Execute Disable Bit Functionality Blocks Malware Code Execution

Introduction

Execute Disable Bit capability is an enhancement to 32-bit Intel® architecture. An IA-32 processor with Execute Disable Bit capability can protect data pages against being used by malicious software to execute code. The processor provides page protection in either of the following modes:

- Legacy protected mode, if Physical Address Extension (PAE) is enabled.
- IA-32e mode, when Intel® Extended Memory 64 Technology (Intel® EM64T) is enabled.

Note that entering IA-32e mode requires enabling PAE. While the Execute Disable Bit capability does not introduce new instructions, it does require operating systems to operate in a PAE-enabled environment and to establish a page-granular protection policy for memory.

Execute Disable Bit Capability Overview

Software can detect the presence of the Execute Disable Bit capability using the CPUID instruction with the input value 80000001H in EAX. Presence is indicated by a value returned in EDX. If bit 20 of EDX is set, the Execute Disable Bit is available.

If CPUID extended function 80000001H reports that Execute Disable Bit capability is available and PAE is enabled, software can enable the Execute Disable Bit capability by setting the NXE bit to 1 in IA32_EFER MSR (address C0000080H). IA32_EFER is available if bit 20 or bit 29 of the EDX register returned by CPUID-extended function 80000001H is 1.

When Physical Address Extension is enabled (either in IA-32e mode or in legacy protected mode), Execute Disable Bit capability is enabled by setting bit 11 of IA32_EFER to 1. If CPUID extended function 80000001H reports Execute Disable Bit capability is not available, bit 11 of IA32_EFER is reserved. A write to IA32_EFER.NXE will produce a #GP exception.

Table 1. Extended Feature Enable MSR (IA32_EFER):

63:12	**11**	**10**	**9**	**8**	**7:1**	**0**
Reserved	Execute Disable Bit Enable (NXE)	IA-32e mode Active (LMA)	Reserved	IA-32e mode Enable (LME)	Reserved	SysCall Enable (SCE)

Execute Disable Bit capability introduces no new instructions. It enhances PAE-enabled paging operations. If the capability is enabled (IA32_EFER.NXE = 1), then a new attribute bit (referred to as the Execute Disable Bit) is defined in paging structures for address translation. If the Execute Disable Bit of a memory page is set to one, that page can be used only as data. An

attempt to execute code from a memory page with the Execute Disable Bit set to 1 will cause a page-fault exception.

The page sizes and physical address sizes supported by Execute Disable Bit capability are shown in Table 2. Existing page-level protection mechanisms continue to apply to memory pages, independent of the Execute Disable Bit setting for that memory page.

Table 2. Paging Data Structures:

PG Flag, CR0	PAE Flag, CR4	PS Flag, PDE	CPUID Feature Flag ECX[IA-32e]	Page Size	Physical Address Size
1	1	0	0	4 KBytes	**implementation specific**
1	**1**	**1**	**0**	**2 MBytes**	**implementation specific**
1	1	0	1	4 KBytes	40 Bits
1	**1**	**1**	**1**	**2 MBytes**	**40 Bits**

For IA-32 processors that support Intel EM64T, paging data structures include a new table in the page-translation hierarchy. The table is called the Page Map Level 4 (PML4) table. The PML4 table sits above the Page Directory Pointer (PDP) table in the page-translation hierarchy. It is used in page translation only when IA-32e mode is activated. It is not used when IA-32e mode is disabled, regardless of whether or not PAE is enabled.

The interaction between the Execute Disable Bit capability and linear-to-physical address translation mechanisms (for IA-32e mode and legacy modes) are described separately.

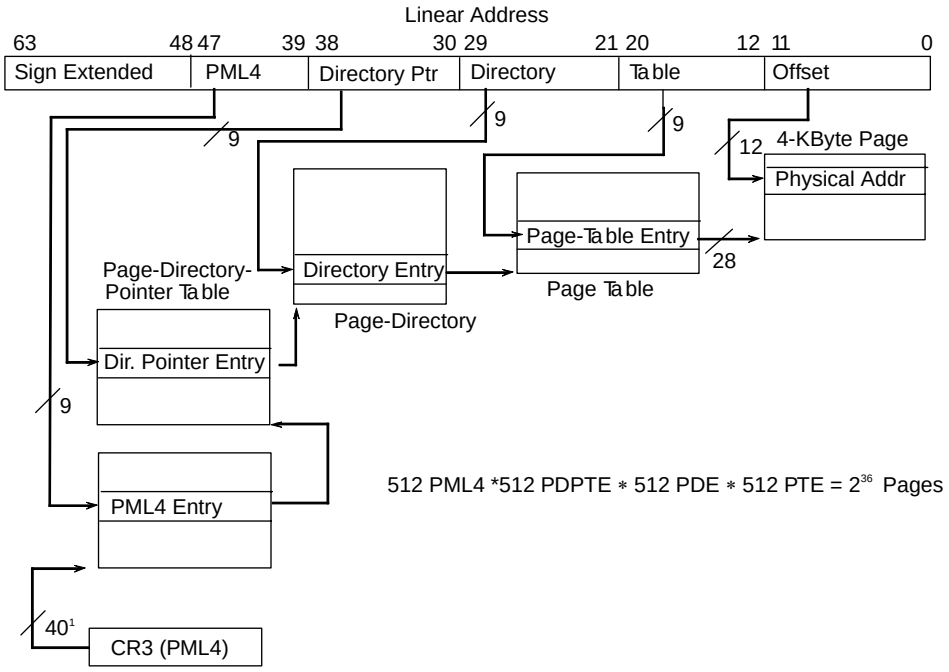
Linear Address Translation in IA-32e Mode

Intel EM64T expands Physical Address Extension (PAE) paging structures to potentially support mapping a 64-bit linear address to a 52-bit physical address. In the first implementation of Intel EM64T, PAE paging structures are extended to support translation of a 48-bit linear address to a 40-bit physical address. The page-translation hierarchy is shown in Figure 1.

PML4 contains 512 eight-byte entries, with each entry pointing to a Page Directory Pointer (PDP) table. Nine linear-address bits are used to index into PML4. The existing page-directory pointer table is expanded in Intel EM64T to 512 eight-byte entries from four entries. As a result, nine bits of the linear address are used to index into a PDP table, rather than two bits.

The size of Page-Directory Entry (PDE) tables and Page-Table Entry (PTE) tables remains at 512 eight-byte entries, each indexed by nine linear-address bits. The total of linear-address index bits into the collection of paging data structures (PML4 + PDP + PDE + PTE + page offset) defined above is 48. The method for translating the high-order 16 linear-address bits into a physical address is currently reserved.

Figure 1. IA-32e Mode Paging Structures (4K pages):



NOTES:
1. 40 bits aligned onto a 4K-Byte boundary

The PS flag (Bit 7) in the Page Directory Entry (PDE.PS) selects between 4KB and 2MB page sizes. Because PDE.PS is used to control large page selection, the CR4.PSE bit is ignored. Tables 3 through 6 show the 64-bit mode PML4, PDP, PDE, and PTE formats when 4KB pages are enabled.

Table 3. IA-32e Mode Page Map Level 4 Entry (PML4 - 4KB Pages):

63	**62:52**	**51:40**	**39:12**	**11:9**	**8:7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Execute Disable Bit or Reserved	Available	Reserved	Page Directory Pointer Base Address	Available	Reserved	Reserved [see note 1]	A	PCD	PWT	U/S	R/W	P

Table 4. IA-32e Mode Page Directory Pointer Table Entry (PDPTE - 4KB Pages):

63	**62:52**	**51:40**	**39:12**	**11:9**	**8:7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Execute Disable Bit or Reserved	Available	Reserved	Page Directory Base Address	Available	Reserved	Reserved [see note 1]	A	PCD	PWT	U/S	R/W	P

Table 5. IA-32e Mode Page Directory Entry (PDE - 4KB Pages):

63	62:52	51:40	39:12	11:9	8	7	6	5	4	3	2	1	0
Execute Disable Bit or Reserved	Available	Reserved	Page Table Base Address	Available	Reserved[1]	0	Reserved [see note 1]	A	PCD	PWT	U/S	R/W	P

Table 6. IA-32e Mode Page Table Entry (PTE - 4KB Pages):

63	62:52	51:40	39:12	11:9	8	7	6	5	4	3	2	1	0
Execute Disable Bit or Reserved	Available	Reserved	Page Base Address	Available	G	PAT	D	A	PCD	PWT	U/S	R/W	P

The physical base-address field in all four table-entry formats is extended by Intel EM64T to bits 51:12. This allows paging tables to be located anywhere in the physical memory supported by a 64-bit implementation. Implementations that do not support the maximum physical-address size reserve the unsupported high-order bits and require that they be cleared to zeros. The physical base-address field in the first implementation of Intel EM64T is specified by bits 39:12.

Software should take care not to locate paging tables above 4G in memory if it is anticipated that mode changes to legacy mode will be needed.

Bit 63 is either reserved or the execute-disable bit, depending whether the Execute Disable Bit capability in PAE is enabled. If PAE's Execute Disable Bit capability is enabled (IA32_EFER.NXE = 1), bit 63 is the execute-disable bit. If PAE's Execute Disable Bit capability is not enabled, bit 63 is reserved.

Bits 62:52 in all page-table entry formats are available for use by system software. In the 64-bit extensions architecture, future implementations leave bits 62:52 available for software use. Other than the extensions made to the base-address field and the addition of the software-available field at bits 62:52, all other PDE and PTE fields are the same as in legacy mode.

Fields within the PDP table entry are similar to legacy-mode PDP table entries with the following exceptions; these reflect the changes necessary to indicate that a higher-level paging structure (PML4) now references the PDP tables:

- Bit 0 is no longer reserved. IA-32e mode defines this bit as the present (P) flag to indicate whether or not the PDE table referenced by the PDP entry is currently stored in physical memory. A page-fault exception (#PF) is generated when the processor accesses a PDP entry with the P flag cleared to 0.
- Bit 1 is no longer reserved. IA-32e mode defines this bit as the read/write (R/W) flag.
- Bit 2 is no longer reserved. IA-32e mode defines this bit as the user/supervisor (U/S) flag.
- Bit 5 is no longer reserved. IA-32e mode defines this bit as the accessed (A) flag.
- The base-address field extensions, as specified above.
- Bits 62:52 are available to software, as specified above. The format of a PML4 table entry is identical to the 64-bit mode PDP table-entry format.
- Bit 63 is the execute-disable bit if IA32_EFER.NXE = 1, otherwise reserved.

Address Translation for 2-Mbyte pages in IA-32e Mode

Tables 7 through 9 show the 64-bit mode PML4, PDP, and PDE formats when 2MB pages are enabled. As with legacy mode, 2MB pages are enabled by setting the PDE page-size bit to 1 (PDE.PS = 1). Page table is not used in address translation for 2MB pages. Control of 2MB page sizes does not depend on CR4.PSE.

Table 7. IA-32e Mode Page Map Level 4 Entry (PML4 - 2MB Pages):

63	**62:52**	**51:40**	**39:12**	**11:9**	**8:7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Execute Disable Bit or Reserved	Available	Reserved	Page Directory Pointer Base Address	Available	Reserved	Reserved [see note 1]	A	PCD	PWT	U/S	R/W	P

Table 8. IA-32e Mode Page Directory Pointer Table Entry (PDPTR - 2MB Pages):

63	**63:52**	**51:40**	**39:12**	**11:9**	**8:7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Execute Disable Bit or Reserved	Available	Reserved	Page Directory Base Address	Available	Reserved	Reserved [see note 1]	A	PCD	PWT	U/S	R/W	P

Table 9. IA-32e Mode Page Directory Entry (PDE - 2MB Pages):

63	62:52	51:40	39:21	20:13	12	11:9	8	7	6	5	4	3	2	1	0
Execute Disable Bit or Reserved	Available	Reserved	Page Base Address	Reserved	PAT	Available	G	1	D	A	PCD	PWT	U/S	R/W	P

The physical base-address field in all three table-entry formats is extended by the 64-bit extension architecture to bits 51:12. This allows paging tables to be located anywhere in the physical memory supported by a 64-bit-mode implementation. Implementations that do not support the maximum physical-address size reserve the unsupported high-order bits and require that they be cleared to zeros.

The physical base-address field in the first implementation of Intel EM64T is specified by bits 39:12. Bits 63:52 in all page-table entry formats are available for use by system software. In the 64-bit extension architecture, future implementations will leave bits 62:52 available for software use. When 2MB pages are selected, the PDE points directly to the physical page, and not to a PTE. Other than the extensions made to the base-address field and the addition of the software-available field at bits 62:52, all other PDE fields are the same as in legacy mode.

Fields within the PDP table entry are similar to legacy-mode PDP table entries with the following exceptions; the exceptions reflect changes necessary to indicate that a higher-level paging structure (PML4) now references the PDP tables:

- Bit 0 is no longer reserved. IA-32e mode defines this bit as the present (P) flag to indicate whether or not the PDE table referenced by the PDP entry is currently stored in physical memory. A page-fault exception (#PF) is generated when the processor accesses a PDP entry with the P flag cleared to 0.

- Bit 1 is no longer reserved. IA-32e mode defines this bit as the read/write (R/W) flag.

- Bit 2 is no longer reserved. IA-32e mode defines this bit as the user/supervisor (U/S) flag.

- Bit 5 is no longer reserved. IA-32e mode defines this bit as the accessed (A) flag.

- The base-address field extensions, as specified above.

- Bits 62:52 available to software, as specified above.

- Bit 63 is the execute-disable bit if IA32_EFER.NXE = 1, otherwise reserved.

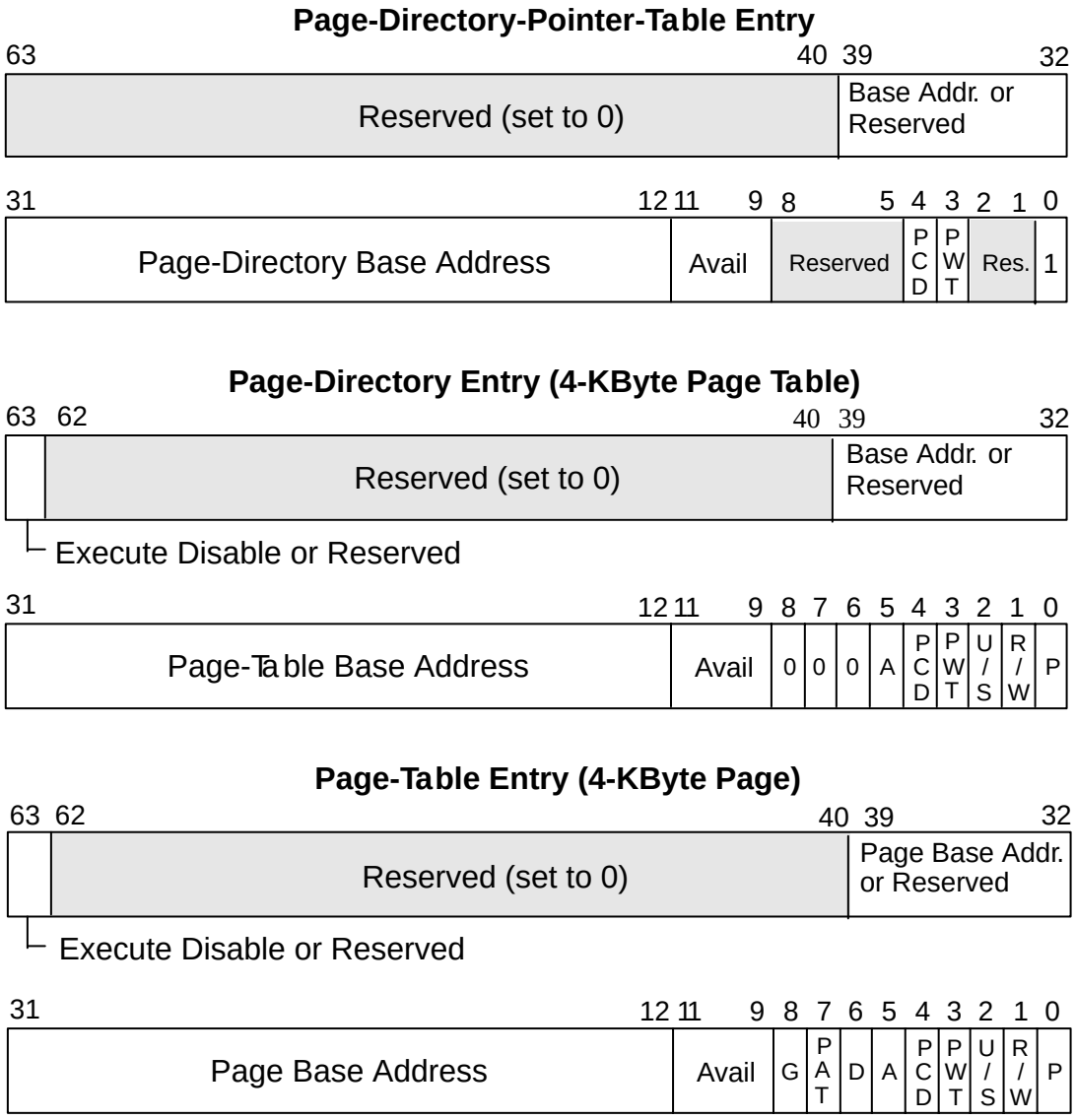
The format of a PML4 table entry is identical to the 64-bit mode PDP table-entry format.

Legacy PAE-Enabled Address Translation

Execute Disable Bit capability is also supported in legacy PAE-enabled paging operation. The address-translation mechanism is the same as documented in the IA-32 Intel Architecture Software Developer's Manual, Volume 3. In legacy mode, PML4 is not available, and page-directory pointer table has four entries. For 4KB pages, bit 63 of each entry in the page table and page directory is the execute-disable bit if IA32_EFER.NXE = 1. Otherwise, bit 63 is reserved. For processors that support 40-bit physical address space, bits 39:12 of each entry contains a base address. For processors that support 36 (or 32)-bit physical address space, bits 35:12 (or 31:12) of each entry contains a base address.

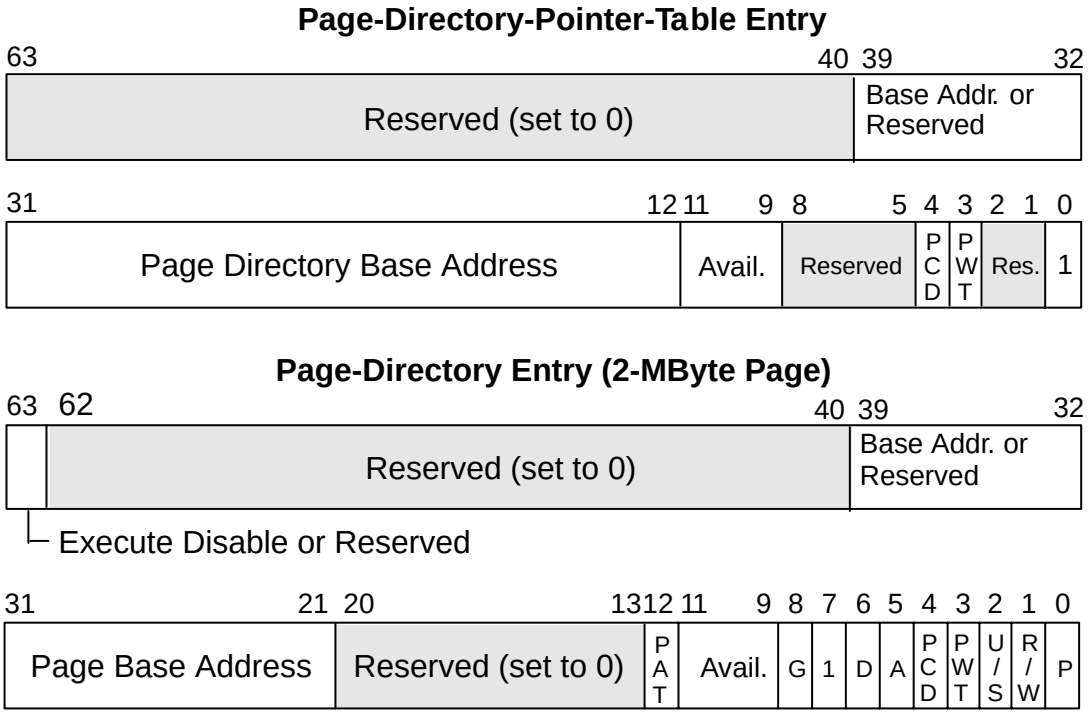
Note: In PAE mode, whether page-directory-pointer table entries are cached or not is model-specific.

Figure 2. Format of Page-Directory-Pointer-Table, Page-Directory, and Page-Table Entries for 4-KByte Pages with PAE Enabled:



For 2MB pages, the page table is not used in address translation. Bit 63 of each entry in the page directory is the execute-disable bit if IA32_EFER.NXE = 1. Otherwise, bit 63 is reserved. For processors that support 40-bit physical address space, bits 39:12 of each entry contains a base address. For processors that support 36 (or 32)-bit physical address space, bits 35:12 (or 31:12) of each entry contains a base address.

Figure 3. Format of Page-Directory-Pointer-Table and Page-Directory Entries for 2-MByte Pages with PAE Enabled:



Execute Disable Bit Page Protection

The Execute Disable Bit in the paging structures enhances page protection for data pages. Memory pages that contain data cannot be used to execute code if IA32_EFER.NXE =1 and the execute-disable bit of the memory page is set. Table 10 shows the effect of setting the Execute Disable Bit in IA-32e mode for code and data pages.

Table 10. IA-32e Mode Page Level Protection Matrix with Execute Disable Bit Capability (continued):

Execute Disable Bit Value (Bit 63)				**Valid Usage**
PML4	PDP	PDE	PTE	
Bit 63=1	*	*	*	Data
*	Bit 63=1	*	*	Data
*	*	Bit 63=1	*	Data
*	*	*	Bit 63=1	Data
Bit 63=0	Bit 63=0	Bit 63=0	Bit 63=0	Data/Code

In legacy PAE-enabled mode, Tables 11 and 12 show the effect of setting the execute-disable bit for code and data pages.

Table 11. Legacy PAE-enabled 4KB Page Level Protection Matrix with Execute Disable Bit Capability (continued):

Execute Disable Bit Value (Bit 63)		Valid Usage
PDE	PTE	
Bit 63=1	*	Data
*	Bit 63=1	Data
Bit 63=0	Bit 63=0	Data/Code

Table 12. Legacy PAE-enabled 2MB Page Level Protection with Execute Disable Bit Capability

Execute Disable Bit Value (Bit 63	Valid Usage
PDE	
Bit 63=1	Data
Bit 63=0	Data/Code

Reserved Bit Checking

The processor enforces reserved bit checking in the paging-data-structure entries. The identity of the reserved bits being checked varies with different paging mode and may vary if the size of physical address space varies. Table 13 shows the reserved bits that are checked when the Execute Disable Bit capability is enabled (i.e., CR4.PAE = 1 and IA32_EFER.NXE= 1). In legacy paging modes:

- Non-PAE 4KB paging: 4KB-page only paging (CR4.PAE = 0, CR4.PSE = 0)
- PSE36: 4KB and 4MB pages (CR4.PAE = 0, CR4.PSE = 1)
- PAE: 4KB and 2MB pages (CR4.PAE = 1, CR4.PSE = x).

In legacy PAE-enabled paging, some processors may only support 36- (or 32)-bit physical address size, and reserved bit checking also applies to bit 39:36 (or 39:32).

Table 13. Reserved Bit Checking When Execute Disable Bit Capability is Enabled (continued):

Mode	Paging Mode	Check Bits
Legacy	4KB paging (non-PAE)	No reserved bits checked
	PSE36 - PDE, 4MB page	Bit [21]
	PSE36 - PDE, 4KB page	No reserved bits checked
	PSE36 - PTE	No reserved bits checked
	PAE - PDP table entry	Bits [63:40] & [8:5] & [2:1] [see note 2]
	PAE - PDE, 2MB page	Bits [62:40] & [20:13] [1]
	PAE - PDE, 4KB page	Bits [62:40] [1]
	PAE - PTE	Bits [62:40] [1]
64-bit	PML4E	Bits [51:40]
	PDPTE	Bits [51:40]
	PDE, 2MB page	Bits [51:40] & [20:13]
	PDE, 4KB page	Bits [51:40]
	PTE	Bits [51:40]

If Execute Disable Bit capability is not enabled or not available, reserved bit checking in 64-bit mode includes bit 63 and the bits described in Table 14.

Table 14. Reserved Bit Checking When Execute Disable Bit Capability is not Enabled:

Mode	Paging Mode	Check Bits
Legacy	KB paging (non-PAE)	No reserved bits checked
	PSE36 - PDE, 4MB page	Bit [21]
	PSE36 - PDE, 4KB page	No reserved bits checked
	PSE36 - PTE	No reserved bits checked
	PAE - PDP table entry	Bits [63:40] & [8:5] & [2:1] [see note 2]
	PAE - PDE, 2MB page	Bits [63:40] & [20:13][1]
	PAE - PDE, 4KB page	Bits [63:40] [1]
	PAE - PTE	Bits [63:40] [1]
64-bit	PML4E	Bit [63], bits [51:40]
	PDPTE	Bit [63], bits [51:40]
	PDE, 2MB page	Bit [63], bits [51:40] & [20:13]
	PDE, 4KB page	Bit [63], bits [51:40]
	PTE	Bit [63], bits [51:40]

Exception Handling

When Execute Disable Bit capability is enabled (IA32_EFER.NXE = 1), conditions for a page fault to occur include the same conditions that apply to an IA-32 processor without Execute Disable Bit capability, plus the following new condition: *An instruction fetch to a linear address that translates to physical address in a memory page that has the execute-disable bit set.*

An Execute Disable Bit page fault can occur at all privilege levels. It can occur on any instruction fetch, including (but not limited to): near branches, far branches, call/ret/int/iret execution, sequential instruction fetches, task switches, and so forth. The execute-disable bit in the page translation mechanism is checked only when:

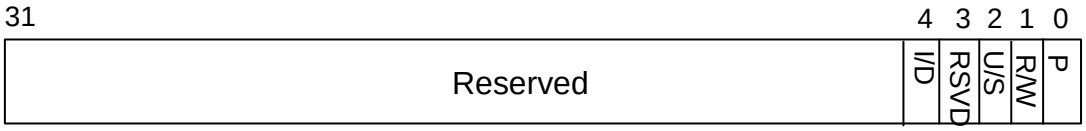
- IA32_EFER.NXE is set.
- The instruction translation look-aside buffer (ITLB) is loaded with a page that is not already present in the ITLB.

The processor provides the page-fault handler with two items of information to aid in diagnosing the exception and potentially recovering from it. The exception-error-code format for page fault is enhanced to include a bit indicating whether a page fault was caused by an instruction fetch. The processor also loads the CR2 register with the linear address that generated the exception.

The page-fault handler can use this address to locate the corresponding entries in the paging data structures.

The format of the page-fault-exception code is shown in Figure 4. A page fault caused by an instruction fetch is indicated by bit 4 in the enhanced page fault exception error code format. If IA32_EFER.NXE is not set, bit 4 of the page fault error code is reserved.

Figure 4. Page Fault Error Code Format when IA32_EFER.NXE = 1:



P	0	The fault was caused by a non-present page.
	1	The fault was caused by a page-level protection violation.
R/W	0	The access causing the fault was a read.
	1	The access causing the fault was a write.
U/S	0	The access causing the fault originated when the processor was executing in supervisor mode.
	1	The access causing the fault originated when the processor was executing in user mode.
RSVD	0	The fault was not caused by reserved bit violation.
	1	The fault was caused by reserved bits set to 1 in a page entry.
I/D	0	The fault was not caused by an instruction fetch.
	1	The fault was caused by an instruction fetch.

Conclusion

Execute-Disable Bit capability is a robust hardware feature, detectable using the CPUID instruction, that protects against malicious software executing code on IA-32 systems. Fundamental software support for this capability includes enabling Physical Address Extension (PAE) functionality and defining a memory-protection policy that is granular down to the page level.

Software developers who implement this capability in their software can deliver a valuable security feature to their users. As users become more knowledgeable and security-conscious, they increasingly include such security features as high-priority components to purchasing decisions.

Notes

[1] The processor will not do reserved bit checking for this bit.

[2] Reserved bit checking also applies to bits 39:36 for processors that support only 36 bits of physical address. For processor that support only 32 bits of physical address, reserved bit checking also applies to bits 39:32.

Additional Resources

Intel, the world's largest chipmaker, also provides an array of value-added products and information to software developers:

- The Early Access Program <link: http://www.intel.com/ids/eap> provides software vendors with Intel's latest technologies, helping member companies to improve product lines and grow market share.

- Intel® Developer Services <link: http://www.intel.com/ids> offers free articles and training to help software developers maximize code performance and minimize time and effort.

- Intel Software Development Products <link: http://www.intel.com/software/products> include Compilers, Performance Analyzers, Performance Libraries and Threading Tools.

- Intel® Solution Services <link: http://www.intel.com/internetservices/intelsolutionservices> is a worldwide consulting organization that helps solution providers, solution developers, and end customers develop cost-effective e-Business solutions to complex business problems.

- IT@Intel <link: http://www.intel.com/ebusiness/it/>, through a series of white papers, case studies, and other materials, describes the lessons it has learned in identifying, evaluating, and deploying new technologies.

- The Intel® Software College <link: http://www.intel.com/software/college/> provides a one-stop shop at Intel for training developers on leading-edge software-development technologies. Training consists of online and instructor-led courses covering all Intel® architectures, platforms, tools, and technologies.