Native IDE mode; it may only be accessed using legacy access methods. As such the ***PI*** register is 'read-only' in this configuration.

5.3.2 SATA Mode Considerations

When the SATA HBA is operating in the Enhanced, AHCI configuration, the ***PI*** register is not programmable since in this configuration the HBA is only accessible using the Native IDE or AHCI memory mapped BARs. Legacy mode is not supported.

Table 7. PI – Programming Interface for Mass Storage, SATA Device

Bit	Type	Reset	Description
07:00	RO	01h	**Interface (IF):** Indicates the SATA Controller is AHCI 1.0 compliant.

Note: The AHCI Specification reserves ***PI*** values of 00h-1fh for SATA HBA implementations that are compliant with the AHCI specification. For ICH6, the SATA HBA ***PI*** register (when ***CC.SCC*** == ***06h***) will always return 01h when read by system software.

5.3.3 RAID Mode Considerations

When the SATA HBA is operating in the Enhanced, AHCI configuration and RAID is selected (i.e. ***CC.SCC*** == ***04h***), the ***PI*** register is not programmable since in this configuration the HBA is only accessible using the Native IDE BARs or AHCI memory mapped BARs. Legacy mode is not supported.

Table 8. PI – Programming Interface for Mass Storage, RAID Device

Bit	Type	Reset	Description
07:00	RO	00h	**Interface (IF):** Indicates that there is no programming interface when the HBA is configured as RAID. Internally under this condition, the SATA HBA is in native mode and its I/O spaces are only accessible through the I/O BARs (BARs 0-3 and BAR 5)

Note: ***PI*** register bits 2 and 0 are write-able at all times (so the value of this field may be 8Ah, 8Bh, 8Dh, or 8Fh), although the write only affects the SATA controller's operation when ***CC.SCC*** is 01h (IDE). If ***PI*** is written while ***CC.SCC*** is 04h (RAID), the value shown will not be 8Ah when ***CC.SCC*** is written from 04h to 01h, but rather the value of the last write to ***PI***. BIOS shall always check and set ***PI*** to the desired mode of operation (Legacy or Native IDE) when ***CC.SCC*** is 01h.

5.4 Port Control and Status Register (PCS)

The ***PCS*** register provides port level control and status for system software. The ***PCS*** register implements the following important features, which are discussed in subsequent sections:

- Port enablement/disablement
- Port level presence detect
- Staggered spin up support

These capabilities are illustrated in Table 9. Port Control and Status (PCS) Register – PCI Configuration Offset 92h:

Note: The ***PCS*** register is managed primarily by the platform BIOS and ASL code; operating system software should never need to access this register.

Table 9. Port Control and Status (PCS) Register – PCI Configuration Offset 92h

Bit	Type	Reset	Description
15:08	RWC	0	*Reserved*
7	RO	0	**Port 3 Present (P3P):** Same as P0P, except for port 3.
6	RO	0	**Port 2 Present (P2P):** Same as P0P, except for port 2.
5	RO	0	**Port 1 Present (P1P):** Same as P0P, except for port 1.
4	RO	0	**Port 0 Present (P0P):** When set, the SATA controller has detected the presence of a device on port 0. It may change at any time. This bit is cleared when the port is disabled via P0E. This bit is not cleared upon surprise removal of a device.
3	RW	0	**Port 3 Enabled (P3E):** Same as P0E, except for port 3. This bit takes precedence over P3CMD.SUD.
2	RW	0	**Port 2 Enabled (P2E):** Same as P0E, except for port 2. This bit takes precedence over P2CMD.SUD.
1	RW	0	**Port 1 Enabled (P1E):** Same as P0E, except for port 1. This bit takes precedence over P1CMD.SUD.
0	RW	0	**Port 0 Enabled (P0E):** When set, the port is enabled. When cleared, the port is disabled. When enabled, the port can transition between the on, partial, and slumber states and can detect devices. When disabled, the port is in the 'off' state and cannot detect any devices. This bit takes precedence over P0CMD.SUD.

NOTES:
1. Refer to the Serial ATA Advanced Host Controller Interface Specification for details on the usage of this register bit.

5.4.1 Port Enabling/Disabling

By default, the SATA ports are set (by hardware) to the disabled state as a result of an initial power on reset. When enabled, the ports can transition between the on, partial, and slumber states and can detect devices. When disabled, the port is in the 'off' state and cannot detect any devices.

System software may choose to (keep) disable a port as a result of a device being disconnected from a port(s). Overall power consumption can be reduced if system software only enables those ports that have SATA devices attached. This is especially beneficial to mobile systems.

A specific port may be enabled and disabled by system software through the manipulation of the appropriate ***PCS.PxE*** (where ***x*** is 0, 1, 2, 3 for Port 0, Port 1, Port 2, and Port 3 respectively) bit where setting the bit to '1' enables the port and setting the bit to '0' disables the port.

Because the AHCI HBA natively (through memory mapped registers) provides mechanism for enabling and disabling individual ports, the remainder of this section applies to system BIOS' and operating system that are not AHCI aware. AHCI specific mechanisms for enabling and disabling individual ports are discussed in the SATA AHCI (Advanced Host Controller Interface) specification.

The following are general guidelines to be used when enabling and disabling the SATA ports:

- If an AHCI-aware or RAID enabled operating system is being booted then system BIOS shall insure that all supported SATA ports are enabled prior to passing control to the OS. Once the AHCI aware OS is booted it becomes the enabling/disabling policy owner for the individual SATA ports.

- A port on the ICH6 SATA HBA shall be enabled in order for its associated ***PCS.PxP*** bit to be accurate. See Section *5.4.2.1 Hardware and Software Considerations.*

- **Applicable to non-AHCI aware system operating systems:** As part of a robust power conservation strategy, ports on the ICH6 SATA host controller shall be disabled when the SATA host controller is to transition to a D3 power state from the D0 power state. This can be accomplished via platform ASL code by implementing the _PS3 control method for the device. *An exception to this occurs when a port is indicated to be hot-pluggable but does not support an interlock switch*

- **Applicable to non-AHCI aware system operating systems:** As part of a robust power conservation strategy, a port on the ICH6 SATA host controller should be disabled when a SATA device is not physically present (due to a hot-swap event or due to the device not being present during POST). *An exception to this occurs when a port is indicated to be hot-pluggable but does not support an interlock switch.*

- Ports on the ICH6 SATA HBA shall be enabled as part of staggered spin-up processing. This applies to both AHCI aware, non-AHCI aware and RAID enabled operating systems and system software. Staggered spin-up is discussed in section *6 Staggered Spin Up.*

Note: Because the ICH6 SATA host controller is designed with backward compatibility in mind, it is expected that existing operating system software designed for P-ATA host controllers should never have to modify the ***PCS.PxE*** bits directly; this shall be done by the system BIOS and/or associated ACPI control methods.

5.4.1.1 BIOS Considerations

To insure that SATA devices are functional following certain system power state transitions, system BIOS shall enable (where appropriate) the SATA ports under the following conditions:

- Any APM supported system state transition where the ICH6 SATA host controller is reset.
- A device power state transition where the operating system **cannot** enable the SATA ports (due to lack of knowledge of the ICH6 SATA host controller's programming interface or due to lack of ACPI support). Because the ICH6 SATA host controller does not reset its registers following a D3hot -> D0 transition, this case is typically limited to those cases where the host controller experiences a D3cold->D0 transition.

Failure by the system BIOS to provide the required support will result in device inaccessibility and/or loss of OS functionality.

5.4.2 Device Presence Detect

The ICH6 SATA host controller provides bits in the ***PCS*** register (***P0P, P1P, P2P, P3P***) that can be used by system software to detect the presence (or lack) of SATA device(s) connected to the SATA host controller.

Device presence detection has two main benefits:

- Assists in quicker boot times (both BIOS POST and OS) as it can eliminate the need for time-consuming device detection algorithms.
- Assists in providing better power management. A port with no attached device can be disabled; a disabled SATA port consumes less power than one that is enabled. See section: *5.4.1 Port Enabling/Disabling*.

Note: The ***PxP*** bits are for use by non-AHCI aware system software (i.e. system BIOS and ASL control methods). AHCI aware system software will typically use the facilities provided by the AHCI memory space. Refer to the SATA AHCI (Advanced Host Controller Interface) specification for details.

5.4.2.1 Hardware and Software Considerations

The value of the ***PxP*** bits is valid only under the following conditions:

- The SATA host controller is in the D0 power state.
- The port to be checked is enabled (***PCS.PxE == '1'***).
- The port's associated ***PxCMD.SUD*** bit is set to '1'

Note: There is no requirement that all of the ports are enabled when checking a ***PxP*** bit; only the port being examined needs to be enabled.

After a port is enabled, system software shall examine the associated ***PxP*** bit (via polling) to determine if a device is present (connected) on the port. System software shall poll the ***PxP*** bit for the maximum amount of time as specified in section *11.2.2.2 OOB Signaling Sequence Failure* of the *Serial ATA: High Speed Serialized AT Attachment v1.0* specification (this document is available from the Serial ATA Working Group at: http://www.serialata.org/). If the ***PxP*** bit has

not made a '0' -> '1' transition in the specified amount of time, then it can be assumed that no device is present on the port. It is important to note that the ***PxP*** bit may transition from '0' -> '1' in less than the specified period - this is permissible.

5.4.2.1.1 Device Ready Detection

The ***PxP*** bits only indicate device presence; device readiness (i.e. able to receive commands) shall be determined by an examination of the status register (contained within the appropriate emulated channel Control register). The device indicates ‘readiness’ when the device is 1) Not busy (BSY == 0) and 2) is ready (DRDY == 1).

Note: The maximum amount of time that a SATA device is permitted to take before indicating readiness is specified in the ATA/ATAPI-6 specification whish is available from the T13 Technical Committee at www.t13.org),

This is identical to how device presence/readiness is determined for P-ATA devices. AHCI aware system software shall determine a device’s readiness through an examination of the associated port’s Port Task File Data (***PxTFD***) register. Refer to the SATA AHCI (Advanced Host Controller Interface) specification for details.

§

6 Staggered Spin Up

ICH6 based platforms that include numerous Serial ATA hard disk drives may be presented with power system design issues related to the electrical current load presented during system power-up. Staggered spin up provides a simple mechanism by which the ICH6 SATA HBA can sequence disk drive initialization and spin up. Specific details with regards to SATA and staggered spin up can be found in *Serial ATA II: Extensions to Serial ATA 1.0 Rev 1.0* specification (this document is available from the Serial ATA Working Group at: http://www.serialata.org/).

The ICH6 SATA HBA allows staggered spin up to be supported by system software that is either non-AHCI aware or AHCI aware.

6.1 Supporting Staggered Spin up during POST

To support staggered spin up during POST, the system BIOS shall perform the following steps:

1. Shall insure that ***CAP.SS*** == '1'. A ***CAP.SS*** transition from 0->1causes each port's ***PxCMD.SUD*** bit to clear to '0'.

2. Shall insure that all implemented SATA ports are enabled. This is done by setting each port's enable bit (***PCS.PxE***) to '1'. Note that the port enable/disable bits for all ports can be set simultaneously.

3. For each implemented port, system BIOS shall begin the device spin up by performing the following steps:

    a. System BIOS shall set the port's ***PxCMD.SUD*** bit to '1'. This results in a COMRESET and causes the attached device to rotate its spindle.

    b. System BIOS shall wait for a positive indication that a device is attached to the port.

        i. If the system BIOS is AHCI aware, then this is accomplished by an examination (via polling) of the port's respective ***PxSSTS.DET*** bit. If this bit returns a value of '1' when read then a device is present. Otherwise, no device is present.

        Note: The maximum amount of time to poll the ***PxSSTS.DET*** bits is specified in the Serial ATA: High Speed Serialized AT Attachment v1.0a specification. This specification is available from www.serialata.org

        ii. If the system BIOS is not AHCI aware, then this is accomplished by an examination of the port's respective ***PCS.PxP*** bit. Refer to section *5.4.2 Device Presence Detect.*

c. If no device is present, then system BIOS shall skip this step and continue to the Cleanup step. Otherwise, system BIOS shall wait for indication that the attached SATA drive is ready (see section *5.4.2.1.1 Device Ready Detection*). If the SATA device indicates ready, then the system BIOS shall continue the staggered spin up sequence using the next available port (go to step a.).

d. Cleanup

i. If booting to a non-AHCI aware OS and no device is detected then system BIOS may choose to either leave the port enabled or it may choose to disable it – this is dependent of the platform power policy. Refer to section *5.4.1 Port Enabling/Disabling.*

ii. If booting to a non-AHCI aware OS and the device does not indicate 'ready', then the system BIOS shall leave the port enabled and shall leave the port's ***PxCMD.SUD*** bit set to '1'

iii. If booting to a AHCI or RAID aware OS and no device is present and/or the device is not ready, then the system BIOS shall leave the port enabled and shall leave the port's ***PxCMD.SUD*** set to ' 1'.

System BIOS shall continue the staggered spin up sequence using the next available port (go to step a).

6.2 Supporting Staggered Spin Up during Resume from the S3 System Power State

This section describes how system BIOS supports staggered spin up during resume from the S3 system power state.

6.2.1 HBA Behavioral Assumptions

The ICH6 SATA HBA enters the D3cold device power state whenever the system enters the S3 power state. Note that this is also true when starting from the S4 and S5 states.

Upon resuming from the S3 system power state, the following is true:

- Results in the AHCI SATA HBA being placed into a D0unitialized state (causes a change in device power state, D3cold->D0uninitialized):
  - AHCI SATA HBA PCI Configuration space is reset
  - AHCI SATA HBA memory mapped space is reset
- Register restoration is the responsibility of system BIOS, the OS and device driver software
- System BIOS can participate in resume path processing
- The SATA devices will lose their programming – requires reprogramming by system software.

6.2.2 BIOS Considerations

To support staggered spin up on the platform when resuming from the S3 system power state the system BIOS shall perform the following:

1. Shall insure that ***CAP.SS*** == '1'. Remember that a ***CAP.SS*** transition from 0->1causes each port's ***PxCMD.SUD*** bit to clear to '0'.

2. Shall insure that all implemented SATA ports are enabled. This is done by setting each port's enable bit (***PCS.PxE***) to '1'. Note that the port enable/disable bits for all ports can be set simultaneously.

3. If the BIOS is resuming to an AHCI-aware operating system, then this completes BIOS' participation in the staggered spin up sequence as it is expected that the OS will complete the remaining steps required to spin up the drives. Eliminating the actual spin up of the drives (from a BIOS perspective) will expedite the resume process (e.g. reduce resume latencies). Otherwise, the user could be presented with a blank screen during the interval when the BIOS is sequentially spinning up each drive in the platform.

4. If the BIOS is resuming to a non-AHCI operating system then system BIOS shall complete the steps outline in steps 3a-d in the previous sub-section. BIOS must perform the complete staggered spin up sequence for this case as a non-AHCI OS is not cognizant of the staggered spin up feature.

6.2.3 OS Considerations

To support staggered spin up on the platform when resuming from the S3 system power state an AHCI-aware operating system device driver shall implement one of the of the following strategies:

Note: It is assumed that the system BIOS has not participated in the full staggered spin up sequence during the S3 resume path and as such has not set each port's ***PxCMD.SUD*** bit to '1'.

1. Do not spin up any of the drives until an access request is made (e.g. read/write).
   - Provides best reduction in resume latency.
   - Slight, per device, one time performance hit taken when each spun-down device is eventually accessed.
   - Care must be taken when accessing any remaining spun down drives simultaneously (e.g. needs to be serialized).
2. Spin up only the drive that contains the page file as statistically, accessing this drive first may have the highest probability.
   - Provides good reduction in resume latency in that only a single drive must be spun up during resume.
   - Slight, per device, one time performance hit taken when each spun-down device is eventually accessed.
   - Care must be taken when accessing any remaining spun down drives simultaneously (e.g. needs to be serialized).

3. Sequentially spin up all of the drives during resume phase.
   - Provides least reduction in resume latency.
   - Decreased complexity
   - One-shot event – once the drives are spun up, they will be instantly available (e.g. no performance penalty due to spin-up).

6.3 Supporting Staggered Spin Up During Resume from the S1 System Power State

This section describes how system software supports staggered spin up during resume from the S1 system power state.

6.3.1 HBA Behavioral Assumptions

The ICH6 SATA HBA enters the D3hot device power state whenever the system enters the S1 power state.

Upon resuming from the S1 system power state, the following is true:

- AHCI will experience a D3hot->D0 transition
  - Results in the HBA being placed into a D0initialized state.
  - AHCI SATA HBA PCI Configuration space is not reset.
  - AHCI SATA HBA memory mapped space is reset.
- BIOS cannot participate in resume path processing
- SATA device will not lose power or programming

6.3.2 BIOS Considerations

Since system BIOS does not participate directly in the S1 resume path, it has no special requirements for staggered spin up support.

6.3.3 Non-AHCI Aware OS Considerations

Since SATA devices maintain power in this state, each device will spin up upon being accessed, provided the device was placed into standby/sleep via ATA command. As such, when resuming from S1, the OS must insure that it does not permit simultaneous access to all attached devices until they have been accessed serially, at least once. For non-AHCI OS device drivers, this can be managed if the devices are re-enumerated sequentially during resume processing (e.g., any ATA command will cause a device previously placed into STANDBY/SLUMBER to begin spinning up).

6.3.4 AHCI Aware OS Considerations

Same as resume from S3, except no programming of the SATA devices is required as power to the devices is maintained in S1.

§

7 Device Insertion/Removal Detection

The ICH6 AHCI supports a variety of mechanisms for detecting device insert/removal events; these mechanisms are consistent with the mechanisms described in the AHCI Specification.

The ICH6 AHCI HBA provides support for the following device insertion/removal detection mechanisms:

- Hot-Plug
- Interlock switch (via SCI or interrupt)
- PME#

Note: ICH6 does not support Cold Presence detect as a device insertion/removal detection mechanism. Hot-plug detection is not supported by BIOS and non-AHCI aware operating systems; it can only be supported by a AHCI aware system software and is not described here. Refer to the SATA AHCI specification for details.

7.1 Device Insertion/Removal Detection Support in an AHCI Aware Environment Using Interlock Switches

In an AHCI aware environment (i.e. ***AE == '1'***), system software can use the ability of the ICH6 to generate an interrupt any time an interlock switch event occurs. The exact system software details can be found in the SATA AHCI specification. If the platform BIOS is not AHCI aware, but is booting to an OS that is, then the BIOS must do the following to insure that when the interlock switch logic is activated (i.e. switch opened/closed) the result is an AHCI interrupt and not an SCI.

To enable interrupt generation, the system BIOS shall insure that the following bits in the GPIO_USE_SEL register are set for those ports that implement interlock switch:

- Set GPIO_USE_SEL[26] = '0' if SATA Port 0 implements an interlock switch
- Set GPIO_USE_SEL[29] = '01' if SATA Port 1 implements an interlock switch
- Set GPIO_USE_SEL[30] = '0' if SATA Port 2 implements an interlock switch
- Set GPIO_USE_SEL[31] = '0' if SATA Port 3 implements an interlock switch

GPIO_USE_SEL[26, 29, 30, 31] are located at GPIOBASE + 00h. GPIOBASE is defined in PCI Configuration space offset 048h in Device 31, Function 0 (LPC Interface).

Note: The ICH6 must never be programmed to generate both SCI and interrupts (e.g. Port 0 generates SCI and Port 1 generates an interrupt). This will result in undefined software behavior.

7.2 Device Insertion/Removal Detection Support in a Non-AHCI Aware Environment Using Interlock Switches

In a non-AHCI enabled environment (i.e. ***AE == '0'***), system software can use the ability of the ICH6 to generate SCIs to detect when a device insertion/removal event occurs. SCI support is not directly implemented by a storage device driver; it is implemented by the OS and/or the system BIOS (usually through ACPI). As such, device insertion/removal support in a non-AHCI enabled environment should be consistent with that commonly found in systems with ATA swap bay support.

How SCI support is implemented on the platform is OEM dependent is beyond the scope of this document.

7.2.1 General BIOS Guidelines

- When enabling SCI generation for interlock switch events, the system BIOS shall insure that bits 26, 29, 30 and 31 in the GPIO_USE_SEL register are clear.
- A port need not be enabled if it supports an interlock switch and no device is present on the port. When a device is attached to the port, the specific SCI handler can re-enable the port prior using the device. As stated previously, disabling a port when a device is not attached is good power management policy.
- When the AHCI HBA is placed into a D3 state, the system BIOS shall insure that SCI generation is disabled.

7.3 PME#

The ICH6 AHCI HBA provides the ability to generate a power management event (PME#) whenever a SATA device is inserted (but not removed) or whenever a SATA ATAPI device that supports AN (ATAPI Notification) signals that the eject button was pressed. PME# can be used to wake the AHCI HBA from the D3hot power state. Waking from a D3cold state is not supported by ICH6.

System BIOS is not responsible for arming the AHCI HBA to generate PME#; this is the responsibility of an operating system and as such is not discussed in this document.

7.3.1 General BIOS Guidelines

To enable the events that can cause a PME#, system BIOS shall insure that:

- The individual ports are enabled (i.e. ***PCS.PxE == '1'***) if the port allows device insertion/removal and the target operating system is non-AHCI aware.
- The individual ports are enabled (i.e. ***PCS.PxE == '1'***) regardless of whether or not the port allows device insertion/removal, if the target operating system is AHCI aware. As discussed earlier in this document, AHCI aware system software can use AHCI mechanisms for disabling the SATA ports.

When the AHCI HBA is placed into a D3 state, the system BIOS shall insure that interlock switch SCI generation is disabled (if applicable).

Appendix A - Initializing the Hardware

For ICH6 SKUs that support the RAID and /or AHCI features, system BIOS shall initialize (and enable when appropriate) the following registers during POST and when resuming from the S3 and S4 power states:

- Native IDE BARS:
  - Primary Command (AHCI PCI Configuration offset 10h).
  - Primary Control (AHCI PCI Configuration offset 14h).
  - Secondary Command (AHCI PCI Configuration offset 18h).
  - Secondary Control (AHCI PCI Configuration offset 1Ch).
  - Bus Master IDE (AHCI PCI Configuration offset 20h).
- AHCI BAR (ABAR)(AHCI PCI configuration offset 24h).
- Capability and Port specific registers. Some of the bits in these registers are platform specific and shall be programmed in accordance with the requirements of the platform. The exact details regarding how these registers shall be programmed can be found in the Serial ATA Advanced Host Controller Interface (AHCI) specification. **These registers shall only be programmed when the ICH6 is programmed to function in RAID or AHCI mode. These registers shall not be programmed when the ICH6 is configured to function in IDE mode.**

Note: Because the following memory mapped registers are implemented as read/write-once (R/WO), it is a requirement that the system BIOS programs each bit at least once, even if the default setting of the bit is the desired value (i.e. BIOS must write '0' to a bit even if the hardware reset value is '0'). Doing so will insure that the bit is unchangeable by non-BIOS software.

Note: Registers containing multiple R/WO bits must be programmed in a single atomic write. Failure to do so will result in non-programmable bits.

- **Supports Command Queue Acceleration (SCQA) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00, bit 30. Indicates that the S-ATA controller supports Serial-ATA command queuing via the First Party DMA FIS. **BIOS shall always set this bit to a value of '1'**.
- **Supports Interlock Switch (SIS) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00, bit 28. This bit is used to indicate whether or not one or more ports on the AHCI implement an interlock switch. Also see ***ISP***.
- **Support Staggered Spin-up (SSS) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00h, bit 27. This bit is used to indicate whether or not the platform supports the staggered spin-up feature.

- **Supports Aggressive Link Power Management (SALP) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00h, bit 26 of this register. Indicates that the S-ATA controller supports auto-generating link requests to the partial or slumber states when there are no commands to process. **BIOS shall always set this bit to a value of '1'**.

- **Supports Port Multiplier (PMS) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00h, bit 17. Indicates whether or not the host controller provides port multiplier support. The ICH6 AHCI HBA *does not* provide port multiplier support. **BIOS shall always set this bit to a value of '0'**.

- **Slumber State Capable (SSC) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00h, bit 14. Indicates whether or not the host controller supports the slumber state. **BIOS shall always set this bit to a value of '1'**.

- **Partial State Capable (PSC) bit.** This bit is located in the AHCI HBA Capabilities register at ABAR + 00h, bit 13. Indicates whether or not the host controller supports the partial state. **BIOS shall always set this bit to a value of '1'**.

- **Ports Implemented (PI)** register. This register is located at ABAR + 0Ch, bits 3:0. This register provides an indication to system software as to which ports are implemented on the AHCI HBA. If a bit is set to '1', the corresponding port is available for software to use. If a bit is cleared to '0', the port is not available for software to use. The maximum number of bits set to '1' shall not exceed 4 (ports implemented >= 1, but < 4) as this is the maximum number of ports implemented by the ICH6.

- **Interlock Switch Attached to Port (ISP)** registers. This register is located in each implemented port's Command Register (PxCMD), bit 19. It provides an indication to system software as which ports support an interlock switch. BIOS shall properly program the GPIO_USE_SEL register whenever the platform implements an interlock switch. See section: *7.1 Device Insertion/Removal Detection Support in an AHCI Aware Environment Using Interlock Switches.* Programming this bit to '1' also requires that BIOS program the **HPCP** and **SIS** bits to a '1'.

- **Hot Plug Capable (HPCP)** registers. This register is located in each implemented port's Command Register (PxCMD), bit 18. It provides an indication to system software as to which ports support hot-plug insertion/removal. If set to '1', the platform has exposed the port to the user for hot plug insertion/removal. If cleared to '0', the platform has not exposed this port to the user for hot plug insertion/removal. If **ISP** (see above) is set to '1', then this bit should be set to '1'. If the device connected to this port is attached to the system chassis such that it is not hot pluggable then this bit should be cleared to '0'.

Note: PxCMD registers for each port are located as follows: (Port 0) P0CMD: ABAR + 118h, (Port 1) P1CMD: ABAR + 198h, (Port 2) P2CMD: ABAR + 218h, (Port 3) P3CMD: ABAR + 298h.

- **Aggressive Link Power Management Enable (ALPE)** register. This register is located in each implemented port's Command Register (PxCMD), bit 26. Programming of this register is optional. When set, the HBA will aggressively enter a lower link power state (partial or slumber) based upon the setting of the **ASP** bit. This bit shall be set/clear based on a platform's particular policy.

- **Aggressive Slumber/Partial (ASP)** register. This register is located in each implemented port's Command Register (PxCMD), bit 26. Programming of this register is optional. When this bit is set and **ALPE** is set, the HBA will aggressively enter the slumber state when it clears the PxCI register and the PxSACT register is cleared. When this bit is clear and **ALPE** bit is set, the HBA will aggressively enter the partial state when the HBA has completed any pending commands. This bit shall be set/clear based on a platform's particular policy.

§

Appendix B – Restoring AHCI Memory Mapped Space Registers and PCI Configuration Space Register When Resuming from S3

As per the PCI Power Management specification, when transitioning from the D3cold device power state (the AHCI is placed into a D3cold device power state whenever the system transitions into the S3 system power states) into the D0 power state, the device is placed into an un-initialized state. As such, it is the responsibility of system BIOS to restore certain AHCI memory mapped registers and well as certain PCI Configuration registers. The following is a list of those memory mapped and PCI Configuration space registers that must be restored (when applicable) by the system BIOS as a result of an S3->S0 system power state transition:

Note: The following memory mapped registers and bits *do not* need to be restored when the ICH6 is configured for IDE mode of operation (i.e. SCC = 01h).

- **Supports Command Queue Acceleration (SCQA)** bit.
- **Supports Interlock Switch (SIS)** bit.
- **Support Staggered Spin-up (SSS)** bit.
- **Supports Aggressive Link Power Management (SALP)** bit.
- **Supports Port Multiplier (PMS)** bit.
- **Slumber State Capable (SSC)** bit.
- **Partial State Capable (PSC)** bit.
- **Ports Implemented (PI)** register.
- **Interlock Switch Attached to Port (ISP)** registers.
- **Hot Plug Capable (HPCP)** registers.
- **Aggressive Link Power Management Enable (ALPE)** register.
- **Aggressive Slumber/Partial (ASP)** register.

Note: The following PCI Configuration registers shall always be restored, regardless of how the ICH6 is configured (i.e. applies all **SCC** values).

- **Offset 0Ah: CC – Class Code.** Refer to section: *5.2 Class Code Register (CC).*
- **Offset 90h: MAP – Port Mapping Register.** Refer to section: *4.2.3.1 Offset 90h: MAP – Port Mapping Register*.
- **Offset 92h: PCS – Port Control and Status.** Refer to section: *5.4 Port Control and Status Register (PCS).*

§