Processor Family

BIOS Writers Guide

Centaur Technology
Confidential

Trademarks:

AMD, the AMD logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Am486 is a registered trademark, and AMD-K5 is a trademark of Advanced Micro Devices, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Intel, the Intel logo, and combinations thereof are trademarks of the Intel Corporation. MMX is a trademark of the Intel Corporation. Intel Pentium and Intel Processor Pro are registered trademarks of the Intel Corporation.

Cyrix, the Cyrix logo, and combinations thereof are trademarks of the Cyrix Corporation. Cyrix 6x86 is a registered trademark of the Cyrix Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

INTRODUCTION

The IDT WinChip C6 is an x86-compatible CPU with unique features and restrictions. This guide should serve as a starting point for any BIOS programmers adapting their BIOS code to recognize and fully utilize the IDT WinChip C6. This is not a stand-alone document; it is meant as a companion to the IDT WinChip C6 Data Sheet. The Data Sheet provides the detailed information. This document serves to point out areas where a BIOS writer needs to be aware that differences exist.

GENERAL COMPATIBILITY

The IDT WinChip C6 is highly compatible with standard ("586-class") x86 implementations. The handling of SMI compatible with the Intel Pentium (P54C). However, only integral clock speed multiples are supported. Currently the IDT WinChip C6 supports multipliers of 2, 3, 4 and 5x. The following table summarizes the IDT WinChip C6 usage of the BF pins as compared to the P54C. Please refer to the Data Sheet for more detailed information.

	Pins			**Clock Multiplier**	
	BF2	BF1	BF0	IDT WinChip C6	Intel P54C
Bus Frequency Multipliers	1	0	0	reserved	5/2 x
	1	0	1	3 x	3 x
	1	1	0	2 x	2 x
	1	1	1	4 x	3/2 x
	0	0	0	reserved	
	0	0	1	5 x	
	0	1	0	4 x	
	0	1	1	reserved	

The Machine-Specific Registers (MSRs) are not currently compatible with a P54C except for the Timestamp Counter (more on that later). When an IDT WinChip C6 CPU is recognized, writes to and reads from MSRs should be avoided (with the exception of MSR 10h, the Timestamp Counter). There are some performance-enhancing features that can and should be enabled. Write-combining is such a feature (described later).

The IDT WinChip C6 does not use a "P-rating" of any kind. Although "MMX-compatible" instructions are supported, the string "MMX" should not be displayed with the processor name. Furthermore, the "-S" string that Intel uses to identify CPUs that support SMI should not be used, even though SMI is supported. The only identifier that should be displayed is "IDT WinChip C6" and a megahertz rating. This megahertz rating is the actual internal clock frequency.

IDT WINCHIP C6 RECOGNITION

The first step is to recognize the IDT WinChip C6 and its revision level. Because each revision level can have unique features, both steps are necessary.

CPU recognition is accomplished through the use of the CPUID instruction. The CPUID instruction on the IDT WinChip C6 is compatible with that of other x86-compatibles, but with added functionality.

Operation of CPUID:

Input value: eax: 0

Output values:

eax: 1

ebx: 0x746E6543

edx: 0x48727561

ecx: 0x736C7561

Note that this makes the vendor ID string "CentaurHauls"

Input value: eax: 1

Output values:

eax[3:0] Stepping ID

eax[7:4] Model

eax[11:8] Family

eax[31:12] Reserved

ebx Reserved

ecx Reserved

edx Feature flags

The stepping ID, model, family and feature flags are defined compatibly.

Input value: eax: 0xC0000000

Output values:

eax: 0xC0000000 Release 1 feature set

other values reserved

CLOCK FREQUENCY DETERMINATION

The best algorithms for determining the clock frequency make use of the RDTSC instruction. This instruction yields a running 64-bit count of the number of processor clocks since powerup. This count can be run against either of two independent clocks in the PC architecture. These two clock sources can be programmed to generate interrupts and the TSC count can be compared between two successive interrupts. This count is accurate enough to place the CPU into one of several "megahertz bins." A system will generally support only several bus speeds, such as 50Mhz, 60Mhz, 66Mhz, etc. These frequencies, multiplied by the available multipliers of 2, 3, 4 and 5, yield the aforementioned megahertz bins.

The two sources are the real-time clock, which will generate the so-called "periodic interrupt" and the timer tick, with its frequency of close to 18 times per second.

WRITE-COMBINING

This is a feature that is described in detail in the Data Sheet. Briefly, when this feature is enabled, the IDT WinChip C6 will attempt to accumulate memory writes of less than a full dword. This reduces the memory access overhead and improves performance. This feature should be enabled when the IDT WinChip C6 is recognized. The actual implementation varies with the Feature Set Level, so the Feature Set Level should be checked with CPUID, eax = 0xC0000000, before attempting to enable this feature.

In the current release, as determined by the return value of eax being unchanged in CPUID with input value eax = 0xC0000000, the following algorithm will set up write-combining (within certain limitations) for total memory capacities in megabytes that are a power of two. This is a very basic algorithm, and obviously is not optimal for all memory capacities. Refinements to this algorithm can be made easily to accommodate other memory capacities. Please refer to the data sheet for more detailed information.

The simple algorithm shown below is designed for coding simplicity and is not optimal. Areas of improvement to this very basic algorithm might include better support for memory sizes not a power of two megabytes. The algorithm shown treats anything greater than the closest power of two megabytes as non-write-combining space, so for example if the memory size is 40 MB then the region from 32-40MB is not enabled for write-combining. Better support for the so-called “OS/2 memory hole” is also possible. When enabled, this hole is from 15-16MB. The sample algorithm will create a non-write-combining hole from 12-16MB.

Simple Sample Algorithm to Enable Write-Combining

```
;=============================================================
; IDT WinChip C6 write combining support
;=============================================================

; base/mask pairs for memory ranges

; 0 - 512K
base_0          equ     00000000000000000000000000000000b
mask_512K       equ     11111111111110000000000000000111b

; 512K 640K
base_512K       equ     00000000000010000000000000000000b
mask_640K       equ     11111111111111100000000000000111b
```

```
; 1M - 2M
base_1          equ     00000000000100000000000000000000b
mask_2          equ     11111111111100000000000000000111b

; 2M - 4M
base_2          equ     00000000001000000000000000000000b
mask_4          equ     11111111111000000000000000000111b

; 4M - 8M
base_4          equ     00000000010000000000000000000000b
mask_8          equ     11111111110000000000000000000111b

; 8M - 16M
base_8          equ     00000000100000000000000000000000b
mask_16         equ     11111111100000000000000000000111b
mask_12         equ     11111111110000000000000000000111b

; 16M - 32M
base_16         equ     00000001000000000000000000000000b
mask_32         equ     11111111000000000000000000000111b

; 32M - 64M
base_32         equ     00000010000000000000000000000000b
mask_64         equ     11111110000000000000000000000111b

; 64M - 128M
base_64         equ     00000100000000000000000000000000b
mask_128        equ     11111100000000000000000000000111b

; 128M - 256M
base_128        equ     00001000000000000000000000000000b
mask_256        equ     11111000000000000000000000000111b

; 256M - 512M
base_256        equ     00010000000000000000000000000000b
```

```
mask_512        equ     11110000000000000000000000000111b

; 512M - 1024M
base_512        equ     00100000000000000000000000000000b
mask_1024       equ     11100000000000000000000000000111b

; 1024M - 2048M
base_1024       equ     01000000000000000000000000000000b
mask_2048       equ     11000000000000000000000000000111b

; 2048M - 4096M
base_2048       equ     10000000000000000000000000000000b
mask_4096       equ     10000000000000000000000000000111b

; programming table for memory range registers

IDT_MCR_table   label   dword
                dd      base_2048
                dd      mask_4096

                dd      base_1024
                dd      mask_2048

                dd      base_512
                dd      mask_1024

                dd      base_256
                dd      mask_512

                dd      base_128
                dd      mask_256

                dd      base_64
                dd      mask_128
```

```
                        dd      base_32
                        dd      mask_64

                        dd      base_16
                        dd      mask_32

                        dd      base_8
                        dd      mask_16

                        dd      base_4
                        dd      mask_8

                         dd      base_2
                        dd      mask_4

                        dd      base_1
                        dd      mask_2

                        dd      base_512K
                        dd      mask_640K

                        dd      -1

MCRbaseno       equ     110h
MCRcontrol      equ     120h

MCRvalue        equ     11111000000000000000001111b

BCRno           equ     145h
TLOCKbit        equ     3


;=======================================================
;Name     :     Enable_C6_Write_Combining
;
;Function:      To enable the IDT C6 CPU's write combining
```

```
;               This code is version-specific!!!!!!!
;
;  *  It should only be run after the C6 is recognized
;
;Inputs:  edi contains memory size in MB
;         bl contains OS/2 memory hole state: 0 for no hole,
;           1 if the 15MB-16MB hole exists in memory map
;
;Uses:    eax, ecx, edx, esi
;         These registers are saved and restored
;=============================================================


Enable_C6_Write_Combining       Proc
                push    eax
                push    ecx
                push    edx
                push    esi

                mov     eax, 0C0000000h
                cpuid
                cmp     eax, 0C0000000h
                jne     No_V2_Write_Comb


        ; always set up 0-512K region

                mov     ecx, MCRbaseno

                mov     edx, base_0
                mov     eax, mask_512K

                wrmsr

                mov     esi, offset IDT_MCR_table
```

```
        ; find largest power of two <= memory size
                mov     ecx, 4096
range_comp:
                cmp     edi, ecx
                jge     short prog_mcrs
                add     esi, 8
                shr     ecx, 1
                jnz     range_comp

; program the Memory Control Registers from the table
prog_mcrs:
                mov     ecx, MCRbaseno+1
program_one_mcr:
                mov     edx, cs:dword ptr [si]

                cmp     edx, -1      ; stop if we are done!
                je      short mcr_done

                mov     eax, cs:dword ptr [si+4]

        ; special check for OS/2 memory hole
                cmp     edx, base_8
                jne     short around_hole_check
                or      bl, bl
                jz      short around_hole_check
                mov     eax, mask_12   ; if hole enabled

around_hole_check:

                wrmsr

                add     si, 8
                inc     ecx
                cmp     ecx, MCRbaseno+8  ; stop if we run
;                                          ;  out of MSRs
```

```
                jb      program_one_mcr
mcr_done:

        ; finally set up MCR_CTRL

                mov     ecx, 120h
                mov     eax, MCRvalue
                mov     edx, 0
                wrmsr

                mov     ecx, BCRno
                rdmsr
                or      al, 1 shl TLOCKbit
                wrmsr


No_V2_Write_Comb:

;       for future use - check for other versions.
;

No_Write_Comb:

                pop     esi
                pop     edx
                pop     ecx
                pop     eax

                ret

Enable_C6_Write_Combining       Endp
```