Processor Recognition

Transmeta Corporation
3940 Freedom Circle
Santa Clara, CA 95054

Title: Processor Recognition
Date: 2002/05/07
Purpose: This document describes how to detect the presence of a Transmeta processor, and how to determine the features it supports. This document also describes the CPUID instruction, including all of its functions and return values.

Processor Recognition

Property of:
Transmeta Corporation
3940 Freedom Circle
Santa Clara, CA 95054
USA
(408) 919-3000
http://www.transmeta.com

Processor Recognition

Introduction

The CPUID instruction is designed to aid the programmer in detecting the processor and its supported features. If the ID bit in the EFLAGS register can be toggled, then the CPUID instruction is supported. Transmeta processors do support toggling the ID bit, as well as executing the CPUID instruction.

The CPUID instruction is non-privileged; therefore it can be executed not only by operating systems, but also by application programs. The instruction is serializing, and it supports multiple functions that are grouped into several function ranges. This design allows for future expansion, as it becomes necessary.

The CPUID instruction expects its input value – the function number – in the EAX register. The output or return values are loaded into the EAX, EBX, ECX, and EDX registers, depending on the invoked function. The following is a list of CPUID function, which are supported by Transmeta processors.

Reserved Functions

Function xxxx_xxxxh		**reserved**
EAX	0000_0000h	reserved
EBX	0000_0000h	reserved
ECX	0000_0000h	reserved
EDX	0000_0000h	reserved

Standard Functions

Function 0000_0000h		**get maximum supported standard function and standard vendor ID string**
EAX	0000_0003h or 0000_0001h	This is the maximum supported standard function. The reported value depends on whether the PSN is enabled or disabled.
EBX	756E_6547h	This is the standard vendor ID string. It consists of twelve ASCII characters (EBX-EDX-ECX), which are programmable through MSRs. The default string is "GenuineTMx86".
ECX	3638_784Dh	
EDX	5465_6E69h	

Function 0000_0001h		**get standard processor type, family (incl. extended), model (incl. extended), and stepping, and standard feature flags**
EAX	0000_054xh	This value represents the extended family (bits 27…20), extended model (bits 19…16), type (bits 13 and 12), family (bits 11…8), model (bits 7…4), and stepping (bits 3…0) of the CPU. The actual stepping value depends on the revision of the processor. The low 16 bits (starting with CMS 4.3: all 32 bits) of the returned value can be programmed through a MSR. x=2 TM3x00 x=3 TM5x00
EBX	0000_0000h	This value represents the local APIC's physical ID (bits 31…24), the virtual processor count (bits 23…16), the CLFLUSH line size in qwords (bits 15…8), and the brand ID (bits 7…0).The local APIC's physical ID is only valid, if a local APIC is supported. The virtual processor count is only valid, if Hyper-Threading Technology is supported. The CLFLUSH line size is only valid, if the instruction is supported. The brand ID is only valid, if it is non-zero.
ECX	0000_0000h	reserved

EDX	xxxx_xxxxh	These are the standard feature flags. A set bit indicates the presence of a feature, while a cleared bit indicates either the absence of a feature, or a reserved flag. bit 23 MMX MMX Instructions bit 18 PSN Processor Serial Number bit 15 CMOV Conditional Move Instructions bit 11 SEP Fast System Call Extension (CMS 4.2 +) bit 8 CX8 CMPXCHG8B Instruction bit 5 MSR Model Specific Registers bit 4 TSC Time Stamp Counter bit 3 PSE Page Size Extension bit 2 DE Debug Extensions bit 1 VME Virtual Mode Extensions bit 0 FPU Floating-Point Unit All bits can be forced to 0 through a MSR. Note that CMS 4.1 forces bit 8 to 0 by default, due to an issue with the Microsoft Windows NT operating system.
	0084_803Fh 0080_803Fh 0084_813Fh 0080_813Fh 0084_883Fh 0080_883Fh 0084_893Fh 0080_893Fh	PSN enabled, CX8 masked (default for CMS 4.1) PSN disabled, CX8 masked PSN enabled, CX8 unmasked PSN disabled, CX8 unmasked PSN enabled, SEP supported, CX8 masked PSN disabled, SEP supported, CX8 masked PSN enabled, SEP supported, CX8 unmasked (default for CMS 4.2 +) PSN disabled, SEP supported, CX8 unmasked

Function 0000_0002h		**reserved**
EAX	0000_0000h	This function is included for compatibility purposes.
EBX	0000_0000h	
ECX	0000_0000h	
EDX	0000_0000h	

Function 0000_0003h		**get processor serial number (PSN)**
EAX	xxxx_xxxxh	This function returns the PSN. If the PSN is disabled, then the returned register values are zero. Due to technical reasons it can't be guaranteed that the PSN is truly unique. In fact, registers ECX and EDX may return a value of zero on evaluation parts. OEMs have the option to permanently disable the PSN. The TM3200 processor does not support the PSN. The following format should be used when printing: EAX-EBX-ECX-EDX (hexadecimal, upper case letters), ie. "xxxxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxx"
EBX	xxxx_xxxxh	
ECX	xxxx_xxxxh	
EDX	xxxx_xxxxh	

Extended Functions

Function 8000_0000h		**get maximum supported extended function and extended vendor ID string**
EAX	8000_0006h	This is the maximum supported extended function.
EBX	6E61_7254h	This is the extended vendor ID string. It consists of twelve ASCII characters (EBX-EDX-ECX), and reads "TransmetaCPU".
ECX	5550_4361h	
EDX	7465_6D73h	

Processor Recognition

Function 8000_0001h		get extended processor type, family, model, and stepping, and extended feature flags
EAX	0000_054xh	This value represents the type (bits 13 and 12), family (bits 11...8), model (bits 7...4), and stepping (bits 3...0) of the CPU. The actual stepping value depends on the revision of the processor. x=2 TM3x00 x=3 TM5x00
EBX	0000_0000h	reserved
ECX	0000_0000h	reserved
EDX	xxxx_xxxxh	These are the extended feature flags. A set bit indicates the presence of a feature, while a cleared bit indicates either the absence of a feature, or a reserved flag. bit 23 MMX MMX Instructions bit 16 FCMOV FP Conditional Move Instructions bit 15 CMOV Conditional Move Instructions bit 8 CX8 CMPXCHG8B Instruction bit 5 MSR Model Specific Registers bit 4 TSC Time Stamp Counter bit 3 PSE Page Size Extension bit 2 DE Debug Extensions bit 1 VME Virtual Mode Extensions bit 0 FPU Floating-Point Unit
	0081_813Fh 0081_813Fh	TM3x00 TM5x00

Function 8000_0002h Function 8000_0003h Function 8000_0004h		get processor name
EAX	xxxx_xxxxh	This is the processor name, which may consist of up to 48 ASCII characters (3x EAX-EBX-ECX-EDX). Unused characters are filled with a value of 00h. It reads "Transmeta(tm) Crusoe(tm) Processor TMx" whereas the x stands for the actual model number (eg. "3200"), which can be up to 11 characters long.
EBX	xxxx_xxxxh	
ECX	xxxx_xxxxh	
EDX	xxxx_xxxxh	

Function 8000_0005h		get processor characteristics
EAX	0000_0000h	reserved
EBX	04FF_04FFh	4-way 256 entry data TLB, 4-way 256 entry code TLB Note that CPUID reports 255 instead of 256 entries, because the field is limited to 8 bits. Also note that the actual 256-entry TLB is reported twice, for compatibility purposes: once for the code TLB, and once for the data TLB.
ECX	2008_0120h 4010_0120h	32 KB 8-way 1 line/tag 32 bytes/line L1 data cache (TM3x00) 64 KB 16-way 1 line/tag 32 bytes/line L1 data cache (TM5x00)
EDX	4008_0140h 4008_0140h	64 KB 8-way 1 line/tag 64 bytes/line L1 code cache (TM3x00) 64 KB 8-way 1 line/tag 64 bytes/line L1 code cache (TM5x00)

Function 8000_0006h		get processor characteristics
EAX	0000_0000h	reserved
EBX	0000_0000h	reserved
ECX	0000_0000h 0100_4180h 0200_4180h	no integrated L2 cache (TM3200) 256 KB 4-way 1 line/tag 128 bytes/line L2 cache (TM5400/TM5500) 512 KB 4-way 1 line/tag 128 bytes/line L2 cache (TM5600/TM5800)
EDX	0000_0000h	reserved

Transmeta Functions

Function 8086_0000h		get maximum supported Transmeta function and Transmeta vendor ID string
EAX	8086_0007h	This is the maximum supported Transmeta function.
EBX	6E61_7254h	This is the Transmeta vendor ID string. It consists of twelve ASCII characters (EBX-EDX-ECX), and reads "TransmetaCPU".
ECX	5550_4361h	
EDX	7465_6D73h	

Function 8086_0001h		get Transmeta processor information
EAX	0000_054xh	This value represents the type (bits 13 and 12), family (bits 11...8), model (bits 7...4), and stepping (bits 3…0) of the CPU. The actual stepping value depends on the revision of the processor. x=2 TM3x00 x=3 TM5x00
EBX	aabb_ccddh	This is the Transmeta processor revision ID. It consists of the major (bits 31…24) and minor (bits 23…16) processor version, and the major (bits 15…8) and minor (bits 7…0) processor mask revision. The following format should be used when printing: EBX-ECX (decimal), ie. "a.b-c.d-x"
	0101_xxyyh 0102_xxyyh 0103_xxyyh 0103_00yyh 0104_xxyyh 0105_xxyyh 0200_0000h	TM3200 TM5400 TM5400 or TM5600 TM5500 or TM5800 TM5500 or TM5800 TM5500 or TM5800 see CPUID function 8086_0002h, register EAX Use the reported L2 cache size to distinguish a TM5400 from a TM5600, or a TM5500 from a TM5800.
ECX	xxxx_xxxxh	This value represents the nominal processor core clock frequency in MHz. Fractional frequencies may be rounded up or down. The following format should be used when printing: EBX-ECX (decimal), ie. "a.b-c.d-x"
EDX	xxxx_xxxxh	These are the Transmeta feature flags. A set bit indicates the presence of a feature, while a cleared bit indicates either the absence of a feature, or a reserved flag. bit 3 LRTI LongRun Table Interface (CMS 4.2) bit 1 LongRun LongRun bit 0 Recovery recovery CMS is active (after a bad flash) Note that the LongRun feature flag remains set to 1 even if LongRun has effectively been disabled by setting both the maximum and the minimum performance percentage to 100%.

Function 8086_0002h		get Transmeta CMS information
EAX	xxxx_xxxxh	If CPUID function 8086_0001h, register EBX reports 0200_0000h, then xxxx_xxxxh is the Transmeta processor revision ID. The following format should be used when printing:

		EAX (hexadecimal, upper case letters), ie. "xxxxxxxx" A detailed description of the format of this value is beyond the scope of this document.
EBX	aabb_ccddh	This is the first part of the Transmeta CMS revision ID. It consists of four 8-bit wide numbers. For example, 0102_0304h stands for 1.2.3-4. The following format should be used when printing: EBX-ECX (decimal), ie. "a.b.c-d-x"
ECX	xxxx_xxxxh	This is the second part of the Transmeta CMS revision ID. It consists of one 32-bit wide number. The following format should be used when printing: EBX-ECX (decimal), ie. "a.b.c-d-x"
EDX	0000_0000h	reserved

Function 8086_0003h Function 8086_0004h Function 8086_0005h Function 8086_0006h		**get Transmeta information string**
EAX	xxxx_xxxxh	This is the Transmeta information string, which may consist of up to 64 ASCII characters (4x EAX-EBX-ECX-EDX). Unused characters are filled with a value of 00h.
EBX	xxxx_xxxxh	
ECX	xxxx_xxxxh	
EDX	xxxx_xxxxh	

Function 8086_0007h		**get current Transmeta processor information**
EAX	xxxx_xxxxh	This value represents the current processor core clock frequency in MHz. Fractional frequencies may be rounded up or down. A value of 0000_0000h is reported if LongRun is not supported.
EBX	xxxx_xxxxh	This value represents the current processor voltage in millivolts. A value of 0000_0000h is reported if LongRun is not supported.
ECX	xxxx_xxxxh	This value represents the current processor performance percentage (between 0 and 100d). A value of 0000_0000h is reported if LongRun is not supported.
EDX	xxxx_xxxxh	This value represents the current gate delay in femtoseconds. If a gate delay is not applicable (CMS 4.1), then a value of zero is reported.
Though no x86 progress is made during a frequency transition, x86 execution does continue while the voltage is adjusted. Thus it is possible that this CPUID function is queried during a LongRun transition – in which case the returned values may reflect either the old or the new state. (This means that waiting for the new state to be reported does not guarantee that the LongRun transition has been finished.) Furthermore it should be noted that CoolRun takes precedence over LongRun. As a result the reported values may or may not reflect the actual ones.		

Notes

For maximum compatibility Transmeta uses only the most universal subset of ASCII characters in strings: the Unique Graphics Characters of ISO 646:1991 and ECMA–6 (**http://www.ecma.ch**). These properties apply only to Transmeta-supplied defaults. Programmable strings may deviate from this description.

END