Minimal Boot Requirements for the Intel® Atom™ Processor E6xx Series

Application Note

May 2012

Order Number: 326973-001US

Contents

Tables

May 2012
Order Number: 326973-001US

Revision History

Date	Revision	Description
May 2012	001	Initial release

1.0 About This Document

This document provides information to assist customers/vendors new to Intel® architecture in supporting the Intel® Atom™ Processor E6xx Series.

This document describes the functions that the firmware must perform to enable correct operation of the platform and to boot the processor.

This document may 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 Terminology

Term	Description
ASPM	Active State Power Management
DDR2	Second generation Double Data Rate memory
DIMM	Dual In-line memory module
FSB	Front Side Bus
IGD	Internal Graphics Device
LCD	Liquid Crystal Display
LFP	Local Flat Panel
LVDS	Low Voltage Differential Signaling
Rank	Unique independently addressable 64-bit data area of a memory module
RCRB	Root Complex Register Block
SDVO	Serial Digital Video Out
SPD	Serial Presence Detect

1.2 Reference Documents

Document	Document No./Location
Intel® Atom™ Processor E6xx Series – Sightings Report (SR)	CDI #433308
Intel® Atom™ Processor E6xx Series – Specification Update - NDA	CDI #457843
Intel® Atom™ Processor E6xx Series Datasheet	Order #324208
Intel® Atom™ Processor E6xx Series Specification Update	Order #324209

1.3 Related Documents

Document	Document No./Location
Intel Corporation, MultiProcessor Specification	Version 1.4, Order #242016
Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture	Order #253665
Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 2A: Instruction Set Reference, A-M	Order #253666
Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 2B: Instruction Set Reference, N-Z	Order #253667
Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide Part 1	Order #253668
Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide Part 2	Order #253669
Intel® 82093AA I/O Advanced Programmable Interrupt Controller (I/O APIC) Datasheet	Order #290566
Intel® High Definition Audio Specification	Revision 1.0a
Compaq Computer Corporation, Phoenix Technologies, Ltd., Intel Corporation, Plug and Play BIOS Specification	Revision 1.0a, June 7, 2005
Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd., Toshiba Corporation, Advanced Configuration and Power Interface (ACPI)	Version 3.0a
PCI Express* Special Interest Group, PCI Express* Base Specification	Revision 1.1
PCI Express* Special Interest Group, PCI Express* Card Electromechanical Specification	Revision 1.1
PCI Special Interest Group, PCI BIOS Specification	Revision 2.1, August 26, 1994
PCI Special Interest Group, PCI Local Bus Specification	Revision 2.3, March 29, 2002
PCI Special Interest Group, PCI to PCI Bridge Architecture Specifications	Revision 1.2, June, 2003

1.4 Related Websites

Site	Context
http://www.microsoft.com/whdc/archive/pciirq.mspx	$PIRQ routing table information
http://www.intel.com/products/processor/manuals/	Intel® 64 and IA-32 Architectures Software Developer's Manuals
http://www.intel.com/standards/hdaudio/pdf/High_Definition_Audio_1_0a.pdf	*Intel® High Definition Audio Specification*
http://www.microsoft.com/whdc/system/default.mspx	List of power management specifications
http://www.intel.com/design/archives/processors/pro/docs/242016.htm	*MultiProcessor Specification*
http://www.intel.com/technology/iapc/acpi/	Power management, ACPI and related specifications

1.5 Formats and Notations

The target audience for this document is firmware writers. The formats and notations used within this document model those used by firmware vendors. This section describes the formatting and the notations that are followed in this document.

Table 1. Number Format and Notation

Number Format	Notation	Example
Decimal (default)	d	14d
Binary	b	1110b
Hex	h	0Eh

Table 2. Data Type Notation

Data Type	Notation	Size
BIT	b	Smallest unit, 0 or 1
BYTE	B	8 bits
WORD	W	16 bits or 2 bytes
DWORD	DW	32 bits or 4 bytes
QWORD	QW	8 bytes or 4 words
Kilobyte	kB	1024 bytes
Megabyte	MB	1,048,576 bytes
Gigabyte	GB	1024 MB

Table 3. Register Programming Table Abbreviations

Abbreviation	Meaning
B	PCI Bus
D	PCI Device
F	PCI Function
P	Msg Port
R	Register

This document refers to individual bit fields within a register, as well as the registers themselves with their designated acronym, followed by the device, register address and bit field in parenthesis. The reader is expected to be familiar with the register definitions for the processor.

The reader must also be capable of referencing the associated documentation described in Section 1.2 if a more detailed description of the register field is required.

When the document specifies individual register bits to be modified, system firmware must perform a read, modify, write sequence to ensure other bits within the register are not changed.

§ §

2.0 Minimum Boot Checklist

This chapter details the minimum steps necessary to boot using the processor.

2.1 Initialize the Non-Standard BARS

Before invoking the memory initialization sequence, the firmware must program a valid, non-conflicting PCI Express* base address and enable decoding of the range.

Table 4 specifies the base address registers in the processor (excluding PCI standard BARs), along with suggested values.

It is the responsibility of the firmware programmer to ensure that the ranges below do not conflict with any other resources on the platform.

Table 4. Non-Standard Base Address Registers

Region	Base Address Type	BAR Control	Size	Suggested Value
ACPI PM1 block	I/O	D31:F0:R48h	16B	80001000h (Address=1000h)
ACPI P block	I/O	Msg Port 4:R70h	16B (Must be located at PM1 block base + 10h)	80001010h (Address=1010h)
SM Bus	I/O	D31:F0:R40h	64B	80001040h (Address=1040h)
GPIO	I/O	D31:F0:R44h	64B	80001080h (Address=1080h)
GPE0	I/O	D31:F0:R4Ch	64B	800010C0h (Address=10C0h)
WDT	I/O	D31:F0:R84h	64B	80001100h (Address=1100h)
OSPMBA	I/O	Msg Port 4:R78h	64B	80001140h (Address=1140h)
SMM Control	Memory	Msg Port 2:R04h	1 MB	See Note below
Base of Stolen Memory	Memory	D02:F0:R5Ch	8 MB	If IGD is enabled: Top of memory – 8 MB (ex: 1 GB - 8 MB = 3F800000h) Else: Top of memory (ex: 1 GB = 40000000h)
PCI Express*	Memory	Msg Port 0:R00h	256 MB	E0000001h (Address=E0000000h)
PCI Express*	Memory	Msg Port 2:R09h	256 MB	E0000001h (Address=E0000000h)
RCBA	Memory	D31:F0:RF0h	16 kB	FED1C001h (Address=FED1C000h)

Note: Table 5 lists the layout of the Host SMM control register. The "Upper Bound" field of the HSMMCTL register must be programmed to match the upper 12 bits of "Top of Physical Memory – UMA Size – 1 MB".

For example, if 1 GB of memory is present in the system and internal graphics is enabled, then "Top of Physical Memory – UMA Size – 1 MB" = (40000000h – 800000h – 100000h = 3F700000h).

Table 5. Host SMM control (HSMMCTL) register layout

	Bit	Access	Default	Acronym	Description
Register on Message Bus Port 2 (HSMMCTL)					
04h	31:20	RW	000h	SMM End	This field defines the upper 1 MB aligned value of the protected SMM address range (SMM end). This value is derived from the upper 12 bits of "top of physical memory - UMA size -1 MB".
	19:16	RO	0b	Reserved	Reserved
	15:4	RW	000h	SMMStart	This field defines the lower 1 MB aligned value of the protected SMM address range (SMM start). This value must be programmed to the same value as "upper bound".
	3	RO	0b	Reserved	Reserved
	2	RW	1b	ALLOW_NON_SMM_WRITES_TO_SMM_SPACE_ SMMWRITESOPENSMMWRITESALLOWED	Allow non-SMM SW writes to SMM space
	1	RW	1b	SMMReadsAllowed	Allow non-SMM SW reads to SMM space
	0	RW	0b	SMMLocked	Locks this register and does not allow its fields to be changed until the system is reset. 0: Unlock the register 1: Lock the register

The "Upper Bound" field upper 12 bits (3F7h) are programmed into the "Upper Bound" field. The "Lower Bound" field must be programmed to the same value as "Upper Bound" (i.e. 3F7h). Bits 2:1 should be programmed to allow reads and writes to SMM space.

For the configuration in the above example, the value of the upper 28 bits (31:4) is 3F703F7h.

Additionally, the enhanced configuration space register must be configured. The PCI Express* specification defines a 256 MB block within the memory address space as PCI Express* configuration space addressable through a Bus:Device:Function mapping. The base address of this configuration space is determined by the value programmed in the "Enhanced Configuration Space" register, as shown in Table 6.

Table 6. Enhanced Configuration Space register

	Bit	Access	Default	Acronym	Description
Register offset on Message Bus Port 0					
00h	31:28	RW	0h	BASE	Specifies the 256MB aligned base address of the enhanced configuration (EC) register space. When enabled via EN in this register, memory accesses that occur within this 256MB region starting from BASE are translated into configuration cycles for internal devices and PCI-Express ports. The same value is required to program to Port 2 reg 9h bit 31:28. Recommended value 0xE0000001
	0	RW	0h	EN	When set, the 256MB range specified by BASE is enabled for enhanced configuration decode. When cleared, the range is not enabled. The same value is required to program to Port 2 reg 9h bit Port 2 Reg 9h bit 0.

Once initialized and enabled by firmware, software can use memory instructions to access the PCI Express* configuration space registers by byte, word or dword, though the access may not cross dword boundaries. To maintain the compatibility with the PCI configuration space, the first 256 bytes (offset 00h through FFh) of the configuration space for a Bus:Device:Function can also be accessed via the I/O index/data register pair at CF8h/CFCh as defined in PCI 2.x specification.

2.2 Program GPIO Configuration

Configure GPIO values according to system configuration.

2.3 Memory Configuration Straps

The Intel® Atom™ Processor E6xx Series contains an integrated 32-bit single channel memory controller that supports DDR2 memory in soldered down DRAM configurations only. The memory controller does not support SODIMMs or any other type of DIMMS.

The memory controller supports a data rate of 800 MT/s.

Traditional PCs rely on methods such as SMBus to determine memory configuration based on Serial Presence Detect (SPD) data. Due to the soldered down configuration of memory, the processor does not support SPD detection by means of SMBus.

The processor utilizes hardware straps to describe the soldered down DDR2 memory configuration. These bootstrap details are exposed to the system via the PCI configuration space of the LPC bridge - bits 7:4 of the Misc Control Register (B0:D31:F0:R5Ch), as described in Table 7.

Table 7. LPC Offset 5Ch: MC – Miscellaneous Control

Bit	Bit Description	Bit Reset Value	Bit Access
07	**Bootstrap MEMID3:** Bootstrap for memory controller configuration ID3. Defines the number of ranks enabled. 1: 1 Rank 0: 2 Rank	Strap	RO
06:05	**Bootstrap MEMID2:MEMID1:** Bootstrap for memory controller configuration ID2 and ID1. Defines the memory device densities. [MEMID2: MEMID1] 11: 2 Gb 10: 1 Gb 01: 512 Mb 00: 256 Mb	Strap	RO
04	**Bootstrap MEMID0:** Bootstrap for memory controller configuration ID0. Defines the memory device width. 0: x16 devices 1: x8 devices	Strap	RO

2.4 Initialize the Memory

Prior to accessing system memory, firmware must perform the following tasks:

1. Determine the memory configuration installed on the platform. The DRAM information required for successful initialization of memory is as listed in Table 8.

Table 8. Required DRAM Information

Attribute	Comment
DRAM frequency0	DRAM frequency is directly related to the Intel® Atom™ Processor E6xx Series frequency option
DRAM Type	DDR2
Device Width	Width of the individual DRAM chips
Device Density	Density of the individual DRAM chips
Number of ranks	1 or 2 ranks
tCL	CAS Latency
tRP	Pre-charge to Activate Delay
tRCD	Activate to CAS Delay
Refresh Rate	3.9 us 0r 7.8 us

2. Program the memory controller registers by means of the Message Bus based on the DRAM attributes noted in Step 1. The message bus registers are listed in Table 9:

   Message Control Register at Bus:0 Device:0 Function:0

   A write to this register issues a message on the GMCH internal message network with the fields specified by that write data. All byte enables must be enabled when writing this register

Table 9. Message Control Register at Bus:0 Device:0 Function:0

Size: 32 bit		**Default: 00h**		**Power Well:**
Access	**PCI Configuration B:D:F 0:0:0**			**Offset Start:** **Offset End:**
	Message Bus	**Port:**		**Register Address: D0h**
	Memory Mapped IO BAR			**Offset**
	Fixed IO	**Address**		
	Variable IO	**Base Address**		**Offset**

Bit	Access	Default	Acronym	Description
31:24	WO	00h	Message Opcode	Refer to Table 10
23:16	WO	0b0h	Message Port	Port ID
15:08	WO	000h	Message Target Register Address	Register offset
07:04	WO	00h	Message Write Byte Enables	Active high byte enables which enable each of the corresponding bytes in the MDR when high.
03:00	WO	00h	Reserved	

Table 10. Message Opcode

Default	Acronym		Description
A0h	DRAM JEDEC Init	MsgD	This message enables accessing the DRAM internal registers. Message Data Payload Bits: Bit 31:22 Reserved (set to 0) Bit 21 Rank Select: Determines the rank that will be the target of the initialization command when sent. Setting this field to 1 will target the command to rank 1, and a 0 will target the command to rank 0. Bit 20:6 These fields reflect bit 14:0 of MR/EMR/EMR1/ EMR3 as specified in DDR2 JEDEC spec, if bit 0-2 is 000 when the initialization command is sent. Bit 5:3 Mode Register Set Mode: 000-MR, 001-EMR1, 010-EMR2, 011-EMR3 when the initialization command is sent to DRAM Bit 2:0 Command The supported commands are listed below: 000 - (Extended) Mode Register Set 001 - Refresh 010 - Pre-charge (single or all as specified by MA[10]) 011 - Bank Activate 100 - Reserved 101 - Reserved 110 - Reserved 111 - NOP

Message Data Register at Bus:0 Device:0 Function:0

Provides the means to specify data to be written or retrieving data that was read due to a message operation. For messages with a data payload, MDR must be written with the data to be sent prior to a write to MCR. For messages that return data, MDR contains the data read after the write to MCR completes.

Table 11. Message Data Register at Bus:0 Device:0 Function:0

Size: 32 bit		**Default:** 00h		**Power Well:**
Access	**PCI Configuration B:D:F 0:0:0**			**Offset Start:** **Offset End:**
	Message Bus	**Port:**		**Register Address:** D4h
	Memory Mapped IO BAR			**Offset**
	Fixed IO	**Address**		
	Variable IO	**Base Address**		**Offset**
Bit	**Access**	**Default**	**Acronym**	**Description**
31:00	WO	00h	Message Data	

3. The registers in Table 12 can be programmed in any order. It is important only that these registers be programmed to valid values prior to performing JEDEC initialization of the DRAM components. The following registers must be programmed:

Table 12. Register offset on Message Bus Port 1 (Sheet 1 of 2)

	Bit	Access	Default	Acronym	Description
Register offset on Message Bus Port 1					
00h	12	RW	0b	DRAM_TYPE	0 - DDR2;
	6			RANK_1_ENABLED	Should be set to 1 when rank 1 is populated to enable the use of this rank. Otherwise this must be left cleared to 0.Rank 1 should only be Enabled when x16 DRAM devices are populated. When Rank 1 is enabled, the DRAM density populated to Rank 1 must equivalent to the ones in Rank 0. Rank 1 only supports DRAM device density of 512Mb, 1Gb and 2Gb.
	4:3				This sets the density of the DRAMs populated in rank 0:; 00 - 256Mb; 01 - 512Mb; 10 - 1Gb; 11 - 2Gb
	1	RW	0	RANK_0_ DEVICE_WIDTH	Indicates the width of the DRAMs populated in rank 0 0 - x8 Devices; 1 - x16 Devices
	0	RW	0	RANK_0_ENABLED	Should be set to 1 when rank 0 is populated to enable the use of this rank. Otherwise this must be left cleared to 0.

Table 12. Register offset on Message Bus Port 1 (Sheet 2 of 2)

	Bit	Access	Default	Acronym	Description
01h	10:9	RW	0h	CAS_LATENCY_CL	This specifies the delay from issuing a Read command until data return begins for a read. The delay from a Write command until data is sampled by the device is also related to this and is equal to CL-1. 00 - 3 DRAM Clocks (DDR2-400); 01 - 4 DRAM Clocks (DDR2-800, 667, 400); 10 - 5 DRAM Clocks (DDR2- 800, 667); 11 - 6 DRAM Clocks (DDR2-800)
	7:6	RW	0h	ACTIVATE_TO_CAS_ DELAY_TRCD	This specifies the delay required from when an Activate command is sent until a Read or a Write command may be sent to the same bank.; 00 - 3 DRAM Clocks (DDR2-400) 01 - 4 DRAM Clocks (DDR2-800, 667, 400) 10 - 5 DRAM Clocks (DDR2-800, 667); 11 - 6 DRAM Clocks (DDR2- 800)
	4:3		0h	PRECHARGE_TO_ ACTIVATE_DELAY_TRP	This specifies the delay required from when a Pre-charge command is sent until an Activate command may be sent to the same bank; 00 - 3 DRAM Clocks (DDR2-400); 01 - 4nDRAM Clocks (DDR2-800, 667, 400); 10 - 5 DRAM Clocks (DDR2-800, 667); 11 - 6 DRAM Clocks (DDR2-800)
	2:0		0h	DRAM_FREQUENCY	This specifies the frequency that is used for computing proper cycle to cycle timings. It has no control on the actual clock frequency. 000 - DDR2-400 001 - Reserved 010 - DDR2-667; 011 - DDR2-800; 100-111 are reserved.
02h	0000_0200h				
03h	Set bits [15:14] to 11b for refresh period Set bits [8] and [12] to 1b for ODT enable and rank swap Set bits [5:4]=2b for single rank Set bits [5:4]=0b for dual rank				
04 h	0000_0003h				
20h	0000_333Bh				
21h	1800_3018h				
23h	0000_1818h				
30h	Set bit [24] to 1b to indicate DDR2				
11h	0h				

4. Disable self-refresh by clearing bit 15:14 of port1 offset 3 to 0
5. Delay 200 us
6. Send a WAKE message to the memory controller by programming the Message Control Register at PCI (B0:D0:F0:RD0h) to 0201_00F0h.
7. Delay 3 microseconds.
8. For each DDR2 rank, initialize the DRAM components according to the JEDEC DDR2 SDRAM Specification (JESD79-2E, available at jedec.org) via message bus at B:0 D:0 F:0 with opcode A0h. Utilize 75 Ω ODT. Table 13 lists the reference JEDEC initialization sequence and value used.

Table 13. JEDEC initialization sequence

Programming Sequence	Description
1.	Apply NOP
2.	Delay minimum 400ns
3.	Apply pre-charge all
4.	Apply EMR2 to first rank, EMR2 bit A0-15 are 0.
5.	Apply EMR3 to first rank, EMR3 bit A0-15 are 0.
6.	Apply EMR1 to first rank, EMR1 bit A0-15 are 0x404.
7.	Apply MR to first rank, MR bit A0-15 are 0x362.
8.	Apply pre-charge all
9.	Apply refresh
10.	Apply refresh
11.	Apply MR to first rank, MR bit A0-15 are 0x262.
12.	Delay minimum 200 clocks
13.	Apply EMR1 to first rank, EMR1 bit A0-15 are 0x784.
14.	Apply EMR1 to first rank, EMR1 bit A0-15 are 0x404.
15.	Apply NOP.
16.	Delay minimum 400 ns
17.	Apply pre-charge all
18.	Apply EMR2 to second rank, EMR2 bit A0-15 are 0.
19.	Apply EMR3 to second rank, EMR3 bit A0-15 are 0
20.	Apply EMR1 to second rank, EMR1 bit A0-15 are 0x404.
21.	Apply MR to second rank, MR bit A0-15 are 0x362.
22.	Apply pre-charge all
23.	Apply refresh
24.	Apply refresh
25.	Apply MR to second rank, MR bit A0-15 are 0x262.
26.	Delay minimum 200 clock
27.	Apply EMR1 to second rank, EMR1 bit A0-15 are 0x784.
28.	Apply EMR1 to second rank, EMR1 bit A0-15 are 0x404.

9. Send a CLEAR FIFO message to the memory controller by programming the Message Control Register at PCI (B0:D0:F0:RD0h) to 4001_00F0h.
10. Program the self refresh bits via MsgBus (Port 01: Register 04h) [19, 17, 7] to 1b.
11. Set the Initialization Complete (IC) bit in the DRAM Controller Operation Register (DCO) register at MsgBus (Port 01: Register 03h) [31] to 1b.
12. Program the Refresh Period field in the DRAM Controller Operation Register (DCO) register at MsgBus (Port 01: Register 03h) [15:14] to 11b for a refresh period of 7.8 us.
13. Program the Host Memory Bound register (Port 02: Register 08h) [31:27] to represent the total memory installed in the system, as shown in Table 14.

Table 14. Host Memory Bound Register

[31:27]	Total System Memory
00010b	256 MB
00100b	512 MB
01000b	1 GB
10000b	2 GB

14. After completion of the above sequence, the DRAM is ready for normal operation.
15. If 0xE0000 or 0xF0000 segment is required, refer to Table 15 for enabling.

Table 15. Enabling segment 0xE0000 or 0xF0000

Register Offset on Message Bus Port 0	
03h	Set bit 2 : 1 to enable 0xF0000 segment Set bit 1 : 1 to enable 0xE0000 segment

2.5 CMC Relocation

After main memory has been either initialized or restored after leaving a suspend state, the CMC binary should be relocated into main system memory to increase system performance. The CMC binary relocation procedure is as follows:

1. Set Port 04h Reg 71h Bit 01h to 1 before CMC is copied from flash to system memory
2. Copy the 64K CMC binary from flash to main memory.
3. Now clear Port 04h Reg 71h Bit 01h to 0
4. Execute the WBINVD instruction to ensure the CMC binary copy is flushed from the processor cache.
5. Indicate to the chipset that the CMC binary has been copied via the DRAM Base Address Ready register [Opcode B8h:Port 04h, Register 00h]. The data payload should contain the upper 16 bits of the 32 bit DRAM address where the CMC's 64K block is located as in Table 16.

Table 16. DRAM Base Address Ready

Bit	Description
31:16	Bit 31:16 of the CMC's 64KB code block in DRAM
15:08	00h
07:00	00h

6. After this procedure is executed, the image located in main memory will be used by the system.

Note: For best system performance, the CMC should be relocated as early as possible after main memory is available.

7. After shadowing the CMC to system memory, enable the Intel® Atom™ Processor E6xx Series fencing mechanism via overlay message bus accessed at Bus:0 Dev:1 Function:0, to program Port 85h Reg 70h with value 1DFFFH.
8. Then read out the value of Bus:0 Dev:31 function:0 reg 5ch, bitwise and the value with FFFF3FFEH, write it back to Bus:0 Dev:31 function:0 reg 5ch.

9. Firmware must ensure that any OS software does not disturb the integrity of the CMC binary after it has been relocated. The 64K region that was reserved for the CMC binary must be reported as Reserved in the INT15h E820h memory map.

2.6 Shadow Firmware to System Memory

System memory is now ready and firmware can be shadowed to the system memory.

2.7 SMI

Relocate SMBase, clear SMI status bits at GPE0BASE + 14h, and enable SMIs.

2.8 Hide Unused PCI Function

The processor provides a mechanism for hiding the PCI Configuration space of the internal PCI devices. Devices that are not used in the system can be hidden before firmware loads the OS to prevent the OS from erroneously loading drivers. Devices should be hidden before firmware assigns system resources to prevent unused and unnecessary holes from being created in the system address space. Devices can be hidden by setting the proper bit in their respective PCI configuration space.

There are a few considerations when disabling internal devices of the processor:

1. When hiding (disabling) a function, only the configuration space is disabled by internal logic. Software must ensure that all functionality within the function (such as memory spaces, I/O spaces, and DMA engines) is disabled and PCI command register 04h is cleared prior to disabling the function.
2. The PCI specification requires that a multi-function device must implement function 0. As a general rule, firmware must ensure that if function 0 of a multi-function device is disabled (hidden) via the Function Disable register, then all the other functions of the same device must be disabled as well to be compliant with PCI specification.
3. The function disable register for a device resides in the PCI configuration space of that device, and the PCI configuration space is disabled once the device is disabled. So once a device is disabled it cannot be re-enabled until a system reset is performed.

2.9 Chipset Registers that Need To Be Initialized Before PCI Enumeration

Before PCI enumeration, the below static chipset registers need to be programmed to a fixed value during POST. However it does not include a list of registers/bit-fields whose values may change based on platform configuration.

Table 17. Chipset Registers to be Initialized

Name	Access Type	Address [Bits]	Value
PCIe* Slot Capabilities	PCI	D23:F0:R54h[31:0]	80500h
PCIe* Slot Capabilities	PCI	D24:F0:R54h[31:0]	100500h
PCIe* Slot Capabilities	PCI	D25:F0:R54h[31:0]	180500h
PCIe* Slot Capabilities	PCI	D26:F0:R54h[31:0]	200500h

2.10 Perform PCI Enumeration

System firmware is required to perform standard PCI/PnP enumeration and initialize all

PCI devices/functions discovered. As part of this process, all PCI-to-PCI bridges (Device 23- 26:F0) must be initialized with secondary/subordinate bus numbers assigned and bridge windows programmed properly. In addition, I/O, memory, and interrupt resource allocation to PCI devices should be completed.

Note: MMIO BARs MUST NOT be mapped below physical address 80000000h. If less than 2GB of system memory is populated, firmware must adjust its PCI enumeration algorithm to ensure that MMIO BARs are not mapped between top of memory and physical address 7FFFFFFFh. Firmware must also reserve the range from the top of memory through 7FFFFFFFh via PNP device node and ACPI _CRS. This reservation ensures that the operating system will not relocate BARs below 2 GB.

2.11 Enable Thermal Monitor 1 or Thermal Monitor 2

TM1 is enabled by setting the thermal-monitor enable flag (bit 3) in IA32_MISC_ENABLE.

TM2 is enabled if the TM_SELECT flag (bit 16) of the MSR_THERM2_CTL register is set to 1 and bit 3 of the IA32_MISC_ENABLE register is set to 1.

3.0 Conclusion

In summary, this document supplements the information provided in the *Intel® Atom™ Processor E6xx Series Datasheet* for use by bootloader/firmware vendors and Intel customers developing their own bootloader/firmware using the Intel® Atom™ Processor E6xx Series. Using the boot checklist in this document, readers can ensure smooth functioning of the platform and boot the processor.

§ §