Preliminary Information

Processor

Data Sheet

This is **Version 1.00** of the IDT WinChip C6 processor datasheet.

The latest versions of this datasheet may be obtained from

www.winchip.com

Trademarks

CONTENTS

1. INTRODUCTION

The IDT WinChip™ C6™ processor, designed by Centaur Technology Inc., is a plug-compatible alternative to the Intel® Pentium® processor and to the new Intel Pentium processor with MMX™ technology (also known informally as the *P54* and the *P55* processors).

The IDT WinChip C6 processor family is based on a unique Centaur-developed design approach and is manufactured in the IDT 0.35-micron CMOS technology. This combination provides high-performance, low-cost, and low-power solutions to the desktop and mobile personal computer market.

In addition to the Intel Pentium family, the IDT WinChip C6 processor also directly competes with the AMD-K6™ and Cyrix® 6x86MX™ processors. When considered individually, the performance, cost, and power dissipation of the IDT WinChip C6 processor family are all very competitive. When considered as a whole, the IDT WinChip C6 processor family offers a breakthrough level of *value*.

1.1 BASIC FEATURES

The IDT WinChip C6 processor family comprises several versions. All family versions share the following common features:

Notable Features

- Plug-compatible with the Intel Pentium processor—bus, electrical interface, and physical package (“socket 7”).
- Software-compatible with Intel Pentium processors and the thousands of X86 software applications
- Software-compatible with the new Intel MMX technology
- Two large (32-KB each) on-chip caches.
- Performance optimized for business applications (as represented by the ZD Winstone97 Business benchmark)
- Lower power than the equivalent Pentium processor.
- Low cost due to a very small die (88 mm^2) and low-cost technology.

1.2 PROCESSOR VERSIONS

The IDT WinChip C6 processor family consists of a *single* functional version that is suitable for both desktop and mobile applications. This basic version is offered in several internal speed ranges and two different voltage settings.

Desktop Features

- Plug-compatible with the Intel P54 processor in a Ceramic PGA package—bus, electrical interface, and physical package.
- Operates in a Socket 7 (the standard socket that accepts both a P54 and a P55 processor).
- Like the P54, uses only one system-board voltage (versus the P55 which requires two voltages).

Mobile Features

- Same package and interface as for desktop
- Provides lower power consumption than the corresponding Mobile Pentium processor with MMX technology.

Speed Versions

- The IDT WinChip C6 processor is available in four speed grades:

  180-MHz (60-MHz bus),

  200-MHz (66-MHz bus)

  225-MHz (75-MHz bus)

  240-MHz (60-MHz bus)

Voltage Versions

- IDT WinChip C6 processors support one of two voltage ranges:

  3.52V (3.45V–3.6V)

  3.3V (3.135V–3.465V)

1.3 COMPETITIVE COMPARISONS

The following tables summarize the major features of the IDT WinChip C6 processor and its primary competitors. The competitive information is as specified in the competitive processor's data sheets and is accurate only as of the time this datasheet was written. The features are those that characterize the primary capabilities of an x86 processor.

The major themes of this summary are:

- The IDT WinChip C6 processor has equivalent or better cache and TLB capabilities. These are critical to system performance for modern PC operating systems and applications. (See Table 1-1)
- The IDT WinChip C6 processor has a very simple instruction execution architecture. This results in a very small die size for the functions provided (and thus facilitates its low price and low power). (See Table 1-2)
- The IDT WinChip C6 processor has very competitive internal frequency. This high frequency, combined with the large caches, yields very good *system* performance. (See Table 1-3)
- The IDT WinChip C6 processor has a much smaller die size in the same basic technology. This small size benefits the user by facilitating low price and low power consumption. (See Table 1-4)
- The IDT WinChip C6 processor has superior (reduced) power dissipation. (See Table 1-5)

Table 1-1. Cache and TLB Characteristics.

Major Features		C6	P55	K6	M2
I-Cache	*Size*	32 KB	16 KB	32 KB	256
	Data Ways	2	4	2	assoc
D-Cache	*Size*	32 KB	16 KB	32 KB	64 KB unified I & D cache
	Data Ways	2	4	2	4
TLB	*Size (I / D)*	64 / 64	32 / 64	64 /128	16
	Ways	4	assoc	?	direct
	L2 TLB	N	N	N	64 x 6
Page Dir Cache		8 entries	N	N	N

Table 1-2. Microarchitecture Characteristics.

Major Features	C6	P55	K6	M2
Decode	Single In-order	2 Insts In-order	3 Insts In-order	2 Insts In-order
Issue & Execute	Single In-order	2 Insts In-order	3 insts Out-order	2 Insts Out-order
Branch Prediction	N	Y	Y	Y
Call/Return Stack	8 entries	Y	16 entries	N

Table 1-3. Performance Characteristics

Major Features	C6	P55	K6	M2
Clock Multiplier Ratios (vs. Bus frequency)	2x, 3x, 4x 5x	2x, 3x, 3/2x, 5/2x	2x, 3x, 3/2x, 5/2x	2x, 3x, 3/2x, 5/2x
Bus Speed (MHz)s	60,66, 75	60,66	66	66,75
Max Internal Freq	240 MHz	233 MHz	233	187.5 (233PR)

Table 1-5. Technology

Major Features		C6	P55	K6	M2
Technology	*Metal*	0.35µ	0.35µ	0.35µ	0.35µ
	Poly	0.28µ			
		4LM	4LM	5LM+LI	4LM
Voltage	*I/O*	3.3/3.52	3.3	3.3	3.3
	Core	same	2.8/2.45	2.9/3.2	2.9
Die Size		88 mm^2	128 mm^2	162 mm^2	197 mm^2

Table 1-6. Power Dissipation

Major Features	C6	P55	K6	M2
MHz	200	200	200	166 (PR200)
Worst Case W		15.7 (2.8V)	20.0 (2.9V)	20.2 (2.9V)
3.52V	10.4			
3.3V	8.9			

1.4 COMPATIBILITY

The IDT WinChip C6 processor is compatible with both the Intel Pentium processor and the new Intel Pentium processor with MMX technology. However, the IDT WinChip C6 processor is specifically optimized for desktop and mobile PC configurations (as opposed to server environments).

An IDT WinChip C6 processor can plug into existing Pentium processor-based desktop and portable system boards and can operate without requiring changes to the system hardware. In some cases, a special BIOS may be needed (due to possible use by the BIOS of Pentium processor-unique machine specific registers). Currently, BIOS support for the IDT WinChip C6 processor is available from Award, AMI, and SystemSoft.

The IDT WinChip C6 processor does not provide Pentium-compatible multiprocessing (neither do the mobile Pentium processor, the AMD-K6 processor, and the Cyrix 6x86MX processors).

Note that *all* processors developed for use in PCs ("x86" processors) have some differences in low-level functions. (These include differences between the various Intel processors; and between these processors and the equivalent Cyrix and the AMD processors.) The IDT WinChip C6 processor has similar differences.

Centaur has performed extensive testing of hundreds of PC boards, peripherals, software applications, and operating systems to confirm the IDT WinChip C6 processor's compatibility.

Indicative of this compatibility, the IDT WinChip C6 processor has obtained the XXCAL Inc. Platinum Certification (their highest compatibility rating) and Microsoft Windows 95 certification.

1.5 DATA SHEET ASSUMPTIONS

The IDT WinChip C6 processor specifications are directly based upon the Pentium processor's external specifications as defined by: (1) publicly available Intel publications, and (2) by the actual behavior (derived from testing) of the Pentium processor. This datasheet book provides only minimal descriptions of these Pentium-compatible functions. The major emphasis in this document is to describe differences from the explicit and implicit (behavioral) Pentium specifications.

The intent of these specifications is to make it easy for a board designer, system designer, or BIOS developer to utilize the IDT WinChip C6 processor in place of the Pentium processor or the Pentium processor with MMX technology. (This, of course, makes it trivially easy for the end-user to be able to exploit the advantages of the IDT WinChip C6 processor.) We assume that the reader is a potential direct user of the IDT WinChip C6 processor and is thus familiar with the specifications of the Pentium processor.

Table 1-7 lists some relevant documents that define the reference x86 architecture.

Table 1-7. x86 Architecture Specification Documents

Document Title	*Intel Order #*	*Version*
Intel Architecture Software Developer's Manual, Vol. 1	243190	001
Intel Architecture Software Developer's Manual, Vol. 2	243191	001
Intel Architecture Software Developer's Manual, Vol. 3	243192	001
Pentium Processor Family Developer's Manual	241428	005
Pentium Processor with MMX Technology	243185	004
Pentium Processor Specification Update	242480	027

2. ARCHITECTURE

2.1 INTRODUCTION

Even though the IDT WinChip C6 processor is externally compatible with the Pentium processor, the internal architecture and design of the IDT WinChip C6 processor is very different from that of the Pentium processor and other contemporary x86 processors such as the AMD-K6 and Cyrix 6x86MX processors. The IDT WinChip C6 processor uses a unique design approach that provides significant benefits to the end-user.

This design approach provides high performance at low cost and low power using a relatively simple architecture that runs at high internal clock frequencies (MHz), includes large on-chip caches, and is extensively optimized for the target PC environment. The resulting IDT C6 processor is smaller (die size is only 88 mm^2 in 0.35µ geometry technology) and uses less power than any contemporary X86 processor.

Philosophically, this is a return to the same basic concepts of RISC design that allowed microprocessor performance breakthroughs in the 1980's. Recently, however, contemporary x86 processors have followed a different path using very complex internal designs employing advanced architecture concepts such as superscalar execution, out-of-order instruction execution, reorder buffers, non-blocking caches, and so forth (these terms are all found in the datasheets of competitive products).

Unfortunately, while these advanced technical concepts make for good technical reading, the real bottom-line benefit that they provide to the end-user has been limited; especially when considering the resultant large chip sizes (resulting in high costs) and high power consumption. No such advanced technical hocus-pocus is to be found on an IDT WinChip C6 processor—it merely offers compatibility with good performance, very low costs, and very low power consumption.

2.2 KEY CONCEPTS

The key concepts underlying the IDT WinChip C6 processor design are:

- *Simple instructions (load, store, branch, ALU) dominate instruction execution time.* This is the basic RISC design concept, which is also true in the x86 architecture: over 90% of instructions executed come from these basic categories. Of course, "simple" x86 instructions are more complex than corresponding RISC architecture instructions.

  The IDT WinChip C6 processor optimizes performance of these types of basic x86 instructions while minimizing the hardware provided for other little-used x86 functions. The little-used instructions are primarily implemented in microcode with minimal hardware support.

- *Improving clock frequency has higher leverage than improving CPI.* The result of advanced computer design approaches over the last few years has been that the improvements in cycles-per-instruction (CPI) come at the expense of cycle-time improvements such that *total* performance (a linear function of frequency and CPI) of complex designs ("brainiacs") hasn't improved as fast as performance of simpler, yet high clock frequency, designs ("speed demons").

  The IDT WinChip C6 processor optimizes *total system* performance by generally optimizing for highest clock frequency, even if this means reduced CPI for some functions versus more complex processor designs.

- *Memory performance is the limiting CPI performance factor.* Due to the high ratio of internal clock speed versus the relatively limited PC processor-bus speed, off-chip memory-access performance is the primary factor in processor CPI performance (as opposed to internal instruction execution performance).

  The IDT WinChip C6 processor addresses this phenomenon by providing very large on-chip caches that run at the high internal processor clock frequency. In addition, sophisticated TLB and cache management algorithms are included to further reduce bus activity.

- *Optimize the design for the target user environment.* The IDT WinChip C6 processor implements very specific and detailed design tradeoffs to provide high performance with low cost. Minimal hardware is provided for functions that are not heavily used or that are not critical to performance in the target environments (low-end desktop and mobile systems). These design optimizations are based on extensive and detailed analysis of the actual behavior of Windows operating systems and applications
- *Small is beautiful.* The IDT WinChip C6 processor is highly optimized for small physical size and few logic transistors. In addition to the obvious cost benefits, this small size provides secondary benefits of low power consumption and improved reliability.

2.3 COMPONENT SUMMARY

2.3.1 General Architecture

Figure 2-1 illustrates the basic components of the IDT WinChip C6 processor.

Fundamentally, the IDT WinChip C6 processor's internal design is a relatively simple five-stage pipeline execution core with an additional instruction translation stage to translate x86 instructions coming from the fetch stage into the internal micro-instruction format.

Fetching and translating x86 instructions is asynchronous to the internal execution pipeline. Instructions are issued one at a time in program order. Instructions are executed and retired in order. Cache misses stall the pipeline until the data is available for the requesting instruction.

In spite of this very basic micro-architecture, the IDT WinChip C6 processor achieves very high performance through several mechanisms:

- High internal clock frequency. The design is heavily optimized to provide high frequency.
- Good CPI on highly used instructions. The IDT WinChip C6 processor implements specific design features to reduce the number of cycles for heavily used instructions — including complex functions such as protect-mode segment-register loads and string instructions.
- Large and fast on-chip caches and TLBs.
- Lots of fine-tuning and low-level optimizations. This includes such items as fast unaligned data access and fewer pipeline interlocks than the Pentium processor.

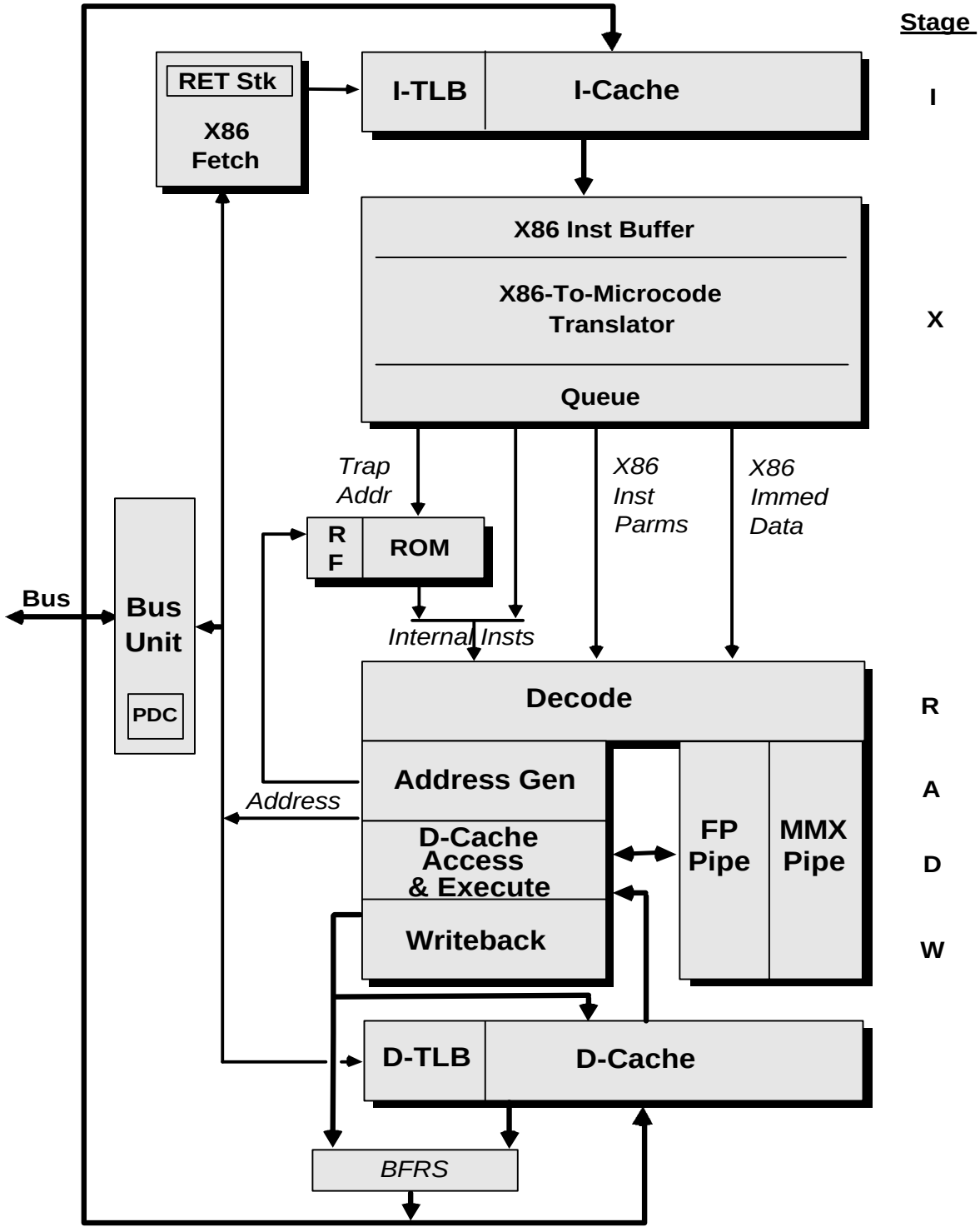


Figure 2-1. IDT WinChip C6 Processor Data Flow

2.3.2 I-Cache

The I-cache contains 32 KB organized as two-way set associative with 32-byte lines. An LRU replacement algorithm is used. The associated I-TLB contains 64 entries organized as a 4-way set associative with a pseudo-LRU replacement algorithm. This large cache has a one-clock access time and operates at the high clock frequencies of the IDT WinChip C6 processor.

The I-TLB utilizes an eight-entry unified page directory cache that significantly reduces the TLB miss penalty. In addition, the I-cache control logic includes several innovative features that minimize cache invalidates and unnecessary bus fetches.

As opposed to many other contemporary x86 processors, the data in the I-cache is exactly what came from the bus; that is, there are no “hidden” pre-decode bits. This facilitates the provision of large cache capacity in a small physical size.

The I-cache is dynamically turned off when not used to reduce power requirements.

2.3.3 Translator Unit

The I-cache or bus unit delivers 16 or 8 bytes per clock to an x86 instruction buffer in the translator unit. The translator converts x86 instructions to internal instruction and data forms. Assuming that the instruction is in the x86 instruction buffer at the start of the cycle, the translator translates an entire x86 instruction in one clock. Instruction prefixes require an additional translator cycle for each prefix. However, due to the asynchronous fetch and “lookahead” capability of the translator, these extra cycles for prefixes rarely result in a bubble in the execution pipeline.

The output of the translator is: (1) the internal micro-instruction stream to perform the x86 instruction function, (2) the immediate data fields from the x86 instruction, and (3) various x86 state information used to control execution (for example, operand size). The internal instruction stream for an x86 instruction can consist of micro-instructions directly generated by the translator, or micro-instructions from the on-chip ROM, or both. For performance-sensitive instructions, there is no delay due to access of micro-code from ROM.

The microcode ROM capacity is larger than most x86 microcode ROMs to allow more unimportant (relative to performance) functions to be performed in microcode (versus in hardware), to allow extensive self-test microcode, and to allow extensive built-in debugging aids (for processor design debug).

Instruction fetch and translator operation is made asynchronous from micro-instruction execution via a three-entry translated-instruction queue between the translator and the execution unit. This queue allows the translator to "look-ahead" and continue translating x86 instructions even though the execution unit is stalled or is busy with a microcode sequence. The translator can also overlap generation of multiple internal instructions with translating prefixes on the subsequent instruction.

2.3.4 Execution Unit

Internal micro-instructions are executed in a tightly coupled four-stage pipeline that is very similar in structure to a basic RISC pipeline:

- **Decode stage (R)**: Micro-instructions are decoded, register files are accessed, resource dependencies evaluated, and so forth.
- **Addressing stage (A)**: Memory addresses are calculated and sent to the cache units. The IDT WinChip C6 processor is capable of calculating most x86 instruction address forms in one clock; forms containing two registers (or a shifted index register) require two clocks.
- **Execute stage (D):** ALU operations or load accesses to the D-cache are performed. All basic ALU functions take one clock except multiply and divide.
- **Write-back stage (W)**: The results of operations are committed to the registers and store data is written to the D-cache or external write buffers.

Although the pipeline structure is similar to non-x86 processors, the micro-instructions and associated execution units are highly tuned to the x86 architecture. The micro-instructions closely resemble the corresponding x86 instructions. Examples of specialized hardware features supporting the x86 architecture are: hardware handling of the x86 condition code, segment descriptor decode and manipulation instructions, hardware to automatically save the x86 floating-point environment, and so forth.

2.3.5 D-Cache

The D-cache is very similar to the I-cache: 32 KB organized as two-way set associative with 32-byte lines. An LRU replacement algorithm is used. The associated D-TLB contains 64 entries organized as 4-way set associative with a pseudo-LRU replacement algorithm. This large cache has a one-clock access time and is designed to operate at the high clock frequencies of the IDT WinChip C6 processor. The D-TLB shares the 8-entry unified page directory cache which reduces the TLB miss penalty. The D-cache is dynamically turned off when not used to reduce power requirements.

2.3.6 X86 Fetch Unit

One of the most visible size (and frequency)-versus-CPI tradeoffs made by the IDT WinChip C6 processor is that there is very limited branch prediction: only a special 8-entry Call/Return stack. This computer-science heresy (no branch prediction) helps support the high clock frequency and small die size but means that, *on average,* simple branches normally take slightly more clocks on the IDT WinChip C6 processor than on other contemporary x86 processors.

However, the clock difference is not as high as might be expected due to: (1) Return instructions are accurately predicted with the special Call/Return stack mechanism, (2) the total performance of complex branches (such as far calls) includes more than the actual branch time (which is overlapped with the additional operations on the IDT WinChip C6 processor), and (3) the short IDT WinChip C6 processor pipeline.

Actually, in a simple pipeline processor like the IDT WinChip C6 processor, the CPI contribution of branches is not very high. The main need for branch prediction is in deeply pipelined superscalar designs where branches happen late in the pipeline flow and are highly disruptive to the multiple issue strategy. In the IDT WinChip C6 processor branches happen early in the short pipeline and invalidate few instructions in the single issue pipeline.

The most important consideration relative to the IDT WinChip C6 processor's lack of branch prediction is that the gain from branch prediction is relatively small compared to the total time waiting on bus activity in high-frequency x86 processors. The IDT WinChip C6 processor design focuses on using its transistors to minimize this bus wait time: large caches, large TLBs, smart cache management algorithms, and so forth, as opposed to a large branch prediction array.

2.3.7 FP-Unit

In addition to the basic integer execution unit, the IDT WinChip C6 processor has a separate 80-bit floating-point execution unit that can execute floating-point instructions in parallel with integer instructions.

The floating-point unit is designed to maximize clock frequency and to minimize chip size while providing adequate levels of floating-point performance for typical desktop use. Some floating-point instructions are pipelined, but some are only partially pipelined.

The IDT WinChip C6 processor issues only one instruction per clock but most integer instructions and most floating-point instructions can execute in parallel.

The floating-point unit is dynamically turned off when not used to reduce power requirements.

2.3.8 MMX-Unit

The IDT WinChip C6 processor contains a separate execution unit for the new MMX-compatible instructions. The MMX architecture registers are the same as the floating-point registers, otherwise the MMX execution unit has its own adder, multiplier and shifter separate from the floating-point unit.

The MMX unit is dynamically turned off when not used to reduce power requirements.

2.3.9 Bus-Unit

The IDT WinChip C6 processor bus unit provides an external bus interface compatible with the Pentium processor. In addition to the expected bus control functions, the bus unit implements a large page-directory cache to reduce the impact of TLB misses as well as several special optimizations intended to reduce cache misses. Four 64-bit write buffers allow internal execution to proceed overlapped with waiting for external stores to complete.

The IDT WinChip C6 processor bus unit contains many special features designed to reduce bus traffic and cache disruption. Examples include store byte-combining function, cache cast-out snarfing, “smart” lock management mechanisms, and so forth.

3. PROGRAMMING INTERFACE

3.1 GENERAL

In general, the IDT WinChip C6 processor is compatible with both the bus and software-visible architecture of the Intel Pentium processor with MMX technology. That is, a program that executes on a Pentium processor should generally execute on an IDT WinChip C6 processor and produce the same results (with the exceptions as noted in this datasheet).

The IDT WinChip C6 processor's Pentium-compatible functions include:

- All basic X86 instructions, registers, and functions
- All floating-point (numeric processor) instructions, registers and functions
- All new Pentium processor instructions and registers (CMPXCHG8B, RDMSR, WRMSR, RDTSC, CPUID, RSM, MOV CR4)
- All basic operating modes: real mode, protect mode, virtual-8086 mode
- System Management Interrupt (SMI) and the associated System Management Mode (SMM)
- All interrupt and exception functions
- All debug functions (including the new I/O breakpoint function)
- All input/output functions
- All tasking functions (TSS, task switch, etc.)
- Processor initialization behavior

The IDT WinChip C6 processor also includes instructions compatible with the MMX instructions in the Pentium processor with MMX technology

However, there are some differences between the IDT WinChip C6 processor and the Pentium processor. These differences fall into four groups:

- **Additional IDT WinChip C6 processor functions**. Examples are memory range registers that allow different attributes for each range. These additional functions are provided through Machine Specific Registers such that compatibility is not affected.
- **Implementation-specific differences**. Examples are cache and TLB testing features, and performance monitoring features that expose the internal implementation features. These types of functions are incompatible among *all* different x86 implementations.—including the Intel486, the Pentium, and the Pentium Pro processors.
- **Omitted functions**. Some Pentium processor functions are not provided on the IDT WinChip C6 processor because they aren't used or aren't needed in the targeted PC systems. Examples are some specific bus functions such as functional redundancy checking and performance monitoring.

  These types of differences are similar to those among various versions of the Pentium processor (for example, the mobile Pentium processor also omits the same bus functions as omitted by the IDT WinChip C6 processor), and among the Pentium processor and the AMD-K6 and Cyrix 6x86MX processors.
- **Low-level behavioral differences**. A few low-level IDT WinChip C6 processor functions are different from the Pentium because the results are (1) documented in the Intel documentation as *undefined*, and (2) known to be different for different x86 implementations (in particular, different among the Intel i486, the Pentium, and the Pentium Pro processors). That is, compatibility with the Pentium processor for these functions is clearly not needed for software compatibility (or they wouldn't be different across different implementations). Where the Pentium and Pentium Pro processor results differ, the IDT WinChip C6 processor often provides the Pentium Pro result.

This chapter summarizes the first three types of differences: additional functions, implementation-specific functions, and omitted functions. *Appendix A* contains more details on machine-specific functions. *Appendix B* contains details on low-level differences.

In some areas, we also include comparative information about the Pentium Pro, AMD-K6, and Cyrix 6x86MX processors. This information is taken from the data sheets of these products and has not been verified by Centaur Technology. Our Pentium processor information, however, is based on detailed testing.

3.2 ADDITIONAL FUNCTIONS

The IDT WinChip C6 processor provides some memory range management functions. These are similar in concept, but different in specifics, to memory range registers in the Pentium Pro, AMD-K6, and Cyrix 6x86MX processors (all of which are different from each other). These functions are provided via Machine Specific Registers. *Appendix A* provides specifics on the IDT WinChip C6 Machine Specific Registers.

3.3 MACHINE-SPECIFIC FUNCTIONS

3.3.1 General

All x86 processor implementations provide a variety of *machine-specific functions*. Examples are cache and TLB testing features, and performance monitoring features that expose the internal implementation features. These types of functions are different and incompatible among all different x86 implementations—including the Intel i486, the Pentium, and the Pentium Pro processors, and between these processors and competitive processors from Cyrix and AMD. The Intel documentation clearly identifies these types of functions as machine-specific and warns of possible changes in new implementations.

This section describes the IDT WinChip C6 processor machine-specific functions that are most likely used by software and compares them to related processors where applicable. *Appendix A* describes the IDT WinChip C6 processor machine-specific registers (*MSR*s).

This section covers those features of Pentium-compatible processors that are used to commonly identify and control processor features. All Pentium-compatible processors have the same mechanisms, but the bit-specific data values often differ.

3.3.2 CPUID Instruction

The CPUID instruction is available on all contemporary x86 processors. The CPUID instruction has two basic functions requested via the EAX register. The first function returns a vendor identification string in registers EBX, ECX and EDX. The second CPUID function returns an assortment of bits in EAX and EDX that identify the chip version and describe the specific features available.

The EAX:EBX:ECX:EDX return values of the CPUID instruction executed with EAX == 0 are:

Table 3-1

Register[bits] - Meaning	***C6***	***P54***	***P55***	***K6***	***M2***
EAX (highest EAX input value understood by CPUID)	1	1	1		
EBX:EDX:ECX (vendor ID string)	"Centaur Hauls"	"Genuine Intel"	"Genuine Intel"	"Authentic AMD"	"Cyrix Instead"

The EAX return values of the CPUID instruction executed with EAX == 1 are:

Table 3-2

EAX Bits - Meaning	***C6***	***P54***	***P55***	***K6***	***M2***
3:0 - Stepping ID	*Same as the return value in EDX after Reset (see next section)*				
7:4 - Model ID					
11:8 - Family ID					
13:12 - Type ID					

The EDX return values of the CPUID instruction with EAX == 1 are:

Table 3-3

EAX Bits - Meaning	***C6***	***P54***	***P55***	***K6***	***M2***	***Notes***
0 - FP present	1	1	1	1	1	
1 - VM86 Extensions (VME)	0	1	1	1	0	1
2 - Debugging Extensions	1	1	1	1	1	
3 - Page Size Extensions (4MB)	0	1	1	1	0	1
4 - Time Stamp Counter (TSC) supported	1	1	1	1	1	2
5 - Model Specific Registers present	1	1	1	1	1	3
6 - PAE supported (P6 Function)	0	0	0	0	0	4
7 - Machine Check Exception	1/0	1	1	1	0	5
8 - CMPXCHG8B instruction	0/1	1	1	1	1	6
9 - APIC supported	0	1	1	0	0	7
10:11 -Reserved						
12- Memory Range Registers	0	0	0	0	0	8
13 - PTE Global Bit supported	0	0	0	0	1	4
14- Machine Check Architecture supported	0	0	0	0	0	4
15- Conditional Move supported	0	0	0	0	1	4
16:22 - Reserved						
23 - MMX supported	1/0	0	1	1	1	98
24:31 - Reserved						

Notes On CPUID Feature Flags

General: an "x/y" entry means that the default setting of this bit is x but the bit (and the underlying function) can be set to y using the FCR MSR.

1. These "Appendix H" functions are not provided on the IDT WinChip C6 processor since they are not used by the target operating systems. They are also not provided on the Cyrix 6x86MX processor.
2. The IDT WinChip C6 processor implementation varies slightly from that of the Pentium processor in a way that should have no practical impact.
3. Every system has different MSRs—addresses and contents.
4. This is a function introduced with the Pentium Pro processor and is generally not provided on Pentium-compatible processors.
5. The Machine Check exception is defined by Intel documentation as machine-specific. The IDT WinChip C6 processor's Machine Check has slightly different specifics than the Pentium processor's Machine Check function. The Machine Check support can be enabled or disabled by a bit in the FCR MSR. The CPUID bit reports the current setting of this enable control.
6. The Pentium processor-compatible CMPXCHG8B instruction is provided and always enabled. However, the default for the corresponding CPUID function bit is 0 (due to a bug found in Windows NT). This default can be changed via a bit in the FCR MSR.
7. This is a Intel-specific multiprocessing function. None of the other Pentium-compatible processors provide this function since it has no utility in the target system environment.
8. The IDT WinChip C6 and Cyrix 6x86MX processors have memory range registers, but the specifics are not compatible with the P6 MRRs. The Pentium processor has no memory range registers.

9. The IDT WinChip C6 processor's MMX-technology compatible instruction support can be enabled or disabled by a bit in the FCR. The CPUID bit reports the current setting of this enable control.

3.3.3 Extended CPUID Instruction Functions

The IDT WinChip C6 processor provides several machine-specific features. Some of these features are compatible with those provided by P55 and are identified by the standard CPUID function EAX=1.

Other machine-specific features described in this datasheet have no P55 equivalent. These features are controlled by IDT WinChip C6 MSRs.

System software must not assume that all future processors in the IDT WinChip C6 family will implement all of the same machine-specific features or even that these features will be implemented in a backward-compatible manner. In order to determine if the processor supports particular machine-specific features, system software should follow this procedure:

1. Identify the processor as an IDT WinChip C6 by using CPUID with EAX=0
2. Execute the CPUID instruction with extended function EAX = C000_0000h. This function is unique to IDT WinChip family processors.
   - If the output in EAX is C000_0000h (unchanged) then the IDT WinChip C6 supports the machine-specific features as described in this datasheet.
   - If EAX output is something other than C000_0000h then the processor is a later version of the WinChip C6 and system software must not attempt to activate any machine-specific feature.

3.3.4 EDX Value After Reset.

As for other x86 processors, after reset the EDX register holds a component identification number as follows:

	31:14	13:12	11:8	7:4	3:0
EDX	reserved	Type ID	Family ID	Model ID	Stepping ID
	18	2	4	4	4

The specific values for the various IDT WinChip C6 processor types are:

Processor	***Type ID***	***Family ID***	***Model ID***	***Stepping ID***
C6	0	5	4	varies

For comparison, following are the values for other X86 processors:

Processor	***Type ID***	***Family ID***	***Model ID***	***Stepping ID***
P54C	0	5	2	varies
P55	0	5	4	varies
Cyrix 6x86MX	0	6	0	varies
AMD-K6	0	6	6-9	varies

3.3.5 CR4

Control register 4 (CR4) is a new feature of the Pentium processor that controls some of its advanced features. The IDT WinChip C6 processor provides a CR4 with the following specifics:

CR4 Bits - Meaning	*C6*	*P54*	*P55*	*K6*	*M2*	*Notes*
0: VME: Enables VME feature	0	0/1	0/1	0/1	0	1
1: PVI: Enables PVI feature	0	0/1	0/1	0/1	0	1
2: TSD: Makes RDTSC inst privileged	0/1	0/1	0/1	0/1	0/1	
3: DE: Enables I/O breakpoints	0/1	0/1	0/1	0/1	0/1	
4: PSE: Enables 4-MB pages	0	0/1	0/1	0/1	0	1
5: PAE: Enables addr extensions	r	r	r	r	r	2
6: MCE: Enables machine check exception	0/1	0/1	0/1	0/1	0	3
7: PGE: Enables global page feature	r	r	r	r	0/1	2
8: PCE: Enables RDPMC for all levels	0/1	r	0/1	r	0/1	
31:9 - reserved	r	r	r	r	r	

Notes On CR4

General: a "0/1" means that the default setting of this bit is 0 but the bit can be set to (1). A "0" means that the bit is always 0; it cannot be set. An "r" means that this bit is reserved. It appears as a 0 when read, and a GP exception is signaled if an attempt is made to write a 1 to this bit.

1. The IDT WinChip C6 processor does not provide this "Appendix H" function and this CR4 bit cannot be set. However, no GP exception occurs if an attempt is made to set this bit. The Cyrix 6x86MX processor also does not provide this function.
2. This is a Pentium-Pro processor function that is typically not provided on P55-compatible processor.
3. The IDT WinChip C6 processor Machine Check has slightly different specifics than the P54C Machine Check function

3.3.6 Machine-Specific Registers

The IDT WinChip C6 processor implements the Pentium family concept of Machine Specific Registers (MSRs). RDMSR and WRMSR instructions are provided and the CPUID instruction identifies that the IDT WinChip C6 processor supports MSRs. However, the IDT WinChip C6 processor MSRs are different from the Pentium and Pentium Pro processors (which are different from each other, and from the Cyrix 6x86MX and AMD-K6 processors).

In general, the MSRs have no usefulness to application or operating system software and are not used. (This is to be expected since the MSRs are different on each processor). *Appendix A* contains a detailed description of the IDT WinChip C6 processor's MSRs.

3.4 OMITTED FUNCTIONS

This section summarizes those functions that are included in some Pentium processor versions, but are not in the IDT WinChip C6 processor.

3.4.1 Pentium Appendix H Enhancements

The infamous *Appendix H* functions are those Pentium functions that are documented in Appendix H (Advanced Functions) of Volume 3 of *the Pentium Processor Family Developer's Manual*.

Unfortunately, Appendix H is only available to those with the "appropriate non-disclosure agreements in place". However, most of these functions are now publicly documented in the Pentium Pro processor documentation.

The Appendix H features are identified as "optional in future processors" and *are specifically identified as being supported or not by the CPUID instruction*.

Due to the limited utility of these advanced functions (they are complex operating system functions), there are few programs that utilize these features. In particular, these functions are either not used at all, or are conditionally used if present, by Microsoft desktop operating systems.

The IDT WinChip C6 processor does not provide the following Appendix H functions.

Virtual Memory Enhancements (4-MB Pages).

These Pentium processor enhancements provide the ability to optionally define 4-MB virtual memory pages in addition to the usual 4-KB page size. A bit in the feature identification return from the CPUID instruction indicates whether this feature is present or not. This enhancement is not provided on the IDT WinChip C6 processor since it is not used by the target operating systems: Windows 95 and Windows 98. Note that this function is also not provided on the Cyrix 6x86MX processor.

Virtual-8086 Mode Enhancements (VME)

These Pentium processor enhancements provide potential performance improvements to mode-switching operations while operating in VM86 mode. A bit in the feature identification return from the CPUID instruction indicates whether this feature is present or not. This enhancement is not provided on the IDT WinChip C6 processor since it is not used by the target Microsoft operating systems. Note that this function is also not provided on the Cyrix 6x86MX processor.

3.4.2 Other Functions

The IDT WinChip C6 processor also omits the software interface to the Intel-proprietary symmetric multiprocessing support: *APIC*. This bus function is omitted since the target market for the IDT WinChip C6 processor is portables and typical desktop systems (which do not support APIC multiprocessing).

A bit in the feature identification return from the CPUID instruction indicates whether this feature is present or not. This enhancement is not provided on the IDT WinChip C6 processor (as it is not on the mobile Pentium processor and on the AMD-K6 and Cyrix 6x86MX processors).

4. HARDWARE INTERFACE

4.1 BUS INTERFACE

The IDT WinChip C6 processor bus interface is compatible with the Pentium processor and the Pentium processor with MMX technology. This behavior is specified in *Pentium Processor Family Developer's Manual*.

The majority of the pins within the bus interface are involved with the physical memory and I/O interface. These pins and this interface perform the same functions as in the Pentium processor. The remaining pins are power and ground pins, test and debug support pins and various ancillary control functions. Most of these pins are identical to the Pentium processor. Others are associated with functions that behave slightly differently from the Pentium processor on the IDT WinChip C6 processor. Still others behave differently among the various versions of the Pentium processor, and thus require clarification on the IDT WinChip C6 processor. Lastly there are several Pentium processor functions which are completely omitted on the IDT WinChip C6.

4.1.1 Differences

The areas where the IDT WinChip C6 processor differs from the Pentium processor are not anticipated to cause operational compatibility issues. These differences are:

- Bus Frequency Control
- Machine Check Exceptions on BUSCHK# and PEN#
- Drive Strengths
- Probe Mode / JTAG / TAP Port (see *Test and Debug Section*)

Bus Frequency Control

Because the IDT WinChip C6 is designed to support several frequencies higher than 233 MHz, the IDT WinChip C6 processors interpret the bus frequency control pin slightly differently from the Pentium processor's specification. The IDT WinChip C6's interpretation is shown in Table 4-1. Note that the changes redefine fractional bus frequency multiples (which the IDT WinChip C6 does not support) to higher multiples. A new pin, BF2 (W-35), is also supported to allow for higher frequencies. The IDT WinChip C6 pulls up all three of the BF pins so that the default is 4x multiplier.

Table 4-1. Bus Frequency Ratios

BF2	BF1	BF0	IDT WinChip C6 Clock Ratio	P55 Clock Ratio
1	0	0	*reserved*	5/2x
1	0	1	3x	3x
1	1	0	2x	2x
1	1	1	4x	3/2x
0	0	0	*reserved*	
0	0	1	5x	
0	1	0	4x	
0	1	1	*reserved*	

Machine Check Exceptions on BUSCHK# & PEN#

As in the Pentium processor, the BUSCHK# interrupt causes a Machine Check exception or is ignored based on CR4.MCE. The difference is that the semantics of Machine Check exception are slightly different on a IDT WinChip C6 processor from a Pentium processor (a IDT WinChip C6 processor doesn't save and report the bus address and cycle data). See section 4.4 for further description of Machine Check.

Drive Strength

Desktop Pentium processors have three driver strengths that can be selected at Reset for certain pins (for example ADS#). The driver strength is selected by the BRDYC# and BUSCHK# pins when sampled at RESET deassertion

The IDT WinChip C6 processor has only two driver strengths:

Table 4-2. Drive Strengths

BUSCHK#	*BRDYC#*	*IDT WinChip C6 Driver*	*P54C Driver*
0	0	Medium	Strong
0	1	Medium	Medium
1	0	Typical	Typical
1	1	Typical	Typical

Only ADS#, A[20:3], HITM# and W/R# are configurable. All other drivers are typical strength. The AC characteristics of both drive strengths are described in Chapter 5, Electrical Specifications.

4.1.2 Clarifications

Power Supply Voltage

The IDT WinChip C6 processor operates with a unified power plane. Depending on the version, the processor requires either 3.3 Volts or 3.52 Volts at its VCC inputs.

The IDT WinChip C6 package is compatible with Socket 7, in that the VCC2DET# pin is internally no-connected. Flexible socket 7 motherboards can use the fact that VCC2DET# is not internally connected to force the motherboard core and pad regulators to produce the same voltage.

5V Tolerance

Like the P55, the IDT WinChip C6 processor's CLK input is not 5 Volt tolerant. It should be driven by a 3.3 Volt device.

4.1.3 Omissions

Advanced Peripheral Interrupt Controller (APIC)

The APIC is not supported by the IDT WinChip C6. The APIC pins (PICCLK, PICD0, and PICD1) are classified as reserved, and should not be connected on the motherboard.

(The APIC is also not supported in the mobile Pentium processor, the Cyrix 6x86MX and AMD-K6 processors.)

Dual Processor Interface

The IDT WinChip C6 processors do not support the dual processor interface. The associated pins (D/P#, PBGNT#, PBREQ#, PHIT#, and PHITM#) are classified as reserved, and should not be connected on the motherboard.

(The DP interface is also not supported in the mobile Pentium processor, the Cyrix 6x86MX and AMD K6.)

Functional Redundancy Checking Mode

The IDT WinChip C6 processors do not support the functional redundancy checking mode. The FRCMC# pin is classified as reserved, and should not be connected on the motherboard

(The functional redundancy checking mode is also not supported in the mobile Pentium processors, the Pentium processors with MMX technology, and the Cyrix 6x86MX.)

Breakpoint and Performance Monitoring Signals

The IDT WinChip C6 processors internally support instruction and data breakpoints. However, the IDT WinChip C6 does not support the Pentium processor's external indication of breakpoint matches via the BP3-BP0 pins. Similarly, the IDT WinChip C6 contains performance monitoring hooks internally, but it does not support the Pentium processor's external indication of performance monitoring events on PM1-PM0. The associated pins are unconnected on the IDT WinChip C6 package.

4.2 SIGNAL SUMMARY

Table 4-3 summarizes the bus interface signals of a Pentium and which signals are provided on a IDT WinChip C6 processor: an '×' in each processor's column indicates that the pin is supported by that processor.

Table 4-3. Signal Summary

Signal	*Type*	*P55 (TCP)*	*P55 (PPGA)*	*P54C (TCP)*	*P54C (PPGA)*	*C6 (CPGA)*
A20M#	I	×	×	×	×	×
A31-A3	I/O	×	×	×	×	×
ADS#	O	×	×	×	×	×
ADSC#	O		×		×	×
AHOLD	I	×	×	×	×	×
AP	I/O	×	×	×	×	×
APCHK#	I	×	×	×	×	×
APICEN/PICD 1	I/O		×		×	
BE7#-BE0#	O	×	×	×	×	×
APICID[3:0]	-	×	×		×	
BF[2:0]	I	×	×	×	×	×
BOFF#	I	×	×	×	×	×
BP[3:2] PM/BP[1:0]	O	×	×	×	×	
BRDY#	I	×	×	×	×	×
BRDYC#	I		×		×	×
BREQ	O	×	×	×	×	×
BUSCHK#	I	×	×	×	×	×
CACHE#	O	×	×	×	×	×
CLK	I	×	×	×	×	×
CPUTYP	I		×		×	
D/C#	O	×	×	×	×	×
D63-D0	I/O	×	×	×	×	×
D/P#	-		×		×	
DP7-DP0	I/O	×	×	×	×	×
DPEN# - PICD0	-		×		×	
EADS#	I	×	×	×	×	×
EWBE#	I	×	×	×	×	×

Signal	Type	P55 (TCP)	P55 (PPGA)	P54C (TCP)	P54C (PPGA)	C6 (CPGA)
FERR#	O	×	×	×	×	×
FLUSH#	I	×	×	×	×	×
FRCMC#	-			×	×	
HIT#	I	×	×	×	×	×
HITM#	I	×	×	×	×	×
HLDA	O	×	×	×	×	×
HOLD	I	×	×	×	×	×
IERR#	O	×	×	×	×	×
IGNNE#	I	×	×	×	×	×
INIT	I	×	×	×	×	×
INV	I	×	×	×	×	×
KEN#	O	×	×	×	×	×
INTR	I	×	×	×	×	×
NMI	I	×	×	×	×	×
LOCK#	O	×	×	×	×	×
M/IO#	O	×	×	×	×	×
NA#	I	×	×	×	×	×
PBGNT#	-		×		×	
PBREQ#	-		×		×	
PCD	O	×	×	×	×	×
PCHK#	O	×	×	×	×	×
PEN#	I	×	×	×	×	×
PHIT#	-		×		×	
PHITM#	-		×		×	
PICCLK	-		×		×	
PRDY	O	×	×	×	×	×
PWT	O	×	×	×	×	×
R/S#	I	×	×	×	×	×
RESET	I	×	×	×	×	×
SCYC	O	×	×	×	×	×

Signal	*Type*	*P55 (TCP)*	*P55 (PPGA)*	*P54C (TCP)*	*P54C (PPGA)*	*C6 (CPGA)*
SMI#	I	×	×	×	×	×
SMIACT#	O	×	×	×	×	×
STPCLK#	I	×	×	×	×	×
TCK	I	×	×	×	×	×
TDI	I	×	×	×	×	×
TDO	O	×	×	×	×	×
TMS	I	×	×	×	×	×
TRST#	I	×	×	×	×	×
VCC2DET#	-		×		×	×
W/R#	O	×	×	×	×	×
WB/WT#	O	×	×	×	×	×

4.3 POWER MANAGEMENT

The IDT WinChip C6 processor provides both static and dynamic power management.

4.3.1 Static Power Management

The IDT WinChip C6 processor supports the five power management modes of the Pentium processor: NORMAL state, STOP CLOCK state, STOP GRANT state, STOP CLOCK SNOOP state, and AUTOHALT state. These are described in the *Pentium Family Developer's Manual.*

4.3.2 Dynamic Power Management

The IDT WinChip C6 processor uses dynamic power management techniques to reduce power consumption in the NORMAL state. In NORMAL state, the on-chip arrays, selected datapaths, and the associated control logic are powered down when not in use.

4.4 TEST & DEBUG

4.4.1 Machine Check

IDT WinChip C6 Processors provide an Machine Check exception function (INT 18) that is slightly different than the Pentium processor or Pentium Pro processor Machine Check function (which are different from each other, of course). These differences are reasonable and expected since Intel documentation specifies that the Machine Check architecture is processor-specific.

In both the Pentium processor and IDT WinChip C6 processor, the Machine Check exception must be enabled by setting the MCE bit in CR4. If not enabled, the conditions (below) causing a Machine Check are ignored and no processor action is taken.

Both the IDT WinChip C6 processor and Pentium processor cause a Machine Check, if enabled, when:

- BUSCHK# is asserted
- PEN# is asserted and a data parity error is detected (PCHK# is asserted)

The differences between the IDT WinChip C6 processor and the Pentium processor are:

- The Pentium processor reports specifics about the bus cycle in MSR's 0 and 1. The IDT WinChip C6 processor does not provide this bus-cycle data.
- The IDT WinChip C6 processor default behavior for internally detected processor errors is (like the Pentium processor) to assert IERR and (normally) perform to a Shutdown bus cycle. However, if the EMCIE bit in the FCR is set, then internal errors on a IDT WinChip C6 processor cause a Machine Check exception.

4.4.2 BIST

A Built-in Self Test (BIST) can be requested as part of the IDT WinChip C6 processor reset sequence using exactly the same mechanism as used on the Pentium processor (INIT asserted as RESET deasserted).

The IDT WinChip C6 processor BIST performs the following general functions:

- A hardware-implemented exhaustive test of (1) all internal microcode ROM, and (2) the X86 instruction decode, instruction generation and entry point generation logic.
- An extensive microcode test of all internal registers and datapaths.
- An extensive microcode test of data and instruction caches, their tags, and associated TLB's.

BIST requires about two million internal clocks.

EAX Value After Reset

The results of a BIST is indicated by a code in EAX. Normally EAX is zero after reset. If a BIST is requested as part of the Reset sequence, EAX contains the BIST results. A 0 in EAX after BIST Reset means that no failures were detected. Any value other than zero indicates an error has occurred during BIST.

4.4.3 Internal Error Detection

During normal execution, the IDT WinChip C6 processor detects parity errors in both caches. In addition, certain "impossible" internal states are detected by microcode. These errors are normally reported via the same mechanism as in the Pentium processor: the IERR bus signal is asserted and (normally) a Shutdown occurs. (The Shutdown can be suppressed on both processors via a control bit in the TR1 MSR.)

Alternatively, an optional feature (a control bit in the FCR MSR) allows internal errors to be reported as Machine Check exceptions.

4.4.4 JTAG

The IDT WinChip C6 processor has a JTAG scan interface which is used for test functions and the proprietary Debug Port. However, unlike the Pentium processor, the IDT WinChip C6 processor does not provide a fully compatible IEEE 1149.1 JTAG function. In particular, the boundary scan function is not provided.

From a practical user viewpoint, JTAG does not exist and the associated pins (TCK, and so forth) should not be used.

4.4.5 Debug Port

The Pentium processor (and other processors such as the AMD-K6) have a proprietary Debug Port which uses the JTAG scan mechanism to control internal debug features ("probe mode"). These interfaces are not documented and are available (if at all) only under a non-disclosure agreement.

Similarly, the IDT WinChip C6 processor has an undocumented and proprietary debug interface.

5. ELECTRICAL SPECIFICATIONS

5.1 AC TIMING TABLES FOR 75-MHZ BUS

Table 5-1. AC Specifications for 75-Mhz Bus

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
f	CLK Frequency	37.5	75	MHz		
t_{1a}	CLK Period	13 1/3	26 2/3	ns		
t_{1b}	CLK Period Stability		±250	ps		Adjacent Clocks
t_2	CLK High Time	4.0		ns		2V
t_3	CLK Low Time	4.0		ns		0.8V
t_4	CLK Fall Time	0.15	1.5	ns		2V-0.8V
t_5	CLK Rise Time	0.15	1.5	ns		0.8V-2V
t_{6a}	PWT, PCD, CACHE# Valid Delay	1.0	7.0	ns		(1)
t_{6b}	AP Valid Delay	1.0	8.5	ns		(1)
t_{6c}	BE0-7#, LOCK# Valid Delay	0.9	7.0	ns		(1)
t_{6d}	ADS# Valid Delay	0.8	6.0	ns		(1)
t_{6e}	ADC#, D/C#, W/R#, SCYC Valid Delay	0.8	7.0	ns		(1)
t_{6f}	M/IO# Valid Delay	0.8	5.9	ns		(1)
t_{6g}	A3-A16 Valid Delay	0.5	6.3	ns		(1)
t_{6h}	A17-A31	0.6	6.3	ns		(1)
t_7	ADS#, ADSC#, AP, A3-A31, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10.0	ns		(1)
t_{8a}	APCHK#, IERR#, FERR# Valid Delay	1.0	8.3	ns		(1)
t_{8b}	PCHK# Valid Delay	1.0	7.0	ns		(1)
t_{9a}	BREQ Valid Delay	1.0	8.0	ns		(1)
t_{9b}	SMIACT# Valid Delay	1.0	7.3	ns		(1)
t_{9c}	HLDA Valid Delay	1.0	6.8	ns		(1)
t_{10a}	HIT# Valid Delay	1.0	6.8	ns		(1)
t_{10b}	HITM# Valid Delay	0.7	6.0	ns		(1)
t_{11b}	PRDY Valid Delay	1.0	8.0	ns		(1)
t_{12}	D0-D63, DP0-7 Valid Delay	1.3	7.5	ns		(1)
t_{13}	D0-D63, DP0-7 Float Delay		10.0	ns		(2)
t_{14}	A5-A31 Setup Time	3.3		ns		
t_{15}	A5-A31 Hold Time	1.0		ns		

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
t_{16a}	INV, AP Setup Time	3.3		ns		
t_{16b}	EADS# Setup Time	3.3		ns		
t_{17}	EADS#, INV, AP Hold Time	1.0		ns		
t_{18a}	KEN# Setup Time	3.0		ns		
t_{18b}	NA#, WB/WT#, NA# Setup Time	3.3		ns		
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.0		ns		
t_{20}	BRDY#, BRDYC# Setup Time	3.3		ns		
t_{21}	BRDY#, BRDYC# Hold Time	1.0		ns		
t_{22}	AHOLD, BOFF# Setup Time	3.3		ns		
t_{23}	AHOLD, BOFF# Hold Time	1.0		ns		
t_{24a}	BUSCHK#, EWBE#, HOLD Setup Time	3.3		ns		
t_{24b}	PEN# Setup Time	3.3		ns		
t_{25a}	BUSCHK#, EWBE#, PEN# Hold Time	1.0		ns		
t_{25b}	HOLD Hold Time	1.5		ns		
t_{26}	A20M#, INTR, STPCLK# Setup Time	3.3		ns		(4)
t_{27}	A20M#, INTR, STPCLK# Hold Time	1.0		ns		(5)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	3.3		ns		(4)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.0		ns		(5)
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width (Async)	2.0		CLK's		(6)
t_{31}	R/S# Setup Time	3.3		ns		
t_{32}	R/S# Hold Time	1.0		ns		
t_{33}	R/S# Pulse Width (Async)	2.0		ns		
t_{34}	D0-63, DP0-7 Read Data Setup Time	2.8		ns		
t_{35}	D0-63, DP0-7 Read Data Hold Time	1.5		ns		
t_{36}	RESET Setup Time	3.3		ns		(4)
t_{37}	RESET Hold Time	1.0		ns		(5)
t_{38}	RESET Pulse Width, Vcc & CLK Stable	15.0		CLK's		
t_{39}	RESET Pulse Width after Vcc & CLK Stable	1.0		ms		Power Up
t_{40}	RESET Configuration Signals (INIT, FLUSH#) Setup Time	3.3		ns		
t_{41}	RESET Configuration Signals (INIT, FLUSH#) Hold Time	1.0		ns		
t_{42a}	RESET Configuration Signals (INIT, FLUSH#) Setup Time (Async)	2.0		CLK's		To RESET Falling Edge

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
t_{42b}	RESET Configuration Signals (INIT, FLUSH#, BRDYC#, BUSCHK#) Hold Time (Async)	2.0		CLK's		To RESET Falling Edge
t_{42c}	RESET Configuration Signals (BRDYC#, BUSCHK#) Setup Time (Async)	3.0		CLK's		To RESET Falling Edge
t_{42d}	RESET Configuration Signals (BRDYC#, BUSCHK#) Hold Time (Sync)	1.0		ns		To RESET Falling Edge
t_{43a}	BF0, BF1, BF2 Setup Time	1.0		ms		To RESET Falling Edge
t_{43b}	BF0, BF1, BF2 Hold Time	2.0		CLK's		To RESET Falling Edge

Notes:

1. *C_L = 0 pF*
2. *Not 100% tested. Guaranteed by design and characterization*
3. *All outputs are glitch free signals, guaranteed to rise and fall monotonically when driven into capacitive loads. Most system loads must be treated as transmission lines. Depending on the length of the transmission line, loading and impedance mismatches, the signal may not rise or fall monotonically at a given point along the transmission line.*
4. *Setup time must be met to guarantee sampling on a given processor clock. Signals may be driven asynchronously, but, if so, are not guaranteed to be sampled by a specific clock edge.*
5. *Hold time must be met to guarantee sampling on a given processor clock. Signals may be driven asynchronously, but, if so, are not guaranteed to be sampled by a specific clock edge.*
6. *To guarantee proper recognition, signal must be deasserted for two or more processor clocks when driven asynchronously. When driven synchronously, signal must be deasserted for one processor clock to be recognized.*

5.2 AC TIMING TABLES FOR 66-MHZ BUS

Table 5-2. AC Specifications for 66-MHz Bus

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
f	CLK Frequency	33 1/3	66 2/3	MHz		
t1a	CLK Period	15.0	30.0	ns		
t1b	CLK Period Stability		±250	ps		Adjacent Clocks
t_2	CLK High Time	4.0		ns		
t_3	CLK Low Time	4.0		ns		
t_4	CLK Fall Time	0.15	1.5	ns		
t_5	CLK Rise Time	0.15	1.5	ns		
t_{6a}	PWT, PCD, CACHE# Valid Delay	1.0	7.0	ns		(1)
t_{6b}	AP Valid Delay	1.0	8.5	ns		(1)
t_{6c}	BE0-7#, LOCK# Valid Delay	0.9	7.0	ns		(1)
t_{6d}	ADS# Valid Delay	0.8	6.0	ns		(1)
t_{6e}	ADC#, D/C#, W/R#, SCYC Valid Delay	0.8	7.0	ns		(1)
t_{6f}	M/IO# Valid Delay	0.8	5.9	ns		(1)
t_{6g}	A3-A16 Valid Delay	0.5	6.3	ns		(1)
t_{6h}	A17-A31	0.6	6.3	ns		(1)
t_7	ADS#, ADSC#, AP, A3-A31, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10.0	ns		(1)
t_{8a}	APCHK#, IERR#, FERR# Valid Delay	1.0	8.3	ns		(1)
t_{8b}	PCHK# Valid Delay	1.0	7.0	ns		(1)
t_{9a}	BREQ Valid Delay	1.0	8.0	ns		(1)
t_{9b}	SMIACT# Valid Delay	1.0	7.3	ns		(1)
t_{9c}	HLDA Valid Delay	1.0	6.8	ns		(1)
t_{10a}	HIT# Valid Delay	1.0	6.8	ns		(1)
t_{10b}	HITM# Valid Delay	0.7	6.0	ns		(1)
t_{11b}	PRDY Valid Delay	1.0	8.0	ns		(1)
t_{12}	D0-D63, DP0-7 Valid Delay	1.3	7.5	ns		(1)
t_{13}	D0-D63, DP0-7 Float Delay		10.0	ns		(2)
t_{14}	A5-A31 Setup Time	6.0		ns		
t_{15}	A5-A31 Hold Time	1.0		ns		
t_{16a}	INV, AP Setup Time	5.0		ns		
t_{16b}	EADS# Setup Time	5.0		ns		

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
t_{17}	EADS#, INV, AP Hold Time	1.0		ns		
t_{18a}	KEN# Setup Time	5.0		ns		
t_{18b}	NA#, WB/WT#, NA# Setup Time	4.5		ns		
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.0		ns		
t_{20}	BRDY#, BRDYC# Setup Time	5.0		ns		
t_{21}	BRDY#, BRDYC# Hold Time	1.0		ns		
t_{22}	AHOLD, BOFF# Setup Time	5.5		ns		
t_{23}	AHOLD, BOFF# Hold Time	1.0		ns		
t_{24a}	BUSCHK#, EWBE#, HOLD Setup Time	5.0		ns		
t_{24b}	PEN# Setup Time	4.8		ns		
t_{25a}	BUSCHK#, EWBE#, PEN# Hold Time	1.0		ns		
t_{25b}	HOLD Hold Time	1.5		ns		
t_{26}	A20M#, INTR, STPCLK# Setup Time	5.0		ns		(4)
t_{27}	A20M#, INTR, STPCLK# Hold Time	1.0		ns		(5)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5.0		ns		(4)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.0		ns		(5)
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width (Async)	2.0		CLK's		
t_{31}	R/S# Setup Time	5.0		ns		
t_{32}	R/S# Hold Time	1.0		ns		
t_{33}	R/S# Pulse Width (Async)	2.0		ns		
t_{34}	D0-63, DP0-7 Read Data Setup Time	2.8		ns		
t_{35}	D0-63, DP0-7 Read Data Hold Time	1.5		ns		
t_{36}	RESET Setup Time	5.0		ns		
t_{37}	RESET Hold Time	1.0		ns		
t_{38}	RESET Pulse Width, Vcc & CLK Stable	15.0		CLK's		
t_{39}	RESET Pulse Width after Vcc & CLK Stable	1.0		ms		Power Up
t_{40}	RESET Configuration Signals (INIT, FLUSH#) Setup Time	5.0		ns		
t_{41}	RESET Configuration Signals (INIT, FLUSH#) Hold Time	1.0		ns		
t_{42a}	RESET Configuration Signals (INIT, FLUSH#) Setup Time (Async)	2.0		CLK's		To RESET Falling Edge
t_{42b}	RESET Configuration Signals (INIT, FLUSH#, BRDYC#, BUSCHK#) Hold Time (Async)	2.0		CLK's		To RESET Falling Edge

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
t_{42c}	RESET Configuration Signals (BRDYC#, BUSCHK#) Setup Time (Async)	3.0		CLK's		To RESET Falling Edge
t_{42d}	RESET Configuration Signals (BRDYC#, BUSCHK#) Hold Time (Sync)	1.0		ns		To RESET Falling Edge
t_{43a}	BF0, BF1, BF2 Setup Time	1.0		ms		To RESET Falling Edge
t_{43b}	BF0, BF1, BF2 Hold Time	2.0		CLK's		To RESET Falling Edge

See notes after Table 5-1.

5.3 AC TIMING TABLES FOR 60-MHZ BUS

Table 5-3. AC Specifications for 60-MHz Bus

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
f	CLK Frequency	30	60	MHz		
t_{1a}	CLK Period	16 2/3	33 1/3	ns		
t_{1b}	CLK Period Stability		±250	ps		Adjacent Clocks
t_2	CLK High Time	4.0		ns		
t_3	CLK Low Time	4.0		ns		
t_4	CLK Fall Time	0.15	1.5	ns		
t_5	CLK Rise Time	0.15	1.5	ns		
t_{6a}	PWT, PCD, CACHE# Valid Delay	1.0	7.0	ns		
t_{6b}	AP Valid Delay	1.0	8.5	ns		
t_{6c}	BE0-7#, LOCK# Valid Delay	0.9	7.0	ns		
t_{6d}	ADS# , ADC#, D/C#, M/IO#, W/R#, SCYC Valid Delay	0.8	7.0	ns		
t_{6e}	A3-A16 Valid Delay	0.5	6.3	ns		
t_{6f}	A17-A31	0.6	6.3	ns		
t_7	ADS#, ADSC#, AP, A3-A31, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10.0	ns		
t_{8a}	APCHK#, IERR#, FERR# Valid Delay	1.0	8.3	ns		
t_{8b}	PCHK# Valid Delay	1.0	7.0	ns		
t_{9a}	BREQ, HLDA Valid Delay	1.0	8.0	ns		
t_{9b}	SMIACT# Valid Delay	1.0	7.6	ns		
t_{10a}	HIT# Valid Delay	1.0	8.0	ns		
t_{10b}	HITM# Valid Delay	0.7	6.0	ns		
t_{11b}	PRDY Valid Delay	1.0	8.0	ns		
t_{12}	D0-D63, DP0-7 Valid Delay	1.3	7.5	ns		
t_{13}	D0-D63, DP0-7 Float Delay		10.0	ns		
t_{14}	A5-A31 Setup Time	6.0		ns		
t_{15}	A5-A31 Hold Time	1.0		ns		
t_{16a}	INV, AP Setup Time	5.0		ns		
t_{16b}	EADS# Setup Time	5.5		ns		
t_{17}	EADS#, INV, AP Hold Time	1.0		ns		
t_{18a}	KEN# Setup Time	5.0		ns		

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
t_{18b}	NA#, WB/WT#, NA# Setup Time	4.5		ns		
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.0		ns		
t_{20}	BRDY#, BRDYC# Setup Time	5.0		ns		
t_{21}	BRDY#, BRDYC# Hold Time	1.0		ns		
t_{22}	AHOLD, BOFF# Setup Time	5.5		ns		
t_{23}	AHOLD, BOFF# Hold Time	1.0		ns		
t_{24}	BUSCHK#, EWBE#, HOLD, PEN# Setup Time	5.0		ns		
t_{25a}	BUSCHK#, EWBE#, PEN# Hold Time	1.0		ns		
t_{25b}	HOLD Hold Time	1.5		ns		
t_{26}	A20M#, INTR, STPCLK# Setup Time	5.0		ns		(4)
t_{27}	A20M#, INTR, STPCLK# Hold Time	1.0		ns		(5)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5.0		ns		(4)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.0		ns		(5)
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width (Async)	2.0		CLK's		(6)
t_{31}	R/S# Setup Time	5.0		ns		
t_{32}	R/S# Hold Time	1.0		ns		
t_{33}	R/S# Pulse Width (Async)	2.0		ns		
t_{34}	D0-63, DP0-7 Read Data Setup Time	3.0		ns		
t_{35}	D0-63, DP0-7 Read Data Hold Time	1.5		ns		
t_{36}	RESET Setup Time	5.0		ns		
t_{37}	RESET Hold Time	1.0		ns		
t_{38}	RESET Pulse Width, V_{cc} & CLK Stable	15.0		CLK's		
t_{39}	RESET Pulse Width after V_{cc} & CLK Stable	1.0		ms		Power Up
t_{40}	RESET Configuration Signals (INIT, FLUSH#) Setup Time	5.0		ns		
t_{41}	RESET Configuration Signals (INIT, FLUSH#) Hold Time	1.0		ns		
t_{42a}	RESET Configuration Signals (INIT, FLUSH#) Setup Time (Async)	2.0		CLK's		To RESET Falling Edge
t_{42b}	RESET Configuration Signals (INIT, FLUSH#, BRDYC#, BUSCHK#) Hold Time (Async)	2.0		CLK's		To RESET Falling Edge
t_{42c}	RESET Configuration Signals (BRDYC#, BUSCHK#) Setup Time (Async)	3.0		CLK's		To RESET Falling

SYMBOL	PARAMETER	MIN	MAX	UNIT	FIGURE	NOTES
						Edge
t_{42d}	RESET Configuration Signals (BRDYC#, BUSCHK#) Hold Time (Sync)	1.0		ns		To RESET Falling Edge
t_{43a}	BF0, BF1, BF2 Setup Time	1.0		ms		To RESET Falling Edge
t_{43b}	BF0, BF1, BF2 Hold Time	2.0		CLK's		To RESET Falling Edge

See notes after Table 5-1.

5.4 DC SPECIFICATIONS

5.4.1 Recommended Operating Conditions

Functional operation of the IDT WinChip C6 processor is guaranteed if the conditions in Table 5-4 are met. Sustained operation outside of the recommended operating conditions may damage the device.

Table 5-4. Recommended Operating Conditions

Parameter	*Min*	*Max*	*Units*	*Notes*
Operating Case Temperature	0	60	°C	
V_{CC} Voltage (3.3 V)	3.135	3.6	V	
V_{CC} Voltage (VRE)	3.45	3.6	V	
V_{IH} - High Level Input Voltage	2.0	V_{cc} + 0.3	V	
V_{IL} - Low level input voltage	-0.3	0.8	V	
I_{OH} - High level output current (typical drive strength)		8.0	mA	@ V = $V_{OH(min)}$
I_{OH} - High level output current (medium drive strength)		16.0	mA	@ V = $V_{OH(min)}$
I_{OL} - Low level output current (typical drive strength)		-8.0	mA	@ V = $V_{OL(max)}$
I_{OL} - Low level output current (medium drive strength)		-16.0	mA	@ V = $V_{OL(max)}$

5.4.2 Maximum Ratings

While functional operation is not guaranteed beyond the operating ranges listed in Table 5-4, the device may be subjected to the limits specified in Table 5-5 without causing long-term damage.

These conditions must not be imposed on the device for a sustained period—any such sustained imposition may damage the device. Likewise exposure to conditions in excess of the maximum ratings may damage the device.

Table 5-5. Maximum Ratings

Parameter	*Min*	*Max*	*Units*	*Notes*
Operating Case Temperature	-65	110	°C	
Storage Temperature	-65	150	°C	
Supply Voltage (V_{CC})	-0.5	4.0	V	
I/O Voltage	-0.5	V_{CC} + 0.5 or V_{CC}(max)	V	

5.4.3 DC Characteristics

Table 5-4. DC Characteristics

Parameter	*Min*	*Max*	*Units*	*Notes*
V_{OH} - High Level Output Voltage (typical drive strength)	2.4	V_{cc}	V	@ I_{oh} = 8mA
V_{OH} - High Level Output Voltage (medium drive strength)	2.4	V_{cc}	V	@ I_{oh} = 16mA
V_{OL} - Low Level Output Voltage (typical drive strength)	0	0.4	V	@ I_{ol} = -8mA
V_{OL} - Low Level Output Voltage (medium drive strength)	0	0.4	V	@ I_{ol} = -16mA
I_L - Input Leakage Current		$\pm$ 15	µA	
I_{LU} - Input Leakage Current for inputs with pull-ups		200	µA	
I_{LD} - Input Leakage Current for inputs with pull-downs		-400	µA	

5.4.4 Power Dissipation

Tables 5.5 and 5.6 give power consumption for the two voltage ranges. A *typical* power value is not given since there is no industry-standard definition of typical and other processor manufacturers do not provide a public test program for their typical values. Thus, typical values from different manufacturers cannot be compared.

However, IDT WinChip C6 processor power consumption at 200 MHz at 3.52V while running the industry-standard Winstone 97 benchmark is 16% less than the 200 MHz 2.8V P55. Details may be obtained from a sales representative.

Table 5-5. Power Consumption IDT WinChip C6 @ 3.52V

Parameter	Min	Max	Units	Notes
I_{DD} - Normal Mode Operating Current				
180 MHz		9.4	Watts	
200 MHz		10.4	Watts	
225 MHz		12.3	Watts	
240 MHz		13.1	Watts	
I_{DD} - StopGrant / AutoHalt Mode Operating Current				*No snooping activity*
180 MHz		2.3	Watts	
200 MHz		2.5	Watts	
225 MHz		2.8	Watts	
240 MHz		3.0	Watts	
I_{DD} - StopClock Mode Operating Current				
180 MHz		0..4	Watts	
200 MHz		0..4	Watts	
225 MHz		0..4	Watts	
240 MHz		0..4	Watts	

Table 5-6. Power Consumption IDT WinChip C6 @ 3.3V

Parameter	Min	Max	Units	Notes
I_{DD} - Normal Mode Operating Power				
180 MHz		8.5	Watts	
200 MHz		9.4	Watts	
225 MHz		10.6	Watts	
240 MHz		11.3	Watts	
I_{DD} - StopGrant / AutoHalt Mode Operating Power				*No snooping activity*
180 MHz		2.0	Watts	
200 MHz		2.2	Watts	
225 MHz		2.4	Watts	
240 MHz		2.6	Watts	
I_{DD} - StopClock Mode Operating Power				
180 MHz		0.3	Watts	
200 MHz		0.3	Watts	
225 MHz		0.3	Watts	
240 MHz		0.3	Watts	

6. MECHANICAL SPECIFICATIONS

The IDT WinChip C6 processor packaged in a 296-pin ceramic pin grid array (CPGA).

The IDT WinChip C6 processor's CPGA package is mechanically compatible with Intel's ceramic and plastic staggered pin grid array (SPGA and PPGA) packages. See Intel's *Pentium Processor Family Developer's Manual*, for comparison.

Figure 6-1. CPGA Pinout

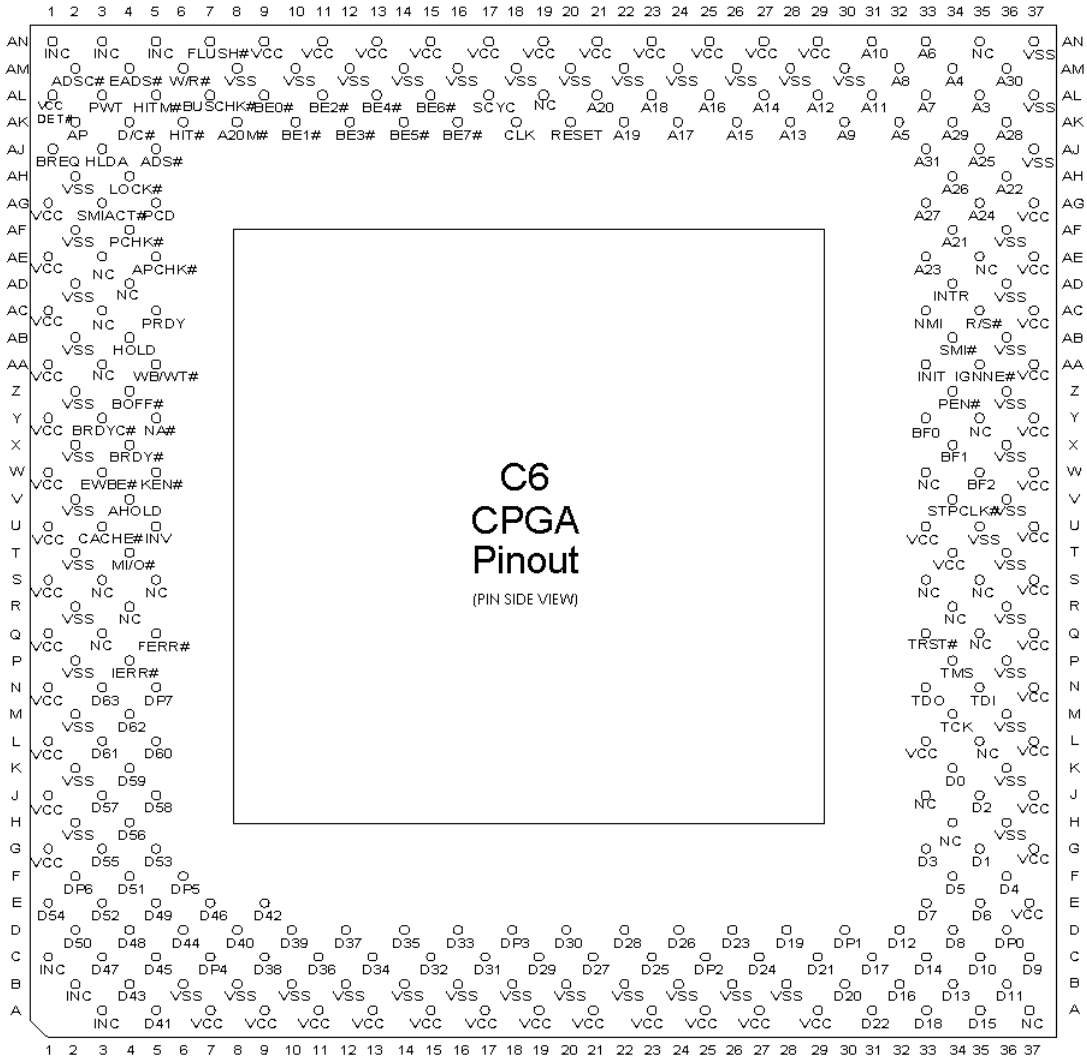


Table 6-1: CPGA Pin Cross Reference

Address		Data		Control		Test		NC	V_{cc}	V_{ss}	Reserved
Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin No.	Pin No.	Pin No.	Pin No.
A3	AL-35	D0	K-34	A20M#	AK-08	TCK	M-34	A-37	A-07	B-06	H-34
A4	AM-34	D1	G-35	ADS#	AJ-05	TDI	N-35	R-34	A-09	B-08	J-33
A5	AK-32	D2	J-35	ADSC#	AM-02	TDO	N-33	S-33	A-11	B-10	L-35
A6	AN-33	D3	G-33	AHOLD	V-04	TMS	P-34	S-35	A-13	B-12	Q-03
A7	AL-33	D4	F-36	AP	AK-02	TRST#	Q-33	W-33	A-15	B-14	Q-35
A8	AM-32	D5	F-34	APCHK#	AE-05			W-35	A-17	B-16	R-04
A9	AK-30	D6	E-35	BE0#	AL-09			AL-19	A-19	B-18	S-03
A10	AN-31	D7	E-33	BE1#	AK-10			AN-01	A-21	B-20	S-05
A11	AL-31	D8	D-34	BE2#	AL-11			AN-35	A-23	B-22	Y-35
A12	AL-29	D9	C-37	BE3#	AK-12				A-25	B-24	AA-03
A13	AK-28	D10	C-35	BE4#	AL-13				A-27	B-26	AC-03
A14	AL-27	D11	B-36	BE5#	AK-14			***INC***	A-29	B-28	AD-04
A15	AK-26	D12	D-32	BE6#	AL-15			A-03	E-37	H-02	AE-03
A16	AL-25	D13	B-34	BE7#	AK-16			B-02	G-01	H-36	AE-35
A17	AK-24	D14	C-33	BF0	Y-33			C-01	G-37	K-02	
A18	AL-23	D15	A-35	BF1	X-34			AN-03	J-01	K-36	
A19	AK-22	D16	B-32	BF2	W-35			AN-05	J-37	M-02	
A20	AL-21	D17	C-31	BOFF#	Z-04				L-01	M-36	
A21	AF-34	D18	A-33	BRDY#	X-04				L-33	P-02	
A22	AH-36	D19	D-28	BRDYC#	Y-03				L-37	P-36	
A23	AE-33	D20	B-30	BREQ	AJ-01				N-01	R-02	
A24	AG-35	D21	C-29	BUSCHK#	AL-07				N-37	R-36	
A25	AJ-35	D22	A-31	CACHE#	U-03				Q-01	T-02	
A26	AH-34	D23	D-26	CLK	AK-18				Q-37	T-36	
A27	AG-33	D24	C-27	D/C#	AK-04				S-01	U-35	
A28	AK-36	D25	C-23	DP0	D-36				S-37	V-02	
A29	AK-34	D26	D-24	DP1	D-30				T-34	V-36	
A30	AM-36	D27	C-21	DP2	C-25				U-01	X-02	
A31	AJ-33	D28	D-22	DP3	D-18				U-33	X-36	
		D29	C-19	DP4	C-07				U-37	Z-02	
		D30	D-20	DP5	F-06				W-01	Z-36	
		D31	C-17	DP6	F-02				W-37	AB-02	
		D32	C-15	DP7	N-05				Y-01	AB-36	
		D33	D-16	EADS#	AM-04				Y-37	AD-02	
		D34	C-13	EWBE#	W-03				AA-01	AD-36	
		D35	D-14	FERR#	Q-05				AA-37	AF-02	

Address		**Data**		**Control**		**Test**		**NC**	**V_{cc}**	**V_{ss}**	**Reserved**
Pin Name	*Pin No.*	*Pin Name*	*Pin No.*	*Pin Name*	*Pin No.*	*Pin Name*	*Pin No.*	*Pin No.*	*Pin No.*	*Pin No.*	*Pin No.*
		D36	C-11	FLUSH#	AN-07				AC-01	AF-36	
		D37	D-12	HIT#	AK-06				AC-37	AH-02	
		D38	C-09	HITM#	AL-05				AE-01	AJ-37	
		D39	D-10	HLDA	AJ-03				AE-37	AL-37	
		D40	D-08	HOLD	AB-04				AG-01	AM-08	
		D41	A-05	IERR#	P-04				AG-37	AM-10	
		D42	E-09	IGNNE#	AA-35				AN-09	AM-12	
		D43	B-04	INIT	AA-33				AN-11	AM-14	
		D44	D-06	INTR	AD-34				AN-13	AM-16	
		D45	C-05	INV	U-05				AN-15	AM-18	
		D46	E-07	KEN#	W-05				AN-17	AM-20	
		D47	C-03	LOCK#	AH-04				AN-19	AM-22	
		D48	D-04	M/IO#	T-04				AN-21	AM-24	
		D49	E-05	NA#	Y-05				AN-23	AM-26	
		D50	D-02	NMI	AC-33				AN-25	AM-28	
		D51	F-04	PCD	AG-05				AN-27	AM-30	
		D52	E-03	PCHK#	AF-04				AN-29	AN-37	
		D53	G-05	PEN#	Z-34						
		D54	E-01	PRDY	AC-05						
		D55	G-03	PWT	AL-03						
		D56	H-04	RESET	AK-20						
		D57	J-03	R/S#	AC-35						
		D58	J-05	SCYC	AL-17						
		D59	K-04	SMI#	AB-34						
		D60	L-05	SMIACT#	AG-03						
		D61	L-03	STPCLK#	V-34						
		D62	M-04	W/R#	AM-06						
		D63	N-03	WB/WT#	AA-05						
				VCC2DET#	AL01						

Figure 6-2. CPGA Package Dimensions

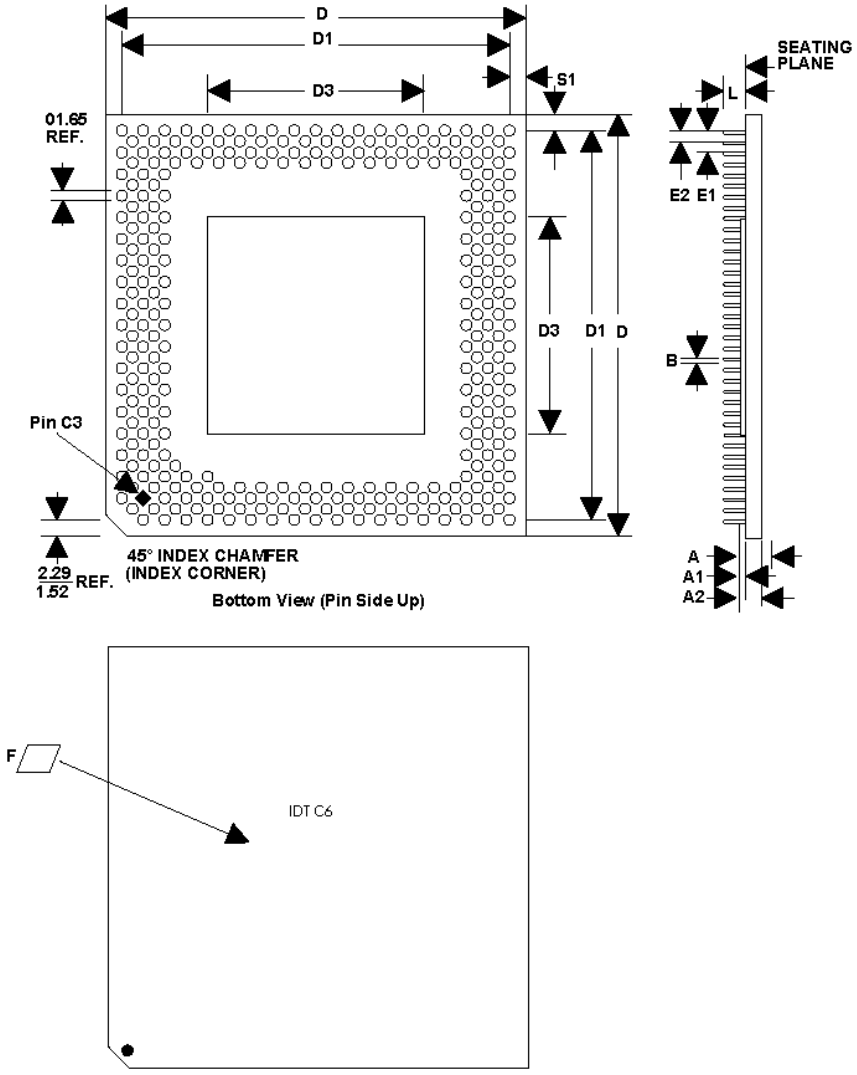


Figure 6-3. CPGA Dimensions

	Millimeters			Inches		
Symbol	**Min**	**Max**	**Notes**	**Min**	**Max**	**Notes**
A	2.62	2.97		0.103	0.117	
A1	0.69	0.84	Lid	0.027	0.033	Lid
A2	3.31	3.81	Lid	0.130	0.150	Lid
B	0.43	0.51		0.017	0.020	
D	49.28	49.78		1.940	1.960	
D1	45.59	45.85		1.795	1.805	
e1	2.29	2.79		0.090	0.110	
L	3.05	3.30		0.120	0.130	
N	296		Lead Count	296		Lead Count
S1	1.52	2.54		0.060	0.100	

7. THERMAL SPECIFICATIONS

7.1 INTRODUCTION

The IDT WinChip C6 is specified for operation with device case temperatures in the range of 0°C to 60°C. Operation outside of this range will result in functional failures and potentially damage the device.

Care must be taken to ensure that the case temperature remains within the specified range at all times during operation. An effective heat sink with adequate airflow is therefore a requirement during operation.

7.2 TYPICAL ENVIRONMENTS

Typical thermal solutions involve three components: a heatsink, an interface material between the heatsink and the package, and a source of airflow. The best thermal solutions rely on the use of all three components. To the extent that any of these components are not used, the other components must be improved to compensate for such omission. In particular, the use of interface material such as thermal grease, silicone paste, or graphite impregnated paper can make a 40°C difference in the case temperature (see Table 7-4). Likewise, the imposition of airflow is realistically a requirement (see Table 7-1).

7.3 MEASURING T_C

The *Intel Pentium Processor Developer's Manual* describes proper thermal measuring techniques in detail in Chapter 10.

The case temperature (T_C) should be measured by attaching a thermocouple to the center of the IDT WinChip C6 package. The heat produced by the processor is very localized so measuring the case temperature anywhere else will underestimate the case temperature.

The presence of a thermocouple is inherently invasive; effort must be taken to minimize the effect of the measurement. Typically the thermocouple should be attached to the processor through a small hole drilled in the heatsink. Thermal grease should be used to ensure that the thermocouple makes good contact with the package.

Physical Test Conditions

Case temperature measurements should be made in the worst case operating environments. Ideally, systems should be maximally configured, and tested at the worst-case ambient temperature.

Test Patterns

During normal operation the processor attempts to minimize power consumption. Consequently, normal power consumption is much lower than the maximum power consumption. Thermal testing should be done while running software which causes the processor to operate at its thermal limits. Your IDT sales representative can supply you with an executable program which will maximize power consumption.

7.4 ESTIMATING T_C

The IDT WinChip C6 processor's case temperature can be estimated based on the general characteristics of the thermal environment. This estimate is not intended as a replacement for actual measurement.

Case temperature can be estimated from Tables 7-1 and 7-2 below, where,

$T_A \equiv$ Ambient Temperature

$T_C \equiv$ Case Temperature

$\theta_{CA} \equiv$ case-to-ambient thermal resistance

$\theta_{JA} \equiv$ junction-to-ambient thermal resistance

$\theta_{JC} \equiv$ junction-to-case thermal resistance

$P \equiv$ power consumption (Watts)

and,

$$T_J = T_C + (P * \theta_{JC})$$

$$T_A = T_J - (P * \theta_{JA})$$

$$T_A = T_C - (P * \theta_{CA})$$

$$\theta_{CA} = \theta_{JA} - \theta_{JC}$$

Table 7-1. CPGA q_{JC} and q_{JA}

		q_{JA} (°C/Watt) vs. Laminar Airflow (linear ft/min)					
Heat Sink in Inches (height)	q_{JC} **(°C/Watt)**	**0**	**100**	**200**	**400**	**600**	**800**
0.25	2.0	10.3	9.2	7.8	5.7	4.8	4.2
0.35	2.0	10.0	8.7	7.2	5.2	4.5	4.0
0.45	2.0	8.6	8.2	6.5	4.8	4.1	3.7
0.55	2.0	9.3	7.7	5.9	4.4	4.8	3.5
0.65	2.0	8.9	7.2	5.5	4.2	3.6	3.3
0.80	2.0	8.2	6.5	5.1	4.0	3.4	3.2
1.00	2.0	7.5	5.9	4.8	3.8	3.3	3.0
1.20	2.0	7.1	5.5	4.5	3.6	3.2	3.0
1.40	2.0	6.6	5.1	4.2	3.4	3.1	2.9
No Heat Sink	2.5	15.6	14.3	12.9	10.0	8.6	7.7

Environment: these estimates assume the use of thermal grease between the processor and the heatsink. Heatsinks are 1.95" square.

Table 7-2. IDT WinChip C6 Power Consumption

Frequency (MHz)	**VCC (Volts)**	**Max Power (Watts)**
180	3.3	8.1
	3.52	9.4
200	3.3	8.9
	3.52	10.4
225	3.3	10.0
	3.52	11.7
240	3.3	10.7
	3.52	12.5

Example:

For a system with:

- 200 MHz IDT WinChip C6 @ 3.52V
- T_A = 35°C, (ambient air around the heatsink)

- 0.65” Heat Sink w/ 400 fpm laminar air flow

The case temperature can be calculated as

$T_C = T_A + (P * \theta_{CA})$

$T_C = T_A + (P * (\theta_{JA} - \theta_{JC}))$

$T_C = 35 + (10.38 * (4.2-2.0))$

$T_C = 57.84$ °C

This T_C is adequate, but note that the 35°C ambient temperature is probably not a realistic worst case environment.

7.5 RECOMMENDED THERMAL SOLUTIONS

Table 7-3 below offers several off-the-shelf thermal solutions which are known to provide adequate cooling for the IDT WinChip C6 at the specified chassis internal ambient temperature. These kits were measured with silicone paste as an attach media except as noted.

For reference, Table 7-4 shows the effectiveness of the Acadia Technology CPB-15502-02 heatsink/fan with different attach media.

Table 7-3. Heatsink / Fan Kits

Manufacturer	***Part Number***	***Max Chassis T_A***			
		180/3.3V	*180/3.52V*	*200/3.3V*	*200/3.52V*
Aavid[1]	26437	51.0°C	47.6°C	48.3°C	43.1°C
Acadia Technology	CPB-15502-02	56.3°C	52.7°C	53.0°C	49.0°C
Cooler Master	TP5-5015	53.0°C	49.2°C	51.0°C	46.0°C
Cooler Master	TP5-5020	52.3°C	49.0°C	51.1°C	46.0°C

[1] *The Aavid 26437 has an attached Graphite sheet. This eliminates the need for thermal grease or epoxy.*

Table 7-4. Interface Material Effectiveness (CPB-15502-02)

Manufacturer	Part Number	Max Chassis T_A (180/3.3V)	Type
Omega	OT-201	56.3°C	Silicone Paste
-	-	45.0°C	Graphite Paper
-	-	13.6°C	Nothing

7.6 CONTACTS

Table 7-5. Heatsink / Fan Kit Contacts

Manufacturer	Contact	Phone #	Address
Aavid	Chris Chapman	(603)528-3400	One Kool Path PO Box 400 Laconia, NH 03247
Acadia Technology	Chung Bao	(408)747-1349	1010 Morse Ave. Suite 4 Sunnyvale, CA 94089
Cooler Master	Jerry Chen	(510)770-0149	115 Fourier Ave. Fremont, CA 94539

Table 7-6. Interface Material Contacts

Manufacturer	Contact	Phone #	Address
Omega Technologies		(203)359-1660	One Omega Drive Stamford, CT 06907

APPENDIX A. MACHINE SPECIFIC REGISTERS

A.1 GENERAL

Tables A-1 and A-2 summarize the IDT WinChip C6 processor machine-specific registers (MSRs). Further description of each MSR follows the table. MSRs are read using the RDMSR instruction and written using the WRMSR instruction.

There are four basic groups of MSRs (not necessarily with contiguous addresses). Other than as defined below, a reference to an undefined MSR causes a General Protection exception.

1. Those that are very similar in function (but possibly different in some detail) to the Pentium processor MSRs. Generally, the same MSR address is used. These registers can have some utility to low-level programs (like BIOS).

   Note that some of the first sixteen Pentium MSRs (addresses 0 to 15) have no function in the IDT WinChip C6 processor. These MSRs do not cause a GP when used on the IDT WinChip C6 processor; instead, reads to these MSRs return zero, and writes are ignored.

2. Memory Configuration Registers which use MSR addresses that are not used on the Pentium processor. These MSRs define memory ranges with associated attributes. These MSRs are similar to the Pentium Pro processor MTRRs and to the Cyrix 6x86MX processor's ARR registers.

3. MSRs used for cache and TLB testing. These use MSR addresses that are not used on the Pentium. These test functions are very low-level and complicated to use. They are not documented in this datasheet but the information will be provided to customers given an appropriate justification.

4. There are some undocumented internal-use MSRs used for low-level hardware testing purposes. Attempts to read or write these undocumented MSRs cause unpredictable and disastrous results; so don't use MSRs that are not documented in this datasheet!

MSRs are not reinitialized by the bus INIT interrupt; the setting of MSRs is preserved across INIT.

Table A-1. Category 1 MSRs (Functionally Similar to Pentium)

MSR	MSR Name	ECX	EDX	EAX	Type	Notes
no MSR		00h-01h	n/a	n/a	RW	1
TR1	Test Register 1	02h	n/a	Control bits	RW	3
no MSR		03h-0Dh	n/a	n/a	RW	1
TR12	Test Register 12	0Eh	n/a	Control bits	RW	3
no MSR		0Fh	n/a	n/a	RW	1
TSC	Time Stamp Counter	10h	Count[63:32]	Count[31:0]	RW	2
EC_CTRL	Event Counter Control	11h	n/a	Control bits	RW	2
EC0	Event Counter 0	12h	Count[39:32]	Count[31:0]	RW	2
EC1	Event Counter 1	13h	Count[39:32]	Count[31:0]	RW	2
FCR	Feature Control Reg	107h	n/a	FCR value	RW	4
FCR2	Feature Control Reg 2	108h	FCR2_Hi	FCR2 value	RW	5
FCR3	Feature Control Reg 3	109h	FCR3_Hi	FCR3 value	WO	5

Notes

1. Pentium processors have MSRs at these addresses. On the IDT WinChip C6 processor, reads to these addresses return zero and writes are ignored.

2. Functionally similar to the same Pentium MSR. However, some minor details are different.

3. A subset of the same Pentium MSR—only those bits meaningful to the IDT WinChip C6 processor have any effect; the rest read as 0 and are ignored when written.

4. Conceptually similar to the Pentium MSR 0Eh (“TR12”) that controls detailed functions like disabling the caches. The IDT WinChip C6 processor controls are different, thus the FCR is placed at a different address than the Pentium TR12 register.

5. FCR2 and FCR3 provide system software with the ability to specify the Vendor ID string returned by the CPUID instruction .

Table A-2. Category 2 MSRs (Memory Configuration Registers)

MSR	ECX	EDX	EAX	Type	Notes
MCR 0	110h	Base Address [31:12]	Address Mask [31:12] & Ctrl Value [11:0]	WO	
MCR 1	111h	same as above	same as above	WO	
MCR 2	112h	same as above	same as above	WO	
MCR 3	113h	same as above	same as above	WO	
MCR 4	114h	same as above	same as above	WO	
MCR 5	115h	same as above	same as above	WO	
MCR 6	116h	same as above	same as above	WO	
MCR 7	117h	same as above	same as above	WO	
MCR_CTRL	120h	-	control value	WO	

A.2 CATEGORY 1 MSRS

02h: TR1 (Pentium Processor Parity Reversal Register)

31:2	1	0
Reserved (Ignored on write; returns 0 on read)	NS	*Res*
30	1	1

Both the IDT WinChip C6 processor and the Intel Pentium processor have a MSR 02 bit 1 that performs the same function on an IDT WinChip C6 processor as on a Pentium processor. Other bits return 0 when read and are ignored when written.

NS: 0 = Assert #IERR and cause Shutdown on internal parity error

1 = Assert #IERR and *do not* cause Shutdown on internal parity error

0Eh: TR12 (Pentium Processor Feature Control)

31:10	9	8:7	6	5:4	3	2:0
Reserved (Ignored on write; returns 0 on read)	ITR	*Res*	AHD	*Res*	CI	*Res*
22	1	2	1	2	1	3

Both the IDT WinChip C6 processor and Pentium processor have MSR 0E bits 3, 6, and 9 that perform the same functions on an IDT WinChip C6 processor as on a Pentium processor. Other bits return 0 when read and are ignored when written.

CI: 0 = Ignored.

1 = Same as for the Pentium processor: Cache line fills (to both caches) are suppressed; all cache misses are performed as single transfer cycles. The PCD output pin is not affected. Note that the caches are not flushed.

AHD: 0 = Ignored.

1 = Same as for the Pentium processor: disable Autohalt Powerdown function

ITR: 0 = Ignored.

1 = Same as for the Pentium processor: enable SMM I/O Restart function

10h: TSC (Time Stamp Counter)

Both the IDT WinChip C6 processor and the Pentium processor have a 64-bit MSR that materializes the Time Stamp Counter (TSC). Both systems increment the TSC once per processor clock.

On the IDT WinChip C6 processor, the MSR (and the value returned by the RDTSC instruction) is an alias for the internal event-counter MSRs (CTR0/CTR1). In normal system operation, the TSC register counts internal processor clocks.

However, if the user code changes the item that CTR0 or CTR1 is counting (see the counter MSR descriptions), then the TSC register also changes what it is counting. There is no practical reason why the machine-specific event counting should be changed by software. On a Pentium processor, the TSC is a separate counter from CTR0/CTR1.

11h: CESR (Control & Event Select Register)

31:24	23:16	15:8	7:0
Reserved	CTR1 Control	*Reserved*	CTR0 Control
8	8	8	8

Both the IDT WinChip C6 processor and Pentium have an MSR that contains bits defining the behavior of the two hardware event counters: CTR0 and CTR1.

The CTR0 and CTR1 control fields define which of several possible events can be counted for each counter. Each counter has the same set of possible events.

The events that can be counted, and their identification numbers, are different from the Pentium processor events (which are different from the Pentium Pro processor events). The Pentium processor has only six bits to identify the event counter, but has additional controls (such as event versus clock counting) in bits 9-6 of each control field.

The CESR should be written before the associated CTR0 and CTR1 are written to initialize the counters. The counts are not necessarily perfectly exact; the counters are intended for use over a large number of events and may differ by one or two counts from what might be expected.

Most counter events are internal implementation-dependent debug functions having no meaning to software. The counters that can have end-user utility are:

Event	Description
0	Internal clocks (default event for CTR0)
1	valid cycles reaching writebacks
2	X86 instructions
71	data read cache misses
74	data write cache misses
99	instruction fetch cache miss

12h-13h: CTR0 & CTR1 (Event Counters 0 & 1)

Both the IDT WinChip C6 processor and Pentium processor have two 40-bit hardware event counters (bits 31:8 of EDX are ignored).

107h: FCR (Feature Control Register)

The FCR controls the major optional feature capabilities of the IDT WinChip C6 processor. It is analogous to the Pentium processor TR12 (actually MSR 0Eh) that controls things like BTB enable, cache enable, and so forth. The Cyrix 6x86MX processor's CCRs (Configuration Control Registers) perform a similar function, as does the AMD-K6 processor's HDCR MSR.

Table A-2 contains the bit values for the FCR. The defaults settings shown for the FCR bits are not necessarily exact. The actual settings can be changed as part of the manufacturing process and thus a particular IDT WinChip C6 processor version can have slightly different default settings than shown here. All reserved bit values of the FCR must be preserved by using a read-modify-write sequence to update the FCR.

Table A-2. FCR Bit Assignments

Bit	Name	Description	Default
0		*Reserved*	0
1	ECX8	Enables CPUID reporting CX8	0
2	EIERRINT	Enables INT18 (Machine Check) for internal errors	0
3	DPM	Disable dynamic power management	0
4	DMCE	Disables Machine Check Exception	0
5	DSTPCLK	Disables supporting STPCLK	0
6	ELINEAR	Enables Linear Burst Mode	0
7	DSMC	Disables strict cache coherency (self-modifying-code)	0
8	EDCTLB	Enables D-Cache for updates to PDE/PTE	0
9	EMMX	Enables MMX-compatible instructions.	1
10		*Reserved*	0
11	DPDC	Disables Page Directory cache	0
12		*Reserved*	0
13	DIC	Disables I-Cache.	0
14	DDC	Disables D-Cache.	0
15	DNA	Disables bus pipelining (NA response)	0
16	ERETSTK	Enables CALL-RET Stack operation	1
17		*Reserved*	0
18		*Reserved*	0
19		*Reserved*	0
20		*Reserved*	0
21		*Reserved*	0
22:25	SID	Stepping ID	0
26		*Reserved*	0
27		*Reserved*	0
28	ADIVFLG	Selects alternate EFLAGS results for divide	0
29	DCPUID	Disables CPUID instruction	0
30	EMOVTR	Enables move-to-test register instructions	0
31		*Reserved*	0

ECX8: 0 = The CPUID instruction does not report the presence of the CMPXCHG8B instruction (CX8 = 0). The instruction actually exists and operates correctly, however.

1 = The CPUID instruction reports that the CMPXCHG8B instruction is supported (CX8 = 1).

EIERRINT: 0 = Normal internal error behavior (IERR and possible Shutdown).

1 = Causes INT18 instead of Shutdown for internal error.

DPM: 0 = Normal dynamic power management behavior.

1 = Disables all dynamic power management.

DMCE: 0 = Machine Check exception enabled.

1 = Disable Machine Check exception; a bus check or internal error condition does not cause an exception.

DSTPCLK: 0 = STPCLK interrupt properly supported.

1 = Ignores SPCLK interrupt.

ELINEAR: 0 = Interleaved burst ordering is enabled.

1 = Linear burst ordering is enabled.

DSMC: 0 = Strict cache coherency is enabled to support Pentium processor style self-modifying code.

1 = Disables strict cache coherency. I-cache/D-cache coherent only if branch is taken after store instruction which modifies instructions to be executed subsequently.

EDCTLB: 0 = Updates to the accessed and dirty bits in PDE/PTE entries are performed using the locked read-modify-write semantics which flushes the data from the D-Cache (like Pentium processor).

1 = Enables D-Cache for updates to accessed and dirty bits in PDE/PTE.

EMMX: 0 = Disables MMX-compatible instructions: they decode as invalid instructions.

1 = Enables MMX-compatible instructions.

DPDC: 0 = Enables use of internal Page Directory Cache.

1 = Disables use of internal Page Directory Cache.

DIC: 0 = Enables use of I-Cache.

1 = Disables use of I-Cache: cache misses are performed as single transfer bus cycles, PCD is de-asserted. This overrides any setting of CR0.CD and CR0.NW.

DDC: 0 = Enables use of D-Cache.

1 = Disables use of D-Cache: same semantics as for DIC except for D-Cache.

DNA: 0 = Enables bus pipelining operation.

1 = Disables bus pipelining operation: bus signal NA is ignored.

ERETSTK: 0 **=** Disables CALL-RETurn stack function.

1 = Enables CALL-RETurn stack function: RET branch target prediction is performed.

ADIVFLG: 0 = The EFLAGS setting after integer divide instructions is unique to the IDT WinChip C6 processor. These flag settings are undefined in the x86 architecture.

1 = The EFLAGS setting after integer divide of 5 by 2 is the same as on a Pentium processor (this is used by some old code to identify the processor)

DCPUID: 0 **=** The CPUID instruction is supported.

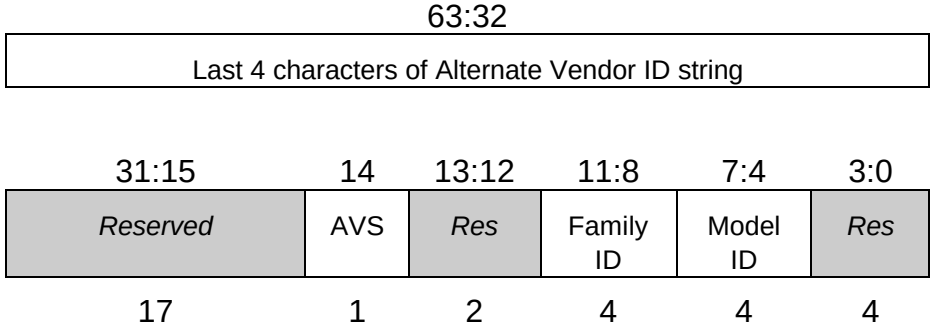
1 = The CPUID instruction is disabled and causes an invalid instruction exception.

EMOVTR: 0 = The Intel486 move-to/from-test register instructions are not supported and their behavior is the same as on a Pentium processor (invalid instruction exception).

1 = The test register instructions do not cause an invalid instruction exception but rather are treated as NOPs.

108h: FCR2 (Feature Control Register 2)

This MSR contains more feature control bits — many of which are undefined. It is important that all reserved bits are preserved by using a read-modify-write sequence to update the MSR.



AVS: 0 = The CPUID instruction vendor ID is "CentaurHauls"

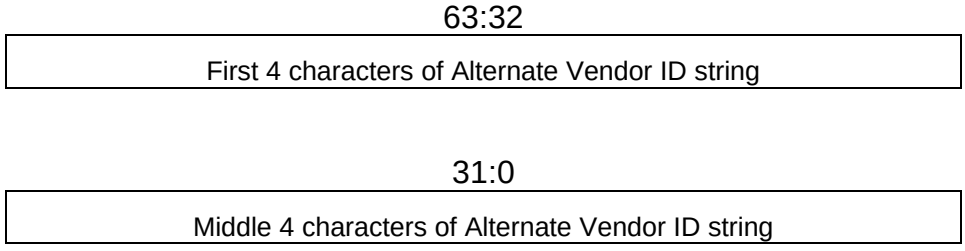
1 = The CPUID instruction returns the alternate Vendor ID. The first 8 characters of the alternate Vendor ID are stored in FCR3 and the last 4 characters in FCR2[63:32]. These 12 characters are undefined after RESET and may be loaded by system software using WRMSR.

Family ID: This field will be returned as the family ID field by subsequent uses of the CPUID instruction

Model ID: This field will be returned as the model ID field by subsequent uses of the CPUID instruction

109h: FCR3 (Feature Control Register 3)

This MSR contains the first 8 characters of the alternate Vendor ID. The alternate Vendor ID is returned by the CPUID instruction when FCR2[AVS] is set to 1. FCR3 is a write-only MSR.



A.3 MEMORY CONFIGURATION REGISTERS

General

The IDT WinChip C6 processor provides extensions over the P55 to define variable size memory ranges with associated special attributes. These Memory Configuration Registers (MCRs) are similar to the MTRRs of the Pentium Pro processor and the Address Region Registers (ARRs) of the Cyrix 6x86MX processor. All of these approaches perform similar functions but differ in specifics.

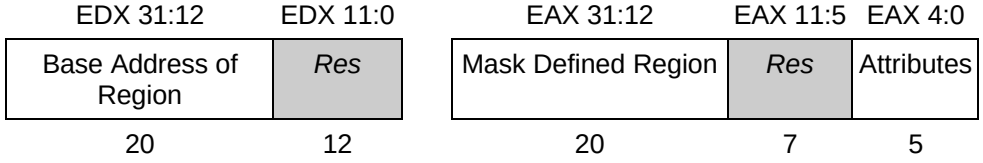
The basic function performed is to define memory regions and special attributes for these regions such as byte-combining and weakly ordered stores. These special attributes are deviations from formal x86 architectural behavior but, in practice, work fine for specific memory regions in a PC architecture. The advantage of these special attributes is improved performance for the affected memory regions.

Memory Configuration Registers

The IDT WinChip C6 processor has eight MCRs, each appearing as a 64-bit MSR. The default value at reset is zero for all fields. This value causes all memory to have normal x86 attributes of no byte-combining and strongly ordered writes.

Note that the MCRs are *write-only*.

The MCR format is:



Base

This is the starting physical address of the memory region. Each definable memory region starts at a 4-KB page boundary; thus the low order 12 bits of the address are ignored.

Mask

This is a bit mask defining the size of the memory region. A memory region hit exists for a MCR if:

Mask address AND memory address = Base address AND mask address in all bit positions (31:12)

Example 1.

For example, consider the memory range from 0x000A0000 to 0x000BFFFF. This is most efficiently done by masking off the low order bits that constitute the range.

Viewing the addresses in binary:

```
0x000A0000  = 0000 0000 0000 1010 0000 0000 0000 0000

0x000BFFFF  = 0000 0000 0000 1011 1111 1111 1111 1111
```

Notice that the upper 15 bits are identical, whereas the lower 17 bits define the address within the range. So a single MRD can describe this range as:

```
MCR Base  = 0000 0000 0000 1010 0000 or 0x000A0

MCR Mask  = 1111 1111 1111 1110 0000 or 0xFFFE0
```

Note that the lower twelve bits of the mask and base are ignored in the match calculation.

Example 2.

Consider a more complex scenario where a range from 0x00080000 to 0x00017FFF is required. Shown in binary as:

```
0x00080000  = 0000 0000 0000 1000 0000 0000 0000 0000

0x0017FFFF  = 0000 0000 0001 0111 1111 1111 1111 1111
```

In this case, the upper 11 bits are identical throughout the range. The next two bits vary, but not all combinations are part of the desired range. The range has to be broken down into two ranges that have common base bits where all combinations of the lower order bits are within the original region. This implies:

```
0x00080000  = 0000 0000 0000 1000 0000 0000 0000 0000

0x000FFFFF  = 0000 0000 0000 1111 1111 1111 1111 1111
```

and

```
0x00100000  = 0000 0000 0001 0000 0000 0000 0000 0000

0x0017FFFF  = 0000 0000 0001 0111 1111 1111 1111 1111
```

These map into MCRs:

```
MCR0 Base = 0000 0000 0000 1000 0000 or 0x00080
MCR0 Mask = 1111 1111 1111 1000 0000 or 0xFFF80
```

and

```
MCR1 Base = 0000 0000 0001 0000 0000 or 0x00100
MCR1 Mask = 1111 1111 1111 1000 0000 or 0xFFF80
```

Attributes

There are five bits of attribute control defined for each memory region as described in Table A.6-1

Table A.6-1

Bit	Description	Default	Notes
0	Enables byte combining on non-stack non-string writes.	0	
1	Enables byte combining on string instruction writes.	0	
2	Enables byte combining on stack instruction writes.	0	
4:3	Defines write-ordering strategy for memory region. 00: All writes strongly ordering 01: String writes weakly ordered 10: Stack writes weakly ordered 11: All writes weakly ordered	00	

MCR Control Register

The MCR_CTRL MSR controls various pervasive behaviors of byte-combining and write-ordering. The store combining definitions for string and non-stack and non-string are defined in the following table.

Table A.7-1

Bit	Description	Default	Notes
1:0	Store combining definition for non-stack & non-string. 00: Forward Combining 01: Forward and Overlapped Combining 10: Forward and Reverse Combining 11: Forward, Reverse, and Overlapped Combining	00	
3:2	Store combining definition for string. 00: Forward Combining 01: Forward and Overlapped Combining 10: Forward and Reverse Combining 11: Forward, Reverse, and Overlapped Combining	00	
4	Weak Write-Ordering Enable	0	
5:19	*Reserved*	0	
24:20	*Reserved*	11111	

Store-Combining

The IDT WinChip C6 processor's store-combining feature allows multiple stores into be combined into a single bus store. This is permissible if the stores are destined to the same 8-byte memory address.

Store-combining can greatly reduce the memory bandwidth requirements of writes that miss the cache. However, if the associated writes are destined to memory mapped I/O locations, problems can arise.

For example, if an ISA bus 8-bit device is controlled with a data register at address 0 and a control register at address 1, and the control register must be written before the data can be written, it is possible for the order of writes to be changed if byte-combining is inappropriately configured.

If, for example, the device writes to address 1 and then to address 0 and the two are combined, a 16-bit word will go to the ISA bus where it will be split for the 8-bit device into two writes. Unfortunately, most ISA bridges split 16-bit operands into two transfers with the low byte first. Consequently, the order of the two writes are reversed.

To eliminate this problem, the IDT WinChip C6 processor is very configurable. Aside from disabling byte-combining, it is also possible to limit the type of instructions that are allowed to combine. Further, it is possible to prevent the processor from combining in reverse address order. Lastly, it is possible to prevent the processor from combine-matching (overlapping) byte addresses.

Weak Write-Ordering

The Pentium processor normally mandates that writes occur in the memory hierarchy in the same order as they occur in the code execution. This is termed strong write-ordering. This restriction can be a performance impact in that it blocks processor execution when a store hits an E or M line in the cache if another store is waiting to be retired on the bus.

Normally systems do not require strong write-ordering unless they have bus-mastering I/O devices that use memory mapped I/O for control purposes. (Most DMA devices are slaves, and do not use memory-mapped I/O. The floppy controller, for example, is a DMA device, but does not use memory-mapped I/O.)

However, since there are devices that could not perform correctly with weak write ordering, this function should only be used in systems where the type of peripherals are tightly controlled and known to not require strongly ordered writes. Weak write ordering should never be turned on by a generic BIOS, for example.

Combining Ranges

It is possible to describe fairly complex ranges with a few descriptors. Generally, this does not involve overlapping MCRs. However, overlapping ranges are permitted and their behavior is useful in some cases.

The behavior of an access to a given memory location is defined by the logical OR of the attribute bits of the MCRs it matches. So, if a memory location does not match any MCRs, its aggregate attribute is 0. If, on the other hand, it matches two MCRs, one with an attribute of 0x18 and the other with an attribute of 0x02, the aggregate attribute is 0x1A. This enables weak writes on all accesses and allows store-combining on stack accesses.

APPENDIX B. COMPATIBILITY

B.1 INTRODUCTION

In general, the IDT WinChip C6 processor is exactly compatible with both the bus and software-visible architecture of the Intel Pentium processor.

An IDT WinChip C6 processor can plug into existing Intel Pentium-based PC system boards and operate without requiring change to the system hardware. Also, an IDT WinChip C6 processor can run all existing industry-standard PC object-code operating systems and application programs.

However, all processors developed for use in PCs (“x86” processors) have some minor incompatibilities in low-level implementation-dependent functions. For example, it is possible to write esoteric software for the Intel 486 processor (cache tests, for example) that produces different results when run on the supposedly compatible Intel Pentium processor.

Similarly, there are low-level incompatibilities between the Intel Pentium and the Intel Pentium Pro processors. Similarly, there are low-level incompatibilities among all x86 “clone” processors such as the AMD-K6 and the Cyrix 6x86MX processors. The IDT WinChip C6 processor has similar low-level differences with the various Intel and x86 clone processors.

Fortunately, these technical incompatibilities among x86 implementations are in areas that have no meaningful use to most programs, and that are well-understood by software developers (and are thus avoided). Therefore, in practice, these types of differences pose no real barriers to program compatibility across various implementations.

This appendix summarizes areas where the IDT WinChip C6 processor differs in behavior from the Intel Pentium processor. These differences are generally “don’t cares”; that is, they are transparent to system hardware and programs.

B.2 BUS COMPATIBILITY

Bus Cycle Activity

When compared cycle by cycle, the IDT WinChip C6 and Intel Pentium processors do not have exactly the same bus cycles. This is anticipated, unavoidable, and desirable (the IDT WinChip C6 processor provides increased bus performance). This difference results from the IDT WinChip C6 processor's different internal architecture and larger cache. This is not an issue because the Intel Pentium processor itself varies at different frequencies and all other competitive processors also have differing bus cycle activity.

Bus Alignment

Although the Intel Pentium processor has a 64-bit bus, it splits loads and stores that cross 32-bit boundaries. The IDT WinChip C6 processor splits memory loads and stores at 64-bit boundaries. However, I/O reads and writes are split on 32-bit boundaries.

Like the Pentium processor, when split cycles are required, the IDT WinChip C6 processor performs the higher address' access first, then performs the lower address' access.

The IDT WinChip C6's bus alignment is not anticipated to be a compatibility issue. The Cyrix 6x86 and AMD-K5 processors also split memory loads and stores at 64-bit boundaries. (Also, the Cyrix 6x86 performs the low addressed access first except for 32-bit misaligned I/O's in which case the higher access is performed first.)

Snoop Responsibility Pickup

The Pentium processor assumes responsibility for incoming cache lines at the point where it determines the cacheability of the line. The cacheability is determined at the first assertion of NA# or BRDY# for the cycle. Subsequent to cacheability determination, the processor will respond to snoops for that line even though it is not completely read into the processor.

The IDT WinChip C6 processor generally mimics this behavior. However, if a bus cycle is pipe-lined over a previous cycle and gets a BOFF# assertion before its cacheability for the second line has been determined, the IDT WinChip C6 processor assumes responsibility for subsequent snoops to *both* lines when cacheability is determined for the *first* retried line.

Descriptor Updates

The exact size of the locked bus transactions used to update the accessed bit in non-accessed descriptors is slightly different between the Pentium processor and the IDT WinChip C6 processor.

8-Byte Writes

Eight-byte aligned locked 8-byte read-modify-write sequences (a locked CMPXCHG8B instruction) are performed as one locked 8-byte read followed by two locked 4-byte writes. The Pentium processor performs the write as an 8-byte locked write.

TLB Retries

Intel processors are not consistent about how and when they take page protection exceptions. The Pentium Pro's operation is architecturally cleaner than the Pentium's operation, so the IDT WinChip C6 mimics the Pentium Pro's behavior. If a memory address hits either of the TLB's and the associated TLB entry would indicate that a protection page fault should be taken, the page tables are retried to ensure that the TLB entry is up to date. Only if the retried TLB entry still indicates that a page protection exception should occur does the exception actually take place.

Table Walk D and A Bit Updates

The IDT WinChip C6 processor does not snoop its instruction cache during table walks. Consequently, if the access (A) or dirty (D) bit need to be updated, and the table or directory entry is in the L1 instruction cache, the L1 instruction cache does not reflect the updated value.

While it is possible to create code that detects this inconsistency, it is highly unlikely that any application or operating system relies on the update.

SCYC

The SCYC signal is loosely defined on the Intel Pentium processor. On the IDT WinChip C6 processor, SCYC is asserted only during external locked read-modify-write cycles that are unaligned.

STPCLK# in Auto Halt State

If the processor sees an assertion on STPCLK# while resting in the Auto Halt Power Down State, it awakens briefly, and goes into the Stop Grant State. This differs from the Intel Pentium processor, which ignores STPCLK# in Auto Halt. The AMD-K6 x86 processor behaves the same as the IDT WinChip C6 processor.

B.3 INTEGER INSTRUCTION COMPATIBILITY

CPUID Vendor ID String

The vendor identification string returned by the CPUID instruction is different, of course, among the various Pentium-compatible processor manufacturers. A very few programs (games) are dependent upon the "GenuineIntel" string returned by the Intel processors and thus will not perform correctly with the default "CentaurHauls" string returned by the IDT WinChip C6 processor.

The IDT WinChip C6 processor has a feature that allows the user to define the vendor ID string returned by the CPUID instruction. This allows programs that are dependent on specific vendor ID strings to be run. This feature is described in Appendix A.

CPUID Feature Flags

As defined in Chapter 3, the feature definition flags returned by the CPUID instruction differ among various Pentium-compatible processors.

Machine-Specific Registers

As defined in Appendix A, the machines-specific registers (MSR's) of a IDT WinChip C6 processor are just that: machine-specific. That is, there are differences between the IDT WinChip C6 processor MSR's and the Pentium processor's MSR's (and all other x86 processors).

Undefined EFLAGS Settings

The Intel Pentium documentation defines the setting of the EFLAGS bits for several instructions as "undefined". In some cases the setting of undefined flags is consistent among Pentium-compatible processors and some programs depend upon these settings. In these cases, the IDT WinChip C6 processor sets the flags the same as does the Pentium processor.

In some cases where the various Pentium-compatible processors differ on these undefined flag settings, the IDT WinChip C6 processor settings differ from the Pentium processor settings.

CMPXCHG8B

The CMPXCHG8B instruction is supported by the IDT WinChip C6 processor. However, it is not reported as being supported in the CPUID instruction. This was required to maintain compatibility with some Windows NT versions.

B.4 FLOATING-POINT COMPATIBILITY

Transcendental Accuracy

The results of the transcendental instructions (FSIN, etc.) are slightly different among the various x86 processors—including between the Intel486 processor and the Pentium processor. The IDT WinChip C6 processor typically gets results within one-half ULP (units in the lower position) average and within one ULP worst-case of the correct mathematical result within the reduced argument range.

Undefined Condition Bits

The Intel Pentium documentation defines the setting of the FPU Status Word condition bits for several instructions as “undefined”. In some cases the setting of undefined flags is consistent among Pentium-compatible processors and some programs depend upon these settings. In these cases, the IDT WinChip C6 processor sets the flags the same as does the Pentium processor.

In some cases where the various Pentium-compatible processors differ on these undefined flag settings, the IDT WinChip C6 processor settings differ from the Pentium processor settings.

APPENDIX C. PROCESSOR ERRATA

C.1 INTRODUCTION

Appendix B describes the expected differences between the IDT WinChip C6 processor and the Intel Pentium processor. This Appendix A describes the IDT WinChip C6 processor *errata*: differences between the actual IDT WinChip C6 processor behavior and the expected results.

There are two current versions or *steppings* of the IDT WinChip C6 processor which have different errata: steppings 0 and 1. These can be identified by the stepping code returned by the CPUID instruction or in EDX following Reset.

Tables C-1 and C-2 describe the codes used to define the status and action plan of each individual erratum.

Table C-1

Status Code	***Description***
X	This version of the processor has the erratum
N/A	This erratum does not apply to this version of the processor
	Fixed in this version of processor

Table C-2

Plan Code	***Description***
Fixed	This erratum is fixed in the latest version of the processor
Fix	This erratum may be fixed in some future version of the processor
NoFix	There are no plans to fix this erratum in future steppings of the IDT WinChip C6 processor

The IDT WinChip C6 processor errata described here is complete and is described in considerable detail. This detailed exposure is consistent with the Centaur philosophy of openness. However, the errata do not, as far as we know, represent a real compatibility exposure. Microsoft and XXCAL did not encounter these during compatibility testing. Two of these errata were found during our extensive beta/system test. The rest were found only by specialized design-verification tests, which perform unnatural acts.

Table C-3 summarizes the errata and provides the codes which describes the status and action plan for each individual erratum. The details of each erratum are described in subsequent sections.

Table C-3

ID	Stepping		Plans	Errata
	0	1		
F-1	X	X	NoFix	FPU environment may be different from P55 in certain cases
F-2	X	X	NoFix	FPU transcendental may return a denormal value without underflow
F-3	X		Fixed	FPU FSINCOS or FCOS may cause DE to be incorrectly set
F-4	X		Fixed	FPU may slow down integer performance with incorrect pipeline slips
F-5	X	X	NoFix	FPU FCOM,FCOMP of denormal memory operand may not set flags correctly
F-6	X		Fixed	FPU FPREM of pseudodenormal does not convert result to Normal
F-7	X		Fixed	FPU instruction may result in spurious exception under certain conditions
M-1	X	X	NoFix	MMX instruction followed by read of FPSW may get wrong top of stack
I-1	X	X	NoFix	INTR is missed if deasserted while processor executes consecutive STIs
I-2	X	X	NoFix	Certain invalid instructions may result in page faults instead of INT6
I-3	X	X	NoFix	BUSCHK# does not work as in P54 in certain cases
I-4	X		Fixed	Wrong EIP pushed when INTR or NMI wakes processor after unmasked FPU exception

ID	Stepping		Plans	Errata
	0	1		
I-5	X		Fixed	INVLPG with linear address close to 4GB may incorrectly GP fault
I-6	X		Fixed	Self-modifying code may not be detected as in P54
I-7	X		Fixed	Time Stamp Counter stops counting when processor in low power standby
I-8	X		Fixed	TR12.AHD does not disable Auto HALT Powerdown as it should
I-9	X	X	NoFix	Inconsistent fault when length limit exceeded with invalid LOCK prefix
B-1	X		Fixed	MCR_CTRL[4] will not enable weak write ordering as it should
B-2	X		Fixed	Data bits with inactive byte enables may differ from P54 during I/O writes
B-3	X		Fixed	TR12.CI does not inhibit cache line fills as it should
B-4	X	X	Fix	Instruction cache does not detect internal parity error
B-5	X	X	Fix	HIT# may incorrectly stay asserted if snoop occurs during INVD / WBINVD
B-6	X	X	Fix	Memory read with disabled cache may result in cache line writeback and/or invalidate

C.2 FLOATING-POINT ERRATA

F-1. FPU environment may be different from P55 in certain cases

PROBLEM: An FSTP QWORD that crosses a page boundary and gets a page fault will set the floating point environment, technically it should not. An FRSTOR that crosses a segment limit of 4GB will partially load floating-point state.

IMPLICATION: These errata are not believed to occur in any application or operating system.

WORKAROUND: None

F-2. FPU transcendental may return a denormal value without underflow

PROBLEM: The transcendental instructions (FSIN, FCOS, FSINCOS, FPTAN, FPATAN, FYL2X, FYL2XP1, F2XM1) may, in some cases, return a denormal value without underflow. This can occur whenever the internal approximation algorithm computes a precise denormal result (i.e. no loss of precision occurred in the computation of the approximation).

In these cases, the transcendental instructions behave like the arithmetic instructions. This is technically incorrect, since transcendental approximations are inherently inexact.

If the C6 transcendental approximation algorithm computes an exact denormal result, UE is not set in FPSW, and if UE is unmasked, the unmasked underflow response does not occur.

Examples:

```
FSIN(denormal x) = x with no underflow
FYL2X(2, denormal y) = y with no underflow
```

IMPLICATION: Applications that run with the underflow exception masked, and do not check the UE bit in FPSW (virtually all applications fall into this category) are unaffected.

WORKAROUND: In applications that do respond to underflow, this condition can be detected by examining result of a transcendental operation with FXAM to see if it is denormal, and if it is, handling that case as underflow.

For applications that expect the unmasked underflow response, the translation of the value to the middle of the normal range can computed without loss of precision in user code:

```
if (denormal result)
result = result * 2 ^ 24576
```

F-3. FPU FSINCOS or FCOS may cause DE to be incorrectly set

PROBLEM: FSINCOS or FCOS on some very small normal values can cause DE to be incorrectly set, which could cause a false exception if DE is unmasked.

IMPLICATION: Since few programs run with DE unmasked, especially among those that use transcendental instructions, this situation is extremely unlikely to cause problems.

WORKAROUND: Run with DE exception masked.

F-4. FPU may slow down integer performance with incorrect pipeline slips

PROBLEM: If Dynamic Power Management is enabled (the default condition) a floating-point status bit may be incorrectly captured when clocks are being deasserted to the FPU. There are other conditions internal to the processor pipeline required for this erratum, which makes it an unlikely event.

IMPLICATION: The FPU will falsely stall the processor resulting in one clock added per micro-instruction until another floating-point instruction, integer divide, or integer multiply is executed.

WORKAROUND: Disable Dynamic Power Management.

F-5. FPU FCOM,FCOMP of denormal memory operand may not set flags correctly

PROBLEM: The C0, C2, C3 flags in the floating-point status word may not be set when an FCOM instruction results in a stack fault. This erratum will occur when the following conditions are met:

- One of the FCOM operands is in memory (single- or double-real memory operand)
- the memory operand is Denormal
- the register operand causes a stack fault (underflow)
- the invalid operation exception is unmasked

The correct processor response is to set C0, C2, C3 flags indicating that the operands were "not comparable."

IMPLICATION: It is unlikely that the exception handler, when dealing with an invalid operation exception, would need the C-bits set as a result of the faulting FCOM. The exception handler will update the floating-point stack and restart the faulting FCOM or terminate the faulting process. It is therefore unlikely that this erratum will cause compatibility problems.

WORKAROUND: Run with invalid operation exception masked.

F-6. FPU FPREM of pseudodenormal does not convert result to Normal

PROBLEM: Pseudodenormals are numbers in the extended format with a zero in the exponent but a '1' in the most significant bit of the significand (in the extended format the leading bit is explicitly stored). The numeric value corresponding to denormals/pseudodenormals is computed as for normals except the exponent is assumed to be 1 (instead of the encoded zero). This means that, by definition, the numeric value of pseudodenormals overlaps the range of Normals for which exponent is 1. Numbers in this range, when represented in the extended format, could be either Normals (exponent of 1) or pseudodenormals (zero exponent). The x86 architecture allows pseudodenormals as operands but requires that the processor generate a Normal rather than a pseudodenormal. Due to this erratum the FPREM instruction will return a pseudodenormal when the input is a pseudodenormal. The correct processor response is to convert the operand into a Normal (make the exponent 1).

IMPLICATION: Pseudodenormals are rarely encountered in applications, and it is unlikely that they would be the input to FPREM. The processor response in this case, although architecturally incorrect, is numerically correct. If the pseudodenormal result of FPREM were to be used in a subsequent computation the answer would be correct. It is therefore not expected to cause problems in any application.

WORKAROUND: None.

F-7. FPU instruction may result in spurious exception under certain conditions

PROBLEM: The processor may generate a spurious floating-point exception under the following conditions:

- An instruction which will fault or modify the instruction stream begins execution
- An FIST or FISTP instruction follows the faulting instruction
- The FIST or FISTP instruction determines there is an overflow
- The fault (or self-modifying store) from the preceding instruction flushes the pipeline
- A subsequent FPU instruction will react to the lingering overflow condition detected by the flushed FIST, possibly generating a spurious floating-point exception

IMPLICATION: Applications which use FIST or FISTP with data which will overflow and which are preceded by instructions which fault or store into the instruction stream are at risk of getting spurious floating-point exceptions.

WORKAROUND: This erratum can be avoided by placing the FIST or FISTP instruction in a subroutine, forcing the processor to branch prior to executing it and thereby clearing the pipeline.

C.3 MMX ERRATA

M-1. MMX instruction followed by read of FPSW may get wrong top of stack

PROBLEM: An MMX instruction followed immediately (without an EMMS) by a read of the Floating Point Status Word (e.g. FSTSW,FDECSTP,FINCSTP,FSAVE) may report the incorrect top of stack (TOS). The correct response is for MMX instructions to clear the TOS.

IMPLICATION: MMX applications should not encounter this erratum. Under the prescribed method of executing MMX instructions after Floating Point routines, Section 3.3.2 of the Intel MMX specification states that EMMS must be used as a barrier between MMX/FPU instructions, or else errors may result.

WORKAROUND: Use EMMS between MMX and FPU instructions as prescribed by the Intel MMX specification.

C.4 INTEGER ERRATA

I-1. INTR is missed if deasserted while processor executes consecutive STIs

PROBLEM: If the processor is executing a sequence of more than one STI and an INTR is asserted and then deasserted before the processor ends the sequence of STIs then INTR will not be recognized.

IMPLICATION:

This situation should not happen on a PC system because:

- INTR is level sensitive, the interrupt controller must latch interrupts from devices and assert INTR until the processor acknowledges.
- There is no reason for software to use consecutive STIs.

WORKAROUND: None needed, see above.

I-2. Certain invalid instructions may result in page faults instead of INT6

PROBLEM: The illegal opcode 0F, 71h 72h and 73h with ModR/m memory (Group A MMX shift) is not flagged as an invalid opcode if the instruction crosses a page boundary and the second page faults, the IDT-C6 processor presents a page fault exception instead

IMPLICATION: If this were to happen the operating system will allocate the second page and resume the faulting task which would then get the invalid opcode exception (INT6).

WORKAROUND: None needed, see above.

I-3. BUSCHK# does not work as in P54 in certain cases

PROBLEM: If BUSHCK# and SMI# are asserted while in HALT state the processor correctly takes BUSCHK# first but Auto Halt Restart slot in SMRAM is not set when SMM is entered. A similar BUSCHK# erratum is that the ESI I/O restart slot may be incorrect if BUSCHK# and SMI# are asserted during an I/O operation.

IMPLICATION: BUSCHK# is not used in PC systems.

WORKAROUND: None needed in a PC system, see above.

I-4. Wrong EIP pushed when INTR or NMI wakes processor after unmasked FPU exception

PROBLEM: The processor will push the wrong EIP on the stack when it services an external interrupt under the following conditions:

- CR0.NE is 0, indicating the processor should not raise INT16 on unmasked FPU exceptions
- an unmasked FPU exception occurs
- IGNNE# is inactive
- the processor does not clear or mask the FPU exception before the next FPU instruction which causes the processor to wait for an external interrupt
- at this point the processor waits for an external interrupt
- INTR or NMI is asserted
- Due to this erratum, the processor pushes the wrong EIP. The correct response is to push the EIP of the FPU instruction which caused the processor to wait for an external interrupt

IMPLICATION: DOS applications that have unmasked FPU exceptions and provide their own interrupt handler are subject to this erratum.

WORKAROUND: None

I-5. INVLPG with linear address close to 4GB may incorrectly GP fault

PROBLEM: A GP fault will result when INVLPG specifies a linear address within the last few bytes of the 32-bit address limit. The specific linear address subject to this erratum depends on the operand size:

- For 16-bit operand size the faulting linear address is 0xFFFFFFFF

- For 32-bit operand size the faulting linear address is in the range {0xFFFFFFFD, 0xFFFFFFFE,0xFFFFFFFF}.

This erratum is independent of the address size specified by the INVLPG.

IMPLICATION: This erratum has not been observed in any operating system.

WORKAROUND: An operating system can avoid this erratum by ensuring that the linear address specified by INVLPG is not in the faulting range. Masking the address so it points to the first byte of the page would meet this requirement.

I-6. Self-modifying code may not be detected as in P54

PROBLEM: The processor may not detect self-modifying code under the following conditions:

- An instruction stores to memory either 1 or 2 bytes (no more)
- The destination byte(s) are contained within the last 3 bytes of a cache line
- The instruction storing to memory is within 3 cache lines of the target
- The target sequentially follows the instruction storing to memory with no branches separating them
- Due to this erratum, the processor may have pre-fetched the target and not detect that it was modified, causing it to execute the previous contents at the target.

The correct response is for the instruction stream to reflect all preceding stores.

IMPLICATION: This erratum does not manifest itself in any known applications.

WORKAROUND: Issue a branch after modifying the target instruction.

I-7. Time Stamp Counter stops counting when processor in low power standby

PROBLEM: The processor will incorrectly stop incrementing the Time Stamp Counter when it enters the low power standby state, i.e. under the following conditions:

- STOP GRANT (STPCLK# asserted)
- SHUTDOWN
- the HALT instruction (Auto HALT Powerdown)
- There is an unmasked floating-point error, CR0.NE is '0', IGNNE# is deasserted

The correct response is for the Time Stamp Counter to continue incrementing at all times.

IMPLICATION: This erratum manifests itself as a spontaneous "Divide by zero" message when Linux is idle for some time.

WORKAROUND: None.

I-8. TR12.AHD does not disable Auto HALT Powerdown as it should

PROBLEM: When TR12.AHD (Auto Halt Disable) is set to '1' the processor should not Powerdown, but due to this erratum it does.

IMPLICATION: This erratum has not been determined to cause any problems during compatibility testing.

WORKAROUND: None.

I-9. Inconsistent fault when length limit exceeded with invalid LOCK prefix

PROBLEM: The LOCK prefix (0xF0) is valid only for certain instructions. The LOCK prefix is not valid, for example, when the operands are all in registers. The correct processor response for such an invalid use of the LOCK prefix is to generate an Invalid Opcode exception (INT 6).

The use of redundant prefixes could result in the instruction length exceeding 15 bytes; the correct processor response is to generate a General Protection fault (INT 13).

When an instruction has an invalid LOCK prefix *and* the length exceeds 15 bytes the resulting exception (INT 6 or INT 13) is implementation-dependent. The Pentium processor (P54) responds with INT 6 whereas the Pentium Processor with MMX Technology (P55) responds with INT 13. The IDT WinChip C6 processor response is not consistent, for some opcodes it responds with INT 6 (like a P54) and for others with INT 13 (like a P55).

IMPLICATION: This erratum has not been determined to cause any problems during compatibility testing.

WORKAROUND: Do not rely on implementation-dependent behavior.

C.5 BUS ERRATA

B-1. MCR_CTRL[4] will not enable weak write ordering as it should

PROBLEM: Weak write ordering is a performance feature of the IDT-C6 processor that is controlled by bit 4 of the MCR_CTRL machine specific register in conjunction with the Memory Range Registers. Due to this erratum the processor may end up with a cache line in the exclusive state instead of modified, leading to stale system memory contents.

IMPLICATION: Memory Range Registers should not specify weak write ordering, and therefore performance of the IDT-C6 processor may not be optimal.

WORKAROUND: Get patch from Centaur to allow Memory Range Registers to specify weak write ordering.

B-2. Data bits with inactive byte enables may differ from P55 during I/O writes

PROBLEM: Bits on the data bus with inactive byte enables have undefined values. During an I/O write bus cycle for an 8-bit or 16-bit operand P54 drives '0' for data bits that are not part of the operand (the operand is zero-extended and rotated so that it is properly aligned on the bus).

IMPLICATION: This erratum is known to cause problems in a system with an Opti Firestar chip set . In this case the ESDI device driver has a long delay because the secondary PCI controller indicates a busy status indefinitely.

WORKAROUND: For P55 compatible behavior zero-extend the operand into EAX and use register operands for the OUT instruction (use OUT DX,AX or OUT DX,AL instead of OUTS).

Example:

```
MOVZX EAX,BX ; Zero-extend to 32-bits
OUT   DX,AX  ; Do not use OUTS
```

B-3. TR12.CI does not inhibit cache line fills as it should

PROBLEM: When TR12.CI (Cache Inhibit) is set to '1' all cache line fills should be inhibited; during cache misses the processor should access memory using single transfer cycles. Due to this erratum TR12.CI is ignored and the caches operate normally regardless of the setting of TR12.CI.

IMPLICATION: If software (like BIOS) relies on TR12.CI to test the second level cache, and if the second level cache is defective, then the defect may go unnoticed. If TR12.CI is used to increase performance when accessing code or data (that will not be referenced again soon), then performance might be adversely affected if a cache line that is frequently referenced is replaced by a cache line that is less frequently referenced. This erratum has not caused problems in compatibility testing.

WORKAROUND: Second level cache tests could be written to avoid TR12.CI, although this might require flushing the internal caches several times and would therefore increase the time required to perform the second level cache test.

B-4. Instruction cache does not detect internal parity errors

PROBLEM: The instruction cache should compute and maintain a parity bit for each 32 bits. These parity bits should be compared to the computed parity whenever the instruction cache is accessed. If a stored parity bit does not match the computed parity bit the correct processor response is to assert IERR# and handle the exception as specified by machine–specific registers (the default is shutdown). Due to this erratum internal parity errors in the instruction cache are not detected.

IMPLICATION: In the unlikely event that a bit stored in the instruction cache spontaneously changes value it may not be detected; the result is unpredictable.

WORKAROUND: None.

B-5. HIT# may incorrectly stay asserted if snoop occurs during INVD / WBINVD

PROBLEM: The HIT# signal is used to indicate that an inquire cycle (external snoop) hits a matching address within a processor cache. The HIT# signal transitions only as a response to an inquire cycle. Due to this erratum the HIT# signal remains asserted when it should have been deasserted as a result of this sequence of events:

- The HIT# signal is asserted as a response to an inquire cycle
- An INVD or WBINVD instruction begins execution
- An inquire cycle occurs during the INVD/WBINVD instruction
- At this point the HIT# signal should transition to deasserted, but due to this erratum it remains asserted

IMPLICATION: Most chipsets in PC systems do not respond to the HIT# signal, and most operating systems do not use the INVD or WBINVD instruction. This erratum has not been observed during compatibility testing.

WORKAROUND: None needed in a PC system, see above.

B-6. Memory read with disabled cache may result in cache line writeback and/or invalidate

PROBLEM: The processor caches are disabled when either CR0.CD=1 or TR12.CI=1. When caches are disabled the processor should not perform cache line fill operations; that is, no new cache lines should be allocated. If an access misses in the cache when caches are disabled, the correct processor response is to issue a single cycle transfer (two for a data access that crosses an 8-byte boundary). Typically no cache line writeback or invalidates occur when caches are disabled. There are special cases where writebacks and/or invalidates occur even though caches are disabled.

Coherency requirements dictate that the processor writeback and/or invalidate a cache line when caches are disabled under the following circumstances (compatible with Pentium):

- CR0.NW=0 and an external snoop hits in the cache, the cache line is written back if modified and invalidated.
- A write misses in the data cache but hits in the instruction cache; the instruction cache line is invalidated.
- An instruction fetch misses in the instruction cache but hits a modified line in the data cache; the cache line is written back and invalidated.
- A locked cycle (read-modify-write sequence) hits a line in the data cache; the cache line is written back if modified and invalidated.

The cache organization of the IDT WinChip C6 processor may result in a writeback and/or invalidate when caches are disabled, a behavior different from Pentium processor, but not considered errata:

- A read or write that hits in the cache uses a virtual alias. This happens when the physical address hits in the cache but there are different values in either bit 13 or 12 of the linear address. If the aliased line is modified then it is written back. The line is invalidated.

There is one remaining case where the processor with disabled caches will writeback and/or invalidate cache lines as a response to a memory read. The behavior in this case is not compatible with Pentium and is the reason for this erratum. The improper writeback and/or invalidate will occur when:

- A read that misses in the cache occurs to an index whose LRU points to a valid line. If the line is modified then it is written back. The line is invalidated. This is a result of the processor ensuring it has an available cache line to allocate and store the data from the bus even though caches are disabled.

IMPLICATION: This erratum has not been determined to cause any problems during compatibility testing. This erratum would have harmful effects only if the processor modifies cache lines for which there is no corresponding writable memory enabled in the system at the time, such that the data from a writeback operation could not be retrieved with a later read cycle.

WORKAROUND: When operating with caches disabled, ensure that the system has writable memory at the physical address corresponding to all modified cache lines.