CPUID Specification

Publication # **25481**	Revision: **2.18**
Issue Date: **January 2006**	

Advanced Micro Devices 

Trademarks

Table of Contents

Revision History

Date	Rev	Description
January 2006	2.18	• Renamed CPUID Fn8000_0007_EDX[8] from TscPStateInvariant to TscInvariant. • Added CPUID Fn8000_0008_ECX[ApicIdCoreIdSize[3:0]].

September 2005	2.16	• Reformatted document for clarity.
		• Moved the chapter titled “Programming The Processor Name String” to the processor revision guide.
		• Added definition for HTT, CmpLegacy, and LogicalProcessorCount for multi-threading.
		  • See “LogicalProcessorCount, CmpLegacy, HTT, and NC” on page 19.
		  • See CPUID Fn8000_0001_ECX for CmpLegacy.
		  • See CPUID Fn0000_0001_EBX for LogicalProcessorCount.
		  • See CPUID Fn0000_0001_EDX for HTT.
		• Extended BrandID (BrandId[15:12]) definition to CPUID Fn8000_0001_EBX.
		• Added SVM. See CPUID Fn8000_0001_ECX.
		• Added SVM definition to CPUID Fn8000_000A.
		• Added CMPXCHG16B. See CPUID Fn0000_0001_ECX.
		• Added AltMovCr8. See CPUID Fn8000_0001_ECX.
		• Added LahfSahf. See CPUID Fn8000_0001_ECX.
		• Added RDTSCP. See CPUID Fn8000_0001_EDX.

See revision 2.15 for the change history prior to rev 2.16.		

1 Overview

This document specifies the CPUID instruction functions and return values in the EAX, EBX, ECX, and EDX registers, for all processors of family 0Fh or greater. The architectural definition of the CPUID instruction is also documented in the section titled “CPUID” in the *AMD64 Architectural Programmer's Manual Volume 3: General-Purpose and System Instructions*, order #24594.

1.1 Reference Documents

The following documents provide background information:

- *AMD64 Architecture Programmer’s Manual*, volumes 1 to 5. (Order #’s 24592, 24593, 24594, 26568, 26569)
- “AMD64 Architecture Programmer’s Manual: Documentation Updates”. (Order# 33633)
- *BIOS and Kernel Developers Guide* (BKDG) for the specific result value for each of the registers affected by the CPUID instruction for each function. The order number will vary by processor family and sometimes by processor model.
- *AMD Processor Recognition Application Note* for the definition of CPUID for processors less than family 0Fh. (Order# 20734)
- *Processor Revision Guide* for how to program the processor name string.

1.2 Conventions

The following conventions are used in this document:

- The convention for referring to CPUID functions is CPUID FnXXXX_XXXX, where the CPUID function is XXXX_XXXXh. E.g., CPUID Fn0000_0001.
- The convention for referring to CPUID capability fields is CPUID FnXXXX_XXXX_RRR[FieldName], where RRR is the register (EAX, EBX, ECX, EDX) and FieldName is the name of the capability field. E.g., CPUID Fn8000_0001_EDX[SVM].
- Unless otherwise specified, the 1-bit feature fields are encoded as 1=Feature is supported by the processor; 0=Feature is not supported by the processor.
- References to the AMD64 Architecture Programmer’s Manual are abbreviated as APMn, where n specifies the volume, from 1 to 5.
- The 8-bit family of a processor (Family[7:0]) is determined by CPUID Fn0000_0001_EAX[ExtendedFamily,BaseFamily].

1.2.1 Numbering

- **Binary numbers**. Binary numbers are indicated by appending a “b” at the end. E.g., 0110b.
- **Decimal numbers**. Unless specified otherwise, all numbers are decimal. Note: this rule does not apply to the register mnemonics; register mnemonics all utilize hexidecimal numbering.
- **Hexidecimal numbers**. Hexidecimal numbers are indicated by appending an “h” to the end. E.g., 45f8h.
- **Underscores in numbers**. Underscores are used to break up numbers to make them more readable. They do not imply any operation, e.g., 0110_1100b.

1.2.2 Arithmetic And Logical Operators

- {} Curly brackets are used to indicate a group of bits that are concatenated together. Each set of bits is separated by a comma, e.g., {10b,01b,1b}=10011b.
- | Logical OR operator.
- & Logical AND operator.

1.3 Definitions

The following defines are used in this document:

- **APMn**. Abbreviation for the document titled “AMD64 Architecture Programmer’s Manual”, where n specifies the volume, from 1 to 5. (Order #’s 24592, 24593, 24594, 26568, 26569)
- **APMU**. Abbreviation for the document titled “AMD64 Architecture Programmer’s Manual: Documentation Updates”. (Order# 33633)
- **BKDG**. BIOS and Kernel Developer’s Guide.
- **CMP**. Chip multi-processing. Refers to processors that include multiple CPU cores.
- **CPU Core**. Executes x86 instructions and contains a set of MSR’s and APIC registers.
- **DW or Doubleword**. Double word. A 32-bit value.
- **Family**. An 8-bit value that identifies one or more processors as belonging to a group that possess some common definition for software or hardware purposes. See CPUID Fn0000_0001_EAX.
- **GB or Gbyte**. Gigabyte; 1,024 Mbytes.
- **HTC**. Hardware thermal control.
- **KB or Kbyte**. Kilobyte; 1024 Bytes.
- **MB or Mbyte**. Megabyte; 1024 Kbytes.
- **Model**. Model specifies one instance of a processor family. See CPUID Fn0000_0001_EAX.
- **MSR**. Model specific register. The CPU includes several MSRs for general configuration and control.
- **NB**. Northbridge. The transaction routing block of the processor.
- **Processor**. A single package that contains one or more CPU cores.
- **QW or Quadword**. Quad word. A 64-bit value.
- **OW or Octword**. Eight word. A 128-bit value.
- **RAZ**. Read as zero. Writes are ignored.
- **Reserved**. Field is reserved for future use. Software may not depend on the state of reserved fields.
- **STC**. Software thermal control.
- **SVM**. Secure virtual machine.
- **Thread**. One architectural context for instruction execution.

1.4 CPUID Function Select

Processor feature capabilities and configuration information are provided through the CPUID instruction. Different information is accessed by (1) setting EAX to the function number, (2) executing the CPUID instruction, and (3) reading the results in EAX, EBX, ECX, and EDX. In the following sections, the phrase *CPUID function X* or *CPUID FnX* refers to the CPUID instruction when EAX is preloaded with X.

1.5 Standard, Extended, and Undefined Functions

The CPUID instruction supports two sets or ranges of functions, 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.

Functions that are neither standard or extended are undefined and should not be relied upon.

2 CPUID Function Specification

This chapter defines each of the supported CPUID functions, both standard and extended.

CPUID Fn0000_0000 Processor Vendor and Largest Standard Function Number

The values returned in EBX, ECX, and EDX for CPUID Fn0000_0000 are the same values returned in EBX, ECX, and EDX for CPUID Fn8000_0000.

Register	Bits	Description
EAX	31:0	The largest CPUID standard-function input value supported by the processor implementation. See “Standard, Extended, and Undefined Functions” on page 8.
EBX, ECX, EDX	31:0	The 12 8-bit ASCII character codes to create the string “AuthenticAMD”. EBX=6874_7541h “h t u A”, ECX=444D_4163h “D M A c”, EDX=6974_6E65h “i t n e”.

CPUID Fn0000_0001_EAX Family, Model, Stepping Identifiers

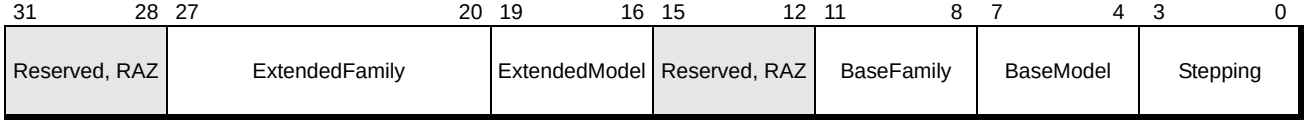
The value returned in EAX is the processor identification signature and is identical for CPUID Fn0000_0001 and CPUID Fn8000_0001. This function is an identical copy of CPUID Fn8000_0001_EAX. Reserved fields should be masked before using the value of EAX for processor identification purposes. Three values are used by software to identify a processor: Family, Model, and Stepping.

The processor family is a way of identifying one or more processors as belonging to a group that possess some common definition for software or hardware purposes. Model specifies one instance of a processor family. 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.

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

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

Stepping is analogous to a revision number.



Bits	Description
31:28	Reserved.
27:20	**ExtendedFamily**: processor extended family. See above for definition of Family[7:0].
19:16	**ExtendedModel**: 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 (revision) for a specific model.

CPUID Fn0000_0001_EBX LocalApicId, LogicalProcessorCount, CLFlush, 8BitBrandId

This function returns miscellaneous information regarding the processor brand, the number of logical threads per processor socket, the CLFLUSH instruction and APIC.

Bits	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 CPU core within a processor and other bits represent the processor ID. See the APIC ID Register in the processor BKDG for details.
23:16	**LogicalProcessorCount**: If CPUID Fn0000_0001_EDX[HTT]=1 then LogicalProcessorCount is the number of threads per CPU core times the number of CPU cores per processor. AMD currently does not support more than 1 thread per CPU core. If CPUID Fn0000_0001_EDX[HTT]=0 then LogicalProcessorCount is reserved. See “LogicalProcessorCount, CmpLegacy, HTT, and NC” on page 19.
15:8	**CLFlush**: CLFLUSH size. Specifies the size of a cache line in quadwords flushed by the CLFLUSH instruction. See the APM3 section titled “CLFLUSH”.
7:0	**8BitBrandId**: 8 bit brand ID. This field, in conjunction with CPUID Fn8000_0001_EBX[BrandId], is used by BIOS to generate the processor name string. See the Processor Revision Guide for how to program the processor name string.

CPUID Fn0000_0001_ECX Feature Identifiers

This function contains the following miscellaneous feature identifiers.

Bits	Description
31	RAZ.
30:14	Reserved.
13	**CMPXCHG16B**: CMPXCHG16B instruction. See the APM3 section titled “CMPXCHG16B”.

Bits	Description
12:1	Reserved.
0	**SSE3**: SSE3 extensions. See the APM3 appendix titled “Instruction Subsets and CPUID Feature Sets” for the list of instructions covered by the SSE3 feature bit. See APM4 for the definition of the SSE3 instructions.

CPUID Fn0000_0001_EDX Feature Identifiers

This function contains the following miscellaneous feature identifiers.

Bits	Description
31:29	Reserved.
28	**HTT**: Hyper-Threading Technology. Indicates either that there is more than 1 thread per CPU core or more than 1 CPU core per processor. AMD currently does not support more than 1 thread per CPU core. See “LogicalProcessorCount, CmpLegacy, HTT, and NC” on page 19.
27	Reserved.
26	**SSE2**: SSE2 extensions. See the APM3 appendix section titled “CPUID Feature Sets”.
25	**SSE**: SSE extensions. See the APM3 appendix titled “CPUID Feature Sets” and the APM1 chapter titled “64-Bit Media Programming”.
24	**FXSR**: FXSAVE and FXRSTOR instructions. See the APM4 sections titled “FXSAVE” and “FXRSTOR”.
23	**MMX**: MMX™ instructions. See the APM3 appendix section titled “CPUID Feature Sets” and the APM1 chapter titled “128-Bit Media and Scientific Programming”.
22:20	Reserved.
19	**CLFSH**: CLFLUSH instruction. See the APM3 section titled “CLFLUSH”.
18	Reserved.
17	**PSE36**: page-size extensions. The PDE[20:13] supplies physical address [39:32]. See the APM2 chapter titled “Page Translation and Protection”.
16	**PAT**: page attribute table. PCD, PWT, and PATi are used to alter memory type. See the APM2 chapter titled “Page-Attribute Table Mechanism”.
15	**CMOV**: conditional move instructions, CMOV, FCMOV. See the APM3 sections titled “CMOV”, “FCMOV“.
14	**MCA**: machine check architecture, MCG_CAP. See the APM2 chapter titled “Machine Check Mechanism”.
13	**PGE**: page global extension, CR4.PGE. See the APM2 chapter titled “Page Translation and Protection”.
12	**MTRR**: memory-type range registers. MTRRcap supported. See the APM2 chapter titled “Page Translation and Protection”.
11	**SysEnterSysExit**: SYSENTER and SYSEXIT instructions. See the APM3 sections titled “SYSENTER”, “SYSEXIT“.
10	Reserved.

Bits	Description
9	**APIC**. advanced programmable interrupt controller (APIC) exists and is enabled. See the APM2 chapter titled “Exceptions and Interrupts”.
8	**CMPXCHG8B**: CMPXCHG8B instruction. See the APM3 section titled “CMPXCHG8B”.
7	**MCE**: machine check exception, CR4.MCE. See the APM2 chapter titled “Machine Check Mechanism”.
6	**PAE**: physical-address extensions (PAE), support for physical addresses >=32b. Number of physical address bits above 32b is implementation specific. See the APM2 chapter titled “Page Translation and Protection”.
5	**MSR**: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions. See the APM2 section titled “Model Specific Registers”.
4	**TSC**: time stamp counter. RDTSC and RDTSCP instruction support. See the APM2 chapter titled “Debug and Performance Resources”.
3	**PSE**: page-size extensions (4 MB pages). See the APM2 chapter titled “Page Translation and Protection”.
2	**DE**: debugging extensions, IO breakpoints, CR4.DE. See the APM2 chapter titled “Debug and Performance Resources”.
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 the APM2 chapter titled “System Resources”.
0	**FPU**: x87 floating point unit on-chip. See the APM1 chapter titled “x87 Floating Point Programming”.

CPUID Fn8000_0000 Processor Vendor and Largest Extended Function Number

The values returned in EBX, ECX, and EDX for CPUID Fn8000_0000 are the same values returned in EBX, ECX, and EDX for CPUID Fn0000_0000.

Register	Bits	Description
EAX	31:0	The largest CPUID extended-function input value supported by the processor implementation. See “Standard, Extended, and Undefined Functions” on page 8.
EBX, ECX, EDX	31:0	The 12 8-bit ASCII character codes to create the string “AuthenticAMD”. EBX=6874_7541h “h t u A”, ECX=444D_4163h “D M A c”, EDX=6974_6E65h “i t n e”.

CPUID Fn8000_0001_EAX AMD Family, Model, Stepping

Same as CPUID Fn0000_0001_EAX.

CPUID Fn8000_0001_EBX BrandId Identifier

This function returns the extended brand ID field.

Bits	Description
31:16	Reserved.
15:0	**BrandId**: brand ID. This field, in conjunction with CPUID Fn0000_0001_EBX[8BitBrandId], is used by BIOS to generate the processor name string. See the Processor Revision Guide for how to program the processor name string.

CPUID Fn8000_0001_ECX Feature Identifiers

This function contains the following miscellaneous feature identifiers.

Bits	Description
31:5	Reserved.
4	**AltMovCr8**: LOCK MOV CR0 means MOV CR8. See the APM3 section titled “MOV(CRn)”.
3	Reserved.
2	**SVM**: secure virtual machine feature. See the APM2 chapter titled “Secure Virtual Machine”.
1	**CmpLegacy**: core multi-processing legacy mode. See “LogicalProcessorCount, CmpLegacy, HTT, and NC” on page 19.
0	**LahfSahf**: LAHF and SAHF instruction support in 64-bit mode. See the APM3 sections titled “LAHF” and “SAHF”.

CPUID Fn8000_0001_EDX Feature Identifiers

This function contains the following miscellaneous feature identifiers.

Bits	Description
31	**3DNow**: 3DNow!™ instructions. See the APM3 appendix section titled “CPUID Feature Sets”.
30	**3DNowExt**: AMD extensions to 3DNow!™ instructions. See the APM3 appendix section titled “CPUID Feature Sets”.
29	**LM**: long mode. See the APM2 section titled “Processor Initialization and Long-Mode Activation”.
28	Reserved.
27	**RDTSCP**: RDTSCP instruction. See the APM3 section titled “RDTSCP”.
26	Reserved.
25	**FFXSR**: FXSAVE and FXRSTOR instruction optimizations. See the APM4 sections titled “FXSAVE” and “FXRSTOR”.
24	**FXSR**: FXSAVE and FXRSTOR instructions. Same as CPUID Fn0000_0001_EDX[FXSR].

Bits	Description
23	**MMX**: MMX™ instructions. Same as CPUID Fn0000_0001_EDX[MMX].
22	**MmxExt**: AMD extensions to MMX™ instructions. See the APM3 appendix section titled “CPUID Feature Sets” and the APM1 chapter titled “128-Bit Media and Scientific Programming”.
21	Reserved.
20	**NX**: no-execute page protection. See the APM2 chapter titled “Page Translation and Protection”.
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 the APM3 sections titled “SYSCALL”, “SYSRET“.
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].
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].

CPUID Fn8000_000[4:2] Processor Name String Identifier

The 3 extended functions from Fn8000_0002 to Fn8000_0004 are initialized to and return a null terminated ASCII string up to 48 characters in length corresponding to the processor name. (The 48 character maximum includes the null character.) The 48 character sequence is ordered first to last 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 processor name string must be programmed by the BIOS during system initialization. See the Processor Revision Guide for information about how to program and display the processor name string.

CPUID Fn8000_0005 L1 Cache and TLB Identifiers

This function contains the processor's first level cache and TLB characteristics for each CPU core.

The *associativity* fields are encoded as follows:

00h: Reserved.
01h: Direct mapped.
02h-FEh: Associativity. (E.g., 04h= 4-way associative.)
FFh: Fully associative.

Register	Bits	Description
EAX	31:24	**L1DTlb2and4MAssoc**. Data TLB associativity for 2 MB and 4 MB pages.
EAX	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.
EAX	15:8	**L1ITlb2and4MAssoc**. Instruction TLB associativity for 2 MB and 4 MB pages.
EAX	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.
EBX	31:24	**L1DTlb4KAssoc**. Data TLB associativity for 4 KB pages.
EBX	23:16	**L1DTlb4KSize**. Data TLB number of entries for 4 KB pages.
EBX	15:8	**L1ITlb4KAssoc**. Instruction TLB associativity for 4 KB pages.
EBX	7:0	**L1ITlb4KSize**. Instruction TLB number of entries for 4 KB pages.
ECX	31:24	**L1DcSize**. L1 data cache size in KB.
ECX	23:16	**L1DcAssoc**. L1 data cache associativity.
ECX	15:8	**L1DcLinesPerTag**. L1 data cache lines per tag.
ECX	7:0	**L1DcLineSize**. L1 data cache line size in bytes.
EDX	31:24	**L1IcSize**. L1 instruction cache size KB.
EDX	23:16	**L1IcAssoc**. L1 instruction cache associativity.
EDX	15:8	**L1IcLinesPerTag**. L1 instruction cache lines per tag.
EDX	7:0	**L1IcLineSize**. L1 instruction cache line size in bytes.

CPUID Fn8000_0006 L2 Cache and L2 TLB Identifiers

This function contains the processor's second level cache and TLB characteristics for each CPU core.

The presence of a unified L2 TLB is indicated by a value of 0000h in the upper 16 bits of the EAX and EBX registers. The unified L2 TLB information is contained in the lower 16 bits of these registers.

The *associativity* fields are encoded as follows:

Table 1: L2 Cache and TLB Associativity Field Definition

Associativity [3:0]	Definition
0h	The L2 cache or TLB is disabled.
1h	Direct mapped.
2h	2-way associative.
4h	4-way associative.
6h	8-way associative.
8h	16-way associative.
Fh	Fully associative.
All other encodings are reserved.	

The L3 cache size field is defined as follows:

Register	Bits	Description
EAX	31:28	**L2DTlb2and4MAssoc**. L2 data TLB associativity for 2 MB and 4 MB pages. (See Table 1)
EAX	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.
EAX	15:12	**L2ITlb2and4MAssoc**. L2 instruction TLB associativity for 2 MB and 4 MB pages. (See Table 1)
EAX	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.
EBX	31:28	**L2DTlb4KAssoc**. L2 data TLB associativity for 4 KB pages. (See Table 1)
EBX	27:16	**L2DTlb4KSize**. L2 data TLB number of entries for 4 KB pages.
EBX	15:12	**L2ITlb4KAssoc**. L2 instruction TLB associativity for 4 KB pages. (See Table 1)
EBX	11:0	**L2ITlb4KSize**. L2 instruction TLB number of entries for 4 KB pages.
ECX	31:16	**L2Size**. L2 cache size in KB.
ECX	15:12	**L2Assoc**. L2 cache associativity. (See Table 1)
ECX	11:8	**L2LinesPerTag**. L2 cache lines per tag.

Register	Bits	Description
ECX	7:0	**L2LineSize**. L2 cache line size in bytes.
EDX	31:0	Reserved.

CPUID Fn8000_0007 Advanced Power Management Information

This function provides advanced power management feature identifiers. Refer to the processor BKDG for a detailed description of the definition of each power management feature and whether that feature is supported.

Register	Bits	Description
EAX	31:0	Reserved.
EBX	31:0	Reserved.
ECX	31:0	Reserved.
EDX	31:9	Reserved.
EDX	8	**TscInvariant**: 1=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.
EDX	7:6	Reserved.
EDX	5	**STC**: software thermal control is supported.
EDX	4	**TM**: hardware thermal control is supported.
EDX	3	**TTP**: THERMTRIP is supported.
EDX	2	**VID**: voltage ID control is supported.
EDX	1	**FID**: frequency ID control.
EDX	0	**TS**: temperature sensor.

CPUID Fn8000_0008 Long Mode Address Size Identifiers

This function returns information about the number of CPU cores and the maximum physical and linear address width (in bits) supported by the processor. The 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 the APM2 section titled “Processor Initialization and Long-Mode Activation”.

Register	Bits	Description
EAX	31:16	Reserved.
EAX	15:8	**LinAddrSize**: maximum linear byte address size in bits.
EAX	7:0	**PhysAddrSize**: maximum physical byte address size in bits.
EBX	31:0	Reserved.
ECX	31:16	Reserved.

Register	Bits	Description

ECX	15:12	**ApicIdCoreIdSize[3:0]**. Indicates the number of least significant bits in the Initial APIC ID that indicate CPU core ID within a processor. A value of zero indicates to use legacy methods to derive maximum-number-of-cores. The size of this field determines the maximum number of cores (MNC) that the processor could theoretically support, not the actual number of cores that are actually implemented or enabled on the processor, as indicated by CPUID Fn8000_0008_ECX[NC].
		
		if (ApicIdCoreIdSize[3:0] == 0){
		// Used by legacy dual-core/single-core processors
		MNC = CPUID Fn8000_0008_ECX[NC] + 1;
		} else {
		// use ApicIdCoreIdSize[3:0] field
		MNC = (2 ^ ApicIdCoreIdSize[3:0]);
		}

ECX	11:8	Reserved.
ECX	7:0	**NC: number of CPU cores - 1**. The number of CPU cores per processor is NC+1. See "LogicalProcessorCount, CmpLegacy, HTT, and NC" on page 19.
EDX	31:0	Reserved.

CPUID Fn8000_0009 Reserved

This function is reserved.

CPUID Fn8000_000A SVM Revision and Feature Identification

This function returns SVM revision and feature information. See the APM2 chapter titled "Secure Virtual Machine". If CPUID Fn8000_0001_ECX[SVM]=0 then CPUID Fn8000_000A is reserved.

Register	Bits	Description
EAX	31:8	Reserved.
EAX	7:0	**SvmRev**: SVM revision.
EBX	31:0	**NASID**: number of address space identifiers (ASID).
ECX	31:0	Reserved.
EDX	31:0	Reserved.

CPUID Fn8000_00[19:0B] Reserved

These functions are reserved.

3 LogicalProcessorCount, CmpLegacy, HTT, and NC

The CPUID identification of total number of CPU cores per processor (c) and threads per processor (t) 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] (number of CPU cores - 1)

Table 3 defines LogicalProcessorCount, HTT, CmpLegacy, and NC as a function of the number of CPU cores per processor (c) and the number of threads per CPU core (t). When HTT=0, LogicalProcessorCount is reserved and the processor contains one CPU core and that one CPU core is single-threaded. When HTT=1 and CmpLegacy=1, LogicalProcessorCount represents the number of CPU cores per processor (c), where each CPU core is single-threaded. When HTT=1 and CmpLegacy=0, LogicalProcessorCount represents the number of total threads for the processor, which is the multiplication of c and t. NC always represents the number of CPU cores per processor minus 1.

Table 3: LogicalProcessorCount, CmpLegacy, HTT, and NC

CPU Cores per Processor (c)	Threads per CPU Core (t)	CmpLegacy	HTT	LogicalProcessorCount	NC
1	1	0	0	Reserved	0
2 or more	1	1	1	c	c-1
1	2 or more	0	1	c*t	0
2 or more	2 or more	0	1	c*t	c-1

AMD currently does not support multiple threads per CPU core. E.g. The number of threads per CPU core is always 1.

The use of CmpLegacy and LogicalProcessorCount for the determination of the number of CPU cores is deprecated. Instead, use NC to determine the number of CPU cores.