IA-32 Intel® Architecture Software Developer's Manual Documentation Changes

Release Date: November 2002

Order Number: 252046-001

CONTENTS

REVISION HISTORY

Date of Revision	Version	Description
November 2002	-001	Initial Release

PREFACE

This document is an update to the information contained in the following document:

- *IA-32 Intel® Architecture Software Developer's Manual:*
  - *Volume 1, Basic Architecture* (Order Number 245470-008)
  - *Volume 2, Instruction Set Reference* (Order Number 245471-008)
  - *Volume 3, System Programming Guide* (Order Number 245472-008)

It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools and contains Documentation Changes.

Nomenclature

Documentation Changes include typographical errors, errors, or omissions from the current published document. These changes will be incorporated in the next release of the document.

Documentation Changes for the IA-32 Intel® Architecture Software Developer's Manual

SUMMARY OF CHANGES

The following table indicates Documentation Changes that apply to the IA-32 Intel® Architecture Software Developer's Manual. This table uses the following notations:

CODES USED IN SUMMARY TABLE

Shaded:	This item is either new or modified from the previous version of the document.

Summary of Documentation Changes

NO.	DOCUMENTATION CHANGES
1	Inaccurate Operation Description for Shift Instructions
2	Documentation Changes to Support the Use of Hyper-Threading Technology
3	IRET/IRETD TASK-RETURN Block has Incorrect Exception Information
4	LGDT Exception Listing in Virtual-8086
5	Various Table B-22 Errors and Omissions
6	Misreporting of Exceptions for MOVDQA
7	Effect of STI on NMI
8	133 MHz PSB Encoding in MSR_EBC_FREQUENCY_ID Register
9	Double Fault Exception
10	Move To/From Control Registers

DOCUMENTATION CHANGES

All Documentation Changes will be incorporated into a future version of the *IA-32 Intel© Architecture Software Developer's Manual*.

1. Inaccurate Operation Description for Shift Instructions

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* Section 3.2 "Instruction Reference" Operation description for the SAL/SAR/SHL/SHR instruction is incorrect. It currently states:

```
(* Determine overflow for the various instructions *)
IF COUNT = 1
        THEN
                IF instruction is SAL or SHL
…
ELSE IF COUNT = 0
        THEN
                All flags remain unchanged;
```

It should state:

```
(* Determine overflow for the various instructions *)
IF (COUNT AND 1FH) = 1
        THEN
                IF instruction is SAL or SHL
…
ELSE IF (COUNT AND 1FH) = 0
        THEN
                All flags remain unchanged;
```

2. Documentation Changes to Support the Use of Hyper-Threading Technology

The *Intel Architecture Software Developer's Manual, Vol 1: Basic Architecture* Chapter 2, and *Vol 3: System Programming Guide* Chapter 7:

Both chapters have been rewritten to include Hyper-Threading Technology terminology and descriptions.

3. IRET/IRETD TASK-RETURN Block has Incorrect Exception Information

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* Chapter 3, Section 3.2 "Instruction Reference" incorrectly describes the TASK-RETURN block of the OPERATION section of the IRET/IRETD instruction. It currently states:

```
IF local/global bit is set to local
        OR index not within GDT limits
                THEN #GP(TSS selector);
…

IF TSS descriptor type is not TSS or if the TSS is marked not busy
        THEN #GP(TSS selector);
```

It should state:

```
IF local/global bit is set to local
        OR index not within GDT limits
                THEN #TS(TSS selector);
…

IF TSS descriptor type is not TSS or if the TSS is marked not busy
        THEN #TS(TSS selector);
```

4. LGDT Exception Listing in Virtual-8086

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* Chapter 3, Section 3.2 "Instruction Reference" incomplete listing of LGDT/LIDT in Virtual-8086 Mode Exceptions. It currently states:

Virtual-8086 Mode Exceptions

#GP(0) The LGDT and LIDT instructions are not recognized in virtual-8086 mode.

It should state:

Virtual-8086 Mode Exceptions

#GP(0) The LGDT and LIDT instructions are not recognized in virtual-8086 mode.

#GP If the current privilege level is not 0.

5. Various Table B-22 Errors and Omissions

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* APPENDIX B, Section B.7 "Floating-point Instruction Formats and Encoding" various errors and omission in Table B-22. It currently states:

FICOMP-Compare Integer and Pop
16-bit memory
32-bit memory
FIDIV
ST(0) ← ST(0) + 16-bit memory
ST(0) ← ST(0) + 32-bit memory
FIDIVR
ST(0) ← ST(0) + 16-bit memory
ST(0) ← ST(0) + 32-bit memory
.
.
.
FIMUL
ST(0) ← ST(0) + 16-bit memory
ST(0) ← ST(0) + 32-bit memory
.
.
.
FISUB
ST(0) ← ST(0) + 16-bit memory
ST(0) ← ST(0) + 32-bit memory

FISUBR
ST(0) ← ST(0) + 16-bit memory
ST(0) ← ST(0) + 32-bit memory
.
.
.
FMULP – Multiply
ST(0) ← ST(0) X ST(i)
.
.
.

It should state:

FICOMP-Compare Integer and Pop
16-bit memory
32-bit memory
FIDIV
ST(0) ← ST(0) ÷ 16-bit memory
ST(0) ← ST(0) ÷ 32-bit memory
FIDIVR
ST(0) ← 16-bit memory ÷ ST(0)
ST(0) ← 32-bit memory ÷ ST(0)
.

.
.
FIMUL
ST(0) ← ST(0) × 16-bit memory
ST(0) ← ST(0) × 32-bit memory
.
.
.
FISUB
ST(0) ← ST(0) - 16-bit memory
ST(0) ← ST(0) - 32-bit memory
FISUBR
ST(0) ← 16-bit memory - ST(0)
ST(0) ← 32-bit memory - ST(0)
.
.
.
FMULP – Multiply
ST(i) ← ST(0) × ST(i)
.
.
.

6. Misreporting of Exceptions for MOVDQA

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* Section 3.2, Instruction Reference, under MOVDQA currently states:

Protected Mode Exceptions

#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit. If memory operand is not aligned on a 16-byte boundary, regardless of segment.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#NM	If TS in CR0 is set.
#UD	If EM in CR0 is set. If OSFXSR in CR4 is 0.

Real-Address Mode Exceptions

#PF(fault-code)	If a page fault occurs.
#GP(0)	If memory operand is not aligned on a 16-byte boundary, regardless of segment. If any part of the operand lies outside of the effective address space from 0 to FFFFH.
#NM	If TS in CR0 is set.
#UD	If EM in CR0 is set. If OSFXSR in CR4 is 0.

It should state:

Protected Mode Exceptions

#PF(fault-code)	If a page fault occurs.
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit. If memory operand is not aligned on a 16-byte boundary, regardless of segment.

#SS(0)	If a memory operand effective address is outside the SS segment limit.
#NM	If TS in CR0 is set.
#UD	If EM in CR0 is set. If OSFXSR in CR4 is 0.

Real-Address Mode Exceptions

#GP(0)	If memory operand is not aligned on a 16-byte boundary, regardless of segment. If any part of the operand lies outside of the effective address space from 0 to FFFFH.
#NM	If TS in CR0 is set.
#UD	If EM in CR0 is set. If OSFXSR in CR4 is 0.

7. Effect of STI on NMI

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* Section 3.2, Instruction Reference, under STI-Set Interrupt Flag currently states:

> The IF flag and the STI and CLI instructions have no affect on the generation of exceptions and NMI interrupts.

It should state:

> The IF flag and the STI and CLI instructions have no affect on the generation of exceptions. NMI interrupts may be blocked for one macroinstruction following an STI.

8. 133 MHz PSB Encoding in MSR_EBC_FREQUENCY_ID Register

The *Intel Architecture Software Developer's Manual, Vol 3: System Programming Guide* Appendix B, Table B-1, the MSR_EBC_FREQUENCY_ID register bits [18:16] currently states:

2CH	44	MSR_EBC_FREQUENCY_ID 20:0 18:16 63:24	Shared	Processor Frequency Configuration. (R) Indicates current processor frequency configuration. **Reserved.** **Scalable Bus Speed.** (R/W) Indicates the intended scalable bus speed: Encoding Scalable Bus Speed 000B 100 MHz All Others Reserved **Reserved.**

It should state:

2CH	44	MSR_EBC_ FREQUENCY_ID	Shared	Processor Frequency Configuration. The bit field layout of this MSR varies according to the MODEL value of the CPUID version information. The following bit field layout applies to Pentium 4 and Xeon Processors with MODEL encoding equal or greater than 2. (R) Indicates current processor frequency configuration.
		15:0		**Reserved.**
		18:16		**Scalable Bus Speed.** (R/W) Indicates the intended scalable bus speed: Encoding Scalable Bus Speed 000B 100 MHz 001B 133 MHz (The actual value of 133.33MHz should be utilized if performing calculation with System Bus Speed, when encoding is 001B.) All Others Reserved
		63:29		**Reserved.**

9. Double Fault Exception

The *Intel Architecture Software Developer's Manual, Vol 3: System Programming Guide* Chapter 5, Section 5.12 "Exception and Interrupt Reference" Interrupt 8 - Double Fault Exception (#DF), the double fault handler description currently states:

> If the shutdown occurs while the processor is executing an NMI interrupt handler, then only a hardware reset can restart the processor.

It should state:

> If the shutdown occurs while the processor is executing an NMI interrupt handler, then only a hardware reset can restart the processor. Likewise, if the shutdown occurs while executing in SMM, a hardware reset should be used to restart the processor.

10. Move To/From Control Registers

The *Intel Architecture Software Developer's Manual, Vol 2: Instruction Set Reference* Section 3.2 "Instruction Reference" Operation description for the Move to/from Control Register instruction is incorrect. It currently states:

> When loading a control register, a program should not attempt to change any of the reserved bits; that is, always set reserved bits to the value previously read.

It should state:

> If a load of CR4 would set any of its reserved bits, a general-protection fault occurs. The reserved bits in CR0 and CR3 remain clear after any load of those registers, although attempts to set such bits do not cause faults. On Intel Pentium 4, Intel Xeon, and P6 family processors, CR0.ET remains set after any load of CR0, although attempts to clear it do not cause faults. When loading a control register, a program should always set reserved bits to the value previously read.