Appendix E Obtaining Processor Information Via the CPUID Instruction

This appendix specifies the information that software can obtain about the processor on which it is running by executing the CPUID instruction. The information in this appendix supersedes the contents of the *CPUID Specification*, order #25481, which is now obsolete.

The CPUID instruction is described on page 171. This appendix does not replace the CPUID instruction reference information presented there.

The CPUID instruction behaves much like a function call. Parameters are passed to the instruction via registers and on execution the instruction loads specific registers with return values. These return values can be interpreted by software based on the field definitions and their assigned meanings.

The first input parameter is the *function number* which is passed to the instruction via the EAX register. Some functions also accept a second input parameter passed via the ECX register. Values are returned via the EAX, EBX, ECX, and EDX registers. Software should not assume that any values written to these registers prior to the execution of CPUID instruction will be retained after the instruction executes (even those that are marked reserved).

The description of each return value breaks the value down into one or more named *fields* which represent a bit position or contiguous range of bits. All bit positions that are not defined as fields are reserved. The value of bits within reserved ranges cannot be relied upon to be zero. Software must mask off all reserved bits in the return value prior to making any value comparisons of represented information.

This appendix applies to all AMD processors with a family designation of 0Fh or greater.

E.1 Special Notational Conventions

The following special notation conventions are used in this appendix:

- The notation (standard throughout this APM) for representing the function number, optional input parameter, and the information returned is as follows:

  CPUID Fn*XXXX_XXXX_RRR*[*FieldName*]_x*YYY*.

  Where:

  - *XXXX_XXXX* is the function number represented in hexadecimal (passed to the instruction in EAX).
  - *RRR* is one of {EDX, ECX, EBX, EAX} and represents a register holding a return value.
  - *YYY* represents the optional input parameter passed in the ECX register expressed as a hexadecimal number. If this parameter is not used, the characters represented by _x*YYY* are omitted from the notation.

  - *FieldName* identifies a specific named element of processor information represented by a specific bit range (1 or more bits wide) within the *RRR* register.
- The notation CPUID Fn*XXXX_XXXX_RRR* is used when referring to one of the registers that holds information returned by the instruction.
- The notation CPUID Fn*XXXX_XXXX* or Fn*XXXX_XXXX* is used to refer to a specific function number.
- Most one-bit fields indicate support or non-support of a specific processor feature. By convention, (unless otherwise noted) a value of 1 means that the feature is supported by the processor and a value of 0 means that the feature is not supported by the processor.

E.2 Standard and Extended Function Numbers

The CPUID instruction supports two sets or ranges of function numbers: standard and extended.

- The smallest function number of the standard function range is Fn0000_0000. The largest function number of the standard function range, for a particular implementation, is returned in CPUID Fn0000_0000_EAX.
- The smallest function number of the extended function range is Fn8000_0000. The largest function number of the extended function range, for a particular implementation, is returned in CPUID Fn8000_0000_EAX.

E.3 Standard Feature Function Numbers

This section describes each of the defined CPUID functions in the standard range.

E.3.1 Function 0h—Maximum Standard Function Number and Vendor String

This function number provides information about the maximum standard function number supported on this processor and a string that identifies the vendor of the product.

CPUID Fn0000_0000_EAX Largest Standard Function Number

The value returned in EAX provides the largest standard function number supported by this processor.

Bits	Field Name	Description
31:0	LFuncStd	Largest standard function. The largest CPUID standard function input value supported by the processor implementation.

CPUID Fn0000_0000_E[D,C,B]X Processor Vendor

The values returned in EBX, EDX, and ECX together provide a 12-character string identifying the vendor of this processor. Each register supplies 4 characters. The leftmost character of each substring

is stored in the least significant bit position in the register. The string is the concatenation of the contents of EBX, EDX, and ECX in left to right order. No null terminator is included in the string.

CPUID Fn8000_0000_E[D,C,B]X return the same values as this function.

Bits	Field Name	Description
31:0	Vendor	Four characters of the 12-byte character string (encoded in ASCII) “AuthenticAMD”. See Table E-1 below.

Table E-1. CPUID Fn0000_0000_E[D,C,B]X values

Register	Value	Description
CPUID Fn0000_0000_EBX	6874_7541h	The ASCII characters “h t u A”.
CPUID Fn0000_0000_ECX	444D_4163h	The ASCII characters “D M A c”.
CPUID Fn0000_0000_EDX	6974_6E65h	The ASCII characters “i t n e”.

E.3.2 Function 1h—Processor and Processor Feature Identifiers

This function number identifies the processor family, model, and stepping and provides feature support information.

CPUID Fn0000_0001_EAX Family, Model, Stepping Identifiers

The value returned in EAX provides the family, model, and stepping identifiers. Three values are used by software to identify a processor: Family, Model, and Stepping.

Bits	Field Name	Description
31:28	—	Reserved
27:20	ExtFamily	Processor extended family. See above for definition of Family[7:0].
19:16	ExtModel	Processor extended model. See above for definition of Model[7:0].
15:12	—	Reserved
11:8	BaseFamily	Base processor family. See above for definition of Family[7:0].
7:4	BaseModel	Base processor model. See above for definition of Model[7:0].
3:0	Stepping	Processor stepping. Processor stepping (revision) for a specific model.

The processor *Family* identifies one or more processors as belonging to a group that possesses some common definition for software or hardware purposes. The *Model* specifies one instance of a processor family. The *Stepping* identifies a particular version of a specific model. Therefore, Family, Model and Stepping, when taken together, form a unique identification or signature for a processor.

The **Family** is an 8-bit value and is defined as: **Family[7:0] =** ({0000b,BaseFamily[3:0]} + ExtFamily[7:0]). For example, if BaseFamily[3:0] = Fh and ExtFamily[7:0] = 01h, then Family[7:0] = 10h. If BaseFamily[3:0] is less than Fh, then ExtFamily is reserved and Family is equal to BaseFamily[3:0].

Model is an 8-bit value and is defined as: **Model[7:0]** = {ExtModel[3:0],BaseModel[3:0]}. For example, if ExtModel[3:0] = Eh and BaseModel[3:0] = 8h, then Model[7:0] = E8h. If BaseFamily[3:0] is less than 0Fh, then ExtModel is reserved and Model is equal to BaseModel[3:0].

The value returned by CPUID Fn8000_0001_EAX is equivalent to CPUID Fn0000_0001_EAX.

CPUID Fn0000_0001_EBX LocalApicId, LogicalProcessorCount, CLFlush

The value returned in EBX provides miscellaneous information regarding the processor brand, the number of logical threads per processor socket, the CLFLUSH instruction, and APIC.

Bits	Field Name	Description
31:24	LocalApicId	Initial local APIC physical ID. The 8-bit value assigned to the local APIC physical ID register at power-up. Some of the bits of LocalApicId represent the core within a processor and other bits represent the processor ID. See the APIC20 “APIC ID” register in the processor BKDG or PPR for details.
23:16	LogicalProcessor Count	Logical processor count. If CPUID Fn0000_0001_EDX[HTT] = 1 then LogicalProcessorCount is the number of logic processors per package. If CPUID Fn0000_0001_EDX[HTT] = 0 then LogicalProcessorCount is reserved. See E.5.1 [Legacy Method].
15:8	CLFlush	CLFLUSH size. Specifies the size of a cache line in quadwords flushed by the CLFLUSH instruction. See “CLFLUSH” on page 150.
7:0	8BitBrandId	8-bit brand ID. This field, in conjunction with CPUID Fn8000_0001_EBX[BrandId], is used by the system firmware to generate the processor name string. See the appropriate processor revision guide for how to program the processor name string.

CPUID Fn0000_0001_ECX Feature Identifiers

The value returned in ECX contains the following miscellaneous feature identifiers:

Bits	Field Name	Description
31	—	RAZ. Reserved for use by hypervisor to indicate guest status.
30	RDRAND	RDRAND instruction support.
29	F16C	Half-precision convert instruction support. See “Half-Precision Floating-Point Conversion” on page 193 of APM Volume 1 and listings for individual F16C instructions in APM Volume 5.
28	AVX	AVX instruction support. See APM Volume 4.
27	OSXSAVE	XSAVE (and related) instructions are enabled. See “OSXSAVE” in APM Volume 2.
26	XSAVE	XSAVE (and related) instructions are supported by hardware. See “XSAVE/XRSTOR Instructions” in APM Volume 2.
25	AES	AES instruction support. See “AES Instructions” in APM Volume 4.

Bits	Field Name	Description
24	—	Reserved
23	POPCNT	POPCNT instruction. See “POPCNT” on page 288.
22		MOVBE: MOVBE instruction support.
21	x2APIC	x2APIC support. See “x2APIC Mode” in APM Volume 2.
20	SSE42	SSE4.2 instruction support. “Determining Media and x87 Feature Support” in APM Volume 2 and individual SSE4.2 instruction listings in APM Volume 4.
19	SSE41	SSE4.1 instruction support. See individual instruction listings in APM Volume 4.
18:14	—	Reserved
13	CMPXCHG16B	CMPXCHG16B instruction support. See “CMPXCHG16B” in APM Volume 3.
12	FMA	FMA instruction support.
11:10	—	Reserved
9	SSSE3	Supplemental SSE3 instruction support.
8:4	—	Reserved
3	MONITOR	MONITOR/MWAIT instructions. See “MONITOR” and “MWAIT” in APM Volume 3.
2	—	Reserved
1	PCLMULQDQ	PCLMULQDQ instruction support. See instruction reference page for the PCLMULQDQ / VPCLMULQDQ instruction in APM Volume 4.
0	SSE3	SSE3 instruction support. See Appendix D “Instruction Subsets and CPUID Feature Sets” in APM Volume 3 for the list of instructions covered by the SSE3 feature bit. See APM Volume 4 for the definition of the SSE3 instructions.

CPUID Fn0000_0001_EDX Feature Identifiers

The value returned in EDX contains the following miscellaneous feature identifiers:

Bits	Field Name	Description
31:29	—	Reserved
28	HTT	Hyper-threading technology. Indicates either that there is more than one thread per core or more than one core per compute unit. See “Legacy Method” on page 671.
27	—	Reserved
26	SSE2	SSE2 instruction support. See Appendix D “CPUID Feature Sets” in APM Volume 3.
25	SSE	SSE instruction support. See Appendix D “CPUID Feature Sets” in APM Volume 3 appendix and “64-Bit Media Programming” in APM Volume 1.
24	FXSR	FXSAVE and FXRSTOR instructions. See “FXSAVE” and “FXRSTOR” in APM Volume 5.
23	MMX	MMX™ instructions. See Appendix D “CPUID Feature Sets” in APM Volume 3 and “128-Bit Media and Scientific Programming” in APM Volume 1.
22:20	—	Reserved
19	CLFSH	CLFLUSH instruction support. See “CLFLUSH” in APM Volume 3.

Bits	Field Name	Description
18	—	Reserved
17	PSE36	Page-size extensions. The PDE[20:13] supplies physical address [39:32]. See “Page Translation and Protection” in APM Volume 2.
16	PAT	Page attribute table. See “Page-Attribute Table Mechanism” in APM Volume 2.
15	CMOV	Conditional move instructions. See “CMOV”, “FCMOV” in APM Volume 3.
14	MCA	Machine check architecture. See “Machine Check Mechanism” in APM Volume 2.
13	PGE	Page global extension. See “Page Translation and Protection” in APM Volume 2.
12	MTRR	Memory-type range registers. See “Page Translation and Protection” in APM Volume 2.
11	SysEnterSysExit	SYSENTER and SYSEXIT instructions. See “SYSENTER”, “SYSEXIT” in APM Volume 3.
10	—	Reserved
9	APIC	Advanced programmable interrupt controller. Indicates APIC exists and is enabled. See “Exceptions and Interrupts” in APM Volume 2.
8	CMPXCHG8B	CMPXCHG8B instruction. See “CMPXCHG8B” in APM Volume 3.
7	MCE	Machine check exception. See “Machine Check Mechanism” in APM Volume 2.
6	PAE	Physical-address extensions. Indicates support for physical addresses ³ 32b. Number of physical address bits above 32b is implementation specific. See “Page Translation and Protection” in APM Volume 2.
5	MSR	AMD model-specific registers. Indicates support for AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions. See “Model Specific Registers” in APM Volume 2.
4	TSC	Time stamp counter. RDTSC and RDTSCP instruction support. See “Debug and Performance Resources” in APM Volume 2.
3	PSE	Page-size extensions. See “Page Translation and Protection” in APM Volume 2.
2	DE	Debugging extensions. See “Debug and Performance Resources” in APM Volume 2.
1	VME	Virtual-mode enhancements. CR4.VME, CR4.PVI, software interrupt indirection, expansion of the TSS with the software, indirection bitmap, EFLAGS.VIF, EFLAGS.VIP. See “System Resources” in APM Volume 2.
0	FPU	x87 floating point unit on-chip. See “x87 Floating Point Programming” in APM Volume 1.

E.3.3 Functions 2h–4h—Reserved

CPUID Fn0000_000[4:2] Reserved

These function numbers are reserved.

E.3.4 Function 5h—Monitor and MWait Features

This function provides feature identifiers for the MONITOR and MWAIT instructions. For more information see the description of the MONITOR instruction on page 424 and the MWAIT instruction on page 430.

CPUID Fn0000_0005_EAX Monitor/MWait

The value returned in EAX provides the following information:

Bits	Field Name	Description
31:16	—	Reserved
15:0	MonLineSizeMin	Smallest monitor-line size in bytes.

CPUID Fn0000_0005_EBX Monitor/MWait

The value returned in EBX provides the following information:

Bits	Field Name	Description
31:16	—	Reserved
15:0	MonLineSizeMax	Largest monitor-line size in bytes.

CPUID Fn0000_0005_ECX Monitor/MWait

The value returned in ECX provides the following information:

Bits	Field Name	Description
31:2	—	Reserved
1	IBE	Interrupt break-event. Indicates MWAIT can use ECX bit 0 to allow interrupts to cause an exit from the monitor event pending state, even if EFLAGS.IF=0.
0	EMX	Enumerate MONITOR/MWAIT extensions: Indicates enumeration MONITOR/MWAIT extensions are supported.

CPUID Fn0000_0005_EDX Monitor/MWait

The value returned in EDX is undefined and is reserved.

E.3.5 Function 6h—Power Management Related Features

This function provides information about the local APIC timer timebase and the effective frequency interface for the processor.

CPUID Fn0000_0006_EAX Local APIC Timer Invariance

The value returned in EAX is undefined and is reserved.

Bits	Field Name	Description
31:3	—	Reserved
2	ARAT	If set, indicates that the timebase for the local APIC timer is not affected by processor p-state.
1:0	—	Reserved

CPUID Fn0000_0006_EBX Reserved

The value returned in EBX is undefined and is reserved.

CPUID Fn0000_0006_ECX Effective Processor Frequency Interface

The value returned in ECX indicates support of the processor effective frequency interface. For more information on this feature, see “Determining Processor Effective Frequency” in APM Volume 2.

Bits	Field Name	Description
31:1	—	Reserved
0	EffFreq	Effective frequency interface support. If set, indicates presence of MSR0000_00E7 (MPERF) and MSR0000_00E8 (APERF).

CPUID Fn0000_0006_EDX Reserved

The value returned in EDX is undefined and is reserved.

E.3.6 Function 7h—Structured Extended Feature Identifiers

Subfunction 0 of Fn0000_0007

CPUID Fn0000_0007_EAX_x0 Structured Extended Feature Identifiers (ECX=0)

Bits	Field Name	Description
31:0	MaxSubFn	Returns the number of subfunctions supported.

CPUID Fn0000_0007_EBX_x0 Structured Extended Feature Identifiers (ECX=0)

Bits	Field Name	Description
31	AVX512VL	AVX512 instructions are extended to 128 and 256 bits
30	AVX512BW	AVX512 Byte/Word Packed Integer instructions

Bits	Field Name	Description
29	SHA	Secure Hash Algorithm instruction extension.
28	AVX512CD	AVX512 Conflict Detection for Vectorizing Loops
27:25	—	Reserved
24	CLWB	CLWB instruction support.
23	CLFLUSHOPT	CLFLUSHOPT instruction support.
22	—	Reserved
21	AVX512_IFMA	AVX512 Integer Fused Multiply-Add instructions support.
20	SMAP	Supervisor mode access prevention.
19	ADX	ADCX, ADOX instruction support.
18	RDSEED	RDSEED instruction support.
17	AVX512DQ	AVX512 Doubleword/Quadword Packed Integer instructions
16	AVX512F	AVX512 Foundation
15	PQE	Platform QOS Enforcement support. See *AMD64 Technology Platform Quality of Service Extensions* in APM Volume 2.
14:13	—	Reserved
12	PQM	Platform QOS Monitoring support. See *AMD64 Technology Platform Quality of Service Extensions* in APM Volume 2.
11	—	Reserved
10	INVPCID	INVPCID instruction support.
9	ERMS	Enhanced REP MOVSB/STOSB support.
8	BMI2	Bit manipulation group 2 instruction support.
7	SMEP	Supervisor mode execution prevention.
6	—	Reserved
5	AVX2	AVX2 instruction subset support.
4	—	Reserved
3	BMI1	Bit manipulation group 1 instruction support.
2	—	Reserved
1	TSCADJUST	TSC Adjust MSR (3Bh) support.
0	FSGSBASE	FS and GS base read/write instruction support.

CPUID Fn0000_0007_ECX_x0 Structured Extended Feature Identifiers (ECX=0)

Bits	Field Name	Description
31:29	—	Reserved
28	MOVDIR64B	MOVDIR64B instruction support.
27	MOVDIRI	MOVDIRI instruction support.
26:25	—	Reserved
24	BUSLOCKTRAP	Bus Lock Trap (#DB) support.
23	—	Reserved

Bits	Field Name	Description
22	RDPID	RDPID instruction and TSC_AUX MSR support.
21:17	—	Reserved
16	LA57	5-Level paging support.
15	—	Reserved
14	AVX512_VPOPCNTDQ	AVX-512 VPOPCNTD/Q instruction support.
13	—	Reserved
12	AVX512_BITALG	AVX512 bit algorithm instructions VPSHUFBITQMB and VPOPCNTB/W support.
11	AVX512_VNNI	AVX512 vector neural network instructions support.
10	VPCMULQDQ	VPCLMULQDQ 256-bit instruction support.
9	VAES	VAES 256-bit instructions support.
8	GFNI	Galois Field New instructions support.
7	CET_SS	Shadow Stacks supported.
6	AVX512_VBMI2	AVX512 vector byte permutation instruction 2 support.
5	—	Reserved
4	OSPKE	OS has enabled Memory Protection Keys and use of the RDPKRU/WRPKRU instructions by setting CR4.PKE=1.
3	PKU	Memory Protection Keys supported.
2	UMIP	User mode instruction prevention support.
1	AVX512_VBMI	AVX512 vector byte permutation instructions support.
0	—	Reserved

CPUID Fn0000_0007_EDX_x0 Structured Extended Feature Identifiers (ECX=0)

Bits	Field Name	Description
31:0	—	Reserved

Subfunction 1of Fn0000_0007

CPUID Fn0000_0007_EAX_x1 Structured Extended Feature Identifiers (ECX=1)

Bits	Field Name	Description
31:6	—	Reserved
5	AVX512_BF16	AVX512 BFloat16 instructions
4	AVX_VNNI	AVX Neural Network instructions
3:0	—	Reserved

CPUID Fn0000_0007_EBX_x1 Structured Extended Feature Identifiers (ECX=1)

Bits	Field Name	Description
31:0	—	Reserved

CPUID Fn0000_0007_ECX_x1 Structured Extended Feature Identifiers (ECX=1)

Bits	Field Name	Description
31:0	—	Reserved

CPUID Fn0000_0007_EDX_x1 Structured Extended Feature Identifiers (ECX=1)

Bits	Field Name	Description
31:0	—	Reserved

E.3.7 Functions 8h–Ah—Reserved

E.3.8 Function Bh — Extended Topology Enumeration

CPUID Fn0000_000B enumerates each level in the processor's topological hierarchy. The hierarchy level is specified by the input value passed in the ECX register.

If this function is executed with an unimplemented level (passed in ECX), the instruction returns all zeros in the EAX register.

Subfunction 0 of Fn0000_000B - Thread Level

Subfunction 0 provides information about the thread-level topology.

CPUID Fn0000_000B_EAX_x0 Extended Topology Enumeration (ECX=0)

Bits	Field Name	Description
31:5	—	Reserved
4:0	ThreadMaskWidth	Number of bits to shift x2APIC_ID right to get to the topology ID of the next level.

CPUID Fn0000_000B_EBX_x0 Extended Topology Enumeration (ECX=0)

Bits	Field Name	Description
31:16	—	Reserved
15:0	NumLogProc	Number of logical processors in a core.

CPUID Fn0000_000B_ECX_x0 Extended Topology Enumeration (ECX=0)

Bits	Field Name	Description
31:16	—	Reserved
15:8	HierarchyLevel	1 (thread level)
7:0	InputEcx	0

CPUID Fn0000_000B_EDX_x0 Extended Topology Enumeration (ECX=0)

Bits	Field Name	Description
31:0	x2APIC_ID	32-bit Extended APIC_ID.

Subfunction 1 of Fn0000_000B - Core Level

Subfunction 1 provides information about the core-level topology.

CPUID Fn0000_000B_EAX_x1 Extended Topology Enumeration (ECX=1)

Bits	Field Name	Description
31:5	—	Reserved
4:0	CoreMaskWidth	Number of bits to shift x2APIC_ID right to get to the topology ID of the next level.

CPUID Fn0000_000B_EBX_x1 Extended Topology Enumeration (ECX=1)

Bits	Field Name	Description
31:16	—	Reserved
15:0	NumLogCores	Number of logical cores in socket.

CPUID Fn0000_000B_ECX_x1 Extended Topology Enumeration (ECX=1)

Bits	Field Name	Description
31:16	—	Reserved
15:8	HierarchyLevel	2
7:0	InputEcx	1

CPUID Fn0000_000B_EDX_x1 Extended Topology Enumeration (ECX=1)

Bits	Field Name	Description
31:0	x2APIC_ID	32-bit Extended APIC_ID.

E.3.9 Function Ch—Reserved

E.3.10 Function Dh—Processor Extended State Enumeration

The XSAVE / XRSTOR instructions are used to save and restore x87/MMX FPU and SSE processor state. These instructions allow processor state associated with specific architected features to be selectively saved and restored. This function provides information about extended state support and save area size requirements.

The function has a number of sub functions specified by the input value passed to the CPUID instruction in the ECX register. If CPUID Fn0000_000D is executed with an unimplemented subfunction (passed in ECX), the instruction returns all zeros in the EAX, EBX, ECX, and EDX registers.

Subfunction 0 of Fn0000_000D

Subfunction 0 provides information about features within the extended processor state management architecture that are supported by the processor.

CPUID Fn0000_000D_EAX_x0 Processor Extended State Enumeration (ECX=0)

The value returned in EAX provides a bit mask specifying which of the features defined by the extended processor state architecture are supported by the processor.

Bits	Field Name	Description
31:0	XFeatureSupportedMask[31:0]	Reports the valid bit positions for the lower 32 bits of the XFeatureEnabledMask register. If a bit is set, the corresponding feature is supported. See “XSAVE/XRSTOR Instructions” in APM Volume 2.

CPUID Fn0000_000D_EBX_x0 Processor Extended State Enumeration (ECX=0)

The value returned in EBX gives the save area size requirement in bytes based on the features currently enabled in the XFEATURE_ENABLED_MASK (XCR0).

Bits	Field Name	Description
31:0	XFeatureEnabledSizeMax	Size in bytes of XSAVE/XRSTOR area for the currently enabled features in XCR0.

CPUID Fn0000_000D_ECX_x0 Processor Extended State Enumeration (ECX=0)

The value returned in ECX gives the save area size requirement in bytes for all extended state management features supported by the processor (whether enabled or not).

Bits	Field Name	Description
31:0	XFeatureSupportedSizeMax	Size in bytes of XSAVE/XRSTOR area for all features that the logical processor supports. See XFeatureEnabledSizeMax.

CPUID Fn0000_000D_EDX_x0 Processor Extended State Enumeration (ECX=0)

The value returned in EDX provides a bit mask specifying which of the features defined by the extended processor state architecture are supported by the processor.

Bits	Field Name	Description
31:0	XFeatureSupportedMask[63:32]	Reports the valid bit positions for the upper 32 bits of the XFeatureEnabledMask register. If a bit is set, the corresponding feature is supported.

See “XSAVE/XRSTOR Instructions” in APM Volume 2 and reference pages for the individual instructions in APM Volume 4.

Subfunction 1 of Fn0000_000D

Subfunction 1 provides additional information about features within the extended processor state management architecture that are supported by the processor.

CPUID Fn0000_000D_EAX_x1 Processor Extended State Enumeration (ECX=1)

Bits	Field Name	Description
31:4		Reserved
3	XSAVES	XSAVES, XRSTOR, and XSS are supported.
2	XGETBV	XGETBV with ECX = 1 supported.
1	XSAVEC	XSAVEC and compact XRSTOR supported.
0	XSAVEOPT	XSAVEOPT is available.

CPUID Fn0000_000D_EBX_x1 Processor Extended State Enumeration (ECX=1)

The value returned on EBX represents the fixed size of the save area (240h) plus the state size of each enabled extended feature:

EBX = 0240h
+ ((XCR0[AVX] == 1) ? 0000_0100h : 0)
+ ((XCR0[MPK] == 1) ? 0000_0008h : 0)
+ ((XSS[CET_U] == 1) ? 0000_0010h : 0)
+ ((XSS[CET_S] == 1) ? 0000_0018h : 0)

CPUID Fn0000_000D_ECX_x1 Processor Extended State Enumeration (ECX=1)

The value returned on ECX returns a 1 for each bit that is settable in the XSS MSR. The following bits are defined:

Bits	Field Name	Description
31:13	—	Reserved
12	CET_S	CET supervisor.
11	CET_U	CET user state.
10:0	—	Reserved

CPUID Fn0000_000D_EDX_x1 Processor Extended State Enumeration (ECX=1)

The value returned in EDX for subfunction 1 is undefined and reserved.

Subfunction 2 of Fn0000_000D

Subfunction 2 provides information about the size and offset of the 256-bit SSE vector floating point processor unit state save area.

CPUID Fn0000_000D_EAX_x2 Processor Extended State Enumeration (ECX=2)

The value returned in EAX provides information about the size of the 256-bit SSE vector floating point processor unit state save area.

Bits	Field Name	Description
31:0	YmmSaveStateSize	YMM state save size. The state save area size in bytes for The YMM registers.

CPUID Fn0000_000D_EBX_x2 Processor Extended State Enumeration (ECX=2)

The value returned in EBX provides information about the offset of the 256-bit SSE vector floating point processor unit state save area from the base of the extended state (XSAVE/XRSTOR) save area.

Bits	Field Name	Description
31:0	YmmSaveStateOffset	YMM state save offset. The offset in bytes from the base of the extended state save area of the YMM register state save area.

CPUID Fn0000_000D_E[D,C]X_x2 Processor Extended State Enumeration (ECX=2)

The values returned in ECX and EDX for subfunction 2 are undefined and are reserved.

Subfunction 11 of Fn0000_000D

Subfunction 11 provides information about the CET user state save area.

CPUID Fn0000_000D_E[A, B, C, D]X_x11 Processor Extended State Emulation (ECX=11)

The value returned in EAX, EBX, ECX and EDX provides information about the CET user state save area.

Register	Bits	Field Name	Description
EAX	31:0	CetUserSize	CET user state save size in bytes.
EBX	31:0	CetUserOffset	CET user state offset from the base of the extended state save area.
ECX	0	U/S	Set to 1, indicating a supervisor state component.
ECX	31:0	—	Cleared to 0.
EDX	31:0	—	Unused, cleared to 0.

Subfunction 12 of Fn0000_000D

Subfunction 12 provides information about the CET supervisor state save area.

CPUID Fn0000_000D_E[A, B, C, D]X_x12 Processor Extended State Emulation (ECX=12)

The value returned in EAX, EBX, ECX and EDX provides information about the CET supervisor state save area.

Register	Bits	Field Name	Description
EAX	31:0	CetSupervisorSize	CET supervisor state save size in bytes.
EBX	31:0	CetSupervisorOffset	CET supervisor state offset from the base of the extended state save area.
ECX	0	U/S	Set to 1, indicating a supervisor state component.
ECX	31:0	—	Cleared to 0.
EDX	31:0	—	Unused, cleared to 0.

Subfunction 3Eh of Fn0000_000D

Subfunction 3Eh provides information about the size and offset of the Lightweight Profiling (LWP) unit state save area.

CPUID Fn0000_000D_EAX_x3E Processor Extended State Enumeration (ECX=62)

The value returned in EAX provides the size of the Lightweight Profiling (LWP) unit state save area.

Bits	Field Name	Description
31:0	LwpSaveStateSize	LWP state save area size. The size of the save area for LWP state in bytes. See “Lightweight Profiling” in APM Volume 2.

CPUID Fn0000_000D_EBX_x3E Processor Extended State Enumeration (ECX=62)

The value returned in EBX provides the offset of the Lightweight Profiling (LWP) unit state save area from the base of the extended state (XSAVE/XRSTOR) save area.

Bits	Field Name	Description
31:0	LwpSaveStateOffset	LWP state save byte offset. The offset in bytes from the base of the extended state save area of the state save area for LWP. See “Lightweight Profiling” in APM Volume 2.

CPUID Fn0000_000D_E[D,C]X_x3E Processor Extended State Enumeration (ECX=62)

The values returned in ECX and EDX for subfunction 3Eh are undefined and are reserved.

Subfunctions of Fn0000_000D greater than 3Eh

For CPUID Fn0000_000D, if the subfunction (specified by contents of ECX) passed as input to the instruction is greater than 3Eh, the instruction returns zero in the EAX, EBX, ECX, and EDX registers.

E.3.11 Function Eh—Reserved

E.3.12 Function Fh—PQOS Monitoring (PQM)

If PQM is supported (CPUID Fn0000_0007_EBX_x0[PQM] = 1), this CPUID function can be used to obtain PQM feature information. If PQM is not supported this function is reserved. For more information on using PQM, see “Platform Quality of Service” in APM Volume 2.

CPUID Fn0000_000F_x0 PQM Capabilities

Subfunction 0 provides information about PQM capabilities.

Register	Bits	Field Name	Description
EAX	31:0	—	Reserved
EBX	31:0	Max_RMID	Largest RMID supported by the system for any resource.
ECX	31:0	—	Reserved
EDX	31:2	—	Reserved
	1	L3CacheMon	L3 Cache Monitoring Support.
	0	—	Reserved

CPUID Fn0000_000F_x1 L3 Cache Monitoring Capabilities

Subfunction 1 provides information about PQM L3 Cache Monitoring capabilities.

Register	Bits	Field Name	Description
EAX	31:7	—	Reserved
	8	OverflowBit	Indicates that MSR QM_CTR bit 61 is a counter overflow bit.
	7:0	CounterSize	QM_CTR counter width, offset from 24 bits. If 0, the family, model, and stepping should be used to determine the counter size. See “Platform Quality of Service” in APM Volume 2.
EBX	31:0	ScaleFactor	Scale factor for the value obtained from QOS_CTR.
ECX	31:0	Max_RMID	Largest RMID supported by the L3CachMon resource.
EDX	31:3	—	Reserved
	2	L3CacheBWMonEvt1	L3 Cache Bandwidth Monitoring Event 1.
	1	L3CacheBWMonEvt0	L3 Cache Bandwidth Monitoring Event 0.
	0	L3CacheOccMon	L3 Cache Occupancy Monitoring Event.

E.3.13 Function 10h—PQOS Enforcement (PQE)

If PQE is supported (CPUID Fn0000_0007_EBX_x0[PQE] = 1), this CPUID function can be used to obtain PQE feature information. If PQE is not supported this function is reserved. For more information on using PQE, see “Platform Quality of Service” in APM Volume 2.

CPUID Fn0000_0010_x0 PQE Capabilities

Subfunction 0 provides information about PQE supported resources.

Register	Bits	Field Name	Description
EAX	31:0	—	Reserved
EBX	31:0	—	Reserved
ECX	31:0	—	Reserved
EDX	31:2	—	Reserved
	1	L3Alloc	L3 Cache Allocation Enforcement Support.
	0	—	Reserved

CPUID Fn0000_0010_x1 L3 Cache Allocation Enforcement Capabilities

Subfunction 1 provides information about PQE L3 cache allocation enforcement capabilities.

Register	Bits	Field Name	Description
EAX	31:5	—	Reserved
	4:0	CBM_LEN	L3 cache capacity bit mask length minus 1.
EBX	31:0	L3ShareAllocMask	L3 cache allocation sharing mask.
ECX	31:3	—	Reserved
	2	CDP	Code-Data Prioritization support.
	1:0	—	Reserved
EDX	31:16	—	Reserved
	15:0	COS_MAX	Maximum COS supported by L3 cache allocation enforcement.

E.3.14 Functions 4000_0000h-4000_00FFh—Reserved for Hypervisor Use

CPUID Fn4000_00[FF:00] Reserved

These function numbers are reserved for use by the virtual machine monitor.

E.4 Extended Feature Function Numbers

This section describes each of the defined CPUID functions in the extended range.

E.4.1 Function 8000_0000h—Maximum Extended Function Number and Vendor String

This function provides information about the maximum extended function number supported on this processor and a string that identifies the vendor of the product.

CPUID Fn8000_0000_EAX Largest Extended Function Number

The value returned in EAX provides the largest extended function number supported by the processor.

Bits	Field Name	Description
31:0	LFuncExt	Largest extended function. The largest CPUID extended function input value supported by the processor implementation.

CPUID Fn8000_0000_E[D,C,B]X Processor Vendor

The values returned in EBX, ECX, and EDX together provide a 12-character string identifying the vendor of this processor. The output string is the same as the one returned by Fn0000_0000. See CPUID Fn0000_0000_E[D,C,B]X on page 622 for more details.

Bits	Field Name	Description
31:0	Vendor	Four characters of the 12-byte character string (encoded in ASCII) “AuthenticAMD”. See Table E-2 below.

Table E-2. CPUID Fn8000_0000_E[D,C,B]X values

Register	Value	Description
CPUID Fn8000_0000_EBX	6874_7541h	The ASCII characters “h t u A”.
CPUID Fn8000_0000_ECX	444D_4163h	The ASCII characters “D M A c”.
CPUID Fn8000_0000_EDX	6974_6E65h	The ASCII characters “i t n e”.

E.4.2 Function 8000_0001h—Extended Processor and Processor Feature Identifiers

CPUID Fn8000_0001_EAX AMD Family, Model, Stepping

The value returned in EAX provides the family, model, and stepping identifiers. Three values are used by software to identify a processor: Family, Model, and Stepping. The value returned in EAX is the same as the value returned in EAX for Fn0000_0001. See CPUID Fn0000_0001_EAX on page 623 for more details on the field definitions.

Bits	Field Names	Description
31:0	Family, Model, Stepping	See: CPUID Fn0000_0001_EAX.

CPUID Fn8000_0001_EBX BrandId Identifier

The value returned in EBX provides package type and a 16-bit processor name string identifiers.

Bits	Field Name	Description
31:28	PkgType	Package type. If (Family[7:0] >= 10h), this field is valid. If (Family[7:0]<10h), this field is reserved.
27:16	—	Reserved
15:0	BrandId	Brand ID. This field, in conjunction with CPUID Fn0000_0001_EBX[8BitBrandId], is used by system firmware to generate the processor name string. See your processor revision guide for how to program the processor name string.

For processor families 10h and greater, PkgType is described in the *BIOS and Kernel Developer's Guide* for the product.

CPUID Fn8000_0001_ECX Feature Identifiers

This function contains the following miscellaneous feature identifiers:

Bits	Field Name	Description
31	—	Reserved
30	AddrMaskExt	Breakpoint Addressing masking extended to bit 31.
29	MONITORX	Support for MWAITX and MONITORX instructions.
28	PerfCtrExtLLC	Support for L3 performance counter extension.
27	PerfTsc	Performance time-stamp counter. Indicates support for MSRC001_0280 [Performance Time Stamp Counter].
26	DataBkptExt	Data access breakpoint extension. Indicates support for MSRC001_1027 and MSRC001_101[B:9].
25	—	Reserved
24	PerfCtrExtNB	NB performance counter extensions support. Indicates support for MSRC001_024[6,4,2,0] and MSRC001_024[7,5,3,1].
23	PerfCtrExtCore	Processor performance counter extensions support. Indicates support for MSRC001_020[A,8,6,4,2,0] and MSRC001_020[B,9,7,5,3,1].
22	TopologyExtensions	Topology extensions support. Indicates support for CPUID Fn8000_001D_EAX_x[N:0]-CPUID Fn8000_001E_EDX.
21	TBM	Trailing bit manipulation instruction support.
20	—	Reserved

Bits	Field Name	Description
19	—	Reserved
18	—	Reserved
17	TCE	Translation Cache Extension support.
16	FMA4	Four-operand FMA instruction support.
15	LWP	Lightweight profiling support. See “Lightweight Profiling” in APM Volume 2 and reference pages for individual LWP instructions in APM Volume 3.
14	—	Reserved
13	WDT	Watchdog timer support. See APM Volume 2 and APM Volume 3. Indicates support for MSRC001_0074.
12	SKINIT	SKINIT and STGI are supported. Indicates support for SKINIT and STGI, independent of the value of MSRC000_0080[SVME]. See APM Volume 2 and APM Volume 3.
11	XOP	Extended operation support.
10	IBS	Instruction based sampling. See “Instruction Based Sampling” in APM Volume 2.
9	OSVW	OS visible workaround. Indicates OS-visible workaround support. See “OS Visible Work-around (OSVW) Information” in APM Volume 2.
8	3DNowPrefetch	PREFETCH and PREFETCHW instruction support. See “PREFETCH” and “PREFETCHW” in APM Volume 3.
7	MisAlignSse	Misaligned SSE mode. See “Misaligned Access Support Added for SSE Instructions” in APM Volume 1.
6	SSE4A	EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support. See “EXTRQ”, “INSERTQ”, “MOVNTSS”, and “MOVNTSD” in APM Volume 4.
5	ABM	Advanced bit manipulation. LZCNT instruction support. See “LZCNT” in APM Volume 3.
4	AltMovCr8	LOCK MOV CR0 means MOV CR8. See “MOV(CRn)” in APM Volume 3.
3	ExtApicSpace	Extended APIC space. This bit indicates the presence of extended APIC register space starting at offset 400h from the “APIC Base Address Register,” as specified in the BKDG.
2	SVM	Secure virtual machine. See “Secure Virtual Machine” in APM Volume 2.
1	CmpLegacy	Core multi-processing legacy mode. See “Legacy Method” on page 671.
0	LahfSahf	LAHF and SAHF instruction support in 64-bit mode. See “LAHF” and “SAHF” in APM Volume 3.

CPUID Fn8000_0001_EDX Feature Identifiers

This function contains the following miscellaneous feature identifiers:

Bits	Field Name	Description
31	3DNow	3DNow!™ instructions. See Appendix D “Instruction Subsets and CPUID Feature Sets” in APM Volume 3.
30	3DNowExt	AMD extensions to 3DNow! instructions. See Appendix D “Instruction Subsets and CPUID Feature Sets” in APM Volume 3.
29	LM	Long mode. See “Processor Initialization and Long-Mode Activation” in APM Volume 2.
28	—	Reserved
27	RDTSCP	RDTSCP instruction. See “RDTSCP” in APM Volume 3.
26	Page1GB	1-GB large page support. See “1-GB Paging Support” in APM Volume 2.
25	FFXSR	FXSAVE and FXRSTOR instruction optimizations. See “FXSAVE” and “FXRSTOR” in APM Volume 5.
24	FXSR	FXSAVE and FXRSTOR instructions. Same as CPUID Fn0000_0001_EDX[FXSR].
23	MMX	MMX™ instructions. Same as CPUID Fn0000_0001_EDX[MMX].
22	MmxExt	AMD extensions to MMX instructions. See Appendix D “Instruction Subsets and CPUID Feature Sets” in APM Volume 3 and “128-Bit Media and Scientific Programming” in APM Volume 1.
21	—	Reserved.
20	NX	No-execute page protection. See “Page Translation and Protection” in APM Volume 2.
19:18	—	Reserved
17	PSE36	Page-size extensions. Same as CPUID Fn0000_0001_EDX[PSE36].
16	PAT	Page attribute table. Same as CPUID Fn0000_0001_EDX[PAT].
15	CMOV	Conditional move instructions. Same as CPUID Fn0000_0001_EDX[CMOV].
14	MCA	Machine check architecture. Same as CPUID Fn0000_0001_EDX[MCA].
13	PGE	Page global extension. Same as CPUID Fn0000_0001_EDX[PGE].
12	MTRR	Memory-type range registers. Same as CPUID Fn0000_0001_EDX[MTRR].
11	SysCallSysRet	SYSCALL and SYSRET instructions. See “SYSCALL” and “SYSRET” in APM Volume 3.
10	—	Reserved
9	APIC	Advanced programmable interrupt controller. Same as CPUID Fn0000_0001_EDX[APIC].
8	CMPXCHG8B	CMPXCHG8B instruction. Same as CPUID Fn0000_0001_EDX[CMPXCHG8B].
7	MCE	Machine check exception. Same as CPUID Fn0000_0001_EDX[MCE].
6	PAE	Physical-address extensions. Same as CPUID Fn0000_0001_EDX[PAE].
5	MSR	AMD model-specific registers. Same as CPUID Fn0000_0001_EDX[MSR].
4	TSC	Time stamp counter. Same as CPUID Fn0000_0001_EDX[TSC].

Bits	Field Name	Description
3	PSE	Page-size extensions. Same as CPUID Fn0000_0001_EDX[PSE].
2	DE	Debugging extensions. Same as CPUID Fn0000_0001_EDX[DE].
1	VME	Virtual-mode enhancements. Same as CPUID Fn0000_0001_EDX[VME].
0	FPU	x87 floating-point unit on-chip. Same as CPUID Fn0000_0001_EDX[FPU].

E.4.3 Functions 8000_0002h–8000_0004h—Extended Processor Name String

CPUID Fn8000_000[4:2]_E[D,C,B,A]X Processor Name String Identifier

The three extended functions from Fn8000_0002 to Fn8000_0004 are programmed to return a null terminated ASCII string up to 48 characters in length corresponding to the processor name.

Bits	Field Name	Description
31:0	ProcName	Four characters of the extended processor name string.

The 48 character maximum includes the terminating null character. The 48 character string is ordered first to last (left to right) as follows:

Fn8000_0002[EAX[7:0],..., EAX[31:24], EBX[7:0],..., EBX[31:24], ECX[7:0],..., ECX[31:24],EDX[7:0],..., EDX[31:24]],
Fn8000_0003[EAX[7:0],..., EAX[31:24], EBX[7:0],..., EBX[31:24], ECX[7:0],..., ECX[31:24], EDX[7:0],..., EDX[31:24]],
Fn8000_0004[EAX[7:0],..., EAX[31:24], EBX[7:0],..., EBX[31:24], ECX[7:0],..., ECX[31:24], EDX[7:0],..., EDX[31:24]].

The extended processor name string is programmed by system firmware. See your processor revision guide for information about how to display the extended processor name string.

E.4.4 Function 8000_0005h—L1 Cache and TLB Information

This function provides first level cache TLB characteristics for the processor that executes the instruction.

CPUID Fn8000_0005_EAX L1 TLB 2M/4M Information

The value returned in EAX provides information about the L1 TLB for 2-MB and 4-MB pages.

Bits	Field Name	Description
31:24	L1DTlb2and4MAssoc	Data TLB associativity for 2-MB and 4-MB pages. Encoding is per Table E-3 below.
23:16	L1DTlb2and4MSize	Data TLB number of entries for 2-MB and 4-MB pages. The value returned is for the number of entries available for the 2-MB page size; 4-MB pages require two 2-MB entries, so the number of entries available for the 4-MB page size is one-half the returned value.
15:8	L1ITlb2and4MAssoc	Instruction TLB associativity for 2-MB and 4-MB pages. Encoding is per Table E-3 below.
7:0	L1ITlb2and4MSize	Instruction TLB number of entries for 2-MB and 4-MB pages. The value returned is for the number of entries available for the 2-MB page size; 4-MB pages require two 2-MB entries, so the number of entries available for the 4-MB page size is one-half the returned value.

The associativity fields (L1DTlb2and4MAssoc and L1ITlb2and4MAssoc) are encoded as follows:

Table E-3. L1 Cache and TLB Associativity Field Encodings

Associativity [7:0]	Definition
00h	Reserved
01h	1 way (direct mapped)
02h–FEh	*n*-way associative. (field encodes *n*)
FFh	Fully associative

CPUID Fn8000_0005_EBX L1 TLB 4K Information

The value returned in EBX provides information about the L1 TLB for 4-KB pages.

Bits	Field Name	Description
31:24	L1DTlb4KAssoc	Data TLB associativity for 4 KB pages. Encoding is per Table E-3 above.
23:16	L1DTlb4KSize	Data TLB number of entries for 4 KB pages.
15:8	L1ITlb4KAssoc	Instruction TLB associativity for 4 KB pages. Encoding is per Table E-3 above.
7:0	L1ITlb4KSize	Instruction TLB number of entries for 4 KB pages.

The associativity fields (L1DTlb4KAssoc and L1ITlb4KAssoc) are encoded as specified in Table E-3 on page 645.

CPUID Fn8000_0005_ECX L1 Data Cache Information

The value returned in ECX provides information about the first level data cache.

Bits	Field Name	Description
31:24	L1DcSize	L1 data cache size in KB.
23:16	L1DcAssoc	L1 data cache associativity. Encoding is per Table E-3.
15:8	L1DcLinesPerTag	L1 data cache lines per tag.
7:0	L1DcLineSize	L1 data cache line size in bytes.

The associativity field (L1DcAssoc) is encoded as specified in Table E-3 on page 645.

CPUID Fn8000_0005_EDX L1 Instruction Cache Information

The value returned in EDX provides information about the first level instruction cache.

Bits	Field Name	Description
31:24	L1IcSize	L1 instruction cache size KB.
23:16	L1IcAssoc	L1 instruction cache associativity. Encoding is per Table E-3.
15:8	L1IcLinesPerTag	L1 instruction cache lines per tag.
7:0	L1IcLineSize	L1 instruction cache line size in bytes.

The associativity field (L1IcAssoc) is encoded as specified in Table E-3 on page 645.

E.4.5 Function 8000_0006h—L2 Cache and TLB and L3 Cache Information

This function provides the second level cache and TLB characteristics for the logical processor that executes the instruction. The EDX register returns the processor's third level cache characteristics that are shared by all logical processors in the package.

CPUID Fn8000_0006_EAX L2 TLB 2M/4M Information

The value returned in EAX provides information about the L2 TLB for 2-MB and 4-MB pages.

Bits	Field Name	Description
31:28	L2DTlb2and4MAssoc	L2 data TLB associativity for 2-MB and 4-MB pages. Encoding is per Table E-4 below.
27:16	L2DTlb2and4MSize	L2 data TLB number of entries for 2-MB and 4-MB pages. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.

Bits	Field Name	Description
15:12	L2ITlb2and4MAssoc	L2 instruction TLB associativity for 2-MB and 4-MB pages. Encoding is per Table E-4 below.
11:0	L2ITlb2and4MSize	L2 instruction TLB number of entries for 2-MB and 4-MB pages. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.

The associativity fields (L2DTlb2and4MAssoc and L2ITlb2and4MAssoc) are encoded as follows:

Table E-4. L2/L3 Cache and TLB Associativity Field Encoding

Associativity [3:0]	Definition
0h	L2/L3 cache or TLB is disabled.
1h	Direct mapped.
2h	2-way associative.
3h	3-way associative.
4h	4 to 5-way associative.
5h	6 to 7-way associative.
6h	8 to 15-way associative.
7h	Permanently reserved
8h	16 to 31-way associative.
9h	Value for all fields should be determined from Fn8000_001D.
Ah	32 to 47-way associative.
Bh	48 to 63-way associative.
Ch	64 to 95-way associative.
Dh	96 to 127-way associative.
Eh	More than 128-way associative but not fully associative.
Fh	Fully associative.

CPUID Fn8000_0006_EBX L2 TLB 4K Information

The value returned in EBX provides information about the L2 TLB for 4-KB pages.

Bits	Field Name	Description
31:28	L2DTlb4KAssoc	L2 data TLB associativity for 4-KB pages. Encoding is per Table E-4 above.
27:16	L2DTlb4KSize	L2 data TLB number of entries for 4-KB pages.
15:12	L2ITlb4KAssoc	L2 instruction TLB associativity for 4-KB pages. Encoding is per Table E-4 above.
11:0	L2ITlb4KSize	L2 instruction TLB number of entries for 4-KB pages.

The associativity fields (L2DTlb4KAssoc and L2ITlb4KAssoc) are encoded per Table E-4 above.

CPUID Fn8000_0006_ECX L2 Cache Information

The value returned in ECX provides information about the L2 cache.

Bits	Field Name	Description
31:16	L2Size	L2 cache size in KB.
15:12	L2Assoc	L2 cache associativity. Encoding is per Table E-4 above.
11:8	L2LinesPerTag	L2 cache lines per tag.
7:0	L2LineSize	L2 cache line size in bytes.

The associativity field (L2Assoc) is encoded per Table E-4 on page 647.

CPUID Fn8000_0006_EDX L3 Cache Information

The value returned in EDX provides the third level cache characteristics shared by all logical processors in the package.

Bits	Field Name	Description
31:18	L3Size	Specifies the L3 cache size range: (L3Size[31:18] * 512KB) ≤ L3 cache size < ((L3Size[31:18]+1) * 512KB).
17:16	—	Reserved
15:12	L3Assoc	L3 cache associativity. Encoded per Table E-4 on page 647.
11:8	L3LinesPerTag	L3 cache lines per tag.
7:0	L3LineSize	L3 cache line size in bytes.

The associativity field (L3Assoc) is encoded per Table E-4 on page 647.

E.4.6 Function 8000_0007h—Processor Power Management and RAS Capabilities

This function provides information about the power management, power reporting, and RAS capabilities of the processor that executes the instruction.There may be other processor-specific features and reporting capabilities not covered here. Refer to the *BIOS and Kernel Developer's Guide* for your specific product to obtain more information.

CPUID Fn8000_0007_EAX Reserved

Bits	Field Name	Description
31:0	—	Reserved

CPUID Fn8000_0007_EBX RAS Capabilities

The value returned in EBX provides information about RAS features that allow system software to detect specific hardware errors.

Bits	Field Name	Description
31:4	—	Reserved
3	ScalableMca	0=MCAX is not supported. 1=MCAX is supported; the MCAX MSR addresses are supported; MCA Extension (MCAX) support. Indicates support for MCAX MSRs. MCA_CONFIG[Mcax] is present in all MCA banks.
2	HWA	Hardware assert support. Indicates support for MSRC001_10[DF:C0].
1	SUCCOR	Software uncorrectable error containment and recovery capability. The processor supports software containment of uncorrectable errors through context synchronizing data poisoning and deferred error interrupts; see APM Volume 2, Chapter 9, “Determining Machine-Check Architecture Support.”
0	McaOverflowRecov	MCA overflow recovery support. If set, indicates that MCA overflow conditions (MCi_STATUS[Overflow]=1) are not fatal; software may safely ignore such conditions. If clear, MCA overflow conditions require software to shut down the system. See APM Volume 2, Chapter 9, “Handling Machine Check Exceptions.”

CPUID Fn8000_0007_ECX Processor Power Monitoring Interface

The value returned in ECX provides information about the implementation of the processor power monitoring interface.

Bits	Field Name	Description
31:0	CpuPwrSampleTimeRatio	Specifies the ratio of the compute unit power accumulator sample period to the TSC counter period. Returns a value of 0 if not applicable for the system.

CPUID Fn8000_0007_EDX Advanced Power Management Features

The value returned in EDX provides information about the advanced power management and power reporting features available. Refer to the *BIOS and Kernel Developer’s Guide* for your specific product for a detailed description of the definition of each power management feature.

Bits	Field Name	Description
31:13	—	Reserved
12	ProcPowerReporting	Processor power reporting interface supported.
11	ProcFeedbackInterface	Processor feedback interface. Value: 1. 1=Indicates support for processor feedback interface. **Note:** This feature is deprecated.

Bits	Field Name	Description
10	EffFreqRO	Read-only effective frequency interface. 1=Indicates presence of MSRC000_00E7 [Read-Only Max Performance Frequency Clock Count (MPerfReadOnly)] and MSRC000_00E8 [Read-Only Actual Performance Frequency Clock Count (APerfReadOnly)].
9	CPB	Core performance boost.
8	TscInvariant	TSC invariant. The TSC rate is ensured to be invariant across all P-States, C-States, and stop grant transitions (such as STPCLK Throttling); therefore the TSC is suitable for use as a source of time. 0 = No such guarantee is made and software should avoid attempting to use the TSC as a source of time.
7	HwPstate	Hardware P-state control. MSRC001_0061 [P-state Current Limit], MSRC001_0062 [P-state Control] and MSRC001_0063 [P-state Status] exist.
6	100MHzSteps	100 MHz multiplier Control.
5	—	Reserved.
4	TM	Hardware thermal control (HTC).
3	TTP	THERMTRIP.
2	VID	Voltage ID control. Function replaced by HwPstate.
1	FID	Frequency ID control. Function replaced by HwPstate.
0	TS	Temperature sensor.

E.4.7 Function 8000_0008h—Processor Capacity Parameters and Extended Feature Identification

This function provides the size or capacity of various architectural parameters that vary by implementation, as well as an extension to the Fn8000_0001 feature identifiers.

CPUID Fn8000_0008_EAX Long Mode Size Identifiers

The value returned in EAX provides information about the maximum host and guest physical and linear address width (in bits) supported by the processor.

Bits	Field Name	Description
31:24	—	Reserved
23:16	GuestPhysAddrSize	Maximum guest physical address size in bits. This number applies only to guests using nested paging. When this field is zero, refer to the PhysAddrSize field for the maximum guest physical address size. See “Secure Virtual Machine” in APM Volume 2.
15:8	LinAddrSize	Maximum linear address size in bits.
7:0	PhysAddrSize	Maximum physical address size in bits. When GuestPhysAddrSize is zero, this field also indicates the maximum guest physical address size.

The address width reported is the maximum supported in any mode. For long mode capable processors, the size reported is independent of whether long mode is enabled. See “Processor Initialization and Long-Mode Activation” in APM Volume 2.

CPUID Fn8000_0008_EBX Extended Feature Identifiers

The value returned in EBX is an extension to the Fn8000_0001 feature flags and indicates the presence of various ISA extensions.

Bit	Field Name	Description
31	—	Reserved
30	IBPB_RET	IBPB clears return address predictor.
29	BTC_NO	The processor is not affected by branch type confusion.
28	PSFD	Predictive Store Forward Disable.
27	CPPC	Collaborative Processor Performance Control.
26	SsbdNotRequired	SSBD not needed on this processor.
25	SsbdVirtSpecCtrl	Use VIRT_SPEC_CTL MSR (C001_011Fh) for SSBD.
24	SSBD	Speculative Store Bypass Disable.
23:22	—	Reserved
21	INVLPGBnestedPages	INVLPGB support for invalidating guest nested translations.
20	EferLmsleUnsupported	EFER.LMSLE is unsupported.
19	IbrsSameMode	IBRS provides same mode speculation limits.
18	IbrsPreferred	IBRS is preferred to software solution.
17	StibpAlwaysOn	Setting STIBP to 1 once is recommended.
16	IbrsAlwaysOn	Setting IBRS to 1 once is recommended.
15	STIBP	Single Thread Indirect Branch Prediction mode.
14	IBRS	Indirect Branch Restricted Speculation.
13	INT_WBINVD	WBINVD/WBNOINVD instructions are interruptible.
12	IBPB	Indirect Branch Prediction Barrier.
11:10	—	Reserved.
9	WBNOINVD	WBNOINVD instruction supported.
8	MCOMMIT	MCOMMIT instruction supported.
7	—	Reserved
6	BE	Bandwidth Enforcement Extension.
5	—	Reserved
4	RDPRU	RDPRU instruction supported.
3	INVLPGB	INVLPGB and TLBSYNC instruction supported.
2	RstrFpErrPtrs	FP Error Pointers Restored by XRSTOR.
1	InstRetCntMsr	Instruction Retired Counter MSR available.
0	CLZERO	CLZERO instruction supported.

CPUID Fn8000_0008_ECX Size Identifiers

The value returned in ECX provides information about the number of cores supported by the processor, the width of the APIC ID, and the width of the performance time-stamp counter.

Bits	Field Name	Description
31:18	—	Reserved
17:16	PerfTscSize	Performance time-stamp counter size. Indicates the size of MSRC001_0280[PTSC]. Bits Description 00b 40 bits 01b 48 bits 10b 56 bits 11b 64 bits

15:12	ApicIdSize	APIC ID size. The number of bits in the initial APIC20[ApicId] value that indicate logical processor ID within a package. The size of this field determines the maximum number of logical processors (MNLP) that the package could theoretically support, and not the actual number of logical processors that are implemented or enabled in the package, as indicated by CPUID Fn8000_0008_ECX[NC]. A value of zero indicates that legacy methods must be used to determine the maximum number of logical processors, as indicated by CPUID Fn8000_0008_ECX[NC].
		
		if (ApicIdSize[3:0] == 0) {
		// Used by legacy dual-core/single-core processors
		MNLP = CPUID Fn8000_0008_ECX[NC] + 1;
		} else {
		// use ApicIdSize[3:0] field
		MNLP = (2 raised to the power of ApicIdSize[3:0]);
		}

11:8	—	Reserved
7:0	NC	Number of physical threads - 1. The number of threads in the processor is NC+1 (e.g., if NC = 0, then there is one thread). See “Legacy Method” on page 671.

CPUID Fn8000_0008_EDX RDPRU Register Identifier Range

The value returned in EDX identifies the maximum recognized register identifier for the RDPRU instruction.

Bits	Field Name	Description
31:16	MaxRdpruID	The maximum ECX value recognized by RDPRU.
15:0	InvlpgbCountMax	Maximum page count for INVLPGB instruction.

E.4.8 Function 8000_0009h—Reserved

CPUID Fn8000_0009 Reserved

This function is reserved.

E.4.9 Function 8000_000Ah—SVM Features

This function provides information about the SVM features that the processor supports. If SVM is not supported (CPUID Fn8000_0001_ECX[SVM] = 0), this function is reserved.

CPUID Fn8000_000A_EAX SVM Revision and Feature Identification

The value returned in EAX provides the SVM revision number.

Bits	Field Name	Description
31:8	—	Reserved
7:0	SvmRev	SVM revision number.

CPUID Fn8000_000A_EBX SVM Revision and Feature Identification

The value returned in EBX provides the number of address space identifiers (ASIDs) that the processor supports.

Bits	Field Name	Description
31:0	NASID	Number of available address space identifiers (ASID).

CPUID Fn8000_000A_ECX SVM Feature Identification

The value returned in EDX provides Secure Virtual Machine architecture feature information. All cross references in the table below are to sections within the “Secure Virtual Machine” on page 497 of APM Volume 2.

Bits	Field Name	Description
31:7	—	Reserved
6	x2AVIC_EXT	4096 vCPUs supported in x2AVIC mode.
5:0	—	Reserved

CPUID Fn8000_000A_EDX SVM Feature Identification

The value returned in EDX provides Secure Virtual Machine architecture feature information. All cross references in the table below are to sections within the “Secure Virtual Machine” on page 497 of APM Volume 2.

Bits	Field Name	Description
31	—	Reserved
30	IdleHltIntercept	Idle HLT intercept.
29	BusLockThreshold	Bus Lock Threshold.
28	NestedVirtVmcbAddrChk	Guest VMCB address check.
27	ExtLvtAvicAccessChg	Extended Interrupt Local Vector Table Register AVIC Access changes. See “Virtual APIC Register Accesses.”
26	IbsVirt	IBS Virtualization. See “Instruction-Based Sampling Virtualization.”
25	VNMI	NMI Virtualization. See “NMI Virtualization.”
24	TlbiCtl	INVLPGB/TLBSYNC hypervisor enable in VMCB and TLBSYNC intercept support.
23	HOST_MCE_OVERRIDE	When host CR4.MCE=1 and guest CR4.MCE=0, machine check exceptions (#MC) in a guest do not cause shutdown and are always intercepted.
22	—	Reserved
21	ROGPT	Read-Only Guest Page Table feature support. See “Nested Table Walk.”
20	SpecCtrl	SPEC_CTRL virtualization.
19	SSSCheck	SVM supervisor shadow stack restrictions. See “Supervisor Shadow Stack Restrictions.”
18	x2AVIC	Support for the AMD advanced virtual interrupt controller for x2APIC mode. See “Advanced Virtual Interrupt Controller.”
17	GMET	Guest Mode Execution Trap.
16	VGIF	Virtualize the Global Interrupt Flag. See “Nested Virtualization.”
15	VMSAVEvirt	VMSAVE and VMLOAD virtualization. See “Nested Virtualization.”
14	—	Reserved
13	AVIC	Support for the AMD advanced virtual interrupt controller. See “Advanced Virtual Interrupt Controller.”
12	PauseFilterThreshold	PAUSE filter threshold. Indicates support for the PAUSE filter cycle count threshold. See “Pause Intercept Filtering.”
11	—	Reserved
10	PauseFilter	Pause intercept filter. Indicates support for the pause intercept filter. See “Pause Intercept Filtering.”
9	—	Reserved
8	PmcVirt	PMC virtualization. See “Performance Monitoring Counter Virtualization.”
7	DecodeAssists	Decode assists. Indicates support for the decode assists. See “Decode Assists.”
6	FlushByAsid	Flush by ASID. Indicates that TLB flush events, including CR3 writes and CR4.PGE toggles, flush only the current ASID's TLB entries. Also indicates support for the extended VMCB TLB_Control. See “TLB Control.”
5	VmcbClean	VMCB clean bits. Indicates support for VMCB clean bits. See “VMCB Clean Bits.”

Bits	Field Name	Description
4	TscRateMsr	MSR based TSC rate control. Indicates support for MSR TSC ratio MSRC000_0104. See “TSC Ratio MSR (C000_0104h).”
3	NRIPS	NRIP save. Indicates support for NRIP save on #VMEXIT. See “State Saved on Exit.”
2	SVML	SVM lock. Indicates support for SVM-Lock. See “Enabling SVM.”
1	LbrVirt	LBR virtualization. Indicates support for LBR Virtualization. See “Hardware Acceleration for LBR Virtualization” in Section 15.23.1 of APM Volume 2
0	NP	Nested paging. Indicates support for nested paging. See “Nested Paging.”

E.4.10 Functions 8000_000Bh–8000_0018h—Reserved

CPUID Fn8000_00[18:0B] Reserved

These functions are reserved.

E.4.11 Function 8000_0019h—TLB Characteristics for 1GB pages

This function provides information about the TLB for 1 GB pages for the processor that executes the instruction.

CPUID Fn8000_0019_EAX L1 TLB 1G Information

The value returned in EAX provides information about the L1 TLB for 1 GB pages.

Bits	Field Name	Description
31:28	L1DTlb1GAssoc	L1 data TLB associativity for 1 GB pages. See Table E-4 on page 647.
27:16	L1DTlb1GSize	L1 data TLB number of entries for 1 GB pages.
15:12	L1ITlb1GAssoc	L1 instruction TLB associativity for 1 GB pages. See Table E-4 on page 647.
11:0	L1ITlb1GSize	L1 instruction TLB number of entries for 1 GB pages.

CPUID Fn8000_0019_EBX L2 TLB 1G Information

The value returned in EBX provides information about the L2 TLB for 1 GB pages.

Bits	Field Name	Description
31:28	L2DTlb1GAssoc	L2 data TLB associativity for 1 GB pages. See Table E-4 on page 647.
27:16	L2DTlb1GSize	L2 data TLB number of entries for 1 GB pages.
15:12	L2ITlb1GAssoc	L2 instruction TLB associativity for 1 GB pages. See Table E-4 on page 647.
11:0	L2ITlb1GSize	L2 instruction TLB number of entries for 1 GB pages.

CPUID Fn8000_0019_E[D,C]X Reserved

The values returned in ECX and EDX for this function are undefined and reserved for future use.

E.4.12 Function 8000_001Ah—Instruction Optimizations

CPUID Fn8000_001A_EAX Performance Optimization Identifiers

This function returns performance related information. For more details on how to use these bits to optimize software, see the *Software Optimization Guide* applicable to your product.

Bits	Field Name	Description
31:3	—	Reserved
2	FP256	The internal FP/SIMD execution data path is 256 bits wide.
1	MOVU	MOVU SSE instructions are more efficient and should be preferred to SSE MOVL/MOVH. MOVUPS is more efficient than MOVLPS/MOVHPS. MOVUPD is more efficient than MOVLPD/MOVHPD.
0	FP128	The internal FP/SIMD execution data path is 128 bits wide.

CPUID Fn8000_001A_E[D,C,B]X Reserved

The values returned in EBX, ECX, and EDX are undefined for this function and are reserved.

E.4.13 Function 8000_001Bh—Instruction-Based Sampling Capabilities

If instruction-based sampling (IBS) is supported (CPUID Fn8000_0001_ECX[IBS] = 1), this CPUID function can be used to obtain IBS feature information. If IBS is not supported (CPUID Fn8000_0001_ECX[IBS] = 0), this function number is reserved. For more information on using IBS, see “Instruction-Based Sampling” in APM Volume 2.

CPUID Fn8000_001B_EAX Instruction-Based Sampling Feature Indicators

The value returned in EAX provides the following information about the specific features of IBS that the processor supports:

Bits	Field Name	Description
31:12		Reserved
11	IbsL3MissFiltering	L3 Miss Filtering for IBS supported. See IBS Filtering in APM Volume 2.
10:9		Reserved
8	OpBrnFuse	Fused branch micro-op indication supported.
7	RipInvalidChk	Invalid RIP indication supported.
6	OpCntExt	IbsOpCurCnt and IbsOpMaxCnt extend by 7 bits.

Bits	Field Name	Description
5	BrnTrgt	Branch target address reporting supported.
4	OpCnt	Op counting mode supported.
3	RdWrOpCnt	Read write of op counter supported.
2	OpSam	IBS execution sampling supported.
1	FetchSam	IBS fetch sampling supported.
0	IBSFFV	IBS feature flags valid.

CPUID Fn8000_001B_E[D,C,B]X Reserved

The values returned in EBX, ECX, and EDX are undefined and are reserved.

E.4.14 Function 8000_001Ch—Lightweight Profiling Capabilities

If lightweight profiling (LWP) is supported (CPUID Fn8000_0001_ECX[LWP] = 1), this CPUID function can be used to obtain information about LWP features supported by the processor. If LWP is not supported (CPUID Fn8000_0001_ECX[LWP] = 0), this function number is reserved. For more information on using LWP, see “Lightweight Profiling” in APM Volume 2.

CPUID Fn8000_001C_EAX Lightweight Profiling Capabilities 0

The value returned in EAX provides the following information about LWP capabilities supported by the processor:

Bits	Field Name	Description
31	LwpInt	Interrupt on threshold overflow available.
30	LwpPTSC	Performance time stamp counter in event record is available.
29	LwpCont	Sampling in continuous mode is available.
28:7	—	Reserved
6	LwpRNH	Core reference clocks not halted event available.
5	LwpCNH	Core clocks not halted event available.
4	LwpDME	DC miss event available.
3	LwpBRE	Branch retired event available.
2	LwpIRE	Instructions retired event available.
1	LwpVAL	LWPVAL instruction available.
0	LwpAvail	The LWP feature is available.

CPUID Fn8000_001C_EBX Lightweight Profiling Capabilities 0

The value returned in EBX provides the following additional information about LWP capabilities supported by the processor:

Bits	Field Name	Description
31:24	LwpEventOffset	Offset in bytes from the start of the LWPCB to the EventInterval1 field.
23:16	LwpMaxEvents	Maximum EventId value supported.
15:8	LwpEventSize	Event record size. Size in bytes of an event record in the LWP event ring buffer.
7:0	LwpCbSize	Control block size. Size in quadwords of the LWPCB.

CPUID Fn8000_001C_ECX Lightweight Profiling Capabilities 0

The value returned in ECX provides the following additional information about LWP capabilities supported by the processor:

Bits	Field Name	Description
31	LwpCacheLatency	Cache latency filtering supported. Cache-related events can be filtered by latency.
30	LwpCacheLevels	Cache level filtering supported. Cache-related events can be filtered by the cache level that returned the data.
29	LwpIpFiltering	IP filtering supported.
28	LwpBranchPrediction	Branch prediction filtering supported. Branches Retired events can be filtered based on whether the branch was predicted properly.
27:24	—	Reserved
23:16	LwpMinBufferSize	Event ring buffer size. Minimum size of the LWP event ring buffer, in units of 32 event records.
15:9	LwpVersion	Version of LWP implementation.
8:6	LwpLatencyRnd	Amount by which cache latency is rounded.
5	LwpDataAddress	Data cache miss address valid. Address is valid for cache miss event records.
4:0	LwpLatencyMax	Latency counter size. Size in bits of the cache latency counters.

CPUID Fn8000_001C_EDX Lightweight Profiling Capabilities 0

The value returned in EDX provides the following additional information about LWP capabilities supported by the processor:

Bits	Field Name	Description
31	LwpInt	Interrupt on threshold overflow supported.
30	LwpPTSC	Performance time stamp counter in event record is supported.
29	LwpCont	Sampling in continuous mode is supported.
28:7	—	Reserved
6	LwpRNH	Core reference clocks not halted event is supported.
5	LwpCNH	Core clocks not halted event is supported.
4	LwpDME	DC miss event is supported.
3	LwpBRE	Branch retired event is supported.

Bits	Field Name	Description
2	LwpIRE	Instructions retired event is supported.
1	LwpVAL	LWPVAL instruction is supported.
0	LwpAvail	Lightweight profiling is supported.

E.4.15 Function 8000_001Dh—Cache Topology Information

CPUID Fn8000_001D reports cache topology information for the cache enumerated by the value passed to the instruction in ECX, referred to as Cache *n* in the following description. To gather information for all cache levels, software must repeatedly execute CPUID with 8000_001Dh in EAX and ECX set to increasing values beginning with 0 until a value of 00h is returned in the field CacheType (EAX[4:0]) indicating no more cache descriptions are available for this processor.

If CPUID Fn8000_0001_ECX[TopologyExtensions] = 0, then CPUID Fn8000_001Dh is reserved. Any value in ECX which does not select an existing cache will return a Null cache type in EAX[4:0].

CPUID Fn8000_001D_EAX_x[N:0] Cache Properties

Bits	Field Name	Description
31:26	—	Reserved
25:14	NumSharingCache	Specifies the number of logical processors sharing the cache enumerated by *N*, the value passed to the instruction in ECX. The number of logical processors sharing this cache is the value of this field incremented by 1. To determine which logical processors are sharing a cache, determine a Share Id for each processor as follows: ShareId = LocalApicId >> log2(NumSharingCache+1) Logical processors with the same ShareId then share a cache. If NumSharingCache+1 is not a power of two, round it up to the next power of two.
13:10	—	Reserved
9	FullyAssociative	Fully associative cache. When set, indicates that the cache is fully associative. If 0 is returned in this field, the cache is set associative.
8	SelfInitialization	Self-initializing cache. When set, indicates that the cache is self initializing; software initialization not required. If 0 is returned in this field, hardware does not initialize this cache.

Bits	Field Name	Description
7:5	CacheLevel	Cache level. Identifies the level of this cache. Note that the enumeration value is not necessarily equal to the cache level. Bits Description 000b Reserved. 001b Level 1 010b Level 2 011b Level 3 111b-100b Reserved.
4:0	CacheType	Cache type. Identifies the type of cache. Bits Description 00h Null; no more caches. 01h Data cache 02h Instruction cache 03h Unified cache 1Fh-04h Reserved

CPUID Fn8000_001D_EBX_x[N:0] Cache Properties

See CPUID Fn8000_001D_EAX_x[N:0].

Bits	Field Name	Description
31:22	CacheNumWays	Number of ways for this cache. The number of ways is the value returned in this field incremented by 1.
21:12	CachePhysPartitions	Number of physical line partitions. The number of physical line partitions is the value returned in this field incremented by 1.
11:0	CacheLineSize	Cache line size. The cache line size in bytes is the value returned in this field incremented by 1.

CPUID Fn8000_001D_ECX_x[N:0] Cache Properties

See CPUID Fn8000_001D_EAX_x[N:0].

Bits	Field Name	Description
31:0	CacheNumSets	Number of ways for set associative cache. Number of ways is the value returned in this field incremented by 1. Only valid for caches that are not fully associative (Fn8000_001D_EAX_x*n*[FullyAssociative] = 0).

CPUID Fn8000_001D_EDX_x[N:0] Cache Properties

See CPUID Fn8000_001D_EAX_x[N:0].

Bits	Field Name	Description
31:2	—	Reserved
1	CacheInclusive	Cache inclusivity. A value of 0 indicates that this cache is not inclusive of lower cache levels. A value of 1 indicates that the cache is inclusive of lower cache levels.
0	WBINVD	Write-Back Invalidate/Invalidate execution scope. A value of 0 returned in this field indicates that the WBINVD/INVD instruction invalidates all lower level caches of non-originating logical processors sharing this cache. When set, this field indicates that the WBINVD/INVD instruction is not guaranteed to invalidate all lower level caches of non-originating logical processors sharing this cache.

E.4.16 Function 8000_001Eh—Processor Topology Information

CPUID Fn8000_001E_EAX Extended APIC ID

If CPUID Fn8000_0001_ECX[TopologyExtensions] = 0, this function number is reserved.

Bits	Field Name	Description
31:0	ExtendedApicId	Extended APIC ID. If MSR0000_001B[ApicEn] = 0, this field is reserved.

CPUID Fn8000_001E_EBX Compute Unit Identifiers

See CPUID Fn8000_001E_EAX.

Bits	Field Name	Description
31:16	—	Reserved
15:8	ThreadsPerComputeUnit	Threads per compute unit (zero-based count). The actual number of threads per compute unit is the value of this field + 1. To determine which logical processors (threads) belong to a given Compute Unit, determine a ShareId for each processor as follows: ShareId = LocalApicId >> log2(ThreadsPerComputeUnit+1) Logical processors with the same ShareId then belong to the same Compute Unit. (If ThreadsPerComputeUnit+1 is not a power of two, round it up to the next power of two).
7:0	ComputeUnitId	Compute unit ID. Identifies a Compute Unit, which may be one or more physical cores that each implement one or more logical processors.

CPUID Fn8000_001E_ECX Node Identifiers

See CPUID Fn8000_001E_EAX.

Bits	Field Name	Description
31:0	—	Reserved
10:8	NodesPerProcessor	Specifies the number of nodes in the package/socket in which this logical processor resides. Node in this context corresponds to a processor die. Encoding is N-1, where N is the number of nodes present in the socket.
7:0	NodeId	Specifies the ID of the node containing the current logical processor. NodeId values are unique across the system.

CPUID Fn8000_001E_EDX Reserved

The value returned in EDX is undefined and is reserved.

E.4.17 Function 8000_001Fh—SEV Capabilities

CPUID Fn8000_001F_EAX SEV Capabilities

Bits	Field Name	Description
31	IbpbOnEntry	IBPB on Entry supported.
30	HvInUseWrAllowed	Writes to Hypervisor-Owned pages are allowed when marked in-use.
29	NestedVirtSnpMsr	VIRT_RMPUPDATE MSR (C001_F001h) and VIRT_PSMASH MSR (C001_F002h) supported.
28	SvsmCommPageMsr	SVSM Communication Page MSR (C001_F000h) is supported.
27	AllowedSevFeatures	Allowed SEV Features supported.
26	SecureAvic	Secure AVIC supported.
25	SmtProtection	SMT Protection supported.
24	VmsaRegProt	VMSA Register Protection supported.
23	SegmentedRmp	Segmented RMP supported.
22	GuestInterceptCtl	Guest intercept control supported.
21	RMPREAD	RMPREAD Instruction supported.
20	PmcVirtGuestCtl	PMC Virtualization supported for SEV-ES and SEV-SNP guests.
19	IbsVirtGuestCtl	IBS Virtualization supported for SEV-ES and SEV-SNP guests.
18	VirtualTomMsr	Virtual TOM MSR supported.
17	VmgexitParameter	VMGEXIT Parameter supported.
16	VTE	Virtual Transparent Encryption supported.
15	PreventHostIbs	Disallowing IBS use by the host supported.
14	DebugVirt	Full debug state virtualization supported for SEV-ES and SEV-SNP guests.

Bits	Field Name	Description
13	AlternateInjection	Alternate Injection supported.
12	RestrictedInjection	Restricted Injection supported.
11	64BitHost	SEV guest execution only allowed from a 64-bit host.
10	HwEnfCacheCoh	Hardware cache coherency across encryption domains enforced.
9	TscAuxVirtualization	TSC AUX Virtualization supported.
8	SecureTsc	Secure TSC supported.
7	VmplSSS	VMPL Supervisor Shadow Stack supported.
6	RMPQUERY	RMPQUERY Instruction supported
5	VMPL	VM Permission Levels supported.
4	SEV-SNP	SEV Secure Nested Paging supported.
3	SEV-ES	SEV Encrypted State supported.
2	PageFlushMsr	Page Flush MSR available.
1	SEV	Secure Encrypted Virtualization supported.
0	SME	Secure Memory Encryption supported.

CPUID Fn8000_001F_EBX SEV Capabilities

Bits	Field Name	Description
31:16	—	Reserved
15:12	NumVMPL	Number of VM Permission Levels supported.
11:6	PhysAddrReduction	Physical Address bit reduction.
5:0	CbitPosition	C-bit location in page table entry.

CPUID Fn8000_001F_ECX SEV Capabilities

Bits	Field Name	Description
31:0	NumEncryptedGuests	Number of encrypted guests supported simultaneously.

CPUID Fn8000_001F_EDX Minimum ASID

Bits	Field Name	Description
31:0	MinSevNoEsAsid	Minimum ASID value for an SEV enabled, SEV-ES disabled guest.

E.4.18 Function 8000_0020—PQOS Extended Features

This CPUID function can be used to obtain PQOS Extended feature information. For more information on using PQOS, see “Platform Quality of Service” in APM Volume 2.

CPUID Fn8000_0020_x0 PQOS Extended Features

Subfunction 0 provides information about PQOS Extended feature capabilities.

Register	Bits	Field Name	Description
EAX	31:0	—	Reserved
EBX	31:7	—	Reserved
	6	SDCIAE	Smart Data Cache Injection (SDCI) Allocation Enforcement.
	5	ABMC	Assignable Bandwidth Monitoring Counters.
	4	L3RR	L3 Range Reservations. See “L3 Range Reservation” in APM Volume 2.
	3	BMEC	Bandwidth Monitoring Event Configuration.
	2	L3SMBE	Slow Memory Bandwidth Enforcement.
	1	L3MBE	Memory Bandwidth Enforcement.
	0	—	Reserved
ECX	31:0	—	Reserved
EDX	31:0	—	Reserved

CPUID Fn8000_0020_x1 L3 Memory Bandwidth Enforcement Information

Subfunction 1 provides information about the L3MBE feature, if this feature is supported (CPUID Fn8000_0020_EBX_x0[L3MBE] = 1). If L3MBE is not supported this function is reserved. For more information on L3MBE, see “Platform Quality of Service” in APM Volume 2.

Register	Bits	Field Name	Description
EAX	31:0	BW_LEN	Identifies the size of the bandwidth specifier field in the L3QOS_BW_Control_n MSRs.
EBX	31:0	—	Reserved
ECX	31:0	—	Reserved
EDX	31:0	COS_MAX	Maximum COS number supported by the L3MBE feature.

CPUID Fn8000_0020_x2 L3 Slow Memory Bandwidth Enforcement Information

Subfunction 2 provides information about the L3SMBE feature, if this feature is supported (CPUID Fn8000_0020_EBX_x0[L3SMBE] = 1). If L3MBE is not supported this function is reserved. For more information on L3SMBE, see “Platform Quality of Service” in APM Volume 2.

Register	Bits	Field Name	Description
EAX	31:0	BW_LEN	Identifies the size of the bandwidth specifier field in the L3QOS_SLOWBW_Control_n MSRs.
EBX	31:0	—	Reserved
ECX	31:0	—	Reserved
EDX	31:0	COS_MAX	Maximum COS number supported by the L3SMBE feature.

CPUID Fn8000_0020_x3 Bandwidth Monitoring Event Counters Information

Subfunction 3 provides information about the BMEC feature, if this feature is supported (CPUID Fn8000_0020_EBX_x0[BMEC] = 1). If BMEC is not supported this function is reserved. For more information on L3MBE, see “Platform Quality of Service” in APM Volume 2.

Register	Bits	Field Name	Description
EAX	31:0	—	Reserved
EBX	31:8	—	Reserved
	7:0	EVT_NUM	Number of configurable bandwidth events.
ECX	31:7	—	Reserved
	6	L3CacheVicMon	Dirty victim writes to all types of memory.
	5	L3CacheRmtSlowBwFIllMon	Reads to remote memory identified as “Slow Memory”.
	4	L3CacheLclSlowBwFIllMon	Reads to local memory identified as “Slow Memory”.
	3	L3CacheRmtBwNtWrMon	Non-temporal writes to remote memory.
	2	L3CacheLclBwNtWrMon	Non-temporal writes to local memory.
	1	L3CacheRmtBwFillMon	Reads to remote DRAM memory.
	0	L3CacheLclBwFillMon	Reads to local DRAM memory.
EDX	31:0	—	Reserved

CPUID Fn8000_0020_x5 Assignable Bandwidth Monitoring Counters Information

Subfunction 5 provides information about the ABMC feature, if this feature is supported (CPUID Fn8000_0020_EAX_x0[ABMC] = 1). If ABMC is not supported this function is reserved. For more information on ABMC see “Platform Quality of Service” in APM Volume 2.

Register	Bits	Field Name	Description
EAX	31:7	—	Reserved
	8	OverflowBit	Indicates that QM_CTR bit 61 is an overflow bit.
	7:0	CounterSize	QM_CTR counter width, offset from 24 bits. If 0, the family, model, and stepping should be used to determine the counter size. See “Platform Quality of Service” in APM Volume 2.

Register	Bits	Field Name	Description
EBX	31:16	—	Reserved
	15:0	MAX_ABMC	Maximum supported ABMC counter ID.
ECX	31:1	—	Reserved
	0	Select_COS	Bandwidth counters can be configured to measure bandwidth consumed by a COS instead of an RMID.
EDX	31:0	—	Reserved

E.4.19 Function 8000_0021—Extended Feature Identification 2

CPUID Fn8000_0021_EAX Extended Feature 2

Bits	Field Name	Description
31	SRSO_MSR_FIX	Software may use MSR_BP_CFG[BpSpecReduce] to mitigate Speculative Return Stack Overflow vulnerability.
30	SRSO_USER_KERNEL_NO	The processor is not affected by Speculative Return Stack Overflow vulnerability across user/kernel boundaries.
29	SRSO_NO	The processor is not affected by Speculative Return Stack Overflow vulnerability.
28	IBPB_BRTYPE	PRED_CMD[IBPB] clears all branch type predictions from the branch predictor.
27	SBPB	Selective Branch Predictor Barrier supported.
26:25	—	Reserved
24	ERAPS	Enhanced Return Address Predictor Security supported.
23:22	—	Reserved
21	FP512_DOWNGRADE	FP512 is downgraded to FP256.
20	PREFETCHI	IC prefetch supported.
19	FAST_REP_SCASB	Fast short REP SCASB supported.
18	EPSF	Enhanced Predictive Store Forwarding supported.
17	CpuidUserDis	CPUID disable for non-privileged software.
16	Opcode0F017Reclaim	0F 01/7 opcode space is reserved for AMD use.
15	AMD_ERMSB	AMD implementation of Enhanced REP MOVSB/STOSB is supported.
14	L2TlbSizeX32	L2TLB sizes are encoded as multiples of 32.
13	PrefetchCtlMsr	Prefetch control MSR supported. See Core::X86::Msr::PrefetchControl in BKDG or PPR for details.
12	Pmc2PreciseRetire	MSR PerfEvtSel2[PreciseRetire] is supported.
11	FastShortRepeCmpsb	Fast short REPE CMPSB supported.
10	FastShortRepStosb	Fast short REP STOSB supported.
9	NoSmmCtlMSR	SMM_CTL MSR (C001_0116h) is not supported.
8	AutomaticIBRS	Automatic IBRS.

Bits	Field Name	Description
7	UpperAddressIgnore	Upper Address Ignore is supported.
6	NullSelectClearsBase	Null segment selector loads also clear the destination segment register base and limit.
5:4	—	Reserved
3	SmmPgCfgLock	SMM paging configuration lock supported.
2	LFenceAlwaysSerializing	LFENCE is always dispatch serializing.
1	FsGsBaseWriteNotSerializing	WRMSR to FS.Base, GS.Base and KernelGSBase MSRs is not serializing.
0	NoNestedDataBp	Processor ignores nested data breakpoints.

CPUID Fn8000_0021_EBX Extended Feature 2

Bits	Field Name	Description
31:24	—	Reserved
23:16	RapSize	Return Address Predictor size
15:0	MicrocodePatchSize	The size of the Microcode patch in 16-byte multiples. If 0, the size of the patch is at most 5568 (15C0h) bytes.

CPUID Fn8000_0021_E[C,D]X Reserved

The values returned in ECX and EDX are undefined and reserved.

E.4.20 Function 8000_0022—Extended Performance Monitoring and Debug

CPUID Fn8000_0022_EAX Extended Performance Monitoring and Debug

Bits	Field Name	Description
31:3	—	Reserved
2	LbrAndPmcFreeze	Freezing Core Performance Counters and LBR Stack on Core Performance Counter overflow supported.
1	LbrStack	Last Branch Record Stack supported.
0	PerfMonV2	Performance Monitoring Version 2 supported. When set, CPUID_Fn8000_0022_EBX reports the number of available performance counters.

CPUID Fn8000_0022_EBX Extended Performance Monitoring and Debug

Bits	Field Name	Description
31:16	—	Reserved
15:10	NumPerfCtrNB	Number of Northbridge Performance Monitor Counters.
9:4	LbrStackSize	Number of Last Branch Record Stack entries.
3:0	NumPerfCtrCore	Number of Core Performance Counters.

CPUID Fn8000_0022_E[C,D]X Reserved

The values returned in ECX and EDX are undefined and are reserved.

E.4.21 Function 8000_0023—Multi-Key Encrypted Memory Capabilities

CPUID Fn8000_0023_EAX Secure Multi-Key Encryption

Bits	Field Name	Description
31:1	—	Reserved
0	MemHmk	Secure Host Multi-Key Memory (MEM-HMK) Encryption Mode Supported.

CPUID Fn8000_0023_EBX Secure Multi-Key Encryption

Bits	Field Name	Description
31:16	—	Reserved
15:0	MaxMemHmkEncrKeyID	Number of simultaneously available host encryption key IDs in MEM-HMK encryption mode.

CPUID Fn8000_0023_E[C,D]X Reserved

The values returned in ECX and EDX are undefined and are reserved.

E.4.22 Function 8000_0024—Reserved

E.4.23 Function 8000_0025—SEV Capabilities 2

CPUID Fn8000_0025_EAX SEV Capabilities 2

Bits	Field Name	Description
31:12	—	Reserved
11:6	MaxRmpSegSize	Maximum supported RMP segment size
5:0	MinRmpSegSize	Minimum supported RMP segment size

CPUID Fn8000_0025_EBX SEV Capabilities 2

Bits	Field Name	Description
31:11	—	Reserved
10	NumSegReduction	Number of RMP segments is reduced
9:0	NumCachedSegments	Number of cached RMP segment definitions

CPUID Fn8000_0025_E[C,D]X Reserved

The values returned in ECX and EDX are undefined and are reserved.

E.4.24 Function 8000_0026—Extended CPU Topology

CPUID Fn8000_0026 reports extended topology information for logical processors, including asymmetric and heterogeneous topology descriptions. Individual logical processors may report different values in systems with asynchronous and heterogeneous topologies.

The topology level is selected by the value passed to the instruction in ECX. To discover the topology of a system, software should execute CPUID Fn8000_0026 with increasing ECX values, starting with a value of zero, until the returned hierarchy level type (CPUID Fn8000_0026_ECX[LevelType]) is equal to zero. It is not guaranteed that all topology level types are present in the system.

Software may use asymmetric and heterogeneous indicators reported by CPUID Fn8000_0026_EAX[31:29] for each hierarchy level to efficiently determine system topology. If CPUID Fn8000_0026_EAX[31:29] is equal to zero at a given hierarchy level, all components at this level are symmetric and homogeneous. If CPUID Fn8000_0026_EAX[31:29] is not equal to zero, software should use APIC ID, APIC ID mask (derived from CPUID Fn8000_0026_EAX[MaskWidth]), and the number of logical processors at a hierarchy level (CPUID Fn8000_0026[NumLogProc]) to determine asymmetric and heterogeneous component properties at this hierarchy level.

CPUID Fn8000_0026_EAX_n[N:0] Extended CPU Topology

Bits	Field Name	Description
31	AsymmetricTopology	Set to 1 if all components at the current hierarchy level do not report the same number of logical processors (NumLogProc).
30	HeterogeneousCores	Set to 1 if all components at the current hierarchy level do not consist of the cores that report the same core type (CoreType).
29	EfficiencyRankingAvailable	Set to 1 if processor power efficiency ranking (PwrEfficiencyRanking) is available and varies between cores. Only valid for LevelType = 1h (Core).
28:5	—	Reserved
4:0	MaskWidth	Number of bits to shift Extended APIC ID right to get a unique topology ID of the current hierarchy level.

CPUID Fn8000_0026_EBX_n[N:0] Extended CPU Topology

Bits	Field Name	Description
31:28	CoreType	Reports a value that may be used to distinguish between cores with different architectural and microarchitectural properties (for example, cores with different performance or power characteristics). Refer to the *Processor Programming Reference Manual* applicable to your product for a list of the available core types. Only valid for LevelType = 1h (Core).
27:24	NativeModelID	Reports a value that may be used to further differentiate implementation specific features. Native mode ID is used in conjunction with the family, model, and stepping identifiers. Refer to the *Processor Programming Reference Manual* applicable to your product for a list of Native Mode IDs. Only valid for LevelType = 1h (Core).
23:16	PwrEfficiencyRanking	Reports a static efficiency ranking between cores of a specific core type, where a lower value indicates comparatively lower power consumption and lower performance. Only valid for LevelType = 1h (Core).
15:0	NumLogProc	Number of logical processors at the current hierarchy level.

CPUID Fn8000_0026_ECX_n[N:0] Extended CPU Topology

Bits	Field Name	Description
31:16	—	Reserved
15:8	LevelType	Encoded hierarchy level type. **Value** — **Description** 0h — Reserved 1h — Core 2h — Complex 3h — Die 4h — Socket FFh-05h — Reserved

Bits	Field Name	Description
7:0	InputEcx	Input ECX[7:0].

CPUID Fn8000_0026_EDX_n[N:0] Extended CPU Topology

Bits	Field Name	Description
31:0	ExtendedApicId	Extended APIC ID of the logical processor.

E.5 Multiple Processor Calculation

Operating systems may use one of two possible methods to calculate the actual number of logical processors per package (NC), and the maximum possible number of logical processors per package (MNLP). The extended method is recommended, but a legacy method is also available.

E.5.1 Legacy Method

The CPUID identification of total number of logical processors per package is derived from information returned by the following fields:

- CPUID Fn0000_0001_EBX[LogicalProcessorCount]
- CPUID Fn0000_0001_EDX[HTT] (Hyper-Threading Technology)
- CPUID Fn8000_0001_ECX[CmpLegacy]
- CPUID Fn8000_0008_ECX[NC]

Table E-5 defines LogicalProcessorCount, HTT, CmpLegacy, and NC as a function of the number of logical processors per package (n).

When HTT = 0, LogicalProcessorCount is reserved and the package contains one logical processor.

When HTT = 1 and CmpLegacy = 1, LogicalProcessorCount represents the number of logical processors per package (n).

Table E-5. LogicalProcessorCount, CmpLegacy, HTT, and NC

Logical Processors per package	CmpLegacy	HTT	LogicalProcessorCount	NC
1	0	0	Reserved	0
2 or more	1	1	n	n-1

The use of CmpLegacy and LogicalProcessorCount for determining the number of logical processors is deprecated. Instead, use NC to determine the number of logical processors per package.

E.5.2 Extended Method (Recommended)

The CPUID identification of total number of logical processors per package is derived from information returned by the CPUID Fn8000_0008_ECX[ApicIdSize[3:0]]. This field indicates the number of least sig-

nificant bits in the CPUID Fn0000_0001_EBX[LocalApicId] that indicates logical processor ID within the package. The size of this field determines the maximum number of logical processors (MNLP) that the package could theoretically support, and not the actual number of logical processors that are implemented or enabled in the package, as indicated by CPUID Fn8000_0008_ECX[NC].

A value of zero for ApicIdSize[3:0] indicates that the legacy method (section E5.1) should be used to derive the maximum number of logical processors:

MNLP = CPUID Fn8000_0008_ECX[NC] + 1.

And for non-zero values of ApicIdSize[3:0]:

MNLP = 2 raised to the power of ApicIdSize[3:0]