

CPUID—CPU Identification

Opcode	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
0F A2	CPUID	Z0	Valid	Valid	Returns processor identification and feature information to the EAX, EBX, ECX, and EDX registers, as determined by input entered in EAX (in some cases, ECX as well).

Instruction Operand Encoding

Op/En	Operand 1	Operand 2	Operand 3	Operand 4
Z0	N/A	N/A	N/A	N/A

Description

The ID flag (bit 21) in the EFLAGS register indicates support for the CPUID instruction. If a software procedure can set and clear this flag, the processor executing the procedure supports the CPUID instruction. This instruction operates the same in non-64-bit modes and 64-bit mode.

CPUID returns processor identification and feature information in the EAX, EBX, ECX, and EDX registers.¹ The instruction's output is dependent on the contents of the EAX register upon execution (in some cases, ECX as well). For example, the following pseudocode loads EAX with 00H and causes CPUID to return a Maximum Return Value and the Vendor Identification String in the appropriate registers:

```
MOV EAX, 00H
CPUID
```

Table 3-17 shows information returned, depending on the initial value loaded into the EAX register.

Two types of information are returned: basic and extended function information. If a value entered for CPUID.EAX is higher than the maximum input value for basic or extended function for that processor then the data for the highest basic information leaf is returned. For example, using some Intel processors, the following is true:

```
CPUID.EAX = 05H (* Returns MONITOR/MWAIT leaf. *)
CPUID.EAX = 0AH (* Returns Architectural Performance Monitoring leaf. *)
CPUID.EAX = 0BH (* Returns Extended Topology Enumeration leaf. *)2
CPUID.EAX = 1FH (* Returns V2 Extended Topology Enumeration leaf. *)2
CPUID.EAX = 80000008H (* Returns linear/physical address size data. *)
CPUID.EAX = 8000000AH (* INVALID: Returns same information as CPUID.EAX = 0BH. *)
```

If a value entered for CPUID.EAX is less than or equal to the maximum input value and the leaf is not supported on that processor then 0 is returned in all the registers.

When CPUID returns the highest basic leaf information as a result of an invalid input EAX value, any dependence on input ECX value in the basic leaf is honored.

CPUID can be executed at any privilege level to serialize instruction execution. Serializing instruction execution guarantees that any modifications to flags, registers, and memory for previous instructions are completed before the next instruction is fetched and executed.

See also:

"Serializing Instructions" in Chapter 10, "Multiple-Processor Management," in the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A.

"Caching Translation Information" in Chapter 4, "Linear-Address Pre-Processing," in the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A.

1. On Intel 64 processors, CPUID clears the high 32 bits of the RAX/RBX/RCX/RDX registers in all modes.

2. CPUID leaf 1FH is a preferred superset to leaf 0BH. Intel recommends first checking for the existence of CPUID leaf 1FH before using leaf 0BH.

Table 3-17. Information Returned by CPUID Instruction

Initial EAX Value	Information Provided about the Processor	
<i>Basic CPUID Information</i>		
0H	EAX	Maximum Input Value for Basic CPUID Information.
	EBX	"Genu"
	ECX	"ntel"
	EDX	"inel"
01H	EAX	Version Information: Type, Family, Model, and Stepping ID (see Figure 3-6).
	EBX	Bits 07-00: Brand Index. Bits 15-08: CLFLUSH line size (Value * 8 = cache line size in bytes; used also by CLFLUSHOPT). Bits 23-16: Maximum number of addressable IDs for logical processors in this physical package*. Bits 31-24: Initial APIC ID**.
	ECX	Feature Information (see Figure 3-7 and Table 3-19).
	EDX	Feature Information (see Figure 3-8 and Table 3-20).
		NOTES: * The nearest power-of-2 integer that is not smaller than EBX[23:16] is the number of unique initial APIC IDs reserved for addressing different logical processors in a physical package. This field is only valid if CPUID.1.EDX.HTT[bit 28]= 1. ** The 8-bit initial APIC ID in EBX[31:24] is replaced by the 32-bit x2APIC ID, available in Leaf 0BH and Leaf 1FH.
02H	EAX	Cache and TLB Information (see Table 3-21).
	EBX	Cache and TLB Information.
	ECX	Cache and TLB Information.
	EDX	Cache and TLB Information.
03H	EAX	Reserved.
	EBX	Reserved.
	ECX	Bits 00-31 of 96-bit processor serial number. (Available in Pentium III processor only; otherwise, the value in this register is reserved.)
	EDX	Bits 32-63 of 96-bit processor serial number. (Available in Pentium III processor only; otherwise, the value in this register is reserved.)
		NOTES: Processor serial number (PSN) is not supported in the Pentium 4 processor or later. On all models, use the PSN flag (returned using CPUID) to check for PSN support before accessing the feature.
CPUID leaves above 2 and below 80000000H are visible only when IA32_MISC_ENABLE[bit 22] has its default value of 0.		
<i>Deterministic Cache Parameters Leaf (Initial EAX Value = 04H)</i>		
04H		NOTES: Leaf 04H output depends on the initial value in ECX.* See also: "INPUT EAX = 04H: Returns Deterministic Cache Parameters for Each Level" on page 258.
	EAX	Bits 04-00: Cache Type Field. 0 = Null - No more caches. 1 = Data Cache. 2 = Instruction Cache. 3 = Unified Cache. 4-31 = Reserved.

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
		<p>Bits 07-05: Cache Level (starts at 1). Bit 08: Self Initializing cache level (does not need SW initialization). Bit 09: Fully Associative cache.</p> <p>Bits 13-10: Reserved. Bits 25-14: Maximum number of addressable IDs for logical processors sharing this cache**, ***. Bits 31-26: Maximum number of addressable IDs for processor cores in the physical package**, ****, *****.</p> <p>EBX Bits 11-00: L = System Coherency Line Size**. Bits 21-12: P = Physical Line partitions**. Bits 31-22: W = Ways of associativity**.</p> <p>ECX Bits 31-00: S = Number of Sets**.</p> <p>EDX Bit 00: Write-Back Invalidate/Invalidate. 0 = WBINVD/INVD from threads sharing this cache acts upon lower level caches for threads sharing this cache. 1 = WBINVD/INVD is not guaranteed to act upon lower level caches of non-originating threads sharing this cache.</p> <p>Bit 01: Cache Inclusiveness. 0 = Cache is not inclusive of lower cache levels. 1 = Cache is inclusive of lower cache levels.</p> <p>Bit 02: Complex Cache Indexing. 0 = Direct mapped cache. 1 = A complex function is used to index the cache, potentially using all address bits.</p> <p>Bits 31-03: Reserved = 0.</p> <p>NOTES:</p> <p>* If ECX contains an invalid sub leaf index, EAX/EBX/ECX/EDX return 0. Sub-leaf index n+1 is invalid if sub-leaf n returns EAX[4:0] as 0.</p> <p>** Add one to the return value to get the result.</p> <p>***The nearest power-of-2 integer that is not smaller than (1 + EAX[25:14]) is the number of unique initial APIC IDs reserved for addressing different logical processors sharing this cache.</p> <p>**** The nearest power-of-2 integer that is not smaller than (1 + EAX[31:26]) is the number of unique Core_IDs reserved for addressing different processor cores in a physical package. Core ID is a subset of bits of the initial APIC ID.</p> <p>***** The returned value is constant for valid initial values in ECX. Valid ECX values start from 0.</p>
MONITOR/MWAIT Leaf (Initial EAX Value = 05H)		
05H	EAX	<p>Bits 15-00: Smallest monitor-line size in bytes (default is processor's monitor granularity). Bits 31-16: Reserved = 0.</p> <p>EBX Bits 15-00: Largest monitor-line size in bytes (default is processor's monitor granularity). Bits 31-16: Reserved = 0.</p> <p>ECX Bit 00: Enumeration of Monitor-Mwait extensions (beyond EAX and EBX registers) supported. Bit 01: Supports treating interrupts as break-event for MWAIT, even when interrupts disabled. Bits 31-02: Reserved.</p> <p>EDX Bits 03-00: Number of C0* sub C-states supported using MWAIT. Bits 07-04: Number of C1* sub C-states supported using MWAIT. Bits 11-08: Number of C2* sub C-states supported using MWAIT. Bits 15-12: Number of C3* sub C-states supported using MWAIT. Bits 19-16: Number of C4* sub C-states supported using MWAIT. Bits 23-20: Number of C5* sub C-states supported using MWAIT. Bits 27-24: Number of C6* sub C-states supported using MWAIT. Bits 31-28: Number of C7* sub C-states supported using MWAIT.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	<p>NOTE: * The definition of C0 through C7 states for MWAIT extension are processor-specific C-states, not ACPI C-states.</p>	
<i>Thermal and Power Management Leaf (Initial EAX Value = 06H)</i>		
06H	EAX	<p>Bit 00: Digital temperature sensor is supported if set. Bit 01: Intel Turbo Boost Technology available (see description of IA32_MISC_ENABLE[38]). Bit 02: ARAT. APIC-Timer-always-running feature is supported if set. Bit 03: Reserved. Bit 04: PLN. Power limit notification controls are supported if set. Bit 05: ECMD. Clock modulation duty cycle extension is supported if set. Bit 06: PTM. Package thermal management is supported if set. Bit 07: HWP. HWP base registers (IA32_PM_ENABLE[bit 0], IA32_HWP_CAPABILITIES, IA32_HWP_REQUEST, IA32_HWP_STATUS) are supported if set. Bit 08: HWP_Notification. IA32_HWP_INTERRUPT MSR is supported if set. Bit 09: HWP_Activity_Window. IA32_HWP_REQUEST[bits 41:32] is supported if set. Bit 10: HWP_Energy_Performance_Preference. IA32_HWP_REQUEST[bits 31:24] is supported if set. Bit 11: HWP_Package_Level_Request. IA32_HWP_REQUEST_PKG MSR is supported if set. Bit 12: Reserved. Bit 13: HDC. HDC base registers IA32_PKG_HDC_CTL, IA32_PM_CTL1, IA32_THREAD_STALL MSRs are supported if set. Bit 14: Intel® Turbo Boost Max Technology 3.0 available. Bit 15: HWP Capabilities. Highest Performance change is supported if set. Bit 16: HWP PECL override is supported if set. Bit 17: Flexible HWP is supported if set. Bit 18: Fast access mode, low latency, and posted IA32_HWP_REQUEST MSR are supported if set. Bit 19: HW_FEEDBACK. IA32_HW_FEEDBACK_PTR MSR, IA32_HW_FEEDBACK_CONFIG MSR, IA32_PACKAGE_THERM_STATUS MSR bit 26, and IA32_PACKAGE_THERM_INTERRUPT MSR bit 25 are supported if set. Bit 20: Ignoring Idle Logical Processor HWP request is supported if set. Bit 21: Reserved. Bit 22: HWP Control MSR Support. The IA32_HWP_CTL MSR is supported if set. Bit 23: Intel® Thread Director supported if set. The IA32_HW_FEEDBACK_CHAR and IA32_HW_FEEDBACK_THREAD_CONFIG MSRs are supported if set. Bit 24: IA32_THERM_INTERRUPT MSR bit 25 is supported if set. Bits 31-25: Reserved.</p>
	EBX	<p>Bits 03-00: Number of Interrupt Thresholds in Digital Thermal Sensor. Bits 31-04: Reserved.</p>
	ECX	<p>Bit 00: Hardware Coordination Feedback Capability (Presence of IA32_MPERF and IA32_APERF). The capability to provide a measure of delivered processor performance (since last reset of the counters), as a percentage of the expected processor performance when running at the TSC frequency. Bits 02-01: Reserved = 0. Bit 03: The processor supports performance-energy bias preference if CPUID.06H:ECX.SETBH[bit 3] is set and it also implies the presence of a new architectural MSR called IA32_ENERGY_PERF_BIAS (1BOH). Bits 07-04: Reserved = 0. Bits 15-08: Number of Intel® Thread Director classes supported by the processor. Information for that many classes is written into the Intel Thread Director Table by the hardware. Bits 31-16: Reserved = 0.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	EDX	<p>Bits 07-00: Bitmap of supported hardware feedback interface capabilities. 0 = When set to 1, indicates support for performance capability reporting. 1 = When set to 1, indicates support for energy efficiency capability reporting. 2-7 = Reserved</p> <p>Bits 11-08: Enumerates the size of the hardware feedback interface structure in number of 4 KB pages; add one to the return value to get the result.</p> <p>Bits 31-16: Index (starting at 0) of this logical processor's row in the hardware feedback interface structure. Note that on some parts the index may be same for multiple logical processors. On some parts the indices may not be contiguous, i.e., there may be unused rows in the hardware feedback interface structure.</p> <p>NOTE: Bits 0 and 1 will always be set together.</p>
<i>Structured Extended Feature Flags Enumeration Leaf (Initial EAX Value = 07H, ECX = 0)</i>		
07H	EAX EBX	<p>Bits 31-00: Reports the maximum input value for supported leaf 7 sub-leaves.</p> <p>Bit 00: FSGSBASE. Supports RDFSBASE/RDGSBASE/WRFSBASE/WRGSBASE if 1. Bit 01: IA32_TSC_ADJUST MSR is supported if 1. Bit 02: SGX. Supports Intel® Software Guard Extensions (Intel® SGX Extensions) if 1. Bit 03: BMI1. Bit 04: HLE. Bit 05: AVX2. Supports Intel® Advanced Vector Extensions 2 (Intel® AVX2) if 1. Bit 06: FDP_EXCPTN_ONLY. x87 FPU Data Pointer updated only on x87 exceptions if 1. Bit 07: SMEP. Supports Supervisor-Mode Execution Prevention if 1. Bit 08: BMI2. Bit 09: Supports Enhanced REP MOVSB/STOSB if 1. Bit 10: INVPCID. If 1, supports INVPCID instruction for system software that manages process-context identifiers. Bit 11: RTM. Bit 12: RDT-M. Supports Intel® Resource Director Technology (Intel® RDT) Monitoring capability if 1. Bit 13: Deprecates FPU CS and FPU DS values if 1. Bit 14: MPX. Supports Intel® Memory Protection Extensions if 1. Bit 15: RDT-A. Supports Intel® Resource Director Technology (Intel® RDT) Allocation capability if 1. Bit 16: AVX512F. Bit 17: AVX512DQ. Bit 18: RDSEED. Bit 19: ADX. Bit 20: SMAP. Supports Supervisor-Mode Access Prevention (and the CLAC/STAC instructions) if 1. Bit 21: AVX512_IFMA. Bit 22: Reserved. Bit 23: CLFLUSHOPT. Bit 24: CLWB. Bit 25: Intel Processor Trace. Bit 26: AVX512PF. (Intel® Xeon Phi™ only.) Bit 27: AVX512ER. (Intel® Xeon Phi™ only.) Bit 28: AVX512CD. Bit 29: SHA. supports Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions) if 1. Bit 30: AVX512BW. Bit 31: AVX512VL.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
ECX	<p>Bit 00: PREFETCHWT1. (Intel® Xeon Phi™ only.)</p> <p>Bit 01: AVX512_VBMI.</p> <p>Bit 02: UMIP. Supports user-mode instruction prevention if 1.</p> <p>Bit 03: PKU. Supports protection keys for user-mode pages if 1.</p> <p>Bit 04: OSPKE. If 1, OS has set CR4.PKE to enable protection keys (and the RDPKRU/WRPKRU instructions).</p> <p>Bit 05: WAITPKG.</p> <p>Bit 06: AVX512_VBMI2.</p> <p>Bit 07: CET_SS. Supports CET shadow stack features if 1. Processors that set this bit define bits 1:0 of the IA32_U_CET and IA32_S_CET MSRs. Enumerates support for the following MSRs: IA32_INTERRUPT_SP_P_TABLE_ADDR, IA32_PL3_SSP, IA32_PL2_SSP, IA32_PL1_SSP, and IA32_PLO_SSP.</p> <p>Bit 08: GFNI.</p> <p>Bit 09: VAES.</p> <p>Bit 10: VPCLMULQDQ.</p> <p>Bit 11: AVX512_VNNI.</p> <p>Bit 12: AVX512_BITALG.</p> <p>Bits 13: TME_EN. If 1, the following MSRs are supported: IA32_TME_CAPABILITY, IA32_TME_ACTIVATE, IA32_TME_EXCLUDE_MASK, and IA32_TME_EXCLUDE_BASE.</p> <p>Bit 14: AVX512_VPOPCNTDQ.</p> <p>Bit 15: Reserved.</p> <p>Bit 16: LA57. Supports 57-bit linear addresses and five-level paging if 1.</p> <p>Bits 21-17: The value of MAWAU used by the BNDLDX and BNDSTX instructions in 64-bit mode.</p> <p>Bit 22: RDPID and IA32_TSC_AUX are available if 1.</p> <p>Bit 23: KL. Supports Key Locker if 1.</p> <p>Bit 24: BUS_LOCK_DETECT. If 1, indicates support for OS bus-lock detection.</p> <p>Bit 25: CLDEMOTE. Supports cache line demote if 1.</p> <p>Bit 26: Reserved.</p> <p>Bit 27: MOVDIRI. Supports MOVDIRI if 1.</p> <p>Bit 28: MOVDIR64B. Supports MOVDIR64B if 1.</p> <p>Bit 29: ENQCMD. Supports Enqueue Stores if 1.</p> <p>Bit 30: SGX_LC. Supports SGX Launch Configuration if 1.</p> <p>Bit 31: PKS. Supports protection keys for supervisor-mode pages if 1.</p>
EDX	<p>Bit 00: Reserved.</p> <p>Bit 01: SGX-KEYS. If 1, Attestation Services for Intel® SGX is supported.</p> <p>Bit 02: AVX512_4VNNIW. (Intel® Xeon Phi™ only.)</p> <p>Bit 03: AVX512_4FMAPS. (Intel® Xeon Phi™ only.)</p> <p>Bit 04: Fast Short REP MOV.</p> <p>Bit 05: UINTR. If 1, the processor supports user interrupts.</p> <p>Bits 07-06: Reserved.</p> <p>Bit 08: AVX512_VP2INTERSECT.</p> <p>Bit 09: SRBDS_CTRL. If 1, enumerates support for the IA32_MCU_OPT_CTRL MSR and indicates its bit 0 (RNGDS_MITG_DIS) is also supported.</p> <p>Bit 10: MD_CLEAR supported.</p> <p>Bit 11: RTM_ALWAYS_ABORT. If set, any execution of XBEGIN immediately aborts and transitions to the specified fallback address.</p> <p>Bit 12: Reserved.</p> <p>Bit 13: If 1, RTM_FORCE_ABORT supported. Processors that set this bit support the IA32_TSX_FORCE_ABORT MSR. They allow software to set IA32_TSX_FORCE_ABORT[0] (RTM_FORCE_ABORT).</p> <p>Bit 14: SERIALIZE.</p> <p>Bit 15: Hybrid. If 1, the processor is identified as a hybrid part. If CPUID.0.MAXLEAF ≥ 1AH and CPUID.1A.EAX ≠ 0, then the Native Model ID Enumeration Leaf 1AH exists.</p> <p>Bit 16: TSXLDTRK. If 1, the processor supports Intel TSX suspend/resume of load address tracking.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
	<p>Bit 17: Reserved. Bit 18: PCONFIG. Supports PCONFIG if 1. Bit 19: Architectural LBRs. If 1, indicates support for architectural LBRs. Bit 20: CET_IBT. Supports CET indirect branch tracking features if 1. Processors that set this bit define bits 5:2 and bits 63:10 of the IA32_U_CET and IA32_S_CET MSRs. Bit 21: Reserved. Bit 22: AMX-BF16. If 1, the processor supports tile computational operations on bfloat16 numbers. Bit 23: AVX512_FP16. Bit 24: AMX-TILE. If 1, the processor supports tile architecture. Bit 25: AMX-INT8. If 1, the processor supports tile computational operations on 8-bit integers. Bit 26: Enumerates support for indirect branch restricted speculation (IBRS) and the indirect branch predictor barrier (IBPB). Processors that set this bit support the IA32_SPEC_CTRL MSR and the IA32_PRED_CMD MSR. They allow software to set IA32_SPEC_CTRL[0] (IBRS) and IA32_PRED_CMD[0] (IBPB). Bit 27: Enumerates support for single thread indirect branch predictors (STIBP). Processors that set this bit support the IA32_SPEC_CTRL MSR. They allow software to set IA32_SPEC_CTRL[1] (STIBP). Bit 28: Enumerates support for L1D_FLUSH. Processors that set this bit support the IA32_FLUSH_CMD MSR. They allow software to set IA32_FLUSH_CMD[0] (L1D_FLUSH). Bit 29: Enumerates support for the IA32_ARCH_CAPABILITIES MSR. Bit 30: Enumerates support for the IA32_CORE_CAPABILITIES MSR.</p> <p>IA32_CORE_CAPABILITIES is an architectural MSR that enumerates model-specific features. A bit being set in this MSR indicates that a model specific feature is supported; software must still consult CPUID family/model/stepping to determine the behavior of the enumerated feature as features enumerated in IA32_CORE_CAPABILITIES may have different behavior on different processor models. Some of these features may have behavior that is consistent across processor models (and for which consultation of CPUID family/model/stepping is not necessary); such features are identified explicitly where they are documented in this manual.</p> <p>Bit 31: Enumerates support for Speculative Store Bypass Disable (SSBD). Processors that set this bit support the IA32_SPEC_CTRL MSR. They allow software to set IA32_SPEC_CTRL[2] (SSBD).</p> <p>NOTE: * If ECX contains an invalid sub-leaf index, EAX/EBX/ECX/EDX return 0. Sub-leaf index n is invalid if n exceeds the value that sub-leaf 0 returns in EAX.</p>
<i>Structured Extended Feature Enumeration Sub-leaf (Initial EAX Value = 07H, ECX = 1)</i>	
07H	<p>NOTES: Leaf 07H output depends on the initial value in ECX. If ECX contains an invalid sub leaf index, EAX/EBX/ECX/EDX return 0.</p> <p>EAX This field reports 0 if the sub-leaf index, 1, is invalid. Bit 00: SHA512. If 1, supports the SHA512 instructions. Bit 01: SM3. If 1, supports the SM3 instructions. Bit 02: SM4. If 1, supports the SM4 instructions. Bit 03: Reserved. Bit 04: AVX-VNNI. AVX (VEX-encoded) versions of the Vector Neural Network Instructions. Bit 05: AVX512_BF16. Vector Neural Network Instructions supporting BFLOAT16 inputs and conversion instructions from IEEE single precision. Bit 06: LASS. If 1, supports Linear Address Space Separation. Bit 07: CMPCCXADD. If 1, supports the CMPccXADD instruction. Bit 08: ArchPerfmonExt. If 1, supports ArchPerfmonExt. When set, indicates that the Architectural Performance Monitoring Extended Leaf (EAX = 23H) is valid. Bit 09: Reserved. Bit 10: If 1, supports fast zero-length REP MOVSB. Bit 11: If 1, supports fast short REP STOSB. Bit 12: If 1, supports fast short REP CMPSB, REP SCASB.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
	<p>Bits 18-13: Reserved. Bit 19: WRMSRNS. If 1, supports the WRMSRNS instruction. Bit 20: Reserved. Bit 21: AMX-FP16. If 1, the processor supports tile computational operations on FP16 numbers. Bit 22: HRESET. If 1, supports history reset via the HRESET instruction and the IA32_HRESET_ENABLE MSR. When set, indicates that the Processor History Reset Leaf (EAX = 20H) is valid. Bit 23: AVX-IFMA. If 1, supports the AVX-IFMA instructions. Bits 25-24: Reserved. Bit 26: LAM. If 1, supports Linear Address Masking. Bit 27: MSRLIST. If 1, supports the RDMSRLIST and WRMSRLIST instructions and the IA32_BARRIER MSR. Bits 29-28: Reserved. Bit 30: INVD_DISABLE_POST_BIOS_DONE. If 1, supports INVD execution prevention after BIOS Done. Bit 31: Reserved.</p> <p>EBX This field reports 0 if the sub-leaf index, 1, is invalid. Bit 00: Enumerates the presence of the IA32_PPIN and IA32_PPIN_CTL MSRs. If 1, these MSRs are supported. Bits 02-01: Reserved. Bit 03: CPUIDMAXVAL_LIM_RMV. If 1, IA32_MISC_ENABLE[bit 22] cannot be set to 1 to limit the value returned by CPUID.00H:EAX[bits 7:0]. Bits 31-04: Reserved.</p> <p>ECX This field reports 0 if the sub-leaf index, 1, is invalid; otherwise it is reserved.</p> <p>EDX This field reports 0 if the sub-leaf index, 1, is invalid. Bits 03-00: Reserved. Bit 04: AVX-VNNI-INT8. If 1, supports the AVX-VNNI-INT8 instructions. Bit 05: AVX-NE-CONVERT. If 1, supports the AVX-NE-CONVERT instructions. Bits 09-06: Reserved. Bit 10: AVX-VNNI-INT16. If 1, supports the AVX-VNNI-INT16 instructions. Bits 13-11: Reserved. Bit 14: PREFETCHI. If 1, supports the PREFETCHIT0/1 instructions. Bits 16-15: Reserved. Bit 17: UIRET_UIF. If 1, UIRET sets UIF to the value of bit 1 of the RFLAGS image loaded from the stack. Bit 18: CET_SSS. If 1, indicates that an operating system can enable supervisor shadow stacks as long as it ensures that a supervisor shadow stack cannot become prematurely busy due to page faults (see Section 18.2.3 of the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 1). When emulating the CPUID instruction, a virtual-machine monitor (VMM) should return this bit as 1 only if it ensures that VM exits cannot cause a guest supervisor shadow stack to appear to be prematurely busy. Such a VMM could set the "prematurely busy shadow stack" VM-exit control and use the additional information that it provides. Bit 19: AVX10. If 1, supports the Intel® AVX10 instructions and indicates the presence of CPUID Leaf 24H, which enumerates version number and supported vector lengths. Bits 31-20: Reserved.</p>
<i>Structured Extended Feature Enumeration Sub-leaf (Initial EAX Value = 07H, ECX = 2)</i>	
07H	<p>NOTES: Leaf 07H output depends on the initial value in ECX. If ECX contains an invalid sub leaf index, EAX/EBX/ECX/EDX return 0.</p> <p>EAX This field reports 0 if the sub-leaf index, 2, is invalid; otherwise it is reserved.</p> <p>EBX This field reports 0 if the sub-leaf index, 2, is invalid; otherwise it is reserved.</p> <p>ECX This field reports 0 if the sub-leaf index, 2, is invalid; otherwise it is reserved.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	EDX	<p>This field reports 0 if the sub-leaf index, 2, is invalid.</p> <p>Bit 00: PSFD. If 1, indicates bit 7 of the IA32_SPEC_CTRL MSR is supported. Bit 7 of this MSR disables Fast Store Forwarding Predictor without disabling Speculative Store Bypass.</p> <p>Bit 01: IPRED_CTRL. If 1, indicates bits 3 and 4 of the IA32_SPEC_CTRL MSR are supported. Bit 3 of this MSR enables IPRED_DIS control for CPL3. Bit 4 of this MSR enables IPRED_DIS control for CPL0/1/2.</p> <p>Bit 02: RRSBA_CTRL. If 1, indicates bits 5 and 6 of the IA32_SPEC_CTRL MSR are supported. Bit 5 of this MSR disables RRSBA behavior for CPL3. Bit 6 of this MSR disables RRSBA behavior for CPL0/1/2.</p> <p>Bit 03: DDPD_U. If 1, indicates bit 8 of the IA32_SPEC_CTRL MSR is supported. Bit 8 of this MSR disables Data Dependent Prefetcher.</p> <p>Bit 04: BHI_CTRL. If 1, indicates bit 10 of the IA32_SPEC_CTRL MSR is supported. Bit 10 of this MSR enables BHI_DIS_S behavior.</p> <p>Bit 05: MCDT_NO. Processors that enumerate this bit as 1 do not exhibit MXCSR Configuration Dependent Timing (MCDT) behavior and do not need to be mitigated to avoid data-dependent behavior for certain instructions.</p> <p>Bit 06: If 1, supports the UC-lock disable feature and it causes #AC.</p> <p>Bit 07: MONITOR_MITG_NO. If 1, indicates that the MONITOR/UMONITOR instructions are not affected by performance or power issues due to MONITOR/UMONITOR instructions exceeding the capacity of an internal monitor tracking table. If 0, then the product may be affected by this issue.</p> <p>Bits 31-08: Reserved.</p>
<i>Direct Cache Access Information Leaf (Initial EAX Value = 09H)</i>		
09H	EAX	Value of bits [31:0] of IA32_PLATFORM_DCA_CAP MSR (address 1F8H).
	EBX	Reserved.
	ECX	Reserved.
	EDX	Reserved.
<i>Architectural Performance Monitoring Leaf (Initial EAX Value = 0AH)</i>		
0AH	EAX	<p>Bits 07-00: Version ID of architectural performance monitoring.</p> <p>Bits 15-08: Number of general-purpose performance monitoring counter per logical processor.</p> <p>Bits 23-16: Bit width of general-purpose, performance monitoring counter.</p> <p>Bits 31-24: Length of EBX bit vector to enumerate architectural performance monitoring events. Architectural event x is supported if $EBX[x]=0 \ \&\& \ EAX[31:24]>x$.</p>
	EBX	<p>Bit 00: Core cycle event not available if 1 or if $EAX[31:24]<1$.</p> <p>Bit 01: Instruction retired event not available if 1 or if $EAX[31:24]<2$.</p> <p>Bit 02: Reference cycles event not available if 1 or if $EAX[31:24]<3$.</p> <p>Bit 03: Last-level cache reference event not available if 1 or if $EAX[31:24]<4$.</p> <p>Bit 04: Last-level cache misses event not available if 1 or if $EAX[31:24]<5$.</p> <p>Bit 05: Branch instruction retired event not available if 1 or if $EAX[31:24]<6$.</p> <p>Bit 06: Branch mispredict retired event not available if 1 or if $EAX[31:24]<7$.</p> <p>Bit 07: Top-down slots event not available if 1 or if $EAX[31:24]<8$.</p> <p>Bits 31-08: Reserved = 0.</p>
	ECX	<p>Bits 31-00: Supported fixed counters bit mask. Fixed-function performance counter 'i' is supported if bit 'i' is 1 (first counter index starts at zero). It is recommended to use the following logic to determine if a Fixed Counter is supported: $FxCtr[i]_{is_supported} := ECX[i] \ \ (EDX[4:0] > i)$; ¹</p>
	EDX	<p>Bits 04-00: Number of contiguous fixed-function performance counters starting from 0 (if Version ID > 1).¹</p> <p>Bits 12-05: Bit width of fixed-function performance counters (if Version ID > 1).</p> <p>Bits 14-13: Reserved = 0.</p> <p>Bit 15: AnyThread deprecation.</p> <p>Bits 31-16: Reserved = 0.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor									
<i>Extended Topology Enumeration Leaf (Initial EAX Value = 0BH, ECX ≥ 0)</i>										
0BH	<p>NOTES:</p> <p><i>CPUID leaf 1FH is a preferred superset to leaf 0BH. Intel recommends first checking for the existence of Leaf 1FH before using leaf 0BH.</i></p> <p>The sub-leaves of CPUID leaf 0BH describe an ordered hierarchy of logical processors starting from the smallest-scoped domain of a Logical Processor (sub-leaf index 0) to the Core domain (sub-leaf index 1) to the largest-scoped domain (the last valid sub-leaf index) that is implicitly subordinate to the unenumerated highest-scoped domain of the processor package (socket).</p> <p>The details of each valid domain is enumerated by a corresponding sub-leaf. Details for a domain include its type and how all instances of that domain determine the number of logical processors and x2 APIC ID partitioning at the next higher-scoped domain. The ordering of domains within the hierarchy is fixed architecturally as shown below. For a given processor, not all domains may be relevant or enumerated; however, the logical processor and core domains are always enumerated.</p> <p>For two valid sub-leaves N and N+1, sub-leaf N+1 represents the next immediate higher-scoped domain with respect to the domain of sub-leaf N for the given processor.</p> <p>If sub-leaf index “N” returns an invalid domain type in ECX[15:08] (00H), then all sub-leaves with an index greater than “N” shall also return an invalid domain type. A sub-leaf returning an invalid domain always returns 0 in EAX and EBX.</p>									
EAX	<p>Bits 04-00: The number of bits that the x2APIC ID must be shifted to the right to address instances of the next higher-scoped domain. When logical processor is not supported by the processor, the value of this field at the Logical Processor domain sub-leaf may be returned as either 0 (no allocated bits in the x2APIC ID) or 1 (one allocated bit in the x2APIC ID); software should plan accordingly.</p> <p>Bits 31-05: Reserved.</p>									
EBX	<p>Bits 15-00: The number of logical processors across all instances of this domain within the next higher-scoped domain. (For example, in a processor socket/package comprising “M” dies of “N” cores each, where each core has “L” logical processors, the “die” domain sub-leaf value of this field would be M*N*L.) This number reflects configuration as shipped by Intel. Note, software must not use this field to enumerate processor topology*.</p> <p>Bits 31-16: Reserved.</p>									
ECX	<p>Bits 07-00: The input ECX sub-leaf index.</p> <p>Bits 15-08: Domain Type. This field provides an identification value which indicates the domain as shown below. Although domains are ordered, their assigned identification values are not and software should not depend on it.</p> <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: left;"><u>Hierarchy</u></th> <th style="text-align: left;"><u>Domain</u></th> <th style="text-align: left;"><u>Domain Type Identification Value</u></th> </tr> </thead> <tbody> <tr> <td>Lowest</td> <td>Logical Processor</td> <td>1</td> </tr> <tr> <td>Highest</td> <td>Core</td> <td>2</td> </tr> </tbody> </table> <p>(Note that enumeration values of 0 and 3-255 are reserved.)</p> <p>Bits 31-16: Reserved.</p>	<u>Hierarchy</u>	<u>Domain</u>	<u>Domain Type Identification Value</u>	Lowest	Logical Processor	1	Highest	Core	2
<u>Hierarchy</u>	<u>Domain</u>	<u>Domain Type Identification Value</u>								
Lowest	Logical Processor	1								
Highest	Core	2								
EDX	<p>Bits 31-00: x2APIC ID of the current logical processor.</p> <p>NOTES:</p> <p>* Software must not use the value of EBX[15:0] to enumerate processor topology of the system. The value is only intended for display and diagnostic purposes. The actual number of logical processors available to BIOS/OS/Applications may be different from the value of EBX[15:0], depending on software and platform hardware configurations.</p>									
<i>Processor Extended State Enumeration Main Leaf (Initial EAX Value = 0DH, ECX = 0)</i>										
0DH	<p>NOTES:</p> <p>Leaf 0DH main leaf (ECX = 0).</p>									

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	EAX	Bits 31-00: Reports the supported bits of the lower 32 bits of XCRO. XCRO[n] can be set to 1 only if EAX[n] is 1. Bit 00: x87 state. Bit 01: SSE state. Bit 02: AVX state. Bits 04-03: MPX state. Bits 07-05: AVX-512 state. Bit 08: Used for IA32_XSS. Bit 09: PKRU state. Bits 16-10: Used for IA32_XSS. Bit 17: TILECFG state. Bit 18: TILEDATA state. Bits 31-19: Reserved.
	EBX	Bits 31-00: Maximum size (bytes, from the beginning of the XSAVE/XRSTOR save area) required by enabled features in XCRO. May be different than ECX if some features at the end of the XSAVE save area are not enabled.
	ECX	Bit 31-00: Maximum size (bytes, from the beginning of the XSAVE/XRSTOR save area) of the XSAVE/XRSTOR save area required by all supported features in the processor, i.e., all the valid bit fields in XCRO.
	EDX	Bit 31-00: Reports the supported bits of the upper 32 bits of XCRO. XCRO[n+32] can be set to 1 only if EDX[n] is 1. Bits 31-00: Reserved.
<i>Processor Extended State Enumeration Sub-leaf (Initial EAX Value = 0DH, ECX = 1)</i>		
0DH	EAX	Bit 00: XSAVEOPT is available. Bit 01: Supports XSAVEC and the compacted form of XRSTOR if set. Bit 02: Supports XGETBV with ECX = 1 if set. Bit 03: Supports XSAVES/XRSTORS and IA32_XSS if set. Bit 04: Supports extended feature disable (XFD) if set. Bits 31-05: Reserved.
	EBX	Bits 31-00: The size in bytes of the XSAVE area containing all states enabled by XCRO IA32_XSS. NOTES: If EAX[3] is enumerated as 0 and EAX[1] is enumerated as 1, EBX enumerates the size of the XSAVE area containing all states enabled by XCRO. If EAX[1] and EAX[3] are both enumerated as 0, EBX enumerates zero.
	ECX	Bits 31-00: Reports the supported bits of the lower 32 bits of the IA32_XSS MSR. IA32_XSS[n] can be set to 1 only if ECX[n] is 1. Bits 07-00: Used for XCRO. Bit 08: PT state. Bit 09: Used for XCRO. Bit 10: PASID state. Bit 11: CET user state. Bit 12: CET supervisor state. Bit 13: HDC state. Bit 14: UINTR state. Bit 15: LBR state (only for the architectural LBR feature). Bit 16: HWP state. Bits 18-17: Used for XCRO. Bits 31-19: Reserved.
	EDX	Bits 31-00: Reports the supported bits of the upper 32 bits of the IA32_XSS MSR. IA32_XSS[n+32] can be set to 1 only if EDX[n] is 1. Bits 31-00: Reserved.

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
<i>Processor Extended State Enumeration Sub-leaves (Initial EAX Value = 0DH, ECX = n, n > 1)</i>	
0DH	<p>NOTES: Leaf 0DH output depends on the initial value in ECX. Each sub-leaf index (starting at position 2) is supported if it corresponds to a supported bit in either the XCRO register or the IA32_XSS MSR. * If ECX contains an invalid sub-leaf index, EAX/EBX/ECX/EDX return 0. Sub-leaf n ($0 \leq n \leq 31$) is invalid if sub-leaf 0 returns 0 in EAX[n] and sub-leaf 1 returns 0 in ECX[n]. Sub-leaf n ($32 \leq n \leq 63$) is invalid if sub-leaf 0 returns 0 in EDX[n-32] and sub-leaf 1 returns 0 in EDX[n-32].</p> <p>EAX Bits 31-00: The size in bytes (from the offset specified in EBX) of the save area for an extended state feature associated with a valid sub-leaf index, <i>n</i>.</p> <p>EBX Bits 31-00: The offset in bytes of this extended state component's save area from the beginning of the XSAVE/XRSTOR area. This field reports 0 if the sub-leaf index, <i>n</i>, does not map to a valid bit in the XCRO register*.</p> <p>ECX Bit 00 is set if the bit <i>n</i> (corresponding to the sub-leaf index) is supported in the IA32_XSS MSR; it is clear if bit <i>n</i> is instead supported in XCRO. Bit 01 is set if, when the compacted format of an XSAVE area is used, this extended state component located on the next 64-byte boundary following the preceding state component (otherwise, it is located immediately following the preceding state component). Bits 31-02 are reserved. This field reports 0 if the sub-leaf index, <i>n</i>, is invalid*.</p> <p>EDX This field reports 0 if the sub-leaf index, <i>n</i>, is invalid*; otherwise it is reserved.</p>
<i>Intel® Resource Director Technology (Intel® RDT) Monitoring Enumeration Sub-leaf (Initial EAX Value = 0FH, ECX = 0)</i>	
0FH	<p>NOTES: Leaf 0FH output depends on the initial value in ECX. Sub-leaf index 0 reports valid resource type starting at bit position 1 of EDX.</p> <p>EAX Reserved.</p> <p>EBX Bits 31-00: Maximum range (zero-based) of RMID within this physical processor of all types.</p> <p>ECX Reserved.</p> <p>EDX Bit 00: Reserved. Bit 01: Supports L3 Cache Intel RDT Monitoring if 1. Bits 31-02: Reserved.</p>
<i>L3 Cache Intel® RDT Monitoring Capability Enumeration Sub-leaf (Initial EAX Value = 0FH, ECX = 1)</i>	
0FH	<p>NOTES: Leaf 0FH output depends on the initial value in ECX.</p> <p>EAX Bits 07-00: The counter width is encoded as an offset from 24b. A value of zero in this field indicates that 24-bit counters are supported. A value of 8 in this field indicates that 32-bit counters are supported. Bit 08: If 1, indicates the presence of an overflow bit in the IA32_QM_CTR MSR (bit 61). Bit 09: If 1, indicates the presence of non-CPU agent Intel RDT CMT support. Bit 10: If 1, indicates the presence of non-CPU agent Intel RDT MBM support. Bits 31-11: Reserved.</p> <p>EBX Bits 31-00: Conversion factor from reported IA32_QM_CTR value to occupancy metric (bytes) and Memory Bandwidth Monitoring (MBM) metrics.</p> <p>ECX Maximum range (zero-based) of RMID of this resource type.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
	EDX Bit 00: Supports L3 occupancy monitoring if 1. Bit 01: Supports L3 Total Bandwidth monitoring if 1. Bit 02: Supports L3 Local Bandwidth monitoring if 1. Bits 31-03: Reserved.
<i>Intel® Resource Director Technology (Intel® RDT) Allocation Enumeration Sub-leaf (Initial EAX Value = 10H, ECX = 0)</i>	
10H	<p>NOTES: Leaf 10H output depends on the initial value in ECX. Sub-leaf index 0 reports valid resource identification (ResID) starting at bit position 1 of EBX.</p> EAX Reserved. EBX Bit 00: Reserved. Bit 01: Supports L3 Cache Allocation Technology if 1. Bit 02: Supports L2 Cache Allocation Technology if 1. Bit 03: Supports Memory Bandwidth Allocation if 1. Bits 31-04: Reserved. ECX Reserved. EDX Reserved.
<i>L3 Cache Allocation Technology Enumeration Sub-leaf (Initial EAX Value = 10H, ECX = ResID = 1)</i>	
10H	<p>NOTES: Leaf 10H output depends on the initial value in ECX.</p> EAX Bits 04-00: Length of the capacity bit mask for the corresponding ResID. Add one to the return value to get the result. Bits 31-05: Reserved. EBX Bits 31-00: Bit-granular map of isolation/contention of allocation units. ECX Bit 00: Reserved. Bit 01: If 1, indicates L3 CAT for non-CPU agents is supported. Bit 02: If 1, indicates L3 Code and Data Prioritization Technology is supported. Bit 03: If 1, indicates non-contiguous capacity bitmask is supported. The bits that are set in the various IA32_L3_MASK_n registers do not have to be contiguous. Bits 31-04: Reserved. EDX Bits 15-00: Highest Class of Service (CLOS) number supported for this ResID. Bits 31-16: Reserved.
<i>L2 Cache Allocation Technology Enumeration Sub-leaf (Initial EAX Value = 10H, ECX = ResID = 2)</i>	
10H	<p>NOTES: Leaf 10H output depends on the initial value in ECX.</p> EAX Bits 04-00: Length of the capacity bit mask for the corresponding ResID. Add one to the return value to get the result. Bits 31-05: Reserved. EBX Bits 31-00: Bit-granular map of isolation/contention of allocation units. ECX Bits 01-00: Reserved. Bit 02: CDP. If 1, indicates L2 Code and Data Prioritization Technology is supported. Bit 03: If 1, indicates non-contiguous capacity bitmask is supported. The bits that are set in the various IA32_L2_MASK_n registers do not have to be contiguous. Bits 31-04: Reserved. EDX Bits 15-00: Highest CLOS number supported for this ResID. Bits 31-16: Reserved.

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
<i>Memory Bandwidth Allocation Enumeration Sub-leaf (Initial EAX Value = 10H, ECX = ResID =3)</i>	
10H	<p>NOTES: Leaf 10H output depends on the initial value in ECX.</p> <p>EAX Bits 11-00: Reports the maximum MBA throttling value supported for the corresponding ResID. Add one to the return value to get the result. Bits 31-12: Reserved.</p> <p>EBX Bits 31-00: Reserved.</p> <p>ECX Bits 01-00: Reserved. Bit 02: Reports whether the response of the delay values is linear. Bits 31-03: Reserved.</p> <p>EDX Bits 15-00: Highest CLOS number supported for this ResID. Bits 31-16: Reserved.</p>
<i>Intel® SGX Capability Enumeration Leaf, Sub-leaf 0 (Initial EAX Value = 12H, ECX = 0)</i>	
12H	<p>NOTES: Leaf 12H sub-leaf 0 (ECX = 0) is supported if CPUID.(EAX=07H, ECX=0H):EBX[SGX] = 1.</p> <p>EAX Bit 00: SGX1. If 1, Indicates Intel SGX supports the collection of SGX1 leaf functions. Bit 01: SGX2. If 1, Indicates Intel SGX supports the collection of SGX2 leaf functions. Bits 04-02: Reserved. Bit 05: If 1, indicates Intel SGX supports ENCLV instruction leaves EINCVIRTUAL, EDECVIRTUAL, and ESETCONTEXT. Bit 06: If 1, indicates Intel SGX supports ENCLS instruction leaves ETRACKC, ERDINFO, ELDBC, and ELDUC. Bit 07: If 1, indicates Intel SGX supports ENCLU instruction leaf EVERIFYREPORT2. Bits 09-08: Reserved. Bit 10: If 1, indicates Intel SGX supports ENCLS instruction leaf EUPDATESVN. Bit 11: If 1, indicates Intel SGX supports ENCLU instruction leaf EDECCSSA. Bits 31-12: Reserved.</p> <p>EBX Bits 31-00: MISCSELECT. Bit vector of supported extended SGX features.</p> <p>ECX Bits 31-00: Reserved.</p> <p>EDX Bits 07-00: MaxEnclaveSize_Not64. The maximum supported enclave size in non-64-bit mode is 2^(EDX[7:0]). Bits 15-08: MaxEnclaveSize_64. The maximum supported enclave size in 64-bit mode is 2^(EDX[15:8]). Bits 31-16: Reserved.</p>
<i>Intel SGX Attributes Enumeration Leaf, Sub-leaf 1 (Initial EAX Value = 12H, ECX = 1)</i>	
12H	<p>NOTES: Leaf 12H sub-leaf 1 (ECX = 1) is supported if CPUID.(EAX=07H, ECX=0H):EBX[SGX] = 1.</p> <p>EAX Bit 31-00: Reports the valid bits of SECS.ATTRIBUTES[31:0] that software can set with ECREATE.</p> <p>EBX Bit 31-00: Reports the valid bits of SECS.ATTRIBUTES[63:32] that software can set with ECREATE.</p> <p>ECX Bit 31-00: Reports the valid bits of SECS.ATTRIBUTES[95:64] that software can set with ECREATE.</p> <p>EDX Bit 31-00: Reports the valid bits of SECS.ATTRIBUTES[127:96] that software can set with ECREATE.</p>
<i>Intel® SGX EPC Enumeration Leaf, Sub-leaves (Initial EAX Value = 12H, ECX = 2 or higher)</i>	
12H	<p>NOTES: Leaf 12H sub-leaf 2 or higher (ECX >= 2) is supported if CPUID.(EAX=07H, ECX=0H):EBX[SGX] = 1. For sub-leaves (ECX = 2 or higher), definition of EDX,ECX,EBX,EAX[31:4] depends on the sub-leaf type listed below.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
	<p>EAX Bit 03-00: Sub-leaf Type 0000b: Indicates this sub-leaf is invalid. 0001b: This sub-leaf enumerates an EPC section. EBX:EAX and EDX:ECX provide information on the Enclave Page Cache (EPC) section. All other type encodings are reserved.</p> <p>Type 0000b. This sub-leaf is invalid. EDX:ECX:EBX:EAX return 0.</p> <p>Type 0001b. This sub-leaf enumerates an EPC sections with EDX:ECX, EBX:EAX defined as follows. EAX[11:04]: Reserved (enumerate 0). EAX[31:12]: Bits 31:12 of the physical address of the base of the EPC section.</p> <p>EBX[19:00]: Bits 51:32 of the physical address of the base of the EPC section. EBX[31:20]: Reserved.</p> <p>ECX[03:00]: EPC section property encoding defined as follows: If ECX[3:0] = 0000b, then all bits of the EDX:ECX pair are enumerated as 0. If ECX[3:0] = 0001b, then this section has confidentiality, integrity, and replay protection. If ECX[3:0] = 0010b, then this section has confidentiality protection only. If ECX[3:0] = 0011b, then this section has confidentiality and integrity protection. All other encodings are reserved. ECX[11:04]: Reserved (enumerate 0). ECX[31:12]: Bits 31:12 of the size of the corresponding EPC section within the Processor Reserved Memory.</p> <p>EDX[19:00]: Bits 51:32 of the size of the corresponding EPC section within the Processor Reserved Memory. EDX[31:20]: Reserved.</p>
<i>Intel® Processor Trace Enumeration Main Leaf (Initial EAX Value = 14H, ECX = 0)</i>	
14H	<p>NOTES: Leaf 14H main leaf (ECX = 0).</p> <p>EAX Bits 31-00: Reports the maximum sub-leaf supported in leaf 14H.</p> <p>EBX Bit 00: If 1, indicates that IA32_RTIT_CTL.CR3Filter can be set to 1, and that IA32_RTIT_CR3_MATCH MSR can be accessed. Bit 01: If 1, indicates support of Configurable PSB and Cycle-Accurate Mode. Bit 02: If 1, indicates support of IP Filtering, TraceStop filtering, and preservation of Intel PT MSRs across warm reset. Bit 03: If 1, indicates support of MTC timing packet and suppression of COFI-based packets. Bit 04: If 1, indicates support of PTWRITE. Writes can set IA32_RTIT_CTL[12] (PTWEn) and IA32_RTIT_CTL[5] (FUPonPTw), and PTWRITE can generate packets. Bit 05: If 1, indicates support of Power Event Trace. Writes can set IA32_RTIT_CTL[4] (PwrEvtEn), enabling Power Event Trace packet generation. Bit 06: If 1, indicates support for PSB and PMI preservation. Writes can set IA32_RTIT_CTL[56] (InjectPsb-PmiOnEnable), enabling the processor to set IA32_RTIT_STATUS[7] (PendTopaPMI) and/or IA32_RTIT_STATUS[6] (PendPSB) in order to preserve ToPA PMIs and/or PSBs otherwise lost due to Intel PT disable. Writes can also set PendToPAPMI and PendPSB. Bit 07: If 1, writes can set IA32_RTIT_CTL[31] (EventEn), enabling Event Trace packet generation. Bit 08: If 1, writes can set IA32_RTIT_CTL[55] (DisTNT), disabling TNT packet generation. Bit 31-09: Reserved.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	ECX	<p>Bit 00: If 1, Tracing can be enabled with IA32_RTIT_CTL.ToPA = 1, hence utilizing the ToPA output scheme; IA32_RTIT_OUTPUT_BASE and IA32_RTIT_OUTPUT_MASK_PTRS MSRs can be accessed.</p> <p>Bit 01: If 1, ToPA tables can hold any number of output entries, up to the maximum allowed by the MaskOrTableOffset field of IA32_RTIT_OUTPUT_MASK_PTRS.</p> <p>Bit 02: If 1, indicates support of Single-Range Output scheme.</p> <p>Bit 03: If 1, indicates support of output to Trace Transport subsystem.</p> <p>Bit 30-04: Reserved.</p> <p>Bit 31: If 1, generated packets which contain IP payloads have LIP values, which include the CS base component.</p>
	EDX	Bits 31-00: Reserved.
<i>Intel® Processor Trace Enumeration Sub-leaf (Initial EAX Value = 14H, ECX = 1)</i>		
14H	EAX	<p>Bits 02-00: Number of configurable Address Ranges for filtering.</p> <p>Bits 15-03: Reserved.</p> <p>Bits 31-16: Bitmap of supported MTC period encodings.</p>
	EBX	<p>Bits 15-00: Bitmap of supported Cycle Threshold value encodings.</p> <p>Bit 31-16: Bitmap of supported Configurable PSB frequency encodings.</p>
	ECX	Bits 31-00: Reserved.
	EDX	Bits 31-00: Reserved.
<i>Time Stamp Counter and Nominal Core Crystal Clock Information Leaf (Initial EAX Value = 15H)</i>		
15H	<p>NOTES:</p> <p>If EBX[31:0] is 0, the TSC/"core crystal clock" ratio is not enumerated.</p> <p>EBX[31:0]/EAX[31:0] indicates the ratio of the TSC frequency and the core crystal clock frequency.</p> <p>If ECX is 0, the nominal core crystal clock frequency is not enumerated.</p> <p>"TSC frequency" = "core crystal clock frequency" * EBX/EAX.</p> <p>The core crystal clock may differ from the reference clock, bus clock, or core clock frequencies.</p>	
	EAX	Bits 31-00: An unsigned integer which is the denominator of the TSC/"core crystal clock" ratio.
	EBX	Bits 31-00: An unsigned integer which is the numerator of the TSC/"core crystal clock" ratio.
	ECX	Bits 31-00: An unsigned integer which is the nominal frequency of the core crystal clock in Hz.
	EDX	Bits 31-00: Reserved = 0.
<i>Processor Frequency Information Leaf (Initial EAX Value = 16H)</i>		
16H	EAX	<p>Bits 15-00: Processor Base Frequency (in MHz).</p> <p>Bits 31-16: Reserved = 0.</p>
	EBX	<p>Bits 15-00: Maximum Frequency (in MHz).</p> <p>Bits 31-16: Reserved = 0.</p>
	ECX	<p>Bits 15-00: Bus (Reference) Frequency (in MHz).</p> <p>Bits 31-16: Reserved = 0.</p>
	EDX	Reserved.

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	<p>NOTES: * Data is returned from this interface in accordance with the processor’s specification and does not reflect actual values. Suitable use of this data includes the display of processor information in like manner to the processor brand string and for determining the appropriate range to use when displaying processor information e.g. frequency history graphs. The returned information should not be used for any other purpose as the returned information does not accurately correlate to information / counters returned by other processor interfaces.</p> <p>While a processor may support the Processor Frequency Information leaf, fields that return a value of zero are not supported.</p>	
<i>System-On-Chip Vendor Attribute Enumeration Main Leaf (Initial EAX Value = 17H, ECX = 0)</i>		
17H	<p>NOTES: Leaf 17H main leaf (ECX = 0). Leaf 17H output depends on the initial value in ECX. Leaf 17H sub-leaves 1 through 3 reports SOC Vendor Brand String. Leaf 17H is valid if MaxSOCID_Index >= 3. Leaf 17H sub-leaves 4 and above are reserved.</p> <p>EAX Bits 31-00: MaxSOCID_Index. Reports the maximum input value of supported sub-leaf in leaf 17H. EBX Bits 15-00: SOC Vendor ID. Bit 16: IsVendorScheme. If 1, the SOC Vendor ID field is assigned via an industry standard enumeration scheme. Otherwise, the SOC Vendor ID field is assigned by Intel. Bits 31-17: Reserved = 0. ECX Bits 31-00: Project ID. A unique number an SOC vendor assigns to its SOC projects. EDX Bits 31-00: Stepping ID. A unique number within an SOC project that an SOC vendor assigns.</p>	
<i>System-On-Chip Vendor Attribute Enumeration Sub-leaf (Initial EAX Value = 17H, ECX = 1..3)</i>		
17H	EAX EBX ECX EDX	<p>Bit 31-00: SOC Vendor Brand String. UTF-8 encoded string. Bit 31-00: SOC Vendor Brand String. UTF-8 encoded string. Bit 31-00: SOC Vendor Brand String. UTF-8 encoded string. Bit 31-00: SOC Vendor Brand String. UTF-8 encoded string.</p> <p>NOTES: Leaf 17H output depends on the initial value in ECX. SOC Vendor Brand String is a UTF-8 encoded string padded with trailing bytes of 00H. The complete SOC Vendor Brand String is constructed by concatenating in ascending order of EAX:EBX:ECX:EDX and from the sub-leaf 1 fragment towards sub-leaf 3.</p>
<i>System-On-Chip Vendor Attribute Enumeration Sub-leaves (Initial EAX Value = 17H, ECX > MaxSOCID_Index)</i>		
17H	EAX EBX ECX EDX	<p>NOTES: Leaf 17H output depends on the initial value in ECX.</p> <p>Bits 31-00: Reserved = 0. Bits 31-00: Reserved = 0. Bits 31-00: Reserved = 0. Bits 31-00: Reserved = 0.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
<i>Deterministic Address Translation Parameters Main Leaf (Initial EAX Value = 18H, ECX = 0)</i>	
18H	<p>NOTES:</p> <p>Each sub-leaf enumerates a different address translation structure.</p> <p>If ECX contains an invalid sub-leaf index, EAX/EBX/ECX/EDX return 0. Sub-leaf index n is invalid if n exceeds the value that sub-leaf 0 returns in EAX. A sub-leaf index is also invalid if EDX[4:0] returns 0. Valid sub-leaves do not need to be contiguous or in any particular order. A valid sub-leaf may be in a higher input ECX value than an invalid sub-leaf or than a valid sub-leaf of a higher or lower-level structure.</p> <p>* Some unified TLBs will allow a single TLB entry to satisfy data read/write and instruction fetches. Others will require separate entries (e.g., one loaded on data read/write and another loaded on an instruction fetch). See the Intel® 64 and IA-32 Architectures Optimization Reference Manual for details of a particular product.</p> <p>** Add one to the return value to get the result.</p> <p>EAX Bits 31-00: Reports the maximum input value of supported sub-leaf in leaf 18H.</p> <p>EBX Bit 00: 4K page size entries supported by this structure. Bit 01: 2MB page size entries supported by this structure. Bit 02: 4MB page size entries supported by this structure. Bit 03: 1 GB page size entries supported by this structure. Bits 07-04: Reserved. Bits 10-08: Partitioning (0: Soft partitioning between the logical processors sharing this structure). Bits 15-11: Reserved. Bits 31-16: W = Ways of associativity.</p> <p>ECX Bits 31-00: S = Number of Sets.</p> <p>EDX Bits 04-00: Translation cache type field. 00000b: Null (indicates this sub-leaf is not valid). 00001b: Data TLB. 00010b: Instruction TLB. 00011b: Unified TLB*. 00100b: Load Only TLB. Hit on loads; fills on both loads and stores. 00101b: Store Only TLB. Hit on stores; fill on stores. All other encodings are reserved. Bits 07-05: Translation cache level (starts at 1). Bit 08: Fully associative structure. Bits 13-09: Reserved. Bits 25-14: Maximum number of addressable IDs for logical processors sharing this translation cache.** Bits 31-26: Reserved.</p>
<i>Deterministic Address Translation Parameters Sub-leaf (Initial EAX Value = 18H, ECX ≥ 1)</i>	
18H	<p>NOTES:</p> <p>Each sub-leaf enumerates a different address translation structure.</p> <p>If ECX contains an invalid sub-leaf index, EAX/EBX/ECX/EDX return 0. Sub-leaf index n is invalid if n exceeds the value that sub-leaf 0 returns in EAX. A sub-leaf index is also invalid if EDX[4:0] returns 0. Valid sub-leaves do not need to be contiguous or in any particular order. A valid sub-leaf may be in a higher input ECX value than an invalid sub-leaf or than a valid sub-leaf of a higher or lower-level structure.</p> <p>* Some unified TLBs will allow a single TLB entry to satisfy data read/write and instruction fetches. Others will require separate entries (e.g., one loaded on data read/write and another loaded on an instruction fetch). See the Intel® 64 and IA-32 Architectures Optimization Reference Manual for details of a particular product.</p> <p>** Add one to the return value to get the result.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	EAX	Bits 31-00: Reserved.
	EBX	Bit 00: 4K page size entries supported by this structure. Bit 01: 2MB page size entries supported by this structure. Bit 02: 4MB page size entries supported by this structure. Bit 03: 1 GB page size entries supported by this structure. Bits 07-04: Reserved. Bits 10-08: Partitioning (0: Soft partitioning between the logical processors sharing this structure). Bits 15-11: Reserved. Bits 31-16: <i>W</i> = Ways of associativity.
	ECX	Bits 31-00: <i>S</i> = Number of Sets.
	EDX	Bits 04-00: Translation cache type field. 0000b: Null (indicates this sub-leaf is not valid). 0001b: Data TLB. 0010b: Instruction TLB. 0011b: Unified TLB*. All other encodings are reserved. Bits 07-05: Translation cache level (starts at 1). Bit 08: Fully associative structure. Bits 13-09: Reserved. Bits 25-14: Maximum number of addressable IDs for logical processors sharing this translation cache** Bits 31-26: Reserved.
<i>Key Locker Leaf (Initial EAX Value = 19H)</i>		
19H	EAX	Bit 00: Key Locker restriction of CPL0-only supported. Bit 01: Key Locker restriction of no-encrypt supported. Bit 02: Key Locker restriction of no-decrypt supported. Bits 31-03: Reserved.
	EBX	Bit 00: AESKLE. If 1, the AES Key Locker instructions are fully enabled. Bit 01: Reserved. Bit 02: If 1, the AES wide Key Locker instructions are supported. Bit 03: Reserved. Bit 04: If 1, the platform supports the Key Locker MSRs (IA32_COPY_LOCAL_TO_PLATFORM, IA23_COPY_PLATFORM_TO_LOCAL, IA32_COPY_STATUS, and IA32_IWKEYBACKUP_STATUS) and backing up the internal wrapping key. Bits 31-05: Reserved.
	ECX	Bit 00: If 1, the NoBackup parameter to LOADIWKEY is supported. Bit 01: If 1, KeySource encoding of 1 (randomization of the internal wrapping key) is supported. Bits 31-02: Reserved.
	EDX	Reserved.
<i>Native Model ID Enumeration Leaf (Initial EAX Value = 1AH, ECX = 0)</i>		
1AH	<p>NOTES:</p> <p>This leaf exists on all hybrid parts, however this leaf is not only available on hybrid parts. The following algorithm is used for detection of this leaf: If CPUID.0.MAXLEAF ≥ 1AH and CPUID.1A.EAX ≠ 0, then the leaf exists.</p>	

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor
	<p>EAX Enumerates the native model ID and core type. Bits 31-24: Core type* 10H: Reserved 20H: Intel Atom® 30H: Reserved 40H: Intel® Core™ Bits 23-00: Native model ID of the core. The core-type and native model ID can be used to uniquely identify the microarchitecture of the core. This native model ID is not unique across core types, and not related to the model ID reported in CPUID leaf 01H, and does not identify the SOC. * The core type may only be used as an identification of the microarchitecture for this logical processor and its numeric value has no significance, neither large nor small. This field neither implies nor expresses any other attribute to this logical processor and software should not assume any.</p> <p>EBX Reserved.</p> <p>ECX Reserved.</p> <p>EDX Reserved.</p>
<i>PCONFIG Information Sub-leaf (Initial EAX Value = 1BH, ECX ≥ 0)</i>	
1BH	<p>For details on this sub-leaf, see "INPUT EAX = 1BH: Returns PCONFIG Information" on page 3-260.</p> <p>NOTE: Leaf 1BH is supported if CPUID.(EAX=07H, ECX=0H):EDX[18] = 1.</p>
<i>Last Branch Records Information Leaf (Initial EAX Value = 1CH)</i>	
1CH	<p>NOTE: This leaf pertains to the architectural feature.</p> <p>EAX Bits 07-00: Supported LBR Depth Values. For each bit n set in this field, the IA32_LBR_DEPTH.DEPTH value 8*(n+1) is supported. Bits 29-08: Reserved. Bit 30: Deep C-state Reset. If set, indicates that LBRs may be cleared on an MWAIT that requests a C-state numerically greater than C1. Bit 31: IP Values Contain LIP. If set, LBR IP values contain LIP. If clear, IP values contain Effective IP.</p> <p>EBX Bit 00: CPL Filtering Supported. If set, the processor supports setting IA32_LBR_CTL[2:1] to non-zero value. Bit 01: Branch Filtering Supported. If set, the processor supports setting IA32_LBR_CTL[22:16] to non-zero value. Bit 02: Call-stack Mode Supported. If set, the processor supports setting IA32_LBR_CTL[3] to 1. Bits 31-03: Reserved.</p> <p>ECX Bit 00: Mispredict Bit Supported. IA32_LBR_x_INFO[63] holds indication of branch misprediction (MISPRED). Bit 01: Timed LBRs Supported. IA32_LBR_x_INFO[15:0] holds CPU cycles since last LBR entry (CYC_CNT), and IA32_LBR_x_INFO[60] holds an indication of whether the value held there is valid (CYC_CNT_VALID). Bit 02: Branch Type Field Supported. IA32_LBR_INFO_x[59:56] holds indication of the recorded operation's branch type (BR_TYPE). Bits 15-03: Reserved. Bits 19-16: Event Logging Supported bitmap. Bits 31-20: Reserved.</p> <p>EDX Bits 31-00: Reserved.</p>
<i>Tile Information Main Leaf (Initial EAX Value = 1DH, ECX = 0)</i>	
1DH	<p>NOTES: For sub-leaves of 1DH, they are indexed by the palette id. Leaf 1DH sub-leaves 2 and above are reserved.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	EAX	Bits 31-00: max_palette. Highest numbered palette sub-leaf. Value = 1.
	EBX	Bits 31-00: Reserved = 0.
	ECX	Bits 31-00: Reserved = 0.
	EDX	Bits 31-00: Reserved = 0.
<i>Tile Palette 1 Sub-leaf (Initial EAX Value = 1DH, ECX = 1)</i>		
1DH	EAX	Bits 15-00: Palette 1 total_tile_bytes. Value = 8192. Bits 31-16: Palette 1 bytes_per_tile. Value = 1024.
	EBX	Bits 15-00: Palette 1 bytes_per_row. Value = 64. Bits 31-16: Palette 1 max_names (number of tile registers). Value = 8.
	ECX	Bits 15-00: Palette 1 max_rows. Value = 16. Bits 31-16: Reserved = 0.
	EDX	Bits 31-00: Reserved = 0.
<i>TMUL Information Main Leaf (Initial EAX Value = 1EH, ECX = 0)</i>		
1EH	<p>NOTE: Leaf 1EH sub-leaves 1 and above are reserved.</p>	
	EAX	Bits 31-00: Reserved = 0.
	EBX	Bits 07-00: tmul_maxk (rows or columns). Value = 16. Bits 23-08: tmul_maxn (column bytes). Value = 64. Bits 31-24: Reserved = 0.
	ECX	Bits 31-00: Reserved = 0.
	EDX	Bits 31-00: Reserved = 0.
<i>V2 Extended Topology Enumeration Leaf (Initial EAX Value = 1FH, ECX ≥ 0)</i>		
1FH	<p>NOTES:</p> <p><i>CPUID leaf 1FH is a preferred superset to leaf 0BH. Intel recommends using leaf 1FH when available rather than leaf 0BH and ensuring that any leaf 0BH algorithms are updated to support leaf 1FH.</i></p> <p>The sub-leaves of CPUID leaf 1FH describe an ordered hierarchy of logical processors starting from the smallest-scoped domain of a Logical Processor (sub-leaf index 0) to the Core domain (sub-leaf index 1) to the largest-scoped domain (the last valid sub-leaf index) that is implicitly subordinate to the unenumerated highest-scoped domain of the processor package (socket).</p> <p>The details of each valid domain is enumerated by a corresponding sub-leaf. Details for a domain include its type and how all instances of that domain determine the number of logical processors and x2 APIC ID partitioning at the next higher-scoped domain. The ordering of domains within the hierarchy is fixed architecturally as shown below. For a given processor, not all domains may be relevant or enumerated; however, the logical processor and core domains are always enumerated. As an example, a processor may report an ordered hierarchy consisting only of "Logical Processor," "Core," and "Die."</p> <p>For two valid sub-leaves N and N+1, sub-leaf N+1 represents the next immediate higher-scoped domain with respect to the domain of sub-leaf N for the given processor.</p> <p>If sub-leaf index "N" returns an invalid domain type in ECX[15:08] (00H), then all sub-leaves with an index greater than "N" shall also return an invalid domain type. A sub-leaf returning an invalid domain always returns 0 in EAX and EBX.</p>	
	EAX	Bits 04-00: The number of bits that the x2APIC ID must be shifted to the right to address instances of the next higher-scoped domain. When logical processor is not supported by the processor, the value of this field at the Logical Processor domain sub-leaf may be returned as either 0 (no allocated bits in the x2APIC ID) or 1 (one allocated bit in the x2APIC ID); software should plan accordingly. Bits 31-05: Reserved.

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor																									
	EBX	<p>Bits 15-00: The number of logical processors across all instances of this domain within the next higher-scoped domain relative to this current logical processor. (For example, in a processor socket/package comprising “M” dies of “N” cores each, where each core has “L” logical processors, the “die” domain sub-leaf value of this field would be M*N*L. In an asymmetric topology this would be the summation of the value across the lower domain level instances to create each upper domain level instance.) This number reflects configuration as shipped by Intel. Note, software must not use this field to enumerate processor topology*.</p> <p>Bits 31-16: Reserved.</p>																								
	ECX	<p>Bits 07-00: The input ECX sub-leaf index.</p> <p>Bits 15-08: Domain Type. This field provides an identification value which indicates the domain as shown below. Although domains are ordered, as also shown below, their assigned identification values are not and software should not depend on it. (For example, if a new domain between core and module is specified, it will have an identification value higher than 5.)</p> <table style="margin-left: 40px;"> <thead> <tr> <th style="text-align: left;"><u>Hierarchy</u></th> <th style="text-align: left;"><u>Domain</u></th> <th style="text-align: left;"><u>Domain Type Identification Value</u></th> </tr> </thead> <tbody> <tr> <td>Lowest</td> <td>Logical Processor</td> <td>1</td> </tr> <tr> <td>...</td> <td>Core</td> <td>2</td> </tr> <tr> <td>...</td> <td>Module</td> <td>3</td> </tr> <tr> <td>...</td> <td>Tile</td> <td>4</td> </tr> <tr> <td>...</td> <td>Die</td> <td>5</td> </tr> <tr> <td>...</td> <td>DieGrp</td> <td>6</td> </tr> <tr> <td>Highest</td> <td>Package/Socket</td> <td>(implied)</td> </tr> </tbody> </table> <p>(Note that enumeration values of 0 and 7-255 are reserved.)</p> <p>Bits 31-16: Reserved.</p>	<u>Hierarchy</u>	<u>Domain</u>	<u>Domain Type Identification Value</u>	Lowest	Logical Processor	1	...	Core	2	...	Module	3	...	Tile	4	...	Die	5	...	DieGrp	6	Highest	Package/Socket	(implied)
<u>Hierarchy</u>	<u>Domain</u>	<u>Domain Type Identification Value</u>																								
Lowest	Logical Processor	1																								
...	Core	2																								
...	Module	3																								
...	Tile	4																								
...	Die	5																								
...	DieGrp	6																								
Highest	Package/Socket	(implied)																								
	EDX	<p>Bits 31-00: x2APIC ID of the current logical processor. It is always valid and does not vary with the sub-leaf index in ECX.</p> <p>NOTES:</p> <p>* Software must not use the value of EBX[15:0] to enumerate processor topology of the system. The value is only intended for display and diagnostic purposes. The actual number of logical processors available to BIOS/OS/Applications may be different from the value of EBX[15:0], depending on software and platform hardware configurations.</p>																								
<i>Processor History Reset Sub-leaf (Initial EAX Value = 20H, ECX = 0)</i>																										
20H	EAX	Reports the maximum number of sub-leaves that are supported in leaf 20H.																								
	EBX	<p>Indicates which bits may be set in the IA32_HRESET_ENABLE MSR to enable reset of different components of hardware-maintained history.</p> <p>Bit 00: Indicates support for both HRESET’s EAX[0] parameter, and IA32_HRESET_ENABLE[0] set by the OS to enable reset of Intel® Thread Director history.</p> <p>Bits 31-01: Reserved = 0.</p>																								
	ECX	Reserved.																								
	EDX	Reserved.																								
<i>Architectural Performance Monitoring Extended Main Leaf (Initial EAX Value = 23H, ECX = 0)</i>																										
23H		<p>NOTE:</p> <p>Output depends on ECX input value.</p>																								
	EAX	Bits 31-0: If bit <i>n</i> is set, sub-leaf <i>n</i> is supported. (For unsupported sub-leaves, 0 is returned in the registers EAX, EBX, ECX, and EDX.)																								

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	<p>EBX</p> <p>ECX</p> <p>EDX</p>	<p>Bit 00: UnitMask2 supported. If set, the processor supports the UnitMask2 field in the IA32_PERFVTSELx MSRs.</p> <p>Bit 01: EQ-bit supported. If set, the processor supports the equal flag in the IA32_PERFVTSELx MSRs.</p> <p>Bits 31-02: Reserved.</p> <p>Bits 07-00: Number of Top-down Microarchitecture Analysis (TMA) slots per cycle. This number can be multiplied by the number of cycles (from CPU_CLK_UNHALTED.THREAD / CPU_CLK_UNHALTED.CORE or IA32_FIXED_CTR1) to determine the total number of slots.</p> <p>Bits 31-08: Reserved.</p> <p>Bits 31-00: Reserved.</p>
Architectural Performance Monitoring Extended Sub-Leaf (Initial EAX Value = 23H, ECX = 1)		
23H	<p>EAX</p> <p>EBX</p> <p>ECX</p> <p>EDX</p>	<p>Bits 31-00: General counters bitmap. For each bit <i>n</i> set in this field, the processor supports general-purpose performance monitoring counter <i>n</i>.</p> <p>Bits 31-00: Fixed counters bitmap. For each bit <i>m</i> set in this field, the processor supports fixed-function performance monitoring counter <i>m</i>.¹</p> <p>Bits 31-00: Reserved.</p> <p>Bits 31-00: Reserved.</p>
Architectural Performance Monitoring Extended Sub-Leaf (Initial EAX Value = 23H, ECX = 2)		
23H	<p>EAX</p> <p>EBX</p> <p>ECX</p> <p>EDX</p>	<p>Bits 31-00: Bitmap of Auto Counter Reload (ACR) general counters that can be reloaded. For each bit <i>n</i> set in this field, the processor supports ACR for general-purpose performance monitoring counter <i>n</i>.</p> <p>Bits 31-00: Bitmap of Auto Counter Reload (ACR) fixed counters that can be reloaded. For each bit <i>m</i> set in this field, the processor supports ACR for fixed-function performance monitoring counter <i>m</i>.</p> <p>Bits 31-00: Bitmap of Auto Counter Reload (ACR) general counters that can cause reloads. For each bit <i>y</i> set in this field, the processor allows general-purpose performance monitoring counter <i>y</i> to reload all existing general-purpose performance monitoring counters capable of being reloaded.</p> <p>Bits 31-00: Bitmap of Auto Counter Reload (ACR) fixed counters that can cause reloads. For each bit <i>x</i> set in this field, the processor allows fixed-function performance monitoring counter <i>x</i> to reload all existing fixed-function performance monitoring counters capable of being reloaded.</p>
Architectural Performance Monitoring Extended Sub-Leaf (Initial EAX Value = 23H, ECX = 3)		
23H	<p>EAX</p>	<p>NOTE: Architectural Performance Monitoring Events Bitmap. For each bit <i>n</i> set in this field, the processor supports Architectural Performance Monitoring Event of index <i>n</i>.</p> <p>Bit 00: Core cycles.</p> <p>Bit 01: Instructions retired.</p> <p>Bit 02: Reference cycles.</p> <p>Bit 03: Last level cache references.</p> <p>Bit 04: Last level cache misses.</p> <p>Bit 05: Branch instructions retired.</p> <p>Bit 06: Branch mispredicts retired.</p> <p>Bit 07: Topdown slots.</p> <p>Bit 08: Topdown backend bound.</p> <p>Bit 09: Topdown bad speculation.</p>

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
	EBX ECX EDX	Bit 10: Topdown frontend bound. Bit 11: Topdown retiring. Bit 12: LBR inserts. Bits 31-13: Reserved. Bits 31-00: Reserved. Bits 31-00: Reserved. Bits 31-00: Reserved.
<i>Converged Vector ISA Main Leaf (Initial EAX Value = 24H, ECX = 0)</i>		
24H	EAX EBX ECX EDX	NOTE: Output depends on ECX input value. Bits 31-00: Reports the maximum number sub-leaves that are supported in leaf 24H. Bits 07-00: Reports the Intel AVX10 Converged Vector ISA version. Bits 15-08: Reserved. Bit 16: If 1, indicates that 128-bit vector support is present. Bit 17: If 1, indicates that 256-bit vector support is present. Bit 18: If 1, indicates that 512-bit vector support is present. Bits 31-19: Reserved. Bits 31-00: Reserved. Bits 31-00: Reserved.
<i>Unimplemented CPUID Leaf Functions</i>		
21H		Invalid. No existing or future CPU will return processor identification or feature information if the initial EAX value is 21H. If the value returned by CPUID.0:EAX (the maximum input value for basic CPUID information) is at least 21H, 0 is returned in the registers EAX, EBX, ECX, and EDX. Otherwise, the data for the highest basic information leaf is returned.
40000000H — 4FFFFFFFH		Invalid. No existing or future CPU will return processor identification or feature information if the initial EAX value is in the range 40000000H to 4FFFFFFFH.
<i>Extended Function CPUID Information</i>		
80000000H	EAX EBX ECX EDX	Maximum Input Value for Extended Function CPUID Information. Reserved. Reserved. Reserved.
80000001H	EAX EBX ECX	Extended Processor Signature and Feature Bits. Reserved. Bit 00: LAHF/SAHF available in 64-bit mode.* Bits 04-01: Reserved. Bit 05: LZCNT. Bits 07-06: Reserved. Bit 08: PREFETCHW. Bits 31-09: Reserved.

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor																	
	EDX	Bits 10-00: Reserved. Bit 11: SYSCALL/SYSRET.** Bits 19-12: Reserved = 0. Bit 20: Execute Disable Bit available. Bits 25-21: Reserved = 0. Bit 26: 1-GByte pages are available if 1. Bit 27: RDTSCP and IA32_TSC_AUX are available if 1. Bit 28: Reserved = 0. Bit 29: Intel® 64 Architecture available if 1. Bits 31-30: Reserved = 0. NOTES: * LAHF and SAHF are always available in other modes, regardless of the enumeration of this feature flag. ** Intel processors support SYSCALL and SYSRET only in 64-bit mode. This feature flag is always enumerated as 0 outside 64-bit mode.																
80000002H	EAX EBX ECX EDX	Processor Brand String. Processor Brand String Continued. Processor Brand String Continued. Processor Brand String Continued.																
80000003H	EAX EBX ECX EDX	Processor Brand String Continued. Processor Brand String Continued. Processor Brand String Continued. Processor Brand String Continued.																
80000004H	EAX EBX ECX EDX	Processor Brand String Continued. Processor Brand String Continued. Processor Brand String Continued. Processor Brand String Continued.																
80000005H	EAX EBX ECX EDX	Reserved = 0. Reserved = 0. Reserved = 0. Reserved = 0.																
80000006H	EAX EBX ECX EDX	Reserved = 0. Reserved = 0. Bits 07-00: Cache Line size in bytes. Bits 11-08: Reserved. Bits 15-12: L2 Associativity field *. Bits 31-16: Cache size in 1K units. Reserved = 0. NOTES: * L2 associativity field encodings: <table data-bbox="407 1581 1076 1816" style="margin-left: 20px;"> <tr> <td>00H - Disabled</td> <td>08H - 16 ways</td> </tr> <tr> <td>01H - 1 way (direct mapped)</td> <td>09H - Reserved</td> </tr> <tr> <td>02H - 2 ways</td> <td>0AH - 32 ways</td> </tr> <tr> <td>03H - Reserved</td> <td>0BH - 48 ways</td> </tr> <tr> <td>04H - 4 ways</td> <td>0CH - 64 ways</td> </tr> <tr> <td>05H - Reserved</td> <td>0DH - 96 ways</td> </tr> <tr> <td>06H - 8 ways</td> <td>0EH - 128 ways</td> </tr> <tr> <td>07H - See CPUID leaf 04H, sub-leaf 2**</td> <td>0FH - Fully associative</td> </tr> </table> ** CPUID leaf 04H provides details of deterministic cache parameters, including the L2 cache in sub-leaf 2	00H - Disabled	08H - 16 ways	01H - 1 way (direct mapped)	09H - Reserved	02H - 2 ways	0AH - 32 ways	03H - Reserved	0BH - 48 ways	04H - 4 ways	0CH - 64 ways	05H - Reserved	0DH - 96 ways	06H - 8 ways	0EH - 128 ways	07H - See CPUID leaf 04H, sub-leaf 2**	0FH - Fully associative
00H - Disabled	08H - 16 ways																	
01H - 1 way (direct mapped)	09H - Reserved																	
02H - 2 ways	0AH - 32 ways																	
03H - Reserved	0BH - 48 ways																	
04H - 4 ways	0CH - 64 ways																	
05H - Reserved	0DH - 96 ways																	
06H - 8 ways	0EH - 128 ways																	
07H - See CPUID leaf 04H, sub-leaf 2**	0FH - Fully associative																	

Table 3-17. Information Returned by CPUID Instruction (Contd.)

Initial EAX Value	Information Provided about the Processor	
80000007H	EAX EBX ECX EDX	Reserved = 0. Reserved = 0. Reserved = 0. Bits 07-00: Reserved = 0. Bit 08: Invariant TSC available if 1. Bits 31-09: Reserved = 0.
80000008H	EAX EBX ECX EDX	Linear/Physical Address size. Bits 07-00: #Physical Address Bits*. Bits 15-08: #Linear Address Bits. Bits 23-16: #Guest Physical Address Bits. This value applies only to software operating in a virtual machine (Intel processors enumerate this value as zero). When this field is zero, refer to #Physical Address Bits for the number of guest physical address bits. Bits 31-24: Reserved = 0. Bits 08-00: Reserved = 0. Bit 09: WBNOINVD is available if 1. Bits 31-10: Reserved = 0. Reserved = 0. Reserved = 0. NOTES: * If CPUID.80000008H:EAX[7:0] is supported, the maximum physical address number supported should come from this field. If TME-MK is enabled, the number of bits that can be used to address physical memory is CPUID.80000008H:EAX[7:0] - IA32_TME_ACTIVATE[35:32].

NOTES:

1. The valid range of fixed-function counters is 0 through 15.

INPUT EAX = 0: Returns CPUID’s Highest Value for Basic Processor Information and the Vendor Identification String

When CPUID executes with EAX set to 0, the processor returns the highest value the CPUID recognizes for returning basic processor information. The value is returned in the EAX register and is processor specific.

A vendor identification string is also returned in EBX, EDX, and ECX. For Intel processors, the string is “GenuineIntel” and is expressed:

- EBX := 756e6547h (* “Genu”, with G in the low eight bits of BL *)
- EDX := 49656e69h (* “inel”, with i in the low eight bits of DL *)
- ECX := 6c65746eh (* “ntel”, with n in the low eight bits of CL *)

INPUT EAX = 80000000H: Returns CPUID’s Highest Value for Extended Processor Information

When CPUID executes with EAX set to 80000000H, the processor returns the highest value the processor recognizes for returning extended processor information. The value is returned in the EAX register and is processor specific.

IA32_BIOS_SIGN_ID Returns Microcode Update Signature

For processors that support the microcode update facility, the IA32_BIOS_SIGN_ID MSR is loaded with the update signature whenever CPUID executes. The signature is returned in the upper DWORD. For details, see Chapter 11 in the Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 3A.

INPUT EAX = 01H: Returns Model, Family, Stepping Information

When CPUID executes with EAX set to 01H, version information is returned in EAX (see Figure 3-6). For example: model, family, and processor type for the Intel Xeon processor 5100 series is as follows:

- Model — 1111B

- Family — 0101B
- Processor Type — 00B

See Table 3-18 for available processor type values. Stepping IDs are provided as needed.

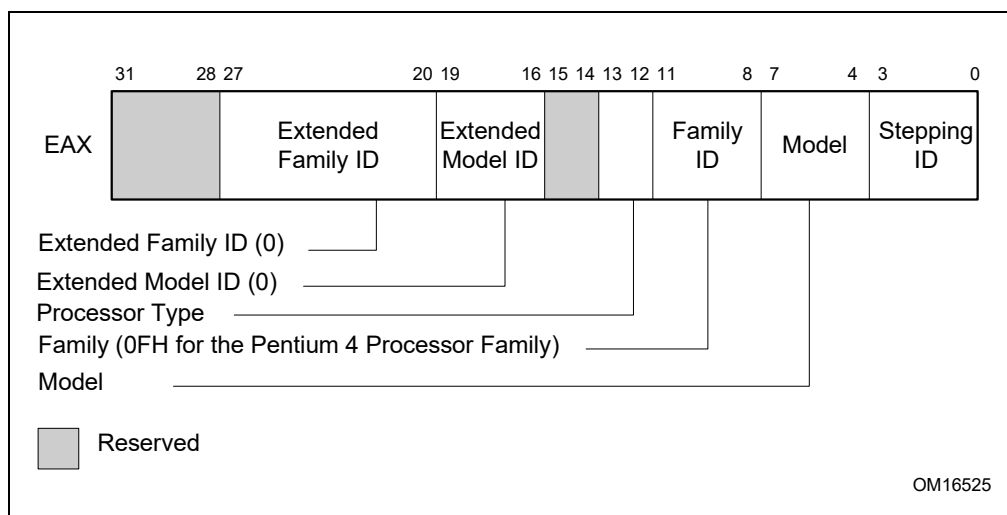


Figure 3-6. Version Information Returned by CPUID in EAX

Table 3-18. Processor Type Field

Type	Encoding
Original OEM Processor	00B
Intel OverDrive™ Processor	01B
Dual processor (not applicable to Intel486 processors)	10B
Intel reserved	11B

NOTE

See Chapter 21 in the Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 1, for information on identifying earlier IA-32 processors.

The Extended Family ID needs to be examined only when the Family ID is 0FH. Integrate the fields into a display using the following rule:

```

IF Family_ID ≠ 0FH
  THEN DisplayFamily = Family_ID;
  ELSE DisplayFamily = Extended_Family_ID + Family_ID;
FI;
(* Show DisplayFamily as HEX field. *)

```

The Extended Model ID needs to be examined only when the Family ID is 06H or 0FH. Integrate the field into a display using the following rule:

```

IF (Family_ID = 06H or Family_ID = 0FH)
  THEN DisplayModel = (Extended_Model_ID « 4) + Model_ID;
  (* Right justify and zero-extend 4-bit field; display Model_ID as HEX field.*)
  ELSE DisplayModel = Model_ID;
FI;

```

(* Show DisplayModel as HEX field. *)

INPUT EAX = 01H: Returns Additional Information in EBX

When CPUID executes with EAX set to 01H, additional information is returned to the EBX register:

- Brand index (low byte of EBX) — this number provides an entry into a brand string table that contains brand strings for IA-32 processors. More information about this field is provided later in this section.
- CLFLUSH instruction cache line size (second byte of EBX) — this number indicates the size of the cache line flushed by the CLFLUSH and CLFLUSHOPT instructions in 8-byte increments. This field was introduced in the Pentium 4 processor.
- Local APIC ID (high byte of EBX) — this number is the 8-bit ID that is assigned to the local APIC on the processor during power up. This field was introduced in the Pentium 4 processor.

INPUT EAX = 01H: Returns Feature Information in ECX and EDX

When CPUID executes with EAX set to 01H, feature information is returned in ECX and EDX.

- Figure 3-7 and Table 3-19 show encodings for ECX.
- Figure 3-8 and Table 3-20 show encodings for EDX.

For all feature flags, a 1 indicates that the feature is supported. Use Intel to properly interpret feature flags.

NOTE

Software must confirm that a processor feature is present using feature flags returned by CPUID prior to using the feature. Software should not depend on future offerings retaining all features.

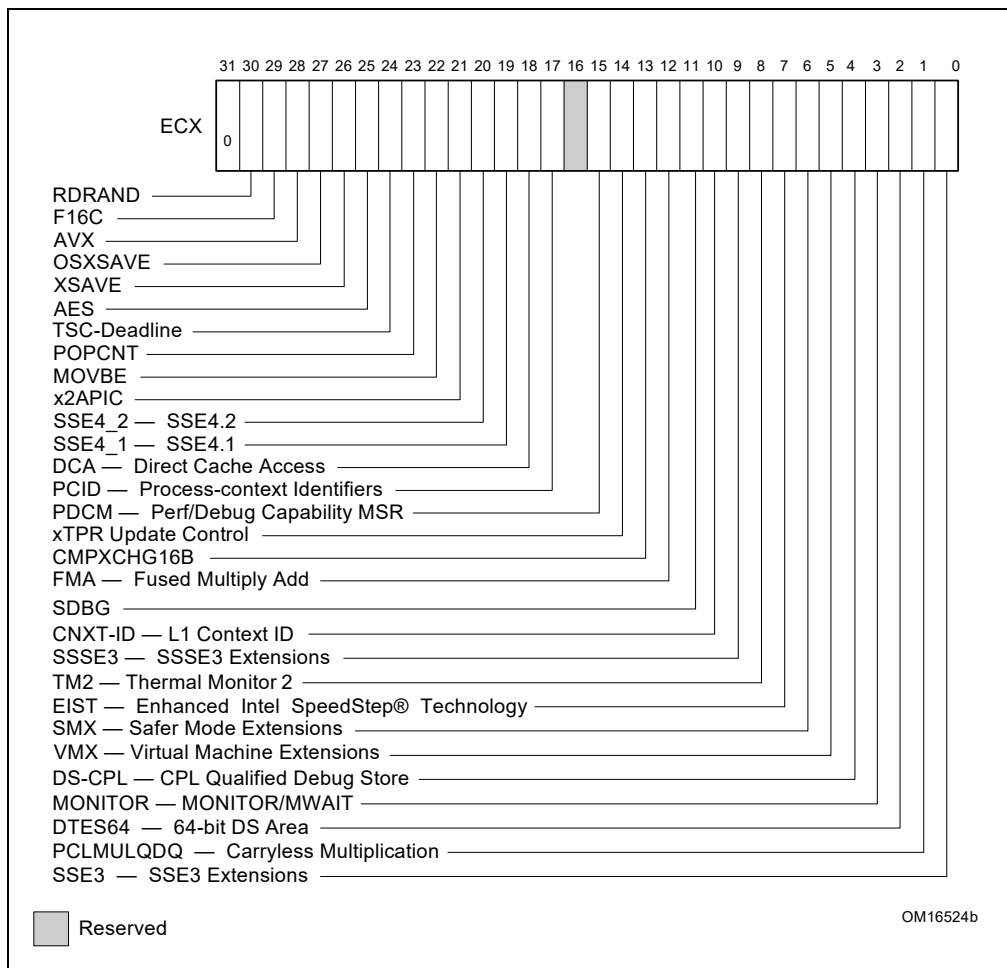


Figure 3-7. Feature Information Returned in the ECX Register

Table 3-19. Feature Information Returned in the ECX Register

Bit #	Mnemonic	Description
0	SSE3	Streaming SIMD Extensions 3 (SSE3). A value of 1 indicates the processor supports this technology.
1	PCLMULQDQ	PCLMULQDQ. A value of 1 indicates the processor supports the PCLMULQDQ instruction.
2	DTES64	64-bit DS Area. A value of 1 indicates the processor supports DS area using 64-bit layout.
3	MONITOR	MONITOR/MWAIT. A value of 1 indicates the processor supports this feature.
4	DS-CPL	CPL Qualified Debug Store. A value of 1 indicates the processor supports the extensions to the Debug Store feature to allow for branch message storage qualified by CPL.
5	VMX	Virtual Machine Extensions. A value of 1 indicates that the processor supports this technology.
6	SMX	Safer Mode Extensions. A value of 1 indicates that the processor supports this technology. See Chapter 7, “Safer Mode Extensions Reference.”
7	EIST	Enhanced Intel SpeedStep® technology. A value of 1 indicates that the processor supports this technology.
8	TM2	Thermal Monitor 2. A value of 1 indicates whether the processor supports this technology.
9	SSSE3	A value of 1 indicates the presence of the Supplemental Streaming SIMD Extensions 3 (SSSE3). A value of 0 indicates the instruction extensions are not present in the processor.
10	CNXT-ID	L1 Context ID. A value of 1 indicates the L1 data cache mode can be set to either adaptive mode or shared mode. A value of 0 indicates this feature is not supported. See definition of the IA32_MISC_ENABLE MSR Bit 24 (L1 Data Cache Context Mode) for details.
11	SDBG	A value of 1 indicates the processor supports IA32_DEBUG_INTERFACE MSR for silicon debug.
12	FMA	A value of 1 indicates the processor supports FMA extensions using YMM state.
13	CMPXCHG16B	CMPXCHG16B Available. A value of 1 indicates that the feature is available. See the “CMPXCHG8B/CMPXCHG16B—Compare and Exchange Bytes” section in this chapter for a description.
14	xTPR Update Control	xTPR Update Control. A value of 1 indicates that the processor supports changing IA32_MISC_ENABLE[bit 23].
15	PDCM	Perfmon and Debug Capability: A value of 1 indicates the processor supports the performance and debug feature indication MSR IA32_PERF_CAPABILITIES.
16	Reserved	Reserved
17	PCID	Process-context identifiers. A value of 1 indicates that the processor supports PCIDs and that software may set CR4.PCIDE to 1.
18	DCA	A value of 1 indicates the processor supports the ability to prefetch data from a memory mapped device.
19	SSE4_1	A value of 1 indicates that the processor supports SSE4.1.
20	SSE4_2	A value of 1 indicates that the processor supports SSE4.2.
21	x2APIC	A value of 1 indicates that the processor supports x2APIC feature.
22	MOVBE	A value of 1 indicates that the processor supports MOVBE instruction.
23	POPCNT	A value of 1 indicates that the processor supports the POPCNT instruction.
24	TSC-Deadline	A value of 1 indicates that the processor’s local APIC timer supports one-shot operation using a TSC deadline value.
25	AESNI	A value of 1 indicates that the processor supports the AESNI instruction extensions.
26	XSAVE	A value of 1 indicates that the processor supports the XSAVE/XRSTOR processor extended states feature, the XSETBV/XGETBV instructions, and XCRO.
27	OSXSAVE	A value of 1 indicates that the OS has set CR4.OSXSAVE[bit 18] to enable XSETBV/XGETBV instructions to access XCRO and to support processor extended state management using XSAVE/XRSTOR.
28	AVX	A value of 1 indicates the processor supports the AVX instruction extensions.

Table 3-19. Feature Information Returned in the ECX Register (Contd.)

Bit #	Mnemonic	Description
29	F16C	A value of 1 indicates that processor supports 16-bit floating-point conversion instructions.
30	RDRAND	A value of 1 indicates that processor supports RDRAND instruction.
31	Not Used	Always returns 0.

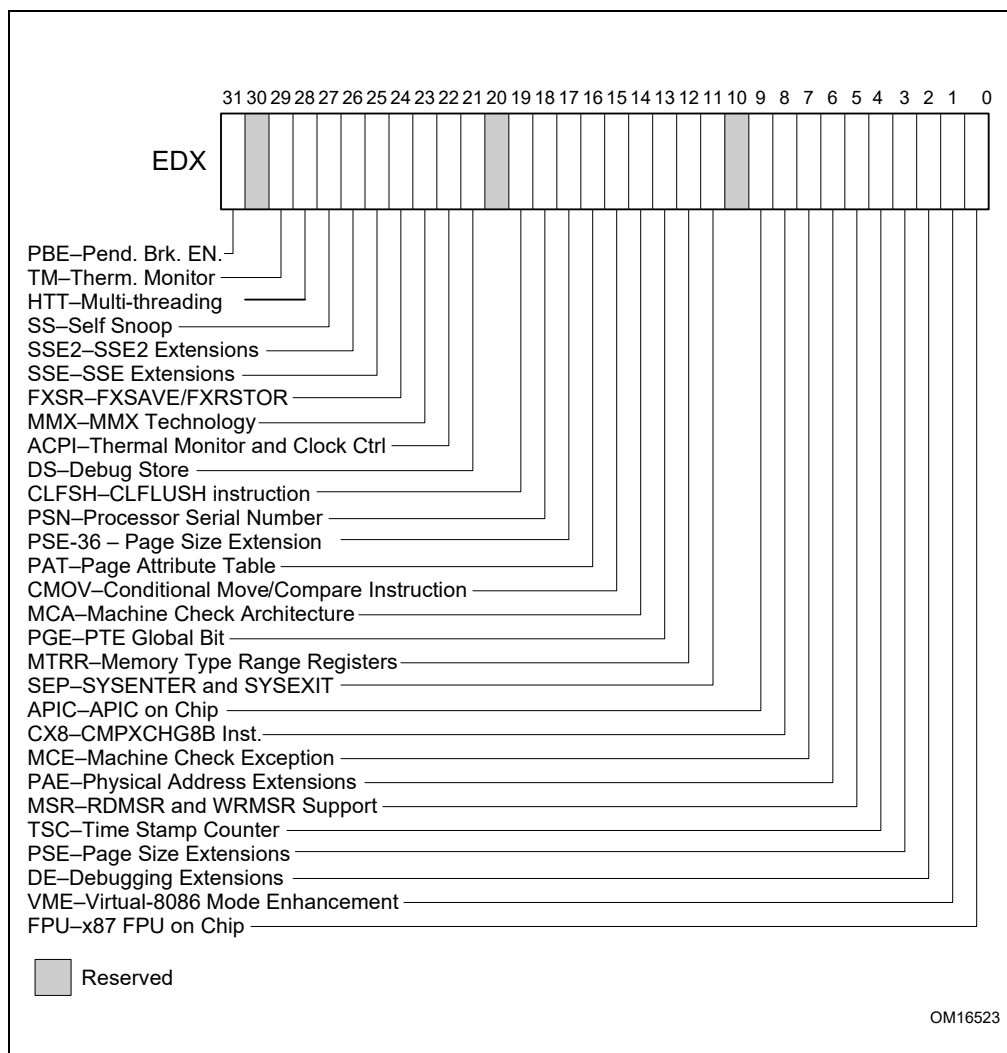


Figure 3-8. Feature Information Returned in the EDX Register

Table 3-20. More on Feature Information Returned in the EDX Register

Bit #	Mnemonic	Description
0	FPU	Floating-Point Unit On-Chip. The processor contains an x87 FPU.
1	VME	Virtual 8086 Mode Enhancements. Virtual 8086 mode enhancements, including CR4.VME for controlling the feature, CR4.PVI for protected mode virtual interrupts, software interrupt indirection, expansion of the TSS with the software indirection bitmap, and EFLAGS.VIF and EFLAGS.VIP flags.
2	DE	Debugging Extensions. Support for I/O breakpoints, including CR4.DE for controlling the feature, and optional trapping of accesses to DR4 and DR5.
3	PSE	Page Size Extension. Large pages of size 4 MByte are supported, including CR4.PSE for controlling the feature, the defined dirty bit in PDE (Page Directory Entries), optional reserved bit trapping in CR3, PDEs, and PTEs.
4	TSC	Time Stamp Counter. The RDTSC instruction is supported, including CR4.TSD for controlling privilege.
5	MSR	Model Specific Registers RDMSR and WRMSR Instructions. The RDMSR and WRMSR instructions are supported. Some of the MSRs are implementation dependent.
6	PAE	Physical Address Extension. Physical addresses greater than 32 bits are supported: extended page table entry formats, an extra level in the page translation tables is defined, 2-MByte pages are supported instead of 4 Mbyte pages if PAE bit is 1.
7	MCE	Machine Check Exception. Exception 18 is defined for Machine Checks, including CR4.MCE for controlling the feature. This feature does not define the model-specific implementations of machine-check error logging, reporting, and processor shutdowns. Machine Check exception handlers may have to depend on processor version to do model specific processing of the exception, or test for the presence of the Machine Check feature.
8	CX8	CMPXCHG8B Instruction. The compare-and-exchange 8 bytes (64 bits) instruction is supported (implicitly locked and atomic).
9	APIC	APIC On-Chip. The processor contains an Advanced Programmable Interrupt Controller (APIC), responding to memory mapped commands in the physical address range FFFE0000H to FFFE0FFFH (by default - some processors permit the APIC to be relocated).
10	Reserved	Reserved
11	SEP	SYSENTER and SYSEXIT Instructions. The SYSENTER and SYSEXIT and associated MSRs are supported.
12	MTRR	Memory Type Range Registers. MTRRs are supported. The MTRRcap MSR contains feature bits that describe what memory types are supported, how many variable MTRRs are supported, and whether fixed MTRRs are supported.
13	PGE	Page Global Bit. The global bit is supported in paging-structure entries that map a page, indicating TLB entries that are common to different processes and need not be flushed. The CR4.PGE bit controls this feature.
14	MCA	Machine Check Architecture. A value of 1 indicates the Machine Check Architecture of reporting machine errors is supported. The MCG_CAP MSR contains feature bits describing how many banks of error reporting MSRs are supported.
15	CMOV	Conditional Move Instructions. The conditional move instruction CMOV is supported. In addition, if x87 FPU is present as indicated by the CPUID.FPU feature bit, then the FCOMI and FCMOV instructions are supported
16	PAT	Page Attribute Table. Page Attribute Table is supported. This feature augments the Memory Type Range Registers (MTRRs), allowing an operating system to specify attributes of memory accessed through a linear address on a 4KB granularity.
17	PSE-36	36-Bit Page Size Extension. 4-MByte pages addressing physical memory beyond 4 GBytes are supported with 32-bit paging. This feature indicates that upper bits of the physical address of a 4-MByte page are encoded in bits 20:13 of the page-directory entry. Such physical addresses are limited by MAXPHYADDR and may be up to 40 bits in size.
18	PSN	Processor Serial Number. The processor supports the 96-bit processor identification number feature and the feature is enabled.
19	CLFSH	CLFLUSH Instruction. CLFLUSH Instruction is supported.
20	Reserved	Reserved

Table 3-20. More on Feature Information Returned in the EDX Register (Contd.)

Bit #	Mnemonic	Description
21	DS	Debug Store. The processor supports the ability to write debug information into a memory resident buffer. This feature is used by the branch trace store (BTS) and processor event-based sampling (PEBS) facilities (see Chapter 25, "Introduction to Virtual Machine Extensions," in the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3C).
22	ACPI	Thermal Monitor and Software Controlled Clock Facilities. The processor implements internal MSRs that allow processor temperature to be monitored and processor performance to be modulated in predefined duty cycles under software control.
23	MMX	Intel MMX Technology. The processor supports the Intel MMX technology.
24	FXSR	FXSAVE and FXRSTOR Instructions. The FXSAVE and FXRSTOR instructions are supported for fast save and restore of the floating-point context. Presence of this bit also indicates that CR4.OSFXSR is available for an operating system to indicate that it supports the FXSAVE and FXRSTOR instructions.
25	SSE	SSE. The processor supports the SSE extensions.
26	SSE2	SSE2. The processor supports the SSE2 extensions.
27	SS	Self Snoop. The processor supports the management of conflicting memory types by performing a snoop of its own cache structure for transactions issued to the bus.
28	HTT	Max APIC IDs reserved field is Valid. A value of 0 for HTT indicates there is only a single logical processor in the package and software should assume only a single APIC ID is reserved. A value of 1 for HTT indicates the value in CPUID.1.EBX[23:16] (the Maximum number of addressable IDs for logical processors in this package) is valid for the package.
29	TM	Thermal Monitor. The processor implements the thermal monitor automatic thermal control circuitry (TCC).
30	Reserved	Reserved
31	PBE	Pending Break Enable. The processor supports the use of the FERR#/PBE# pin when the processor is in the stop-clock state (STPCLK# is asserted) to signal the processor that an interrupt is pending and that the processor should return to normal operation to handle the interrupt.

INPUT EAX = 02H: TLB/Cache/Prefetch Information Returned in EAX, EBX, ECX, EDX

When CPUID executes with EAX set to 02H, the processor returns information about the processor's internal TLBs, cache, and prefetch hardware in the EAX, EBX, ECX, and EDX registers. The information is reported in encoded form and fall into the following categories:

- The least-significant byte in register EAX (register AL) will always return 01H. Software should ignore this value and not interpret it as an informational descriptor.
- The most significant bit (bit 31) of each register indicates whether the register contains valid information (set to 0) or is reserved (set to 1).
- If a register contains valid information, the information is contained in 1 byte descriptors. There are four types of encoding values for the byte descriptor, the encoding type is noted in the second column of Table 3-21. Table 3-21 lists the encoding of these descriptors. Note that the order of descriptors in the EAX, EBX, ECX, and EDX registers is not defined; that is, specific bytes are not designated to contain descriptors for specific cache, prefetch, or TLB types. The descriptors may appear in any order. Note also a processor may report a general descriptor type (FFH) and not report any byte descriptor of "cache type" via CPUID leaf 2.

Table 3-21. Encoding of CPUID Leaf 2 Descriptors

Descriptor Value	Type	Cache or TLB Description
00H	General	Null descriptor, this byte contains no information.
01H	TLB	Instruction TLB: 4 KByte pages, 4-way set associative, 32 entries.
02H	TLB	Instruction TLB: 4 MByte pages, fully associative, 2 entries.
03H	TLB	Data TLB: 4 KByte pages, 4-way set associative, 64 entries.
04H	TLB	Data TLB: 4 MByte pages, 4-way set associative, 8 entries.
05H	TLB	Data TLB1: 4 MByte pages, 4-way set associative, 32 entries.
06H	Cache	1st-level instruction cache: 8 KBytes, 4-way set associative, 32 byte line size.
08H	Cache	1st-level instruction cache: 16 KBytes, 4-way set associative, 32 byte line size.
09H	Cache	1st-level instruction cache: 32KBytes, 4-way set associative, 64 byte line size.
0AH	Cache	1st-level data cache: 8 KBytes, 2-way set associative, 32 byte line size.
0BH	TLB	Instruction TLB: 4 MByte pages, 4-way set associative, 4 entries.
0CH	Cache	1st-level data cache: 16 KBytes, 4-way set associative, 32 byte line size.
0DH	Cache	1st-level data cache: 16 KBytes, 4-way set associative, 64 byte line size.
0EH	Cache	1st-level data cache: 24 KBytes, 6-way set associative, 64 byte line size.
1DH	Cache	2nd-level cache: 128 KBytes, 2-way set associative, 64 byte line size.
21H	Cache	2nd-level cache: 256 KBytes, 8-way set associative, 64 byte line size.
22H	Cache	3rd-level cache: 512 KBytes, 4-way set associative, 64 byte line size, 2 lines per sector.
23H	Cache	3rd-level cache: 1 MBytes, 8-way set associative, 64 byte line size, 2 lines per sector.
24H	Cache	2nd-level cache: 1 MBytes, 16-way set associative, 64 byte line size.
25H	Cache	3rd-level cache: 2 MBytes, 8-way set associative, 64 byte line size, 2 lines per sector.
29H	Cache	3rd-level cache: 4 MBytes, 8-way set associative, 64 byte line size, 2 lines per sector.
2CH	Cache	1st-level data cache: 32 KBytes, 8-way set associative, 64 byte line size.
30H	Cache	1st-level instruction cache: 32 KBytes, 8-way set associative, 64 byte line size.
40H	Cache	No 2nd-level cache or, if processor contains a valid 2nd-level cache, no 3rd-level cache.
41H	Cache	2nd-level cache: 128 KBytes, 4-way set associative, 32 byte line size.
42H	Cache	2nd-level cache: 256 KBytes, 4-way set associative, 32 byte line size.
43H	Cache	2nd-level cache: 512 KBytes, 4-way set associative, 32 byte line size.
44H	Cache	2nd-level cache: 1 MByte, 4-way set associative, 32 byte line size.
45H	Cache	2nd-level cache: 2 MByte, 4-way set associative, 32 byte line size.
46H	Cache	3rd-level cache: 4 MByte, 4-way set associative, 64 byte line size.
47H	Cache	3rd-level cache: 8 MByte, 8-way set associative, 64 byte line size.
48H	Cache	2nd-level cache: 3MByte, 12-way set associative, 64 byte line size.
49H	Cache	3rd-level cache: 4MB, 16-way set associative, 64-byte line size (Intel Xeon processor MP, Family 0FH, Model 06H); 2nd-level cache: 4 MByte, 16-way set associative, 64 byte line size.
4AH	Cache	3rd-level cache: 6MByte, 12-way set associative, 64 byte line size.
4BH	Cache	3rd-level cache: 8MByte, 16-way set associative, 64 byte line size.
4CH	Cache	3rd-level cache: 12MByte, 12-way set associative, 64 byte line size.
4DH	Cache	3rd-level cache: 16MByte, 16-way set associative, 64 byte line size.
4EH	Cache	2nd-level cache: 6MByte, 24-way set associative, 64 byte line size.
4FH	TLB	Instruction TLB: 4 KByte pages, 32 entries.

Table 3-21. Encoding of CPUID Leaf 2 Descriptors (Contd.)

Descriptor Value	Type	Cache or TLB Description
50H	TLB	Instruction TLB: 4 KByte and 2-MByte or 4-MByte pages, 64 entries.
51H	TLB	Instruction TLB: 4 KByte and 2-MByte or 4-MByte pages, 128 entries.
52H	TLB	Instruction TLB: 4 KByte and 2-MByte or 4-MByte pages, 256 entries.
55H	TLB	Instruction TLB: 2-MByte or 4-MByte pages, fully associative, 7 entries.
56H	TLB	Data TLB0: 4 MByte pages, 4-way set associative, 16 entries.
57H	TLB	Data TLB0: 4 KByte pages, 4-way associative, 16 entries.
59H	TLB	Data TLB0: 4 KByte pages, fully associative, 16 entries.
5AH	TLB	Data TLB0: 2 MByte or 4 MByte pages, 4-way set associative, 32 entries.
5BH	TLB	Data TLB: 4 KByte and 4 MByte pages, 64 entries.
5CH	TLB	Data TLB: 4 KByte and 4 MByte pages, 128 entries.
5DH	TLB	Data TLB: 4 KByte and 4 MByte pages, 256 entries.
60H	Cache	1st-level data cache: 16 KByte, 8-way set associative, 64 byte line size.
61H	TLB	Instruction TLB: 4 KByte pages, fully associative, 48 entries.
63H	TLB	Data TLB: 2 MByte or 4 MByte pages, 4-way set associative, 32 entries and a separate array with 1 GByte pages, 4-way set associative, 4 entries.
64H	TLB	Data TLB: 4 KByte pages, 4-way set associative, 512 entries.
66H	Cache	1st-level data cache: 8 KByte, 4-way set associative, 64 byte line size.
67H	Cache	1st-level data cache: 16 KByte, 4-way set associative, 64 byte line size.
68H	Cache	1st-level data cache: 32 KByte, 4-way set associative, 64 byte line size.
6AH	Cache	uTLB: 4 KByte pages, 8-way set associative, 64 entries.
6BH	Cache	DTLB: 4 KByte pages, 8-way set associative, 256 entries.
6CH	Cache	DTLB: 2M/4M pages, 8-way set associative, 128 entries.
6DH	Cache	DTLB: 1 GByte pages, fully associative, 16 entries.
70H	Cache	Trace cache: 12 K- μ op, 8-way set associative.
71H	Cache	Trace cache: 16 K- μ op, 8-way set associative.
72H	Cache	Trace cache: 32 K- μ op, 8-way set associative.
76H	TLB	Instruction TLB: 2M/4M pages, fully associative, 8 entries.
78H	Cache	2nd-level cache: 1 MByte, 4-way set associative, 64byte line size.
79H	Cache	2nd-level cache: 128 KByte, 8-way set associative, 64 byte line size, 2 lines per sector.
7AH	Cache	2nd-level cache: 256 KByte, 8-way set associative, 64 byte line size, 2 lines per sector.
7BH	Cache	2nd-level cache: 512 KByte, 8-way set associative, 64 byte line size, 2 lines per sector.
7CH	Cache	2nd-level cache: 1 MByte, 8-way set associative, 64 byte line size, 2 lines per sector.
7DH	Cache	2nd-level cache: 2 MByte, 8-way set associative, 64byte line size.
7FH	Cache	2nd-level cache: 512 KByte, 2-way set associative, 64-byte line size.
80H	Cache	2nd-level cache: 512 KByte, 8-way set associative, 64-byte line size.
82H	Cache	2nd-level cache: 256 KByte, 8-way set associative, 32 byte line size.
83H	Cache	2nd-level cache: 512 KByte, 8-way set associative, 32 byte line size.
84H	Cache	2nd-level cache: 1 MByte, 8-way set associative, 32 byte line size.
85H	Cache	2nd-level cache: 2 MByte, 8-way set associative, 32 byte line size.
86H	Cache	2nd-level cache: 512 KByte, 4-way set associative, 64 byte line size.
87H	Cache	2nd-level cache: 1 MByte, 8-way set associative, 64 byte line size.

Table 3-21. Encoding of CPUID Leaf 2 Descriptors (Contd.)

Descriptor Value	Type	Cache or TLB Description
A0H	DTLB	DTLB: 4k pages, fully associative, 32 entries.
B0H	TLB	Instruction TLB: 4 KByte pages, 4-way set associative, 128 entries.
B1H	TLB	Instruction TLB: 2M pages, 4-way, 8 entries or 4M pages, 4-way, 4 entries.
B2H	TLB	Instruction TLB: 4KByte pages, 4-way set associative, 64 entries.
B3H	TLB	Data TLB: 4 KByte pages, 4-way set associative, 128 entries.
B4H	TLB	Data TLB1: 4 KByte pages, 4-way associative, 256 entries.
B5H	TLB	Instruction TLB: 4KByte pages, 8-way set associative, 64 entries.
B6H	TLB	Instruction TLB: 4KByte pages, 8-way set associative, 128 entries.
BAH	TLB	Data TLB1: 4 KByte pages, 4-way associative, 64 entries.
C0H	TLB	Data TLB: 4 KByte and 4 MByte pages, 4-way associative, 8 entries.
C1H	STLB	Shared 2nd-Level TLB: 4 KByte/2MByte pages, 8-way associative, 1024 entries.
C2H	DTLB	DTLB: 4 KByte/2 MByte pages, 4-way associative, 16 entries.
C3H	STLB	Shared 2nd-Level TLB: 4 KByte /2 MByte pages, 6-way associative, 1536 entries. Also 1GByte pages, 4-way, 16 entries.
C4H	DTLB	DTLB: 2M/4M Byte pages, 4-way associative, 32 entries.
CAH	STLB	Shared 2nd-Level TLB: 4 KByte pages, 4-way associative, 512 entries.
D0H	Cache	3rd-level cache: 512 KByte, 4-way set associative, 64 byte line size.
D1H	Cache	3rd-level cache: 1 MByte, 4-way set associative, 64 byte line size.
D2H	Cache	3rd-level cache: 2 MByte, 4-way set associative, 64 byte line size.
D6H	Cache	3rd-level cache: 1 MByte, 8-way set associative, 64 byte line size.
D7H	Cache	3rd-level cache: 2 MByte, 8-way set associative, 64 byte line size.
D8H	Cache	3rd-level cache: 4 MByte, 8-way set associative, 64 byte line size.
DCH	Cache	3rd-level cache: 1.5 MByte, 12-way set associative, 64 byte line size.
DDH	Cache	3rd-level cache: 3 MByte, 12-way set associative, 64 byte line size.
DEH	Cache	3rd-level cache: 6 MByte, 12-way set associative, 64 byte line size.
E2H	Cache	3rd-level cache: 2 MByte, 16-way set associative, 64 byte line size.
E3H	Cache	3rd-level cache: 4 MByte, 16-way set associative, 64 byte line size.
E4H	Cache	3rd-level cache: 8 MByte, 16-way set associative, 64 byte line size.
EAH	Cache	3rd-level cache: 12MByte, 24-way set associative, 64 byte line size.
EBH	Cache	3rd-level cache: 18MByte, 24-way set associative, 64 byte line size.
ECH	Cache	3rd-level cache: 24MByte, 24-way set associative, 64 byte line size.
FOH	Prefetch	64-Byte prefetching.
F1H	Prefetch	128-Byte prefetching.
FEH	General	CPUID leaf 2 does not report TLB descriptor information; use CPUID leaf 18H to query TLB and other address translation parameters.
FFH	General	CPUID leaf 2 does not report cache descriptor information, use CPUID leaf 4 to query cache parameters.

Example 3-1. Example of Cache and TLB Interpretation

The first member of the family of Pentium 4 processors returns the following information about caches and TLBs when the CPUID executes with an input value of 2:

```
EAX    66 5B 50 01H
EBX    0H
ECX    0H
EDX    00 7A 70 00H
```

Which means:

- The least-significant byte (byte 0) of register EAX is set to 01H. This value should be ignored.
- The most-significant bit of all four registers (EAX, EBX, ECX, and EDX) is set to 0, indicating that each register contains valid 1-byte descriptors.
- Bytes 1, 2, and 3 of register EAX indicate that the processor has:
 - 50H - a 64-entry instruction TLB, for mapping 4-KByte and 2-MByte or 4-MByte pages.
 - 5BH - a 64-entry data TLB, for mapping 4-KByte and 4-MByte pages.
 - 66H - an 8-KByte 1st level data cache, 4-way set associative, with a 64-Byte cache line size.
- The descriptors in registers EBX and ECX are valid, but contain NULL descriptors.
- Bytes 0, 1, 2, and 3 of register EDX indicate that the processor has:
 - 00H - NULL descriptor.
 - 70H - Trace cache: 12 K- μ op, 8-way set associative.
 - 7AH - a 256-KByte 2nd level cache, 8-way set associative, with a sectored, 64-byte cache line size.
 - 00H - NULL descriptor.

INPUT EAX = 04H: Returns Deterministic Cache Parameters for Each Level

When CPUID executes with EAX set to 04H and ECX contains an index value, the processor returns encoded data that describe a set of deterministic cache parameters (for the cache level associated with the input in ECX). Valid index values start from 0.

Software can enumerate the deterministic cache parameters for each level of the cache hierarchy starting with an index value of 0, until the parameters report the value associated with the cache type field is 0. The architecturally defined fields reported by deterministic cache parameters are documented in Table 3-17.

This Cache Size in Bytes

$$\begin{aligned} &= (\text{Ways} + 1) * (\text{Partitions} + 1) * (\text{Line_Size} + 1) * (\text{Sets} + 1) \\ &= (\text{EBX}[31:22] + 1) * (\text{EBX}[21:12] + 1) * (\text{EBX}[11:0] + 1) * (\text{ECX} + 1) \end{aligned}$$

The CPUID leaf 04H also reports data that can be used to derive the topology of processor cores in a physical package. This information is constant for all valid index values. Software can query the raw data reported by executing CPUID with EAX=04H and ECX=0 and use it as part of the topology enumeration algorithm described in Chapter 10, "Multiple-Processor Management," in the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A.

INPUT EAX = 05H: Returns MONITOR and MWAIT Features

When CPUID executes with EAX set to 05H, the processor returns information about features available to MONITOR/MWAIT instructions. The MONITOR instruction is used for address-range monitoring in conjunction with MWAIT instruction. The MWAIT instruction optionally provides additional extensions for advanced power management. See Table 3-17.

INPUT EAX = 06H: Returns Thermal and Power Management Features

When CPUID executes with EAX set to 06H, the processor returns information about thermal and power management features. See Table 3-17.

INPUT EAX = 07H: Returns Structured Extended Feature Enumeration Information

When CPUID executes with EAX set to 07H and ECX = 0, the processor returns information about the maximum input value for sub-leaves that contain extended feature flags. See Table 3-17.

When CPUID executes with EAX set to 07H and the input value of ECX is invalid (see leaf 07H entry in Table 3-17), the processor returns 0 in EAX/EBX/ECX/EDX. In subleaf 0, EAX returns the maximum input value of the highest leaf 7 sub-leaf, and EBX, ECX & EDX contain information of extended feature flags.

INPUT EAX = 09H: Returns Direct Cache Access Information

When CPUID executes with EAX set to 09H, the processor returns information about Direct Cache Access capabilities. See Table 3-17.

INPUT EAX = 0AH: Returns Architectural Performance Monitoring Features

When CPUID executes with EAX set to 0AH, the processor returns information about support for architectural performance monitoring capabilities. Architectural performance monitoring is supported if the version ID (see Table 3-17) is greater than Pn 0. See Table 3-17.

For each version of architectural performance monitoring capability, software must enumerate this leaf to discover the programming facilities and the architectural performance events available in the processor. The details are described in Chapter 25, "Introduction to Virtual Machine Extensions," in the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3C.

INPUT EAX = 0BH: Returns Extended Topology Information

CPUID leaf 1FH is a preferred superset to leaf 0BH. Intel recommends first checking for the existence of Leaf 1FH before using leaf 0BH.

When CPUID executes with EAX set to 0BH, the processor returns information about extended topology enumeration data. Software must detect the presence of CPUID leaf 0BH by verifying (a) the highest leaf index supported by CPUID is $\geq 0BH$, and (b) CPUID.0BH:EBX[15:0] reports a non-zero value. See Table 3-17.

INPUT EAX = 0DH: Returns Processor Extended States Enumeration Information

When CPUID executes with EAX set to 0DH and ECX = 0, the processor returns information about the bit-vector representation of all processor state extensions that are supported in the processor and storage size requirements of the XSAVE/XRSTOR area. See Table 3-17.

When CPUID executes with EAX set to 0DH and ECX = n ($n > 1$, and is a valid sub-leaf index), the processor returns information about the size and offset of each processor extended state save area within the XSAVE/XRSTOR area. See Table 3-17. Software can use the forward-extendable technique depicted below to query the valid sub-leaves and obtain size and offset information for each processor extended state save area:

```
For i = 2 to 62 // sub-leaf 1 is reserved
  IF (CPUID.(EAX=0DH, ECX=0H):VECTOR[i] = 1 ) // VECTOR is the 64-bit value of EDX:EAX
    Execute CPUID.(EAX=0DH, ECX = i) to examine size and offset for sub-leaf i;
  FI;
```

INPUT EAX = 0FH: Returns Intel Resource Director Technology (Intel RDT) Monitoring Enumeration Information

When CPUID executes with EAX set to 0FH and ECX = 0, the processor returns information about the bit-vector representation of QoS monitoring resource types that are supported in the processor and maximum range of RMID values the processor can use to monitor of any supported resource types. Each bit, starting from bit 1, corresponds to a specific resource type if the bit is set. The bit position corresponds to the sub-leaf index (or ResID) that software must use to query QoS monitoring capability available for that type. See Table 3-17.

When CPUID executes with EAX set to 0FH and ECX = n ($n \geq 1$, and is a valid ResID), the processor returns information software can use to program IA32_PQR_ASSOC, IA32_QM_EVTSEL MSRs before reading QoS data from the IA32_QM_CTR MSR.

INPUT EAX = 10H: Returns Intel Resource Director Technology (Intel RDT) Allocation Enumeration Information

When CPUID executes with EAX set to 10H and ECX = 0, the processor returns information about the bit-vector representation of QoS Enforcement resource types that are supported in the processor. Each bit, starting from bit 1, corresponds to a specific resource type if the bit is set. The bit position corresponds to the sub-leaf index (or ResID) that software must use to query QoS enforcement capability available for that type. See Table 3-17.

When CPUID executes with EAX set to 10H and ECX = n (n >= 1, and is a valid ResID), the processor returns information about available classes of service and range of QoS mask MSRs that software can use to configure each class of services using capability bit masks in the QoS Mask registers, IA32_resourceType_Mask_n.

INPUT EAX = 12H: Returns Intel SGX Enumeration Information

When CPUID executes with EAX set to 12H and ECX = 0H, the processor returns information about Intel SGX capabilities. See Table 3-17.

When CPUID executes with EAX set to 12H and ECX = 1H, the processor returns information about Intel SGX attributes. See Table 3-17.

When CPUID executes with EAX set to 12H and ECX = n (n > 1), the processor returns information about Intel SGX Enclave Page Cache. See Table 3-17.

INPUT EAX = 14H: Returns Intel Processor Trace Enumeration Information

When CPUID executes with EAX set to 14H and ECX = 0H, the processor returns information about Intel Processor Trace extensions. See Table 3-17.

When CPUID executes with EAX set to 14H and ECX = n (n > 0 and less than the number of non-zero bits in CPUID.(EAX=14H, ECX= 0H).EAX), the processor returns information about packet generation in Intel Processor Trace. See Table 3-17.

INPUT EAX = 15H: Returns Time Stamp Counter and Nominal Core Crystal Clock Information

When CPUID executes with EAX set to 15H and ECX = 0H, the processor returns information about Time Stamp Counter and Core Crystal Clock. See Table 3-17.

INPUT EAX = 16H: Returns Processor Frequency Information

When CPUID executes with EAX set to 16H, the processor returns information about Processor Frequency Information. See Table 3-17.

INPUT EAX = 17H: Returns System-On-Chip Information

When CPUID executes with EAX set to 17H, the processor returns information about the System-On-Chip Vendor Attribute Enumeration. See Table 3-17.

INPUT EAX = 18H: Returns Deterministic Address Translation Parameters Information

When CPUID executes with EAX set to 18H, the processor returns information about the Deterministic Address Translation Parameters. See Table 3-17.

INPUT EAX = 19H: Returns Key Locker Information

When CPUID executes with EAX set to 19H, the processor returns information about Key Locker. See Table 3-17.

INPUT EAX = 1AH: Returns Native Model ID Information

When CPUID executes with EAX set to 1AH, the processor returns information about Native Model Identification. See Table 3-17.

INPUT EAX = 1BH: Returns PCONFIG Information

When CPUID executes with EAX set to 1BH, the processor returns information about PCONFIG capabilities. This information is enumerated in sub-leaves selected by the value of ECX (starting with 0).

Each sub-leaf of CPUID function 1BH enumerates its **sub-leaf type** in EAX. If a sub-leaf type is 0, the sub-leaf is invalid and zero is returned in EBX, ECX, and EDX. In this case, all subsequent sub-leaves (selected by larger input values of ECX) are also invalid.

The only valid sub-leaf type currently defined is 1, indicating that the sub-leaf enumerates target identifiers for the PCONFIG instruction. Any non-zero value returned in EBX, ECX, or EDX indicates a valid target identifier of the PCONFIG instruction (any value of zero should be ignored). The only target identifier currently defined is 1, indicating TME-MK. See the “PCONFIG—Platform Configuration” instruction in Chapter 4 of the Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 2B, for more information.

INPUT EAX = 1CH: Returns Last Branch Record Information

When CPUID executes with EAX set to 1CH, the processor returns information about LBRs (the architectural feature). See Table 3-17.

INPUT EAX = 1DH: Returns Tile Information

When CPUID executes with EAX set to 1DH and ECX = 0H, the processor returns information about tile architecture. See Table 3-17.

When CPUID executes with EAX set to 1DH and ECX = 1H, the processor returns information about tile palette 1. See Table 3-17.

INPUT EAX = 1EH: Returns TMUL Information

When CPUID executes with EAX set to 1EH and ECX = 0H, the processor returns information about TMUL capabilities. See Table 3-17.

INPUT EAX = 1FH: Returns V2 Extended Topology Information

When CPUID executes with EAX set to 1FH, the processor returns information about extended topology enumeration data. Software must detect the presence of CPUID leaf 1FH by verifying (a) the highest leaf index supported by CPUID is $\geq 1FH$, and (b) CPUID.1FH:EBX[15:0] reports a non-zero value. See Table 3-17.

INPUT EAX = 20H: Returns History Reset Information

When CPUID executes with EAX set to 20H, the processor returns information about History Reset. See Table 3-17.

INPUT EAX = 23H: Returns Architectural Performance Monitoring Extended Information

When CPUID executes with EAX set to 23H, the processor returns architectural performance monitoring extended information. See Table 3-17.

INPUT EAX = 24H: Returns Intel AVX10 Converged Vector ISA Information

When CPUID executes with EAX set to 24H, the processor returns Intel AVX10 converged vector ISA information. See Table 3-17.

METHODS FOR RETURNING BRANDING INFORMATION

Use the following techniques to access branding information:

1. Processor brand string method.
2. Processor brand index; this method uses a software supplied brand string table.

These two methods are discussed in the following sections. For methods that are available in early processors, see Section: “Identification of Earlier IA-32 Processors” in Chapter 21 of the Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 1.

The Processor Brand String Method

Figure 3-9 describes the algorithm used for detection of the brand string. Processor brand identification software should execute this algorithm on all Intel 64 and IA-32 processors.

This method (introduced with Pentium 4 processors) returns an ASCII brand identification string and the Processor Base frequency of the processor to the EAX, EBX, ECX, and EDX registers.

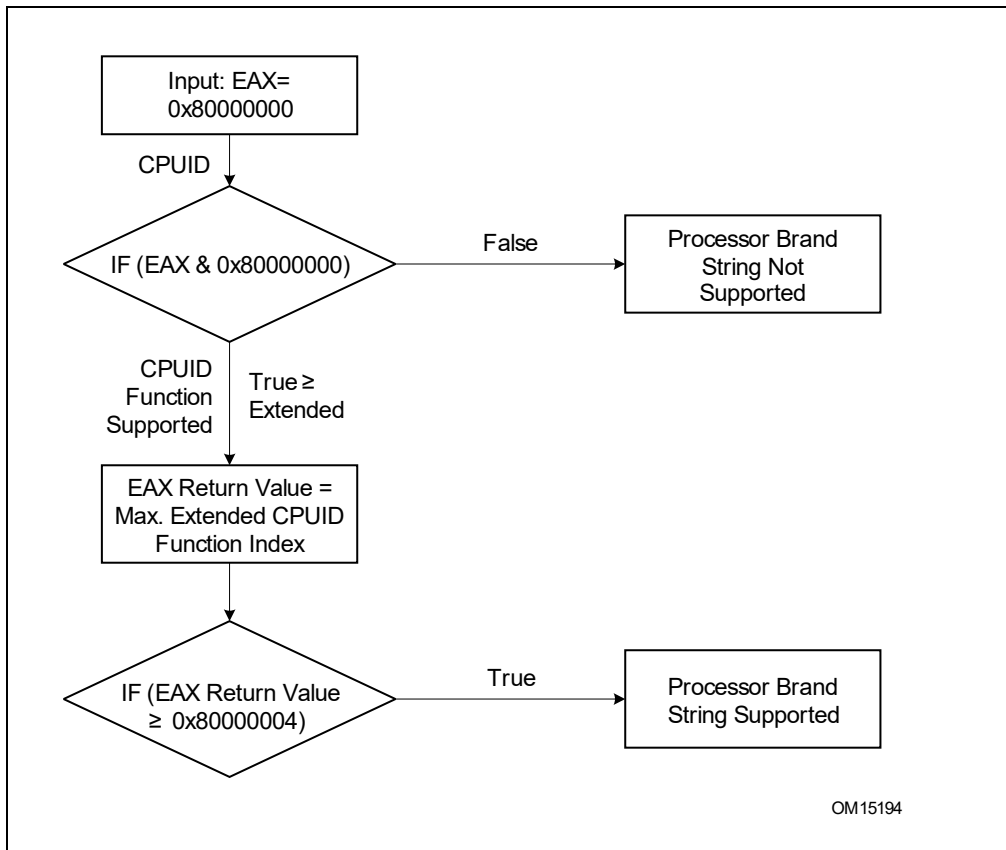


Figure 3-9. Determination of Support for the Processor Brand String

How Brand Strings Work

To use the brand string method, execute CPUID with EAX input of 8000002H through 80000004H. For each input value, CPUID returns 16 ASCII characters using EAX, EBX, ECX, and EDX. The returned string will be NULL-terminated.

Table 3-22 shows the brand string that is returned by the first processor in the Pentium 4 processor family.

Table 3-22. Processor Brand String Returned with Pentium 4 Processor

EAX Input Value	Return Values	ASCII Equivalent
80000002H	EAX = 20202020H EBX = 20202020H ECX = 20202020H EDX = 6E492020H	" " " " " " "nl "
80000003H	EAX = 286C6574H EBX = 50202952H ECX = 69746E65H EDX = 52286D75H	"(let" "P)R" "itne" "R(mu"

Table 3-22. Processor Brand String Returned with Pentium 4 Processor (Contd.)

EAX Input Value	Return Values	ASCII Equivalent
80000004H	EAX = 20342029H EBX = 20555043H ECX = 30303531H EDX = 007A484DH	" 4)" " UPC" "0051" "\0zHM"

Extracting the Processor Frequency from Brand Strings

Figure 3-10 provides an algorithm which software can use to extract the Processor Base frequency from the processor brand string.

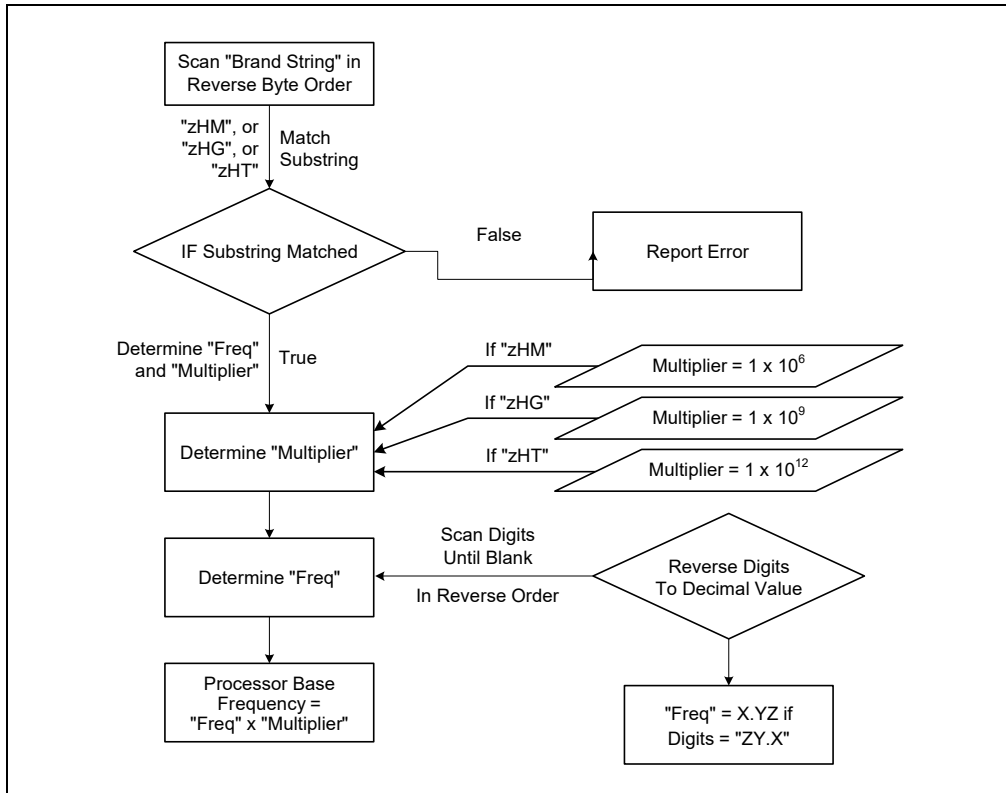


Figure 3-10. Algorithm for Extracting Processor Frequency

The Processor Brand Index Method

The brand index method (introduced with Pentium® III Xeon® processors) provides an entry point into a brand identification table that is maintained in memory by system software and is accessible from system- and user-level code. In this table, each brand index is associate with an ASCII brand identification string that identifies the official Intel family and model number of a processor.

When CPUID executes with EAX set to 1, the processor returns a brand index to the low byte in EBX. Software can then use this index to locate the brand identification string for the processor in the brand identification table. The first entry (brand index 0) in this table is reserved, allowing for backward compatibility with processors that do not support the brand identification feature. Starting with processor signature family ID = 0FH, model = 03H, brand index method is no longer supported. Use brand string method instead.

Table 3-23 shows brand indices that have identification strings associated with them.

Table 3-23. Mapping of Brand Indices; and Intel 64 and IA-32 Processor Brand Strings

Brand Index	Brand String
00H	This processor does not support the brand identification feature
01H	Intel(R) Celeron(R) processor ¹
02H	Intel(R) Pentium(R) III processor ¹
03H	Intel(R) Pentium(R) III Xeon(R) processor; If processor signature = 000006B1h, then Intel(R) Celeron(R) processor
04H	Intel(R) Pentium(R) III processor
06H	Mobile Intel(R) Pentium(R) III processor-M
07H	Mobile Intel(R) Celeron(R) processor ¹
08H	Intel(R) Pentium(R) 4 processor
09H	Intel(R) Pentium(R) 4 processor
0AH	Intel(R) Celeron(R) processor ¹
0BH	Intel(R) Xeon(R) processor; If processor signature = 00000F13h, then Intel(R) Xeon(R) processor MP
0CH	Intel(R) Xeon(R) processor MP
0EH	Mobile Intel(R) Pentium(R) 4 processor-M; If processor signature = 00000F13h, then Intel(R) Xeon(R) processor
0FH	Mobile Intel(R) Celeron(R) processor ¹
11H	Mobile Genuine Intel(R) processor
12H	Intel(R) Celeron(R) M processor
13H	Mobile Intel(R) Celeron(R) processor ¹
14H	Intel(R) Celeron(R) processor
15H	Mobile Genuine Intel(R) processor
16H	Intel(R) Pentium(R) M processor
17H	Mobile Intel(R) Celeron(R) processor ¹
18H - 0FFH	RESERVED

NOTES:

1. Indicates versions of these processors that were introduced after the Pentium III

IA-32 Architecture Compatibility

CPUID is not supported in early models of the Intel486 processor or in any IA-32 processor earlier than the Intel486 processor.

Operation

IA32_BIOS_SIGN_ID MSR := Update with installed microcode revision number;

CASE (EAX) OF

EAX = 0:

EAX := Highest basic function input value understood by CPUID;

EBX := Vendor identification string;

EDX := Vendor identification string;

ECX := Vendor identification string;

BREAK;

EAX = 1H:

EAX[3:0] := Stepping ID;

EAX[7:4] := Model;

EAX[11:8] := Family;

EAX[13:12] := Processor type;
 EAX[15:14] := Reserved;
 EAX[19:16] := Extended Model;
 EAX[27:20] := Extended Family;
 EAX[31:28] := Reserved;
 EBX[7:0] := Brand Index; (* Reserved if the value is zero. *)
 EBX[15:8] := CLFLUSH Line Size;
 EBX[16:23] := Reserved; (* Number of threads enabled = 2 if MT enable fuse set. *)
 EBX[24:31] := Initial APIC ID;
 ECX := Feature flags; (* See Figure 3-7. *)
 EDX := Feature flags; (* See Figure 3-8. *)

BREAK;
 EAX = 2H:
 EAX := Cache and TLB information;
 EBX := Cache and TLB information;
 ECX := Cache and TLB information;
 EDX := Cache and TLB information;

BREAK;
 EAX = 3H:
 EAX := Reserved;
 EBX := Reserved;
 ECX := ProcessorSerialNumber[31:0];
 (* Pentium III processors only, otherwise reserved. *)
 EDX := ProcessorSerialNumber[63:32];
 (* Pentium III processors only, otherwise reserved. *)

BREAK;
 EAX = 4H:
 EAX := Deterministic Cache Parameters Leaf; (* See Table 3-17. *)
 EBX := Deterministic Cache Parameters Leaf;
 ECX := Deterministic Cache Parameters Leaf;
 EDX := Deterministic Cache Parameters Leaf;

BREAK;
 EAX = 5H:
 EAX := MONITOR/MWAIT Leaf; (* See Table 3-17. *)
 EBX := MONITOR/MWAIT Leaf;
 ECX := MONITOR/MWAIT Leaf;
 EDX := MONITOR/MWAIT Leaf;

BREAK;
 EAX = 6H:
 EAX := Thermal and Power Management Leaf; (* See Table 3-17. *)
 EBX := Thermal and Power Management Leaf;
 ECX := Thermal and Power Management Leaf;
 EDX := Thermal and Power Management Leaf;

BREAK;
 EAX = 7H:
 EAX := Structured Extended Feature Flags Enumeration Leaf; (* See Table 3-17. *)
 EBX := Structured Extended Feature Flags Enumeration Leaf;
 ECX := Structured Extended Feature Flags Enumeration Leaf;
 EDX := Structured Extended Feature Flags Enumeration Leaf;

BREAK;
 EAX = 8H:
 EAX := Reserved = 0;
 EBX := Reserved = 0;
 ECX := Reserved = 0;

```

    EDX := Reserved = 0;
BREAK;
EAX = 9H:
    EAX := Direct Cache Access Information Leaf; (* See Table 3-17. *)
    EBX := Direct Cache Access Information Leaf;
    ECX := Direct Cache Access Information Leaf;
    EDX := Direct Cache Access Information Leaf;
BREAK;
EAX = AH:
    EAX := Architectural Performance Monitoring Leaf; (* See Table 3-17. *)
    EBX := Architectural Performance Monitoring Leaf;
    ECX := Architectural Performance Monitoring Leaf;
    EDX := Architectural Performance Monitoring Leaf;
BREAK;
EAX = BH:
    EAX := Extended Topology Enumeration Leaf; (* See Table 3-17. *)
    EBX := Extended Topology Enumeration Leaf;
    ECX := Extended Topology Enumeration Leaf;
    EDX := Extended Topology Enumeration Leaf;
BREAK;
EAX = CH:
    EAX := Reserved = 0;
    EBX := Reserved = 0;
    ECX := Reserved = 0;
    EDX := Reserved = 0;
BREAK;
EAX = DH:
    EAX := Processor Extended State Enumeration Leaf; (* See Table 3-17. *)
    EBX := Processor Extended State Enumeration Leaf;
    ECX := Processor Extended State Enumeration Leaf;
    EDX := Processor Extended State Enumeration Leaf;
BREAK;
EAX = EH:
    EAX := Reserved = 0;
    EBX := Reserved = 0;
    ECX := Reserved = 0;
    EDX := Reserved = 0;
BREAK;
EAX = FH:
    EAX := Intel Resource Director Technology Monitoring Enumeration Leaf; (* See Table 3-17. *)
    EBX := Intel Resource Director Technology Monitoring Enumeration Leaf;
    ECX := Intel Resource Director Technology Monitoring Enumeration Leaf;
    EDX := Intel Resource Director Technology Monitoring Enumeration Leaf;
BREAK;
EAX = 10H:
    EAX := Intel Resource Director Technology Allocation Enumeration Leaf; (* See Table 3-17. *)
    EBX := Intel Resource Director Technology Allocation Enumeration Leaf;
    ECX := Intel Resource Director Technology Allocation Enumeration Leaf;
    EDX := Intel Resource Director Technology Allocation Enumeration Leaf;
BREAK;
EAX = 12H:
    EAX := Intel SGX Enumeration Leaf; (* See Table 3-17. *)
    EBX := Intel SGX Enumeration Leaf;
    ECX := Intel SGX Enumeration Leaf;

```

EDX := Intel SGX Enumeration Leaf;

BREAK;

EAX = 14H:

EAX := Intel Processor Trace Enumeration Leaf; (* See Table 3-17. *)

EBX := Intel Processor Trace Enumeration Leaf;

ECX := Intel Processor Trace Enumeration Leaf;

EDX := Intel Processor Trace Enumeration Leaf;

BREAK;

EAX = 15H:

EAX := Time Stamp Counter and Nominal Core Crystal Clock Information Leaf; (* See Table 3-17. *)

EBX := Time Stamp Counter and Nominal Core Crystal Clock Information Leaf;

ECX := Time Stamp Counter and Nominal Core Crystal Clock Information Leaf;

EDX := Time Stamp Counter and Nominal Core Crystal Clock Information Leaf;

BREAK;

EAX = 16H:

EAX := Processor Frequency Information Enumeration Leaf; (* See Table 3-17. *)

EBX := Processor Frequency Information Enumeration Leaf;

ECX := Processor Frequency Information Enumeration Leaf;

EDX := Processor Frequency Information Enumeration Leaf;

BREAK;

EAX = 17H:

EAX := System-On-Chip Vendor Attribute Enumeration Leaf; (* See Table 3-17. *)

EBX := System-On-Chip Vendor Attribute Enumeration Leaf;

ECX := System-On-Chip Vendor Attribute Enumeration Leaf;

EDX := System-On-Chip Vendor Attribute Enumeration Leaf;

BREAK;

EAX = 18H:

EAX := Deterministic Address Translation Parameters Enumeration Leaf; (* See Table 3-17. *)

EBX := Deterministic Address Translation Parameters Enumeration Leaf;

ECX := Deterministic Address Translation Parameters Enumeration Leaf;

EDX := Deterministic Address Translation Parameters Enumeration Leaf;

BREAK;

EAX = 19H:

EAX := Key Locker Enumeration Leaf; (* See Table 3-17. *)

EBX := Key Locker Enumeration Leaf;

ECX := Key Locker Enumeration Leaf;

EDX := Key Locker Enumeration Leaf;

BREAK;

EAX = 1AH:

EAX := Native Model ID Enumeration Leaf; (* See Table 3-17. *)

EBX := Native Model ID Enumeration Leaf;

ECX := Native Model ID Enumeration Leaf;

EDX := Native Model ID Enumeration Leaf;

BREAK;

EAX = 1BH:

EAX := PCONFIG Information Enumeration Leaf; (* See "INPUT EAX = 1BH: Returns PCONFIG Information" on page 3-260. *)

EBX := PCONFIG Information Enumeration Leaf;

ECX := PCONFIG Information Enumeration Leaf;

EDX := PCONFIG Information Enumeration Leaf;

BREAK;

EAX = 1CH:

EAX := Last Branch Record Information Enumeration Leaf; (* See Table 3-17. *)

EBX := Last Branch Record Information Enumeration Leaf;

ECX := Last Branch Record Information Enumeration Leaf;

EDX := Last Branch Record Information Enumeration Leaf;
BREAK;
EAX = 1DH:
EAX := Tile Information Enumeration Leaf; (* See Table 3-17. *)
EBX := Tile Information Enumeration Leaf;
ECX := Tile Information Enumeration Leaf;
EDX := Tile Information Enumeration Leaf;
BREAK;
EAX = 1EH:
EAX := TMUL Information Enumeration Leaf; (* See Table 3-17. *)
EBX := TMUL Information Enumeration Leaf;
ECX := TMUL Information Enumeration Leaf;
EDX := TMUL Information Enumeration Leaf;
BREAK;
EAX = 1FH:
EAX := V2 Extended Topology Enumeration Leaf; (* See Table 3-17. *)
EBX := V2 Extended Topology Enumeration Leaf;
ECX := V2 Extended Topology Enumeration Leaf;
EDX := V2 Extended Topology Enumeration Leaf;
BREAK;
EAX = 20H:
EAX := Processor History Reset Sub-leaf; (* See Table 3-17. *)
EBX := Processor History Reset Sub-leaf;
ECX := Processor History Reset Sub-leaf;
EDX := Processor History Reset Sub-leaf;
BREAK;
EAX = 23H:
EAX := Architectural Performance Monitoring Extended Leaf; (* See Table 3-17. *)
EBX := Architectural Performance Monitoring Extended Leaf;
ECX := Architectural Performance Monitoring Extended Leaf;
EDX := Architectural Performance Monitoring Extended Leaf;
BREAK;
EAX = 24H:
EAX := Intel AVX10 Converged Vector ISA Leaf; (* See Table 3-17. *)
EBX := Intel AVX10 Converged Vector ISA Leaf;
ECX := Intel AVX10 Converged Vector ISA Leaf;
EDX := Intel AVX10 Converged Vector ISA Leaf;
BREAK;
EAX = 80000000H:
EAX := Highest extended function input value understood by CPUID;
EBX := Reserved;
ECX := Reserved;
EDX := Reserved;
BREAK;
EAX = 80000001H:
EAX := Reserved;
EBX := Reserved;
ECX := Extended Feature Bits (* See Table 3-17.*);
EDX := Extended Feature Bits (* See Table 3-17. *);
BREAK;
EAX = 80000002H:
EAX := Processor Brand String;
EBX := Processor Brand String, continued;
ECX := Processor Brand String, continued;

```

    EDX := Processor Brand String, continued;
BREAK;
EAX = 80000003H:
    EAX := Processor Brand String, continued;
    EBX := Processor Brand String, continued;
    ECX := Processor Brand String, continued;
    EDX := Processor Brand String, continued;
BREAK;
EAX = 80000004H:
    EAX := Processor Brand String, continued;
    EBX := Processor Brand String, continued;
    ECX := Processor Brand String, continued;
    EDX := Processor Brand String, continued;
BREAK;
EAX = 80000005H:
    EAX := Reserved = 0;
    EBX := Reserved = 0;
    ECX := Reserved = 0;
    EDX := Reserved = 0;
BREAK;
EAX = 80000006H:
    EAX := Reserved = 0;
    EBX := Reserved = 0;
    ECX := Cache information;
    EDX := Reserved = 0;
BREAK;
EAX = 80000007H:
    EAX := Reserved = 0;
    EBX := Reserved = 0;
    ECX := Reserved = 0;
    EDX := Reserved = Misc Feature Flags;
BREAK;
EAX = 80000008H:
    EAX := Address Size Information;
    EBX := Misc Feature Flags;
    ECX := Reserved = 0;
    EDX := Reserved = 0;
BREAK;
EAX >= 40000000H and EAX <= 4FFFFFFFH:
DEFAULT: (* EAX = Value outside of recognized range for CPUID. *)
    (* If the highest basic information leaf data depend on ECX input value, ECX is honored. *)
    EAX := Reserved; (* Information returned for highest basic information leaf. *)
    EBX := Reserved; (* Information returned for highest basic information leaf. *)
    ECX := Reserved; (* Information returned for highest basic information leaf. *)
    EDX := Reserved; (* Information returned for highest basic information leaf. *)
BREAK;
ESAC;

```

Flags Affected

None.

Exceptions (All Operating Modes)

#UD

If the LOCK prefix is used.

In earlier IA-32 processors that do not support the CPUID instruction, execution of the instruction results in an invalid opcode (#UD) exception being generated.