ddcpuid User Manual

Exploring your x86 micro-processor

Second Edition

Table of Contents

Preface

This document was written to better understand x86 technologies and features that are referenced from ddcpuid and x86 reference manuals.

It was written by dd86k for the ddcpuid project (https://github.com/dd86k/ddcpuid).

Disclaimer

INFORMATION IN THIS DOCUMENT MAY BE INACCURATE OR MISSING. I AM NOT HELD RESPONSIBLE FOR INACCURATE INFORMATION IN THIS DOCUMENT. PLEASE SEND AN EMAIL OR OPEN AN ISSUE ON GITHUB IF YOU FIND INACCURATE OR MISSING INFORMATION.

1 Introduction

ddcpuid is a simple x86 CPUID information tool that works best with Intel and AMD microprocessors. It solely relies on the CPUID instruction in a Ring 3 context (CPL).

By default, ddcpuid shows basic information, like the CPU vendor string, processor brand string, instruction extensions, cache, and processor technologies.

In advanced mode, it shows most micro-processor features, like debugging features.

1.1 History

The project started in June 26, 2016, in D. It was created as a simple tool to explore x86 technologies and features out of my interest for system programming.

On February 1st, 2017, ddcpuid got 64-bit compilation support.

On December 11, 2017, ddcpuid was rewritten to be compiled with D’s betterC feature.

On June 15, 2018, cache information was added.

1.2 References

This project is based on information from the *Intel® 64 and IA-32 Architectures Software Developer’s Manual* (May 2018, combined volumes) and *AMD64 Architecture Programmer’s Manual Volume 3: General-Purpose and System Instructions* (May 2018) manuals.

2 Command Line Interface

The help screen can be accessed via `-h` or –`help`.

Some options can be combined: `-ro` is the same as `-r -o`.

2.1 Options

`-d`

Advanced mode. Includes advanced information that would be useful for an engineer

`-r`

Show raw CPUID data in a table

`-o`

Override CPUID leaves to 20H and 8000_0020H respectively, only useful with `-r`

`-v`, `--version`

Show version screen and exit

`-h`, `--help`

Show help screen and exit

3 Normal Mode

By default, ddcpuid shows basic information, which includes:

- Vendor string;
- Brand string (model);
- Identifier;
- Extensions (e.g. MMX);
- Cache information;
- And featured technologies.

3.1 Identifier Calculation

On Intel processors, processor identifiers are calculated according to the reference manual (Volume 2A, page 3-206 (778), Figure 3-6):

```
if (BaseFamily != 0)
     Family = BaseFamily;
else
     Family = ExtendedFamily + BaseFamily;

if (BaseFamily == 6 || BaseFamily == 0)
     Model = (ExtendedModel << 4) + BaseModel;
else
     Model = BaseModel;
```

On AMD processors, processor identifiers are calculated according to the reference manual (Volume 3, page 603 (639), §E.3.2, §CPUID Fn0000_0001_EAX):

```
if (BaseFamily < 0xF) {
     Family = BaseFamily;
     Model = BaseModel;
} else {
     Family = ExtendedFamily + BaseFamily;
     Model = (ExtendedModel << 4) + BaseModel;
}
```

3.2 Supported Processor Technologies

3.2.1 Intel Processors

- Enhanced SpeedStep(R) Technology
- TurboBoost

3.2.2 AMD Processors

- Core Performance Boost

4 Advanced Mode

Advanced mode, accessed via `-d`, is the advanced mode which additionally includes:

- Highest processor leaf and extended leaf;
- Processor type;
- FPU features;
- ACPI and APIC features;
- Virtualization features;
- Memory features;
- Debugging features;
- Brand index
- And miscellaneous features.

Features are explained in Processor Features.

4.1 Advanced Identifier Display

In advanced mode, ddcpuid shows the processor identifier in a different way:

```
Family [BaseFamily:ExendedFamily] Model [BaseModel:ExtendedModel]
Stepping
```

All values are hexadecimal numbers.

Identifier calculation is explained in Identifier Calculation.

4.2 Processor Type

Processor type is a deprecated feature in Intel processors and may show within the following values:

- Original OEM Processor
- Intel OverDrive Processor
- Dual Processor
- Reserved (Intel)

All recent Intel processors show Original OEM Processor.

5 Processor Extensions

5.1 MMX

Year introduced	1997
CPUID bit	01h.EDX[23]
Instructions	• MOVD • MOVQ • PACKSSWB • PACKSSDW • PACKUSWB • PUNPCKHBW • PUNPCKHWD • PUNPCKHDQ • PUNPCKLBW • PUNPCKLWD • PUNPCKLDQ • PADDB • PADDW • PADDD • PADDSB • PADDSW • PADDUSB • PADDUSW • PSUBB • PSUBW • PSUBD • PSUBSB • PSUBSW • PSUBUSB • PSUBUSW • PMULHW • PMULLW • PMADDWD • PCMPEQB • PCMPEQW • PCMPEQD • PCMPGTB • PCMPGTW • PCMPGTD • PAND • PANDN • POR • PXOR • PSLLW • PSLLD • PSLLQ • PSRLW • PSRLD • PSRLQ • PSRAW • PSRAD • EMMS

The MMX extension introduced SIMD (single instruction, multiple data) instructions with new registers: MM0 to MM7.

5.2 Extended MMX

Year introduced	1999
CPUID bit	(AMD) 8000_0001h.EDX[22]
Instructions	• PADDSIW • PAVEB • PDISTIB • PMACHRIW • PMAGW • PMULHRW • PMULHRIW • PMVZB • PMVNZB • PMVLZB • PMVGEZB • PSUBSIW

The Extended MMX extension, introduced in AMD Athlon processors, are added SIMD instructions and are distinguished from SSE since AMD did not include some SSE instructions in their Athlon processors.

5.3 3DNow! (3DNow)

Year introduced	1998
CPUID bit	(AMD) 8000_0001h.EDX[31]
Instructions	• PAVGUSB • PMULHRW • PI2FD • PF2ID • PFMAX • PFMIN • PFCMPEQ • PFCMPGE • PFCMPGT • PFADD • PFACC • PFSUB • PFSUBR • PFMUL • PFRCP • PFRSQRT • PFRCPIT1 • PFRCPIT2 • PFRSQIT1

The 3DNow! extension was added by AMD for vector processing, useful for video processing and three-dimensional rendering.

In 2010, AMD deprecated these instructions.

5.4 Extended 3DNow! (3DNowExt)

Year introduced	1999
CPUID bit	(AMD) 8000_0001h.EDX[30]
Instructions	• PF2IW • PI2FW • PSWAPD • PFNACC • PFPNACC

The Extended 3DNow! instruction set is an extension to 3DNow!, added in their Athlon processors.

5.5 Streaming SIMD Extentions (SSE)

Year introduced	1999
CPUID bit	01h.EDX[25]

Instructions	• ADDPS • ADDSS • SUBPS • SUBSS • MULPS • MULSS • DIVPS • DIVSS • RCPPS • RCPSS • SQRTPS • SQRTSS • RSQRTPS • RSQRTSS • MAXPS • MAXSS • MINPS • MINSS • PAVGB • PAVGW • PSADBW • PEXTRW • PINSRW • PMAXSW • PMAXUB • PMINSW • PMINUB • PMOVMSKB • PMULHUW • PSHUFW • ANDNPS • ANDPS • ORPS • XORPS • CMPXXPS • CMPXXSS • COMISS • UCOMISS • EQ • LT • LE • NE • NLT • NLE • ORD • UNORD • CVTPI2PS • CVTPS2PI • CVTSI2SS • CVTSS2SI • CVTTPS2PI • CVTTSS2SI • FXRSTOR • FXSAVE • LDMXCSR • STMXCSR • MOVAPS • MOVHLPS • MOVLHPS • MOVHPS • MOVLPS • MOVMSKPS • MOVSS • MOVUPS • MASKMOVQ • MOVNTPS • MOVNTQ • SHUFPS • UNPCKHPS • UNPCKLPS • PREFETCHT0 • PREFETCHT1 • PREFETCHT2 • PREFETCHNTA • SFENCE

The Streaming SIMD Extentions were added in the Intel Pentium III and AMD AthlonXP processors. It adds registers XMM0 to XMM7 and MXCSR (status register).

5.6 Streaming SIMD Extentions 2 (SSE2)

Year introduced	2000 (Intel), 2003 (AMD)
CPUID bit	01h.EDX[26]

Instructions	• ADDPD • ADDSD • SUBPD • SUBSD • MULPD • MULSD • DIVPD • DIVSD • MAXPD • MAXSD • MINPD • MINSD • PADDB • PADDW • PADDD • PADDQ • PADDSB • PADDSW • PADDUSB • PADDUSW • PSUBB • PSUBW • PSUBD • PSUBQ • PSUBSB • PSUBSW • PSUBUSB • PSUBUSW • PMADDWD • PMULHW • PMULLW • PMULUDQ • RCPPS • RCPSS • SQRTPD • SQRTSD • ANDNPD • ANDNPS • ANDPD • PAND • PANDN • POR • PSLLDQ • PSLLQ • PSLLD • PSLLW • PSRAD • PSRAW • PSRLDQ • PSRLQ • PSRLD • PSRLW • PXOR • ORPD • XORPD • CMPPD • CMPSD • COMISD • UCOMISD • PCMPXXB • PCMPXXW • PCMPXXD • EQ • LT • LE • NE • NLT • NLE • ORD • UNORD • CVTDQ2PD • CVTDQ2PS • CVTPD2PI • CVTPD2DQ • CVTPD2PS • CVTPI2PD • CVTPS2DQ • CVTPS2PD • CVTSD2SI • CVTSD2SS • CVTSI2SD • CVTSI2SS • CVTSS2SD • CVTSS2SI • CVTTPD2PI • CVTTPD2DQ • CVTTPS2DQ • CVTTPS2PI • CVTTSD2SI • CVTTSS2SI • MOVQ • MOVSD • MOVAPD • MOVUPD • MOVHPD • MOVLPD • MOVDQ2Q • MOVQ2DQ • MOVNTPD • MOVNTDQ • MOVNTI • MASKMOVDQU • PMOVMSKB • PSHUFD • PSHUFHW • PSHUFLW • UNPCKHPD • UNPCKLPD • PUNPCKHBW • PUNPCKHWD • PUNPCKHDQ • PUNPCKHQDQ • PUNPCKLBW • PUNPCKLWD • PUNPCKLDQ • PUNPCKLQDQ • PACKSSDW • PACKSSWB • PACKUSWB • CLFLUSH • LFENCE • MFENCE • PAUSE

First introduced in the Intel Pentium 4 processor, the second extension to SSE features the CLFLUSH instruction, and a few more instructions for cache control.

5.7 Streaming SIMD Extentions 3 (SSE3)

Year introduced	2004 (Intel), 2005 (AMD)
CPUID bit	01h.ECX[0]
Instructions	• ADDSUBPD • ADDSUBPS • HADDPD • HADDPS • HSUBPD • HSUBPS • LDDQU • MOVDDUP • MOVSHDUP • MOVSLDUP • FISTTP • MONITOR • MWAIT

Introduced in the Intel Pentium 4 Prescott family, SSE3 is an extension to the SSE family of instructions.

AMD did not implement MONITOR and MWAIT (for process control), since those instructions are only useful with HyperThreading Technology.

5.8 Supplemental Streaming SIMD Extentions 3 (SSSE3)

Year introduced	2006
CPUID bit	01h.ECX[9]
Instructions	• PSIGND • PSIGNW • PSIGNB • PHADDD • PHADDW • PHADDSW • PHSUBD • PHSUBW • PHSUBSW • PMADDUBSW • PABSD • PABSW • PABSB • PMULHRSW • PSHUFB • PALIGNR

Introduced in the Core 2 architecture, SSSE3 is an extension to SSE3.

5.9 Streaming SIMD Extentions 4 (SSE4)

SSE4 is a group of extensions coming in three flavors: SSE4.1, SEE4.2, and SSE4a. All announced in 2006, but implemented in 2007, SSE4 extensions are both supported in recent Intel and AMD processors.

5.9.1 Streaming SIMD Extentions 4.1 (SSE41)

Year introduced	2007
CPUID bit	01h.ECX[15]

Instructions	• MPSADBW • PHMINPOSUW • PMULDQ • PMULLD • DPPS • DPPD • BLENDPS • BLENDPD • BLENDVPS • BLENDVPD • PBLENDVB • PBLENDW • PMINSB • PMAXSB • PMINUW • PMAXUW • PMINUD • PMAXUD • PMINSD • PMAXSD • ROUNDPS • ROUNDSS • ROUNDPD • ROUNDSD • INSERTPS • PINSRB • PINSRD • PINSRQ • EXTRACTPS • PEXTRB • PEXTRW • PEXTRD • PEXTRQ • PMOVSXBW • PMOVZXBW • PMOVSXBD • PMOVZXBD • PMOVSXBQ • PMOVZXBQ • PMOVXSWD • PMOVZXWD • PMOVSXWQ • PMOVZXWQ • PMOVSXDQ • PMOVZXDQ • PTEST • PCMPEQQ • PACKUSDW • MOVNTDQA

Introduced in Intel's Core 2 Penryn architecture.

5.9.2 Streaming SIMD Extentions 4.2 (SSE42)

Year introduced	2008
CPUID bit	01h.ECX[20]
Instructions	• CRC32 • PCMPESTRI • PCMPESTRM • PCMPISTRI • PCMPISTRM • PCMPGTQ • POPCNT

Introduced in Intel's Core (1st generation) Nehalem architecture. It was designed to speed up XML parsing[1] and includes the CRC32 instruction.

5.9.3 Streaming SIMD Extentions 4a (SSE4a)

Year introduced	(AMD) 2007
CPUID bit	8000_0001h.ECX[6]
	• LZCNT • POPCNT • EXTRQ • INSERQ • MOVNTSD • MOVNTSS

Introduced in AMD's Barcelona (Family 10h) architecture.

[1] https://software.intel.com/en-us/articles/xml-parsing-accelerator-with-intel-streaming-simd-extensions-4-intel-sse4/

5.9.4 POPCNT and LZCNT

POPCNT (Intel) and LZCNT (AMD) instructions were added around the time the SSE4.2 and SSE4a were introduced, including them with their extensions.

POPCNT, an instruction that returns the count of number of bits set, can be checked via CPUID.01h.ECX[23].

LZCNT, an instruction that counts the number of leading zero bits, can be checked via CPUID.8000_0001h.ECX[5].

5.10 Advanced Vector Extension (AVX)

Year introduced	2011
CPUID bit	01h.ECX[28]
Instructions	• VBROADCASTSS • VBROADCASTSD • VBROADCASTF128 • VINSERTF128 • VEXTRACTF128 • VMASKMOVPS • VMASKMOVPD • VPERMILPS • VPERMILPD • VPERM2F128 • VZEROALL • VZEROUPPER See other instructions with the VEX prefix.

Introduced in Intel's Sandy Bridge architecture, the AVX extension includes YMM0 through YMM15 256-bit registers, and 12 new instructions for 256-bit processing.

5.11 Advanced Vector Extension 2 (AVX2)

Year introduced	2013
CPUID bit	07h.EBX[5]
Instructions	• VBROADCASTSS • VBROADCASTSD • VPBROADCASTB • VPBROADCASTW • VPBROADCASTD • VPBROADCASTQ • VBROADCASTI128 • VINSERTI128 • VEXTRACTI128 • VGATHERDPD • VGATHERQPD • VGATHERDPS • VGATHERQPS • VPGATHERDD • VPGATHERDQ • VPGATHERQD • VPGATHERQQ • VPMASKMOVD • VPMASKMOVQ • VPERMPS • VPERMD • VPERMPD • VPERMQ • VPERM2I128 • VPBLENDD • VPSLLVD • VPSLLVQ • VPSRLVD • VPSRLVQ • VPSRAVD

Introduced in Intel's Haswell architecture, the AVX2 extension expends most vector integer SSE and AVX instructions to 256 bits.

5.12 Advanced Vector Extension, 512-bit (AVX512F)

Year introduced	(Intel) 2015
CPUID bit	07h.EBX[16]
Instructions	See instructions with EVEX prefix.

Introduced in Intel's Landing Knights (Xeon Phi) and Skylake architectures, the AVX-512 extension may features a few more instructions:

- Exponential and Reciprocal instructions (**AVX512ER**);
- Conflict Detection instructions (**AVX512CD**);
- New Prefetch (**AVX512PF**) instructions;
- DWORD and QWORD extensions (**AVX512DQ**, CPUID.07h.EBX[17]);
- BYTE and WORD extensions (**AVX512BW**, CPUID.07h.EBX[30]);
- And Vector Length extensions (**AVX512VL**).
- Integer Fused Multiply-Add with 52-bits of precision instructions (**AVX512_IFMA**);
- Vector Byte Manipulation Instructions (**AVX512_VBMI**, on-top of AVX512BW);

The AVX-512 extension also adds ZMM0 to ZMM31 512-bit registers, and extends YMM and XMM register counts to 31 each.

In order to have any AVX-512 extensions, the AVX512F (foundation) feature is required to be present within the processor.

5.12.1 AVX512ER

CPUID bit	(Intel) 07h.EBX[27]

Includes AVX-512 Exponential and Reciprocal instructions. If set, VEXP2PD, VEXP2PS, VRCP28xx, and VRSQRT28xx instructions are supported.

5.12.2 AVX512PF

CPUID bit	(Intel) 07h.EBX[26]

Includes AVX-512 Prefetch instructions. If set, VGATHERPF0xxx, VGATHERPF1xxx, VSCATTERPF0xxx, and VSCATTERPF1xxx instructions are supported.

5.12.3 AVX512VL

CPUID bit	07h.ECX[1]

Allows 128-bit (XMM) and 256-bit (YMM) operations. See instructions tagged with the AVX512VL CPUID feature flag to see what instructions are affected.

5.12.4 AVX512_IFMA

CPUID bit	(Intel) 07h.EBX[21]

AVX-512 52-bit precision, where bit 53 is set, makes the VPMADD52HUQ and VPMADD52LUQ instructions available to use.

5.12.5 AVX512_VBMI

CPUID bit	(Intel) 07h.EBX[31]

Denotes 512-bit ZMM register usage is available for instructions VPERMB, VPERMI2B, VPERMT2B, and VPMULTISHIFTQB. Also adds additional capabilities not in AVX512BW.

5.13 Advanced Encryption Standard NI (AES-NI)

Year introduced	2009
CPUID bit	01h.ECX[25]
Instructions	• PCLMULQDQ • AESENC • AESDEC • AESENCLAST • AESDECLAST • AESKEYGENASSIST • AESIMC

Includes instructions to speedup the calculation of AES-related encryption and decryption calculations.

5.14 Fused-Multiply-Add (FMA)

The Fused-Multiply-Add instructions is a series of extensions to perform a single-rounding of a multiplication and addition instruction (e.g. ab + c). These instructions may use the VEX C4h prefix.

5.14.1 FMA4

Year introduced	(AMD) 2011
CPUID bit	8000_0001h.ECX[16]
Instructions	• VFMADDPDx • VFMADDPDy • VFMADDPSx • VFMADDPSy • VFMADDSD • VFMADDSS

FMA4 extends on FMA3. Intel never implemented this extension in their processor.

5.14.2 FMA3

Year introduced	(AMD) 2012, (Intel) 2013
CPUID bit	01h.ECX[22]
Instructions	• VFMADD132PDy • VFMADD132PSy • VFMADD132PDx • VFMADD132PSx • VFMADD132SD • VFMADD132SS • VFMADD213PDy • VFMADD213PSy • VFMADD213PDx • VFMADD213PSx • VFMADD213SD • VFMADD213SS • VFMADD231PDy • VFMADD231PSy • VFMADD231PDx • VFMADD231PSx • VFMADD231SD • VFMADD231SS

Mostly known as FMA, the FMA3 extension is implemented in both Intel and AMD processors.

6 Avanced Processor Features

This section attempts to explain some of the more advanced processor features.

6.1 Virtualization (VMX, SVM)

Year introduced	(Intel) 2005, (AMD) 2006
CPUID bit	(Intel) 01h.ECX[5] (AMD) 8000_0001h.ECX[2]
Instructions	• VMPTRLD • VMPTRST • VMCLEAR • VMREAD • VMWRITE • VMCALL • VMLAUNCH • VMRESUME • VMXOFF • VMXON

Virtualization, known as VT-x from Intel (VMX), AMD-V from AMD (SVM), and VIA VT from VIA, are technologies enabling running many guest operation systems on top of the host operation system efficiently. Type 1 and type 2 hypervisors use this technology.

6.2 FPU Features

Most recent x86 processors include a Floating Point Unit. A processor unit designed to handle decimal values, sometimes known as float numbers, following the IEEE 754 standard.

6.2.1 Float-16 Conversion (F16C)

Year introduced	2009
CPUID bit	01h.ECX[29]
Instructions	• VCVTPH2PS • VCVTPS2PH

If set, 16-bit float conversion is available, such as converting four packed half precision (16-bit) floating-point values or eight packed half precision (16-bit) floating-point values to a packed single-precision float-point value (VCVTPH2PS) and vice versa (VCVTPS2PH).

6.3 Advanced Configuration and Power Interface (ACPI)

CPUID bit	01h.EDX[22]

The Advanced Configuration and Power Interface was introduced to help power and thermal management through the operating system.

If set, ACPI features are present.

6.3.1 Advanced Programmable Interrupt Controller (APIC)

CPUID bit	01h.EDX[9]

Integrated in the micro-processor, the Advanced Programmable Interrupt Controller is an updated standard from Intel from the older PIC standard. It is used to effectively redirect interrupts.

This feature was added Intel Pentium micro-processors.

If set, an APIC is present.

Initial ID refers to the logical core that the program is being run on. On a modern operating system, this will likely be scheduled depending on the operating system strategy.

Max ID refers to the maximum ID that the program can run under. Note that Intel processor usually have this value doubled to the logical core count.

6.3.2 Thermal Monitor (TM)

CPUID bit	(Intel) 01h.EDX[29] (AMD) 8000_0007H.EDX[4]

The first thermal monitor thermally-initiates (on-die) modulations for the stop-clock duty cycle for reduced power consumption.

If set, the first thermal monitor is available.

Requires ACPI feature.

6.3.3 Thermal Monitor 2 (TM2)

CPUID bit	(Intel) 01h.ECX[8]

The second thermal monitor performs frequency transitions for reduced power consumption.

If set, the second thermal monitor is available.

Requires ACPI feature.

6.4 Virtualization Features

6.4.1 Virtual 8086 Mode Enhancements (VME)

CPUID bit	01h.EDX[1]

A number of enhancements were added within the Pentium architecture to the virtual 8086 mode, including virtual interrupts.

If set, the virtual 8086 mode enhancements are available.

6.5 Memory features

6.5.1

6.5.2 Page Size Extension (PSE)

CPUID bit	01h.EDX[3]

Traditionally, memory pages were 4 KiB in size, which usually ended up cluttering the translation look-aside buffer.

Introduced in the Pentium processor, the page size extension let user programs request bigger memory pages.

If set, the processor can support memory pages larger than 4 KiB.

6.5.3 36-bit Page Size Extension (PSE36)

CPUID bit	01h.EDX[17]

The 36-bit page size extension is an extension from 32-bit memory page addressing, allowing to address from 4 GiB to 64 GiB of memory.

If set, the processor may address up to at least 64 GiB of memory.

6.5.4 1 GiB Pages support (Page1GB)

CPUID bit	8000_0001h.EDX[26]

This feature allows the processor to page 1 GiB memory spaces.

If set, the processor can initiate 1 GiB memory pages.

6.5.5 Memory Type Range Registers (MTRR)

CPUID bit	01h.EDX[12]

A set of additional control registers to fine-tune memory regions that should be cached by the processor.

If set, these specific control registers are available.

6.5.6 Page Attribute Table (PAT)

CPUID bit	01h.EDX[16]

The page attribute table allows users to control attributes on a per-page basis. For example, marking memory as should-be cached.

If set, the processor supports the page attribute table feature.

6.5.7 Page Global Bit (PGE)

CPUID bit	01h.EDX[13]

The global bit in a page table is used to prevent the TLB from updating the address in cache if CR3 is reset.

If set, indicates support for the global bit in the page table.

6.5.8 Direct Cache Access (DCA)

CPUID bit	(Intel) 01h.ECX[18]

Direct cache access is an I/O technology that permits other devices to directly place data into the processor’s cache.

If set, direct cache access is available for other devices to use.

6.6 Debugging Features

6.6.1 Machine Check Architecture (MCA)

CPUID bit	01h.EDX[14]

Mechanism that enables hardware error reporting to the operating system.

If set, machine check architecture is available.

6.6.2 Machine Check Exception (MCE)

CPUID bit	01h.EDX[7]

On hardware errors, the processor may throw a machine-check exception, which may include errors about system buses errors, ECC errors, parity errors, etc.

This feature does not define the model-specific implementations of machine-check error logging, reporting, and processor shutdowns.

If set, the processor supports machine-check exceptions.

6.6.3 Debugging Extensions (DE)

CPUID bit	01h.EDX[2]

Defines support for I/O breakpoints.

If set, the processor supports I/O breakpoints.

6.6.4 Debug Store (DS)

CPUID bit	(Intel) 01h.EDX[21]

A debug store is a processor feature where the processor is able to write debugging information to a memory buffer.

If set, the processor may use the debug store.

6.6.5 Debug Store CPL (DS-CPL)

CPUD bit	(Intel) 01h.ECX[4]

The processor may supported the extensions to the debug store feature allowing for branch messages storage qualified by the CPL (Ring).

If set, indicates support for debug store CPL message branching.

6.6.6 64-bit DS Area (DTES64)

CPUID bit	(Intel) 01h.ECX[2]

Allows the use of 64-bit addresses in the debug store area.

If set, the debug store area can be used using 64-bit addresses.

6.6.7 Perfmon And Debug Capability (PDCM)

CPUID bit	(Intel) 01h.ECX[15]

Indicates performance and debug feature indication in IA32_PERF_CAPABILITIES (MSR).

If set, IA32_PERF_CAPABILITIES feature is available.

6.6.8 IA32_DEBUG_INTERFACE (SDBG)

CPUID bit	(Intel) 01h.ECX[11]

Firmware may use IA32_DEBUG_INTERFACE (MSR) for silicon debugging.

If set, silicon debugging is available.

6.7 Other Features

6.7.1 Brand Index

CPUID value	01h.EBX[7:0]

Before the processor brand string, the brand index was used instead. For example, a brand index of 04H indicated the Intel(R) Pentium(R) III processor.

If clear, the processor does not support the brand index identification.

Any values higher than 18H and higher are reserved (Intel).

6.7.2 L1 Context ID (CNXT-ID)

CPUID bit	(Intel) 01h.ECX[10]

The L1 data cache may be adapted to two strategies: adaptive and shared. The L1 context ID serves as setting the L1 data context strategy.

If set, the L1 Context ID feature is available.

6.7.3 xTPR Update Control (xTPR)

CPUID bit	(Intel) 01h.ECX[14]

Intel has yet to disclose information about xTPR Update Control.

If set, the processor may change IA32_MISC_ENABLE[23].

6.7.4 Process-Context Architecture (PCID)

CPUID bit	(Intel) 01h.ECX[17]
Instructions	• INVPCID

Enables up to 4,096 processes to be created an managed via a unique ID (PCID) via the processor.

If set, PCID is available.

6.7.5 Processor Serial Number (PSN)

CPUID bit	(Intel) 01h.EDX[18]

Introduced only in the Pentium III, the processor serial number is supposed to be a unique serial number per processor packages.

If set, the processor has a serial number.

6.7.6 Self Snoop (SS)

CPUID bit	(Intel) 01h.EDX[27]

Management of conflicting memory types may be done by performing a snoop of its own cache structure for transactions issued to the bus, thus self snooping.

If set, memory self snooping is available.

6.7.7 Pending Break Enable (PBE)

CPUID bit	(Intel) 01h.EDX[31]

In interrupt handling, the processor may use FERR# and PBE# pins in a stop-clock state (STPCLK#) that an interrupt in pending and that the processor should return to normal.

If set, supports FERR# and PBE# pins when in a stop-clock state.

6.7.8 Supervisor Mode Execution Protection (SMEP)

CPUID bit	07h.EBX[7]

Instruction fetches from user-mode addresses may be restricted in privileged addresses.

If set, supervisor mode execution protection is available (CR4.SMEP).

6.7.9 Bit manipulation groups

CPUID bits	(BMI1) 07h.EBX[4] (BMI2) 07h.EBX[8]	
Instructions	BMI1 • ANDN • BEXTR • BLSI • BLSMSK • BLSR • TZCNT	BMI2 • BZHI • MULX • PDEP • PEXT • RORX • SARX • SHRX • SHLX

Defines groups of bit manipulation instructions as BMI1 and BMI2. Please note that Advanced Bit Manipulation extension (ABM, AMD) features the LZCNT instruction.

7 Raw CPUID information

The `-r` option in ddcpuid prints a table of raw CPUID data, useful for debugging. Sub-leaf information is not provided.

The `-o` option overrides default leaves obtained by the processor and may be useful in exploring data further than the supported CPUID leaf.

Example (Intel Core i7-3770):

```
| Leaf     | EAX      | EBX      | ECX      | EDX      |
|----------|----------|----------|----------|----------|
|        0 |        D | 756E6547 | 6C65746E | 49656E69 |
|        1 |    306A9 |  5100800 | 7FBAE3FF | BFEBFBFF |
|        2 | 76035A01 |   F0B2FF |        0 |   CA0000 |
|        3 |        0 |        0 |        0 |        0 |
|        4 | 1C004121 |  1C0003F |       3F |        0 |
|        5 |       40 |       40 |        3 |     1120 |
|        6 |       77 |        2 |        9 |        0 |
|        7 |        0 |      281 |        0 |        0 |
|        8 |        0 |        0 |        0 |        0 |
|        9 |        0 |        0 |        0 |        0 |
|        A |  7300403 |        0 |        0 |      603 |
|        B |        1 |        2 |      100 |        3 |
|        C |        0 |        0 |        0 |        0 |
|        D |        7 |      340 |      340 |        0 |
| 80000000 | 80000008 |        0 |        0 |        0 |
| 80000001 |        0 |        0 |        1 | 28100800 |
| 80000002 | 20202020 | 20202020 | 65746E49 | 2952286C |
| 80000003 | 726F4320 | 4D542865 | 37692029 | 3737332D |
| 80000004 | 50432030 | 20402055 | 30342E33 |   7A4847 |
| 80000005 |        0 |        0 |        0 |        0 |
| 80000006 |        0 |        0 |  1006040 |        0 |
| 80000007 |        0 |        0 |        0 |      100 |
| 80000008 |     3024 |        0 |        0 |        0 |
```