Pentium™ Processor User's Manual
Volume 1: Pentium Processor Data Book

intel pentium PROCESSOR

intel i486™

intel i386™

intel

Founded in 1968 to pursue the integration of large numbers of transistors onto tiny silicon chips, Intel's history has been marked by a remarkable number of scientific breakthroughs and innovations. In 1971, Intel introduced the 4004, the first microprocessor. Containing 2300 transistors, this first commercially available computer-on-a-chip is primitive compared with today's million-plus transistor products.

Innovations such as the microprocessor, the erasable programmable read-only memory (EPROM) and the dynamic random access memory (DRAM) revolutionized electronics by making integrated circuits the mainstay of both consumer and business computing products.

Over the last two-and-a-half decades, Intel's business has evolved and today the company's focus is on delivering an extensive line of component, module and system-level building block products to the computer industry. The company's product line covers a broad spectrum, and includes microprocessors, flash memory, microcontrollers, a broad line of PC enhancement and local area network products, multimedia technology products, and massively parallel supercomputers. Intel's 32-bit X86 architecture, represented by the Intel386™ and Intel486™ microprocessor families, is the de facto standard of modern business computing in millions of PCs worldwide.

Intel has over 26,000 employees located in offices and manufacturing facilities around the world. Today, Intel is the largest semiconductor company in the world.

intel®

LITERATURE

To order Intel literature or obtain literature pricing information in the U.S. and Canada call or write Intel Literature Sales. In Europe and other international locations, please contact your **local** sales office or distributor.

INTEL LITERATURE SALES
P.O. Box 7641
Mt. Prospect, IL 60056-7641

In the U.S. and Canada
call toll free
(800) 548-4725
This 800 number is for external customers only.

CURRENT HANDBOOKS

Product line handbooks contain data sheets, application notes, article reprints, and other design information. All handbooks can be ordered individually, and most are available in a pre-packaged set in the U.S. and Canada.

Title	Intel Order Number	ISBN
SET OF TWELVE HANDBOOKS (Available in U.S. and Canada)	**231003**	**N/A**

CONTENTS LISTED BELOW FOR INDIVIDUAL ORDERING:

Title	Intel Order Number	ISBN
CONNECTIVITY	231658	1-55512-174-8
EMBEDDED APPLICATIONS (1993/94)	270648	1-55512-179-9
EMBEDDED MICROCONTROLLERS & PROCESSORS (2 volume set)	270645	1-55512-176-4
MEMORY PRODUCTS	210830	1-55512-172-1
MICROCOMPUTER PRODUCTS	280407	1-55512-173-X
MICROPROCESSORS (2 volume set)	230843	1-55512-169-1
MOBILE COMPUTER PRODUCTS	241420	1-55512-186-1
i750®, i860™, i960® PROCESSORS AND RELATED PRODUCTS	272084	1-55512-185-3
PACKAGING	240800	1-55512-182-9
PERIPHERAL COMPONENTS	296467	1-55512-181-0
PRODUCT OVERVIEW (A guide to Intel Architectures and Applications)	210846	N/A
PROGRAMMABLE LOGIC	296083	1-55512-180-2

ADDITIONAL LITERATURE:
(Not included in handbook set)

Title	Intel Order Number	ISBN
AUTOMOTIVE	231792	1-55512-125-X
COMPONENTS QUALITY/RELIABILITY	210997	1-55512-132-2
CUSTOMER LITERATURE GUIDE	210620	N/A
INTERNATIONAL LITERATURE GUIDE (Available in Europe only)	E00029	N/A
MILITARY AND SPECIAL PRODUCTS (2 volume set)	210461	1-55512-189-6
SYSTEMS QUALITY/RELIABILITY	231762	1-55512-091-9
HANDBOOK DIRECTORY (Index of all data sheets contained in the handbooks)	241197	N/A

intel ®

Pentium™ Processor
User's Manual

Volume 1:

Pentium™ Processor Data Book

NOTE: The *Pentium™ Processor User's Manual* consists of three books: *Pentium™ Processor Data Book*, Order Number 241428; the *82496 Cache Controller and 82491 Cache SRAM Data Book*, Order Number 241429; and the *Architecture and Programming Manual*, Order Number 241430. Please refer to all three volumes when evaluating your design needs.

1993

intel.

PENTIUM™ PROCESSOR

- **Binary Compatible with Large Software Base**
  - MS-DOS, Windows, OS/2, UNIX SVR4, NeXTstep 486, Solaris 2.0
- **32-Bit Microprocessor**
  - 32-Bit Addressing
  - 64-Bit Data Bus
- **Superscalar architecture**
  - Two pipelined integer units
  - Capable of under one Clock per Instruction
  - Pipelined Floating Point Unit
- **Separate Code and Data Caches**
  - 8K Code, 8K Write Back Data
  - 2-way 32-byte Line Size
  - Software Transparent
  - MESI Cache Consistency Protocol
- **Advanced Design Features**
  - Branch Prediction
  - Virtual Mode Extensions

- **273-Pin Grid Array Package**
- **BiCMOS Silicon Technology**
- **Increased Page Size**
  - 4M for Increased TLB Hit Rate
- **Multi-Processor Support**
  - Multiprocessor Instructions
  - Support for Second Level Cache
- **Internal Error Detection**
  - Functional Redundancy Checking
  - Built in Self Test
  - Parity testing and checking
- **IEEE 1149.1 Boundary Scan Compatibility**
- **Performance Monitoring**
  - Counts Occurrence of Internal Events
  - Traces Execution through Pipelines

The Pentium processor provides the new generation of power for high-end workstations and servers. The Pentium processor is compatible with the entire installed base of applications for DOS, Windows, OS/2, and UNIX. The Pentium processor's superscalar architecture can execute two instructions per clock cycle. Branch Prediction and separate caches also increase performance. The pipelined floating point unit of the Pentium processor delivers workstation level performance. Separate code and data caches reduce cache conflicts while remaining software transparent. The Pentium processor has 3.1 million transistors and is built on Intel's .8 Micron BiCMOS silicon technology.

int_el. ®

TABLE OF CONTENTS

Figures

Tables

intel®

1

Pinout

CHAPTER 1
PINOUT

1.1. PINOUT AND PIN DESCRIPTIONS

1.1.1. Pentium™ Processor Pinout

PDB82

Figure 1-1. Pentium™ Processor Pinout (Top View)

Figure 1-2. Pentium™ Processor Pinout (Bottom View)

Table 1-1. Pentium™ Processor Pin Cross Reference Table by Pin Name

Signal	Location	Signal	Location
A3	T17	A20M#	U05
A4	W19	ADS#	P04
A5	U18	AHOLD	L02
A6	U17	AP	P03
A7	T16	APCHK#	W03
A8	U16	BE0#	U04
A9	T15	BE1#	Q04
A10	U15	BE2#	U06
A11	T14	BE3#	V01
A12	U14	BE4#	T06
A13	T13	BE5#	S04
A14	U13	BE6#	U07
A15	T12	BE7#	W01
A16	U12	BOFF#	K04
A17	T11	BP2	B02
A18	U11	BP3	B03
A19	T10	BRDY#	L04
A20	U10	BREQ	V02
A21	U21	BT0	T08
A22	U09	BT1	W21
A23	U20	BT2	T07
A24	U08	BT3	W20
A25	U19	BUSCHK#	T03
A26	T09	CACHE#	J04
A27	V21	CLK	K18
A28	V06	D0	D03
A29	V20	D1	E03
A30	W05	D2	E04
A31	V19	D3	F03

intel®

Table 1-1. Pentium™ Processor Pin Cross Reference Table by Pin Name (Contd.)

Signal	Location	Signal	Location
D4	C04	D33	C17
D5	G03	D34	C19
D6	B04	D35	D17
D7	G04	D36	C18
D8	F04	D37	D16
D9	C12	D38	D19
D10	C13	D39	D15
D11	E05	D40	D14
D12	C14	D41	B19
D13	D04	D42	D20
D14	D13	D43	A20
D15	D05	D44	D21
D16	D06	D45	A21
D17	B09	D46	E18
D18	C06	D47	B20
D19	C15	D48	B21
D20	D07	D49	F19
D21	C16	D50	C20
D22	C07	D51	F18
D23	A10	D52	C21
D24	B10	D53	G18
D25	C08	D54	E20
D26	C11	D55	G19
D27	D09	D56	H21
D28	D11	D57	F20
D29	C09	D58	J18
D30	D12	D59	H19
D31	C10	D60	L19
D32	D10	D61	K19

Table 1-1. Pentium™ Processor Pin Cross Reference Table by Pin Name (Contd.)

Signal	Location		Signal	Location
D62	J19		IV	B01
D63	H18		KEN#	J03
D/C#	V04		LOCK#	V03
DP0	H04		M/IO#	A02
DP1	C05		NA#	K03
DP2	DP3		NMI	N19
DP3	D08		PCD	W04
DP4	D18		PCHK#	R03
DP5	A19		PEN#	M18
DP6	E19		PM0/BP0	D02
DP7	E21		PM1/BP1	C03
EADS#	M03		PRDY	U03
EWBE#	A03		PWT	S03
FERR#	H03		RESET	L18
FLUSH#	U02		R/S#	R18
FRCMC#	M19		SCYC	R04
HIT#	W02		SMI#	P18
HITM#	M04		SMIACT#	T05
HLDA	Q03		TCK	T04
HOLD	V05		TDI	T21
IBT	T19		TDO	S21
IERR#	C02		TMS	P19
IGNNE#	S20		TRST#	S18
INIT	T20		WB/WT#	M02
INTR	N18		W/R#	N03
INV	A01		NC	L03, N04, Q19, R19, S19, T18
IU	J02			

intel®

Table 1-1. Pentium™ Processor Pin Cross Reference Table by Pin Name (Contd.)

Signal	Location
VCC	A04, A05, A06, A07, A08, A11, A12, A13, A14, A15, A16, A17, A18, C01, D01, E01, F01, F21, G01, G21, H01, J21, K21, L21, M21, N01, N21, P01, P21, Q01, Q18, Q21, R01, R21, S01, T01,U01, W06, W07, W08, W09, W10, W11, W12, W13, W14, W15, W16, W17, W18
VSS	B05, B06, B07, B08, B11, B12, B13, B14, B15, B16, B17, B18, E02, F02, G02, G20, H02, H20, J01, J20, K01, K02, K20, L01, L20, M01, M20, N02, N20, P02, P20, Q02, Q20, R02, R20, S02, T02, V07, V08, V09, V10, V11, V12, V13, V14, V15, V16, V17, V18

1.2.   DESIGN NOTES

For reliable operation, always connect unused inputs to an appropriate signal level. Unused active low inputs should be connected to VCC. Unused active HIGH inputs should be connected to GND.

No Connect (NC) pins must remain unconnected. Connection of NC pins may result in component failure or incompatibility with processor steppings.

Note: The No Connect pin located at L03 (BRDYC#) along with BUSCHK# are sampled by the Pentium processor at RESET to configure the I/O buffers of the processor for use with the 82496 Cache Controller/82491 Cache SRAM secondary cache as a chip set (refer to the *82496 Cache Controller/82491 Cache SRAM Data Book for Use with the Pentium™ Processor* for further information).

1.3.   QUICK PIN REFERENCE

This section gives a brief functional description of each of the pins. For a detailed description, see the Hardware Interface chapter in this manual. **Note that all input pins must meet their AC/DC specifications to guarantee proper functional behavior.** In this section, the pins are arranged in alphabetical order. The functional grouping of each pin is listed at the end of this chapter.

The # symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage. When a # symbol is not present after the signal name, the signal is active, or asserted at the high voltage level.

intel®

Table 1-2. Quick Pin Reference

Symbol	Type*	Name and Function
A20M#	I	When the *address bit 20 mask* pin is asserted, the Pentium™ Processor emulates the address wraparound at one Mbyte which occurs on the 8086. When A20M# is asserted, the Pentium processor masks physical address bit 20 (A20) before performing a lookup to the internal caches or driving a memory cycle on the bus. The effect of A20M# is undefined in protected mode. A20M# must be asserted only when the processor is in real mode.
A31-A3	I/O	As outputs, the *address* lines of the processor along with the byte enables define the physical area of memory or I/O accessed. The external system drives the inquire address to the processor on A31-A5.
ADS#	O	The *address status* indicates that a new valid bus cycle is currently being driven by the Pentium processor.
AHOLD	I	In response to the assertion of *address hold*, the Pentium processor will stop driving the address lines (A31-A3), and AP in the next clock. The rest of the bus will remain active so data can be returned or driven for previously issued bus cycles.
AP	I/O	*Address parity* is driven by the Pentium processor with even parity information on all Pentium processor generated cycles in the same clock that the address is driven. Even parity must be driven back to the Pentium processor during inquire cycles on this pin in the same clock as EADS# to ensure that the correct parity check status is indicated by the Pentium processor.
APCHK#	O	The *address parity check* status pin is asserted two clocks after EADS# is sampled active if the Pentium processor has detected a parity error on the address bus during inquire cycles. APCHK# will remain active for one clock each time a parity error is detected.
BE7#-BE0#	O	The *byte enable* pins are used to determine which bytes must be written to external memory, or which bytes were requested by the CPU for the current cycle. The byte enables are driven in the same clock as the address lines (A31-3).
BOFF#	I	The *backoff* input is used to abort all outstanding bus cycles that have not yet completed. In response to BOFF#, the Pentium processor will float all pins normally floated during bus hold in the next clock. The processor remains in bus hold until BOFF# is negated at which time the Pentium processor restarts the aborted bus cycle(s) in their entirety.

intel®

Table 1-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
BP[3:2] PM/BP[1:0]	O	The *breakpoint* pins (BP3-0) correspond to the debug registers, DR3-DR0. These pins externally indicate a breakpoint match when the debug registers are programmed to test for breakpoint matches. BP1 and BP0 are multiplexed with the Performance Monitoring pins (PM1 and PM0). The PB1 and PB0 bits in the Debug Mode Control Register determine if the pins are configured as breakpoint or performance monitoring pins. The pins come out of reset configured for performance monitoring (for more information see Appendix A).
BRDY#	I	The *burst ready* input indicates that the external system has presented valid data on the data pins in response to a read or that the external system has accepted the Pentium processor data in response to a write request. This signal is sampled in the T2, T12 and T2P bus states.
BREQ	O	The *bus request output* indicates to the external system that the Pentium processor has internally generated a bus request. This signal is always driven whether or not the Pentium processor is driving its bus.
BT3-BT0	O	The *branch trace* outputs provide bits 2-0 of the branch target linear address (BT2-BT0) and the default operand size (BT3) during a branch trace message special cycle.
BUSCHK#	I	The *bus check* input allows the system to signal an unsuccessful completion of a bus cycle. If this pin is sampled active, the Pentium processor will latch the address and control signals in the machine check registers. If in addition, the MCE bit in CR4 is set, the Pentium processor will vector to the machine check exception.
CACHE#	O	For Pentium processor-initiated cycles the *cache* pin indicates internal cacheability of the cycle (if a read), and indicates a burst writeback cycle (if a write). If this pin is driven inactive during a read cycle, Pentium processor will not cache the returned data, regardless of the state of the KEN# pin. This pin is also used to determine the cycle length (number of transfers in the cycle).
CLK	I	The *clock* input provides the fundamental timing for the Pentium processor. Its frequency is the internal operating frequency of the Pentium processor and requires TTL levels. All external timing parameters except TDI, TDO, TMS and TRST# are specified with respect to the rising edge of CLK.
D/C#	O	The *Data/Code* output is one of the primary bus cycle definition pins. It is driven valid in the same clock as the ADS# signal is asserted. D/C# distinguishes between data and code or special cycles.
D63-D0	I/O	These are the 64 *data lines* for the processor. Lines D7-D0 define the least significant byte of the data bus; lines D63-D56 define the most significant byte of the data bus. When the CPU is driving the data lines, they are driven during the T2, T12, or T2P clocks for that cycle. During reads, the CPU samples the data bus when BRDY# is returned.

Table 1-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
DP7-DP0	I/O	These are the *data parity* pins for the processor. There is one for each byte of the data bus. They are driven by the Pentium processor with even parity information on writes in the same clock as write data. Even parity information must be driven back to the Pentium processor on these pins in the same clock as the data to ensure that the correct parity check status is indicated by the Pentium processor. DP7 applies to D63-D56, DP0 applies to D7-D0.
EADS#	I	This signal indicates that a *valid external address* has been driven onto the Pentium processor address pins to be used for an inquire cycle.
EWBE#	I	The *external write buffer empty* input, when inactive (high), indicates that a write cycle is pending in the external system. When the Pentium processor generates a write, and EWBE# is sampled inactive, the Pentium processor will hold off all subsequent writes to all E or M-state lines in the data cache until all write cycles have completed, as indicated by EWBE# being active.
FERR#	O	The *floating point error* pin is driven active when an unmasked floating point error occurs. FERR# is similar to the ERROR# pin on the Intel387™ math coprocessor. FERR# is included for compatibility with systems using DOS type floating point error reporting.
FLUSH#	I	When asserted, the *cache flush* input forces the Pentium processor to writeback all modified lines in the data cache and invalidate its internal caches. A Flush Acknowledge special cycle will be generated by the Pentium processor indicating completion of the writeback and invalidation. If FLUSH# is sampled low when RESET transitions from high to low, tristate test mode is entered.
FRCMC#	I	The *Functional Redundancy Checking Master/Checker* mode input is used to determine whether the Pentium processor is configured in master mode or checker mode. When configured as a master, the Pentium processor drives its output pins as required by the bus protocol. When configured as a checker, the Pentium processor tristates all outputs (except IERR# and TDO) and samples the output pins. The configuration as a master/checker is set after RESET and may not be changed other than by a subsequent RESET.
HIT#	O	The *hit* indication is driven to reflect the outcome of an inquire cycle. If an inquire cycle hits a valid line in either the Pentium processor data or instruction cache, this pin is asserted two clocks after EADS# is sampled asserted. If the inquire cycle misses Pentium processor cache, this pin is negated two clocks after EADS#. This pin changes its value only as a result of an inquire cycle and retains its value between the cycles.
HITM#	O	The *hit to a modified line* output is driven to reflect the outcome of an inquire cycle. It is asserted after inquire cycles which resulted in a hit to a modified line in the data cache. It is used to inhibit another bus master from accessing the data until the line is completely written back.

intel®

Table 1-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
HLDA	O	The *bus hold acknowledge* pin goes active in response to a hold request driven to the processor on the HOLD pin. It indicates that the Pentium processor has floated most of the output pins and relinquished the bus to another local bus master. When leaving bus hold, HLDA will be driven inactive and the Pentium processor will resume driving the bus. If the Pentium processor has bus cycle pending, it will be driven in the same clock that HLDA is deasserted.
HOLD	I	In response to the *bus hold request*, the Pentium processor will float most of its output and input/output pins and assert HLDA after completing all outstanding bus cycles. The Pentium processor will maintain its bus in this state until HOLD is deasserted. HOLD is not recognized during LOCK cycles. The Pentium processor will recognize HOLD during reset.
IBT	O	The *instruction branch taken* pin is driven active (high) for one clock to indicate that a branch was taken. This output is always driven by the Pentium processor.
IERR#	O	The *internal error* pin is used to indicate two types of errors, internal parity errors and functional redundancy errors. If a parity error occurs on a read from an internal array, the Pentium processor will assert the IERR# pin for one clock and then shutdown. If the Pentium processor is configured as a checker and a mismatch occurs between the value sampled on the pins and the corresponding value computed internally, the Pentium processor will assert IERR# two clocks after the mismatched value is returned.
IGNNE#	I	This is the *ignore numeric error* input. This pin has no effect when the NE bit in CR0 is set to 1. When the CR0.NE bit is 0, and the IGNNE# pin is asserted, the Pentium processor will ignore any pending unmasked numeric exception and continue executing floating point instructions for the entire duration that this pin is asserted. When the CR0.NE bit is 0, IGNNE# is not asserted, a pending unmasked numeric exception exists (SW.ES = 1), and the floating point instruction is one of FINIT, FCLEX, FSTENV, FSAVE, FSTSW, FSTCW, FENI, FDISI, or FSETPM, the Pentium processor will execute the instruction in spite of the pending exception. When the CR0.NE bit is 0, IGNNE# is not asserted, a pending unmasked numeric exception exists (SW.ES = 1), and the floating point instruction is one other than FINIT, FCLEX, FSTENV, FSAVE, FSTSW, FSTCW, FENI, FDISI, or FSETPM, the Pentium processor will stop execution and wait for an external interrupt.
INIT	I	The Pentium processor *initialization* input pin forces the Pentium processor to begin execution in a known state. The processor state after INIT is the same as the state after RESET except that the internal caches, write buffers, and floating point registers retain the values they had prior to INIT. INIT may NOT be used in lieu of RESET after power-up.
		If INIT is sampled high when RESET transitions from high to low the Pentium processor will perform built-in self test prior to the start of program execution.
INTR	I	An active *maskable interrupt* input indicates that an external interrupt has been generated. If the IF bit in the EFLAGS register is set, the Pentium processor will generate two locked interrupt acknowledge bus cycles and vector to an interrupt handler after the current instruction execution is completed. INTR must remain active until the first interrupt acknowledge cycle is generated to assure that the interrupt is recognized.

Table 1-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
INV	I	The *invalidation* input determines the final cache line state (S or I) in case of an inquire cycle hit. It is sampled together with the address for the inquire cycle in the clock EADS# is sampled active.
IU	O	The *u-pipe instruction complete* output is driven active (high) for 1 clock to indicate that an instruction in the u-pipeline has completed execution. This pin is always driven by the Pentium processor.
IV	O	The *v-pipe instruction complete* output is driven active (high) for one clock to indicate that an instruction in the v-pipeline has completed execution. This pin is always driven by the Pentium processor.
KEN#	I	The *cache enable* pin is used to determine whether the current cycle is cacheable or not and is consequently used to determine cycle length. When the Pentium processor generates a cycle that can be cached (CACHE# asserted) and KEN# is active, the cycle will be transformed into a burst line fill cycle.
LOCK#	O	The *bus lock* pin indicates that the current bus cycle is locked. The Pentium processor will not allow a bus hold when LOCK# is asserted (but AHOLD and BOFF# are allowed). LOCK# goes active in the first clock of the first locked bus cycle and goes inactive after the BRDY# is returned for the last locked bus cycle. LOCK# is guaranteed to be deasserted for at least one clock between back to back locked cycles.
M/IO#	O	The *Memory/Input-Output* is one of the primary bus cycle definition pins. It is driven valid in the same clock as the ADS# signal is asserted. M/IO# distinguishes between memory and I/O cycles.
NA#	I	An active *next address* input indicates that the external memory system is ready to accept a new bus cycle although all data transfers for the current cycle have not yet completed. The Pentium processor will drive out a pending cycle two clocks after NA# is asserted. The Pentium processor supports up to 2 outstanding bus cycles.
NMI	I	The *non-maskable interrupt* request signal indicates that an external non-maskable interrupt has been generated.
PCD	O	The *page cache disable* pin reflects the state of the PCD bit in CR3, the Page Directory Entry, or the Page Table Entry. The purpose of PCD is to provide an external cacheability indication on a page by page basis.
PCHK#	O	The *parity check* output indicates the result of a parity check on a data read. It is driven with parity status two clocks after BRDY# is returned. PCHK# remains low one clock for each clock in which a parity error was detected. Parity is checked only for the bytes on which valid data is returned.

Table 1-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
PEN#	I	The *parity enable* input (along with CR4.MCE) determines whether a machine check exception will be taken as a result of a data parity error on a read cycle. If this pin is sampled active in the clock a data parity error is detected, the Pentium processor will latch the address and control signals of the cycle with the parity error in the machine check registers. If in addition the machine check enable bit in CR4 is set to "1", the Pentium processor will vector to the machine check exception before the beginning of the next instruction.
PM/BP[1:0]B P[3:2]	O	For more information on the *performance monitoring* pins, see Appendix A. The breakpoint pins BP[1:0] are multiplexed with the Performance Monitoring pins PM[1:0]. The PB1 and PB0 bits in the Debug Mode Control Register determine if the pins are configured as breakpoint or performance monitoring pins. The pins come out of reset configured for performance monitoring (for more information see Appendix A).
PRDY	O	The PRDY output pin indicates that the processor has stopped normal execution in response to the R/S# pin going active, or Probe Mode being entered (see Appendix A for more information regarding Probe Mode). This pin is provided for use with the Intel debug port described in the "Debugging" chapter.
PWT	O	The *page write through* pin reflects the state of the PWT bit in CR3, the Page Directory Entry, or the Page Table Entry. The PWT pin is used to provide an external writeback indication on a page by page basis.
R/S#	I	The R/S# input is an asynchronous, edge sensitive interrupt used to stop the normal execution of the processor and place it into an idle state. A high to low transition on the R/S# pin will interrupt the processor and cause it to stop execution at the next instruction boundary. This pin is provided for use with the Intel debug port described in the "Debugging" chapter.
RESET	I	*Reset* forces the Pentium processor to begin execution at a known state. All the Pentium processor internal caches will be invalidated upon the RESET. Modified lines in the data cache are not written back. FLUSH#, FRCMC# and INIT are sampled when RESET transitions from high to low to determine if tristate test mode or checker mode will be entered, or if BIST will be run.
SCYC	O	The *split cycle* output is asserted during misaligned LOCKed transfers to indicate that more than two cycles will be locked together. This signal is defined for locked cycles only. It is undefined for cycles which are not locked.
SMI#	I	The system Management Interrupt causes a system management interrupt request to be latched internally. When the latched SMI# is recognized on an instruction boundary, the processor enters System Management Mode.
SMIACT#	O	An active *system management interrupt active* output indicates that the processor is operating in System Management Mode (SMM).

Table 1-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
TCK	I	The *testability clock* input provides the clocking function for the Pentium processor boundary scan in accordance with the IEEE Boundary Scan interface (Standard 1149.1). It is used to clock state information and data into and out of the Pentium processor during boundary scan.
TDI	I	The *test data input* is a serial input for the test logic. TAP instructions and data are shifted into the Pentium processor on the TDI pin on the rising edge of TCK when the TAP controller is in an appropriate state.
TDO	O	The *test data output* is a serial output of the test logic. TAP instructions and data are shifted out of the Pentium processor on the TDO pin on the falling edge of TCK when the TAP controller is in an appropriate state.
TMS	I	The value of the *test mode select* input signal sampled at the rising edge of TCK controls the sequence of TAP controller state changes.
TRST#	I	When asserted, the *test reset* input allows the TAP controller to be asynchronously initialized.
W/R#	O	*Write/Read* is one of the primary bus cycle definition pins. It is driven valid in the same clock as the ADS# signal is asserted. W/R# distinguishes between write and read cycles.
WB/WT#	I	The writeback/writethrough input allows a data cache line to be defined as write back or write through on a line by line basis. As a result, it determines whether a cache line is initially in the S or E state in the data cache.

*   **NOTE**: the pins are classified as Input or Output based on their function in Master Mode. See the Functional Redundancy Checking section in the 'Error Detection' Chapter for further information.

1.4. PIN REFERENCE TABLES

Table 1-3. Output Pins

Name	Active Level	When Floated
ADS#	LOW	Bus Hold, BOFF#
APCHK#	LOW	
BE7#-BE0#	LOW	Bus Hold, BOFF#
BREQ	HIGH	
BT3-BT0	n/a	
CACHE#	LOW	Bus Hold, BOFF#
FERR#	LOW	
HIT#	LOW	
HITM#	LOW	
HLDA	HIGH	
IBT	HIGH	
IERR#	LOW	
IU	HIGH	
IV	HIGH	
LOCK#	LOW	Bus Hold, BOFF#
M/IO#, D/C#, W/R#	n/a	Bus Hold, BOFF#
PCHK#	LOW	
BP3-2, PM1/BP1, PM0/BP0	HIGH	
PRDY	HIGH	
PWT, PCD	HIGH	Bus Hold, BOFF#
SCYC	HIGH	Bus Hold, BOFF#
SMIACT#	LOW	
TDO	n/a	All states except Shift-DR and Shift-IR

NOTE: All output and input/output pins are floated during tristate test mode and checker mode (except IERR#).

Table 1-4. Input Pins

Name	Active Level	Synchronous/ Asynchronous	Internal resistor	Qualified
A20M#	LOW	Asynchronous		
AHOLD	HIGH	Synchronous		
BOFF#	LOW	Synchronous		
BRDY#	LOW	Synchronous		Bus State T2,T12,T2P
BUSCHK#	LOW	Synchronous	Pullup	BRDY#
CLK	n/a			
EADS#	LOW	Synchronous		
EWBE#	LOW	Synchronous		BRDY#
FLUSH#	LOW	Asynchronous		
FRCMC#	LOW	Asynchronous		
HOLD	HIGH	Synchronous		
IGNNE#	LOW	Asynchronous		
INIT	HIGH	Asynchronous		
INTR	HIGH	Asynchronous		
INV	HIGH	Synchronous		EADS#
KEN#	LOW	Synchronous		First BRDY#/NA#
NA#	LOW	Synchronous		Bus State T2,TD,T2P
NMI	HIGH	Asynchronous		
PEN#	LOW	Synchronous		BRDY#
R/S#	n/a	Asynchronous	Pullup	
RESET	HIGH	Asynchronous		
SMI#	LOW	Asynchronous	Pullup	
TCK	n/a		Pullup	
TDI	n/a	Synchronous/TCK	Pullup	TCK
TMS	n/a	Synchronous/TCK	Pullup	TCK
TRST#	LOW	Asynchronous	Pullup	
WB/WT#	n/a	Synchronous		First BRDY#/NA#

Table 1-5. Input/Output Pins

Name	Active Level	When Floated	Qualified (when an input)
A31-A3	n/a	Address hold, Bus Hold, BOFF#	EADS#
AP	n/a	Address hold, Bus Hold, BOFF#	EADS#
D63-D0	n/a	Bus Hold, BOFF#	BRDY#
DP7-DP0	n/a	Bus Hold, BOFF#	BRDY#

NOTE: All output and input/output pins are floated during tristate test mode (except TDO) and checker mode (except IERR# and TDO).

intel

1.5. PIN GROUPING ACCORDING TO FUNCTION

Table 1-6 organizes the pins with respect to their function.

Table 1-6. Pin Functional Grouping

Function	Pins
Clock	CLK
Initialization	RESET, INIT
Address Bus	A31-A3, BE7# - BE0#
Address Mask	A20M#
Data Bus	D63-D0
Address Parity	AP, APCHK#
Data Parity	DP7-DP0, PCHK#, PEN#
Internal Parity Error	IERR#
System Error	BUSCHK#
Bus Cycle Definition	M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK#
Bus Control	ADS#, BRDY#, NA#
Page Cacheability	PCD, PWT
Cache Control	KEN#, WB/WT#
Cache Snooping/Consistency	AHOLD, EADS#, HIT#, HITM#, INV
Cache Flush	FLUSH#
Write Ordering	EWBE#
Bus Arbitration	BOFF#, BREQ, HOLD, HLDA
Interrupts	INTR, NMI
Floating Point Error Reporting	FERR#, IGNNE#
System Management Mode	SMI#, SMIACT#
Functional Redundancy Checking	FRCMC# (IERR#)
TAP Port	TCK, TMS, TDI, TDO, TRST#
Breakpoint/Performance Monitoring	PM0/BP0, PM1/BP1, BP3-2
Execution Tracing	BT3-BT0, IU, IV, IBT
Probe Mode	R/S#, PRDY

intel ®

1.6. OUTPUT PIN GROUPING ACCORDING TO WHEN DRIVEN

This section groups the output pins according to when they are driven.

Group 1

The following output pins are driven active at the beginning of a bus cycle with ADS#. A31-A3 and AP are guaranteed to remain valid until AHOLD is asserted or until the earlier of the clock after NA# or the last BRDY#. The remaining pins are guaranteed to remain valid until the earlier of the clock after NA# or the last BRDY#:

A31-A3, AP, BE7#-0#, CACHE#, M/IO#, W/R#, D/C#, SCYC, PWT, PCD.

Group 2

As outputs, the following pins are driven in T2, T12, and T2P. As inputs, these pins are sampled with BRDY#:

D63-0, DP7-0.

Group 3

These are the status output pins. They are always driven:

BREQ, HIT#, HITM#, IU, IV, IBT, BT3-BT0, PM0/BP0, PM1/BP1, BP3, BP2, PRDY, SMIACT#.

Group 4

These are the glitch free status output pins.
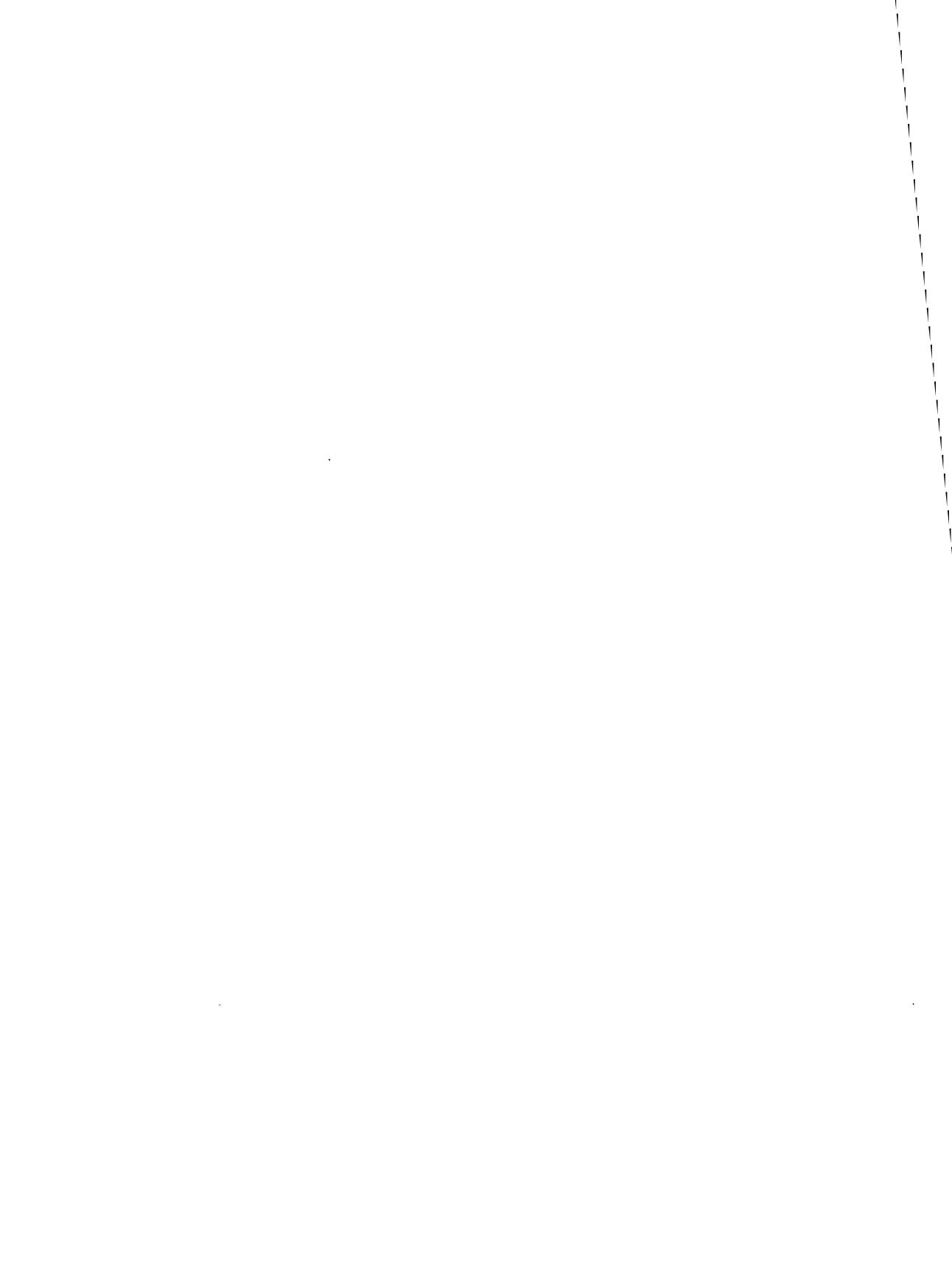
APCHK#, FERR#, HLDA, IERR#, LOCK#, PCHK#.

intel ®

2

Microprocessor Architecture Overview

intel®

CHAPTER 2
MICROPROCESSOR ARCHITECTURE OVERVIEW

The Pentium processor is the next generation member of the Intel386™ and Intel486™ microprocessor family. It is 100% binary compatible with the 8086/88, 80286, Intel386 DX CPU, Intel386 SX CPU, Intel486 DX CPU, Intel486 SX and the Intel486 DX2 CPUs.

The Pentium processor contains all of the features of the Intel486 CPU, and provides significant enhancements and additions including the following:

- Superscalar Architecture
- Dynamic Branch Prediction
- Pipelined Floating-Point Unit
- Improved Instruction Execution Time
- Separate 8K Code and Data Caches
- Writeback MESI Protocol in the Data Cache
- 64-Bit Data Bus
- Bus Cycle Pipelining
- Address Parity
- Internal Parity Checking
- Function Redundancy Checking
- Execution Tracing
- Performance Monitoring
- IEEE 1149.1 Boundary Scan
- System Management Mode
- Virtual Mode Extensions

The application instruction set of the Pentium processor includes the complete Intel486 CPU instruction set with extensions to accommodate some of the additional functionality of the Pentium processor. All application software written for the Intel386 and Intel486 microprocessors will run on the Pentium processor without modification. The on-chip memory management unit (MMU) is completely compatible with the Intel386 and Intel486 CPUs.

The Pentium processor implements several enhancements to increase performance. The two instruction pipelines and floating-point unit on the Pentium processor are capable of independent operation. Each pipeline issues frequently used instructions in a single clock. Together, the dual pipes can issue two integer instructions in one clock, or one floating point instruction (under certain circumstances, 2 floating point instructions) in one clock.

Branch prediction is implemented in the Pentium processor. To support this, the Pentium processor implements two prefetch buffers, one to prefetch code in a linear fashion, and one that prefetches code according to the BTB so the needed code is almost always prefetched before it is needed for execution.

The floating-point unit has been completely redesigned over the Intel486 CPU. Faster algorithms provide up to 10X speed-up for common operations including add, multiply, and load.

The Pentium processor includes separate code and data caches integrated on chip to meet its performance goals. Each cache is 8 Kbytes in size, with a 32-byte line size and is 2-way set associative. Each cache has a dedicated Translation Lookaside Buffer (TLB) to translate linear addresses to physical addresses. The data cache is configurable to be writeback or writethrough on a line by line basis and follows the MESI protocol. The data cache tags are triple ported to support two data transfers and an inquire cycle in the same clock. The code cache is an inherently write protected cache. The code cache tags are also triple ported to support snooping and split line accesses. Individual pages can be configured as cacheable or non-cacheable by software or hardware. The caches can be enabled or disabled by software or hardware.

The Pentium processor has increased the data bus to 64-bits to improve the data transfer rate. Burst read and burst writeback cycles are supported by the Pentium processor. In addition, bus cycle pipelining has been added to allow two bus cycles to be in progress simultaneously. The Pentium processor Memory Management Unit contains optional extensions to the architecture which allow 2 Mbyte and 4 Mbyte page sizes.

The Pentium processor has added significant data integrity and error detection capability. Data parity checking is still supported on a byte by byte basis. Address parity checking, and internal parity checking features have been added along with a new exception, the machine check exception. In addition, the Pentium processor has implemented functional redundancy checking to provide maximum error detection of the processor and the interface to the processor. When functional redundancy checking is used, a second processor, the "checker" is used to execute in lock step with the "master" processor. The checker samples the master's outputs and compares those values with the values it computes internally, and asserts an error signal if a mismatch occurs.

As more and more functions are integrated on chip, the complexity of board level testing is increased. To address this, the Pentium processor has increased test and debug capability. Like many of the Intel486 CPUs, the Pentium processor implements IEEE Boundary Scan (Standard 1149.1). In addition, the Pentium processor has specified 4 breakpoint pins that correspond to each of the debug registers and externally indicate a breakpoint match. Execution tracing provides external indications when an instruction has completed execution in either of the two internal pipelines, or when a branch has been taken.

System management mode has been implemented along with some extensions to the SMM architecture. Enhancements to the Virtual 8086 mode have been made to increase performance by reducing the number of times it is necessary to trap to a virtual 8086 monitor.

Figure 2-1 shows a block diagram of the Pentium processor.
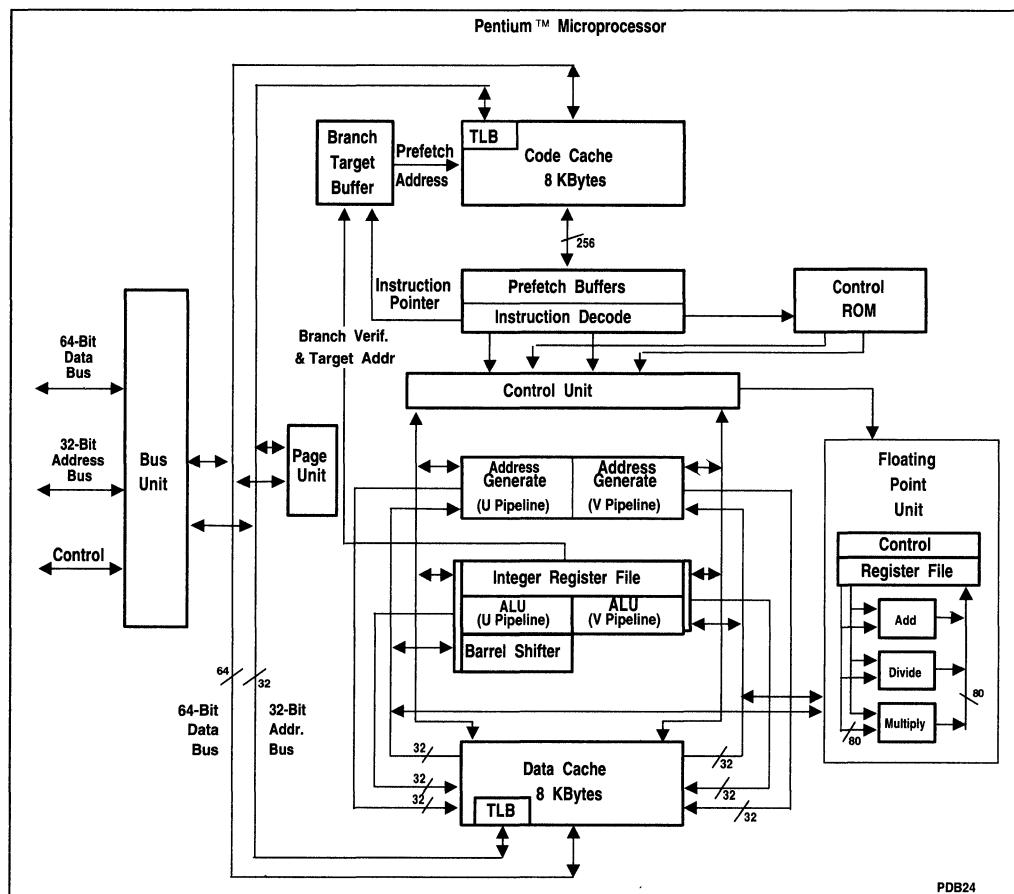
Figure 2-1. Pentium™ Processor Block Diagram

The block diagram shows the two instruction pipelines, the "u" pipe and the "v" pipe. The u-pipe can execute all integer and floating point instructions. The v-pipe can execute simple integer instructions and the FXCH floating point instructions.

The separate caches are shown, the code cache and data cache. The data cache has two ports, one for each of the two pipes (the tags are triple ported to allow simultaneous inquire cycles). The data cache has a dedicated Translation Lookaside Buffer (TLB) to translate linear addresses to the physical addresses used by the data cache.

The code cache, branch target buffer and prefetch buffers are responsible for getting raw instructions into the execution units of the Pentium processor. Instructions are fetched from the code cache or from the external bus. Branch addresses are remembered by the branch target buffer. The code cache TLB translates linear addresses to physical addresses used by the code cache.

The decode unit decodes the prefetched instructions so the Pentium processor can execute the instruction. The control ROM contains the microcode which controls the sequence of operations that must be performed to implement the Pentium processor architecture. The control ROM unit has direct control over both pipelines.

The Pentium processor contains a pipelined floating point unit that provides a significant floating point performance advantage over previous generations of the Pentium processor.

The architectural features introduced in this chapter are more fully described in the "Component Operation" chapter of this document.

intel®

3

**Component
Operation**

intel ®

CHAPTER 3
COMPONENT OPERATION

The Pentium processor has an optimized superscalar micro-architecture capable of executing two instructions in a single clock. A 64-bit external bus, separate 8 KByte data and instruction caches, write buffers, branch prediction, and a pipelined floating point unit combine to sustain the high execution rate. These architectural features and their operation are discussed in this chapter.

3.1. PIPELINE AND INSTRUCTION FLOW

Like the Intel486 CPU, integer instructions traverse a 5 stage pipeline. The pipeline stages are as follows:

PF   Prefetch
D1   Instruction Decode
D2   Address Generate
EX   Execute - ALU and Cache Access
WB   Write Back

Figure 3-1 shows how instructions move through the Intel486 CPU pipeline.
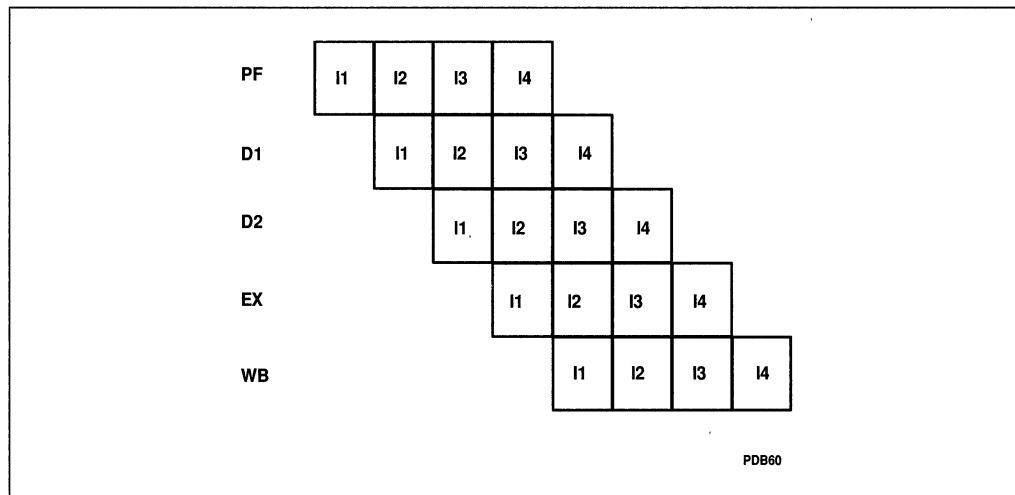
Figure 3-1. Intel486™ CPU Pipeline Execution

intel®

Unlike the Intel486 microprocessor, the Pentium processor is a superscalar machine capable of executing two instructions in parallel. Two five stage pipelines operate in parallel allowing integer instructions to execute in a single clock in each pipeline. Figure 3-2 depicts instruction flow in the Pentium processor.
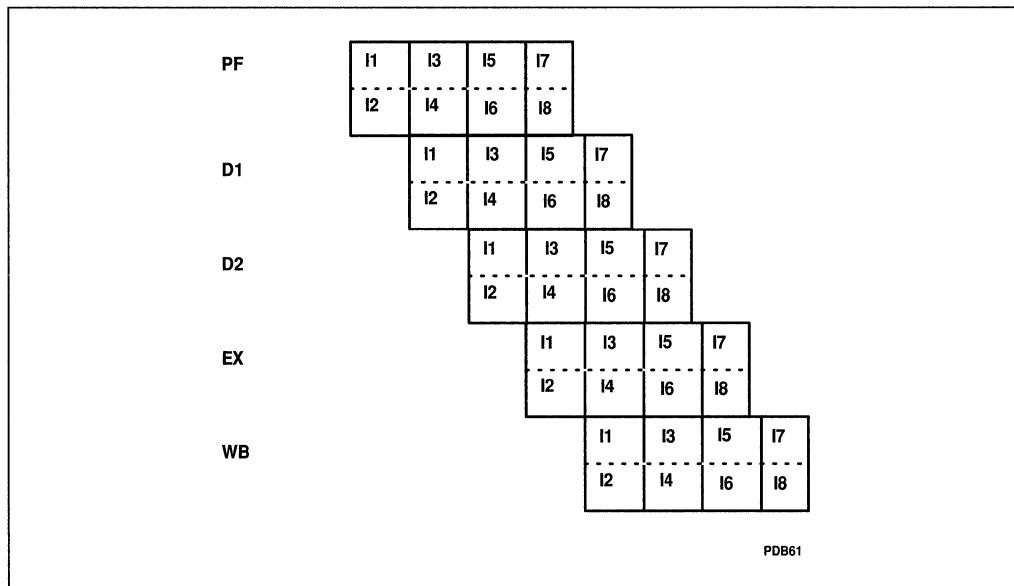
Figure 3-2. Pentium™ Processor Pipeline Execution

The pipelines in the Pentium processor are called the "u" and "v" pipes and the process of issuing two instructions in parallel is termed "pairing." The u-pipe can execute any instruction in the Intel X86 architecture while the v-pipe can execute "simple" instructions as defined in the "Instruction Pairing Rules" Section of this chapter. When instructions are paired, the instruction issued to the v-pipe is always the next sequential instruction after the one issued to the u-pipe.

3.1.1.   Pentium Processor Pipeline Description and Improvements

While the basic pipeline structure is the same as the Intel486 CPU, the Pentium processor pipeline has been optimized to achieve higher throughput.

The first stage of the pipeline is Prefetch (PF) stage in which instructions are prefetched from the on chip instruction cache or memory. Because the Pentium processor has separate caches for instructions and data, prefetches no longer conflict with data references for access to the cache. If the requested line is not in the code cache, a memory reference is made. In the PF stage, two independent pairs of line-size (32-byte) prefetch buffers operate in conjunction with the branch target buffer. This allows one prefetch buffer to prefetch instructions sequentially, while the other prefetches according to the branch target buffer predictions. The prefetch buffers alternate their prefetch paths. See the section titled "Instruction Prefetch" in this chapter

for further details on the Pentium processor prefetch buffers.

The next pipeline stage is Decode1 (D1) in which two parallel decoders attempt to decode and issue the next two sequential instructions. The decoders determine whether one or two instructions can be issued contingent upon the instruction pairing rules described in the section titled "Instruction Pairing Rules". The Pentium processor, similar to the Intel486 CPU, requires an extra D1 clock to decode instruction prefixes. Prefixes are issued to the u-pipe at the rate of one per clock without pairing. After all prefixes have been issued, the base instruction will then be issued and paired according to the pairing rules. The one exception to this is that the Pentium processor will decode near conditional jumps (long displacement) in the second opcode map (0Fh prefix) in a single clock in either pipeline.

The D1 stage is followed by Decode2 (D2) in which the address of memory resident operands are calculated similar to the Intel486 CPU. In the Intel486 CPU, instructions containing both a displacement and an immediate, or instructions containing a base and index addressing mode require an additional D2 clock to decode. The Pentium processor removes both of these restrictions and is able to issue instructions in these categories in a single clock.

Similar to the Intel486 CPU, the Pentium processor uses the Execute (EX) stage of the pipeline for both ALU operations and for data cache access; therefore those instructions specifying both an ALU operation and a data cache access will require more than one clock in this stage. In EX all u-pipe instructions and all v-pipe instructions except conditional branches are verified for correct branch prediction. Microcode is designed to utilize both pipelines and thus those instructions requiring microcode execute faster than on the Intel486 CPU.

The final stage is Writeback (WB) where instructions are enabled to modify processor state and complete execution. In this stage v-pipe conditional branches are verified for correct branch prediction.

During their progression through the pipeline instructions may be stalled due to certain conditions. Both the u-pipe and v-pipe instructions enter and leave the D1 and D2 stages in unison. When an instruction in one pipe is stalled then the instruction in the other pipe is also stalled at the same pipeline stage. Thus both the u-pipe and the v-pipe instructions enter the EX stage in unison. Once in EX if the u-pipe instruction is stalled, then the v-pipe instruction (if any) is also stalled. If the v-pipe instruction is stalled then the instruction paired with it in the u-pipe is allowed to advance. No successive instructions are allowed to enter the EX stage of either pipeline until the instructions in both pipelines have advanced to WB.

3.1.1.1.   INSTRUCTION PREFETCH

In the PF stage, two independent pairs of line-size (32-byte) prefetch buffers operate in conjunction with the branch target buffer. Only one prefetch buffer actively requests prefetches at any given time. Prefetches are requested sequentially until a branch instruction is fetched. When a branch instruction is fetched, the branch target buffer (BTB) predicts whether the branch will be taken or not. If the branch is predicted not taken, prefetch requests continue linearly. On a predicted taken branch the other prefetch buffer is enabled and begins to prefetch as though the branch was taken. If a branch is discovered mis-predicted, the instruction pipelines are flushed and prefetching activity starts over.

intel®

The dynamic branch prediction algorithm speculatively runs code fetch cycles to addresses corresponding to instructions executed some time in the past. Such code fetch cycles are run based on past execution history, regardless of whether the instructions retrieved are relevant to the currently executing instruction sequence.

One effect of the branch prediction mechanism is that the Pentium processor may run code fetch bus cycles to retrieve instructions which are never executed. Although the opcodes retrieved are discarded, the system must complete the code fetch bus cycle by returning BRDY#. It is particularly important that the system return BRDY# for all code fetch cycles, regardless of the address.

Furthermore, it is possible that the Pentium processor may run speculative code fetch cycles to addresses beyond the end of the current code segment. Although the Pentium processor may prefetch beyond the CS limit, it will not attempt to execute beyond the CS limit, it will raise a GP fault instead. Thus, segmentation cannot be used to prevent speculative code fetches to inaccessible areas of memory. On the other hand, the Pentium processor never runs code fetch cycles to inaccessible pages, so the paging mechanism guards against both the fetch and execution of instructions in inaccessible pages.

For memory reads and writes, both segmentation and paging prevent the generation of bus cycles to inaccessible regions of memory.

3.1.2. Instruction Pairing Rules

The Pentium processor can issue one or two instructions every clock. In order to issue two instructions simultaneously they must satisfy the following conditions:

- Both instructions in the pair must be "simple" as defined below
- There must be no read-after-write or write-after-after register dependencies between them
- Neither instruction may contain both a displacement and an immediate
- Instructions with prefixes (other than 0F of JCC instructions) can only occur in the u-pipe

Simple instructions are entirely hardwired; they do not require any microcode control and, in general, execute in one clock. The exceptions are the ALU mem,reg and ALU reg,mem instructions which are two and three clock operations respectively. Sequencing hardware is used to allow them to function as simple instructions. The following integer instructions are considered simple and may be paired:

1. mov reg, reg/mem/imm
2. mov mem, reg/imm
3. alu reg, reg/mem/imm
4. alu mem, reg/imm
5. inc reg/mem
6. dec reg/mem
7. push reg/mem
8. pop reg
9. lea reg,mem

10. jmp/call/jcc near

11. nop

In addition, conditional and unconditional branches may be paired only if they occur as the second instruction in the pair. They may not be paired with the next sequential instruction. Also, SHIFT/ROT by 1 and SHIFT by imm may pair as the first instruction in a pair.

The register dependencies that prohibit instruction pairing include implicit dependencies via registers or flags not explicitly encoded in the instruction. For example, an ALU instruction in the u-pipe (which sets the flags) may not paired with an ADC or a SBB instruction in the v-pipe. There are two exceptions to this rule. The first is the commonly occurring sequence of compare and branch which may be paired. The second exception is pairs of pushes or pops. Although these instructions have an implicit dependency on the stack pointer, special hardware is included to allow these common operations to proceed in parallel.

Although in general two paired instructions may proceed in parallel independently, there is an exception for paired "read-modify-write" instructions. Read-modify-write instructions are ALU operations with an operand in memory. When two of these instructions are paired there is a sequencing delay of two clocks in addition to the three clocks required to execute the individual instructions.

Although instructions may execute in parallel their behavior as seen by the programmer is exactly the same as if they were executed sequentially (as on the Intel486 CPU).

For information on code optimization, please see Appendix A.

3.2.   BRANCH PREDICTION

The Pentium processor uses a Branch Target Buffer to predict the outcome of branch instructions which minimizes pipeline stalls due to prefetch delays.

The processor accesses the BTB with the address of the instruction in the D1 stage. In the event of a correct prediction, a branch will execute without pipeline stalls or flushes. Branches which miss the BTB are assumed to be not taken. Conditional and unconditional near branches and near calls execute in 1 clock and may be executed in parallel with other integer instructions. A mispredicted branch (whether a BTB hit or miss) or a correctly predicted branch with the wrong target address will cause the pipelines to be flushed and the correct target to be fetched. Incorrectly predicted unconditional branches will incur an additional three clock delay, incorrectly predicted conditional branches in the u-pipe will incur an additional three clock delay, and incorrectly predicted conditional branches in the v-pipe will incur an additional four clock delay.

The benefits of branch prediction are illustrated in the following example. Consider the following loop from a benchmark program for computing prime numbers:

```
for(k=i+prime;k<=SIZE;k+=prime)
      flags[k]=FALSE;
```

A popular compiler generates the following assembly code:

(prime is allocated to `ecx,` `k` is allocated to `edx`, and `al` contains the value `FALSE`)

intel®

```
inner_loop:
     mov byte ptr flags[edx],al
     add edx,ecx
     cmp edx, FALSE
     jle inner_loop
```

Each iteration of this loop will execute in 6 clocks on the Intel486 CPU. On the Pentium processor, the `mov` is paired with the `add`; the `cmp` with the `jle`. With branch prediction, each loop iteration executes in 2 clocks.

3.3. WRITE BUFFERS AND MEMORY ORDERING

The Pentium processor has two write buffers, one corresponding to each of the pipelines, to enhance the performance of consecutive writes to memory. These write buffers are one quad-word wide (64-bits) and can be filled simultaneously in one clock e.g., by two simultaneous write misses in the two instruction pipelines. Writes in these buffers are driven out on the external bus in the order they were generated by the processor core. No reads (as a result of cache miss) are reordered around previously generated writes sitting in the write buffers. The implication of this is that the write buffers will be flushed or emptied before a subsequent bus cycle is run on the external bus (unless BOFF# is asserted and a writeback cycle becomes pending, see section 3.3.3.).

The Pentium processor supports strong write ordering only. That is, writes generated by the Pentium processor will be driven to the bus or updated in the cache in the order that they occur. The Pentium processor will not write to E or M-state lines in the data cache if there is a write in either write buffer, if a write cycle is running on the bus, or if EWBE# is active.

Note that only memory writes are buffered and I/O writes are not. There is no guarantee of synchronization between completion of memory writes on the bus and instruction execution after the write. The OUT instruction or a serializing instruction needs to be executed to synchronize writes with the next instruction. Please refer to the "Serializing Operations" section for more information.

No re-ordering of read cycles occurs on the Pentium processor. Specifically, the write buffers are flushed before the IN instruction is executed.

3.3.1. External Event Synchronization

When the system changes the value of NMI, INTR, FLUSH#, SMI# or INIT as the result of executing an OUT instruction, these inputs must be at a valid state three clocks before BRDY# is returned to ensure that the new value will be recognized before the next instruction is executed.

Note that if an OUT instruction is used to modify A20M#, this will not affect previously prefetched instructions. A serializing instruction must be executed to guarantee recognition of A20M# before a specific instruction.

3.3.2. Serializing Operations

After executing certain instructions the Pentium processor serializes instruction execution. This means that any modifications to flags, registers, and memory for previous instructions are completed before the next instruction is fetched and executed. The prefetch queue is flushed as a result of serializing operations.

The Pentium processor serializes instruction execution after executing one of the following instructions: Move to Special Register (except CR0), INVD, INVLPG, IRET, IRETD, LGDT, LLDT, LIDT, LTR, WBINVD, CPUID, RSM and WRMSR.

Notes:

1. The CPUID instruction can be executed at any privilege level to serialize instruction execution.

2. When the Pentium processor serializes instruction execution, it ensures that it has completed any modifications to memory, including flushing any internally buffered stores; it then waits for the EWBE# pin to go active before fetching and executing the next instruction. Pentium processor systems may use the EWBE# pin to indicate that a store is pending externally. In this manner, a system designer may ensure that all externally pending stores will complete before the Pentium processor begins to fetch and execute the next instruction.

3. The Pentium processor does not generally write back the contents of modified data in its data cache to external memory when it serializes instruction execution. Software can force modified data to be written back by executing the WBINVD instruction.

4. Whenever an instruction is executed to enable or disable paging, then that instruction and the following instructions must be located on a page whose linear address is identical to its physical address. Note that the Intel386, Intel486 and Pentium processor have slightly different requirements to enable and disable paging.

5. Whenever an instruction is executed to change the contents of CR3 while paging is enabled, then that instruction and the following instructions should be located on a page whose linear address is mapped to the same physical address by both the old and new values of CR3.

6. The Pentium processor implements branch-prediction techniques to improve performance by prefetching the destination of a branch instruction before the branch instruction is executed. Consequently, instruction execution is not generally serialized when a branch instruction is executed.

7. Although the I/O instructions are not "serializing" because the processor does not wait for these instructions to complete before it prefetches the next instruction, they do have the following properties that cause them to function in a manner that is identical to previous generations. I/O reads are not re-ordered within the processor; they wait for all internally pending stores to complete. Note that the Pentium processor does not sample the EWBE# pin during reads. If necessary, external hardware must ensure that externally pending stores are complete before returning BRDY#. This is the same requirement that exists on

Intel386 and Intel486 systems. The OUT and OUTS instructions are also not "serializing", as they do not stop the prefetcher. They do, however, ensure that all internally buffered stores have completed, that EWBE# has been sampled active indicating that all externally pending stores have completed and that the I/O write has completed before they begin to execute the next instruction. Note that unlike the Intel486 CPU, it is not necessary for external hardware to ensure that externally pending stores are complete before returning BRDY#.

3.3.3. Linefill and Writeback Buffers

In addition to the write buffers corresponding to each of the internal pipelines, the Pentium processor has 3 writeback buffers. Each of the writeback buffers are 1 deep and 32-bytes (1 line) wide.

There is a dedicated replacement writeback buffer which stores writebacks caused by a linefill that replaces a modified line in the data cache. There is one external snoop writeback buffer that stores writebacks caused by an inquire cycle that hits a modified line in the data cache. Finally, there is an internal snoop writeback buffer that stores writebacks caused by an internal snoop cycle that hits a modified line in the data cache (Internal and external snoops are discussed in detail in the Inquire Cycle section of the Bus Functional Description chapter of this document). Write cycles are driven to the bus with the following priority:

- Contents of external snoop writeback buffer
- Contents of internal snoop writeback buffer
- Contents of replacement writeback buffer
- Contents of write buffers.

Note that the contents of whichever write buffer was written into first is driven to the bus first. If both write buffers were written to in the same clock, the contents of the u-pipe buffer is written out first.

The Pentium processor also implements two line fill buffers, one for the data cache and one for the code cache. As information (data or code) is returned to the Pentium processor for a cache line fill, it is written into the line fill buffer. After the entire line has been returned to the processor it is transferred to the cache. Note that the processor requests the needed information first and uses that information as soon as it is returned. The Pentium processor does not wait for the line fill to complete before using the requested information.

If a linefill causes a modified line in the data cache to be replaced, the replaced line will remain in the cache until the line fill is complete. After the line fill is complete, the line being replaced is moved into the replacement writeback buffer and the new line fill is moved into the cache.

3.4. EXTERNAL INTERRUPT CONSIDERATIONS

The Pentium processor recognizes the following external interrupts: BUSCHK#, R/S#, FLUSH#, SMI#, INIT, NMI, and INTR. These interrupts are recognized at instruction boundaries. On the Pentium processor, the instruction boundary is the first clock in the

execution stage of the instruction pipeline. This means that before an instruction is executed, the Pentium processor checks to see if any interrupts are pending. If an interrupt is pending, the processor flushes the instruction pipeline and then services the interrupt. The priority order of external interrupts is as shown below:

- BUSCHK#
- R/S#
- FLUSH#
- SMI#
- INIT
- NMI
- INTR

3.5.   MODEL SPECIFIC REGISTERS

The Pentium processor defines certain Model Specific Registers that are used in execution tracing, performance monitoring, testing, and machine check errors. They are unique to the Pentium processor and may not be implemented, or may not be implemented in the same way in future processors.

Two new instructions, RDMSR and WRMSR (read/write model specific registers) are used to access these registers. When these instructions are executed, the value in ECX specifies which model specific register is being accessed. Table 3-1 lists all model specific registers and the corresponding values (in Hex) that need to be loaded into ECX to access them.

Table 3-1. Model Specific Registers

Value (in Hex)	Register Name	Description
00H	Machine Check Address	Stores address of cycle causing the exception
01H	Machine Check Type	Stores cycle type of cycle causing the exception
0EH	Test Register 12 (TR12)	New feature control

NOTE: Do not execute RDMSR or WRMSR with undefined values in ECX.

Software must not depend on the value of reserved bits in the model specific registers. Any writes to the model specific registers should write "0" into any reserved bits.

3.6.   FLOATING POINT UNIT

The floating point unit (FPU) of the Pentium processor is integrated with the integer unit on the same chip. It is heavily pipelined. The FPU is designed to be able to accept one floating point operation every clock. It can receive up to two floating point instructions every clock, one of which must be an exchange instruction.

For information on code optimization, please see Appendix A.

3.6.1.   Floating Point Pipeline Stages

The Pentium processor FPU has 8 pipeline stages, the first five of which it shares with the integer unit. Integer instructions pass through only the first 5 stages. Integer instructions use the fifth (X1) stage as a WB (write-back) stage. The 8 FP pipeline stages, and the activities that are performed in them are summarized below:

PF   Prefetch;

D1   Instruction Decode;

D2   Address generation;

EX   Memory and register read; conversion of FP data to external memory format and memory write;

X1   Floating Point Execute stage one; conversion of external memory format to internal FP data format and write operand to FP register file; bypass1 (bypass1 described in the "Bypasses" section).

X2   Floating Point Execute stage two;

WF   Perform rounding and write floating-point result to register file; bypass 2 (bypass2 described in the "Bypasses" section).

ER   Error Reporting/Update Status Word.

3.6.2.   Instruction Issue

Described below are the rules of how floating point (FP) instructions get issued on the Pentium processor:

1. FP instructions do not get paired with integer instructions. However, a limited pairing of two FP instructions can be performed.

2. When a pair of FP instructions is issued to the FPU, only the FXCH instruction can be the second instruction of the pair. The first instruction of the pair must be one of a set F where F = [FLD single/double, FLD ST(i), all forms of FADD, FSUB, FMUL, FDIV, FCOM, FUCOM, FTST, FABS, FCHS].

3. FP instructions other than the FXCH instruction and other than instructions belonging to set F (defined in rule 2) always get issued singly to the FPU.

4. FP instructions that are not directly followed by an FP exchange instruction are issued singly to the FPU.

The Pentium processor stack architecture instruction set requires that all instructions have one source operand on the top of the stack. Since most instructions also have their destination as the top of the stack, most instructions see a "top of stack bottleneck". New source operands must be brought to the top of the stack before we can issue an arithmetic instruction on them. This calls for extra usage of the exchange instruction, which allows the programmer to bring an available operand to the top of the stack. The Pentium processor FPU uses pointers to access its registers to allow fast execution of exchanges and the execution of exchanges in parallel with other floating point instructions. An FP exchange that is paired with other FP

instructions takes 0 clocks for its execution. Since such exchanges can be executed in parallel on the Pentium processor, it is recommended that one use them when necessary to overcome the stack bottleneck.

Note that when exchanges are paired with other floating point instructions, they should not be followed immediately by integer instructions. The Pentium processor stalls such integer instructions for a clock if the FP pair is declared safe, or for 4 clocks if the FP pair is unsafe.

Also note that the FP exchange must always follow another FP instruction to get paired. The pairing mechanism does not allow the FP exchange to be the first instruction of a pair that is issued in parallel. If an FP exchange is not paired, it takes 1 clock for its execution.

3.6.3.   Safe Instruction Recognition

The Pentium processor FPU performs Safe Instruction Recognition or SIR in the X1 stage of the pipeline. SIR is an early inspection of operands and opcodes to determine whether the instruction is guaranteed not to generate an arithmetic overflow, underflow, or unmasked inexact exception. An instruction is declared safe if it cannot raise any other floating point exception, and if it does not need microcode assist for delivery of special results. If an instruction is declared safe, the next FP instruction is allowed to complete its E stage operation. If an instruction is declared unsafe, the next FP instruction stalls in the E stage until the current one completes (ER stage) with no exception. This means a 4 clock stall, which is incurred even if the numeric instruction that was declared unsafe does not eventually raise a floating point exception.

For normal data, the rules used on the Pentium processor for declaring an instruction safe are as follows.

If FOP= FADD/FSUB/FMUL/FDIV, the instruction is safe from arithmetic overflow, underflow, and unmasked inexact exceptions if:

1. Both operands have unbiased exponent =< 1FFEh AND
2. Both operands have unbiased exponent >= -1FFEh AND
3. The inexact exception is masked

Note that arithmetic overflow of the double precision format occurs when the unbiased exponent of the result is >= 400h, and underflow occurs when the exponent is <=-3FFh. Hence, the SIR algorithm on the Pentium processor allows improved throughput on a much greater range of numbers than that spanned by the double precision format.

3.6.4.   Bypasses

The following section describes the floating point register file bypasses that exist on the Pentium processor. The register file has two write ports and two read ports. The read ports are used to read data out of the register file in the E stage. One write port is used to write data into the register file in the X1 stage, and the other in the WF stage. A bypass allows data that is about to be written into the register file to be available as an operand that is to be read from the register file by any succeeding floating point instruction. A bypass is specified by a pair of ports (a write port and a read port) that get circumvented. Using the bypass, data is made

available even before actually writing it to the register file.

The following procedures are implemented:

1. Bypass the X1 stage register file write port and the E stage register file read port.
2. Bypass the WF stage register file write port and the E stage register file read port.

With bypass 1, the result of a floating point load (that writes to the register file in the X1 stage) can bypass the X1 stage write and be sent directly to the operand fetch stage or E stage of the next instruction.

With bypass 2, the result of any arithmetic operation can bypass the WF stage write to the register file, and be sent directly to the desired execution unit as an operand for the next instruction.

Note that the FST instruction reads the register file with a different timing requirement, so that for the FST instruction, which attempts to read an operand in the E stage:

1. There is no bypassing the X1 stage write port and the E stage read port, i.e. no added bypass for FLD followed by FST. Thus FLD (double) followed by FST (double) takes 4 clocks (2 for FLD, and 2 for FST).
2. There is no bypassing the WF stage write port and the E stage read port. The E stage read for the FST happens only in the clock following the WF write for any preceding arithmetic operation.

Furthermore, there is no memory bypass for an FST followed by an FLD from the same memory location.

3.6.5. Branching upon Numeric Condition Codes

Branching upon numeric condition codes is accomplished by transferring the floating point SW to the integer FLAGS register and branching on it. The "test numeric condition codes and branch" construct looks like:

FP instruction1; instruction whose effects on the status word are to be examined;

"numeric_test_and_branch_construct":

FSTSW AX; move the status word to the ax register.

SAHF; transfer the value in ax to the upper half of the eflags register.

JC xyz ; jump upon the condition codes in the eflags register.

Note that all FP instructions update the status word only in the ER stage. Hence there is a built-in status word interlock between FP instruction1 and the FSTSW AX instruction. The above piece of code takes 9 clocks before execution of code begins at the target of the jump. These 9 clocks are counted as:

FP instruction1 :   X1, X2, WF, ER (4 E stage stalls for the FSTSWAX);
FSTSW AX :   2 E clocks;
SAHF :   2 E clocks;

JC xyz :   1 clock if no mispredict on branch.

Note that if there is a branch mispredict, there will be a minimum of 3 clocks added to the clock count of 9.

It is recommended that such attempts to branch upon numeric condition codes be preceded by integer instructions, i.e. one should insert integer instructions in between FP instruction1 and the FSTSW AX instruction which is the first instruction of the "numeric test and branch" construct. This allows the elimination of up to 4 clocks (the 4 E-stage stalls on FSTSW AX) from the cost attributed to this construct, so that numeric branching can be accomplished in 5 clocks.

3.7.   ON CHIP CACHES

The Pentium processor implements two internal caches for a total integrated cache size of 16 Kbytes: an 8 Kbyte data cache and a separate 8 Kbyte code cache.   These caches are transparent to application software to maintain compatibility with previous generations of the Intel386/Intel486 architecture.

The data cache fully supports the MESI (modified/exclusive/shared/invalid) writeback cache consistency protocol. The code cache is inherently write protected to prevent code from being inadvertently corrupted, and as a consequence supports a subset of the MESI protocol, the S (shared) and I (invalid) states.

The caches have been designed for maximum flexibility and performance. The data cache is configurable as writeback or writethrough on a line by line basis. Memory areas can be defined as non-cacheable by software and external hardware. Cache writeback and invalidations can be initiated by hardware or software. Protocols for cache consistency and line replacement are implemented in hardware, easing system design.

3.7.1.   Cache Organization

Each of the caches are 8 Kbytes in size and each is organized as a 2-way set associative cache. There are 128 sets in each cache, each set containing 2 lines (each line has its own tag address). Each cache line is 32 bytes wide.

Replacement in both the data and instruction caches is handled by the LRU mechanism which requires one bit per set in each of the caches. A conceptual diagram of the organization of the data and code caches is shown below in Figure 3-3. Note that the data cache supports the MESI writeback cache consistency protocol which requires 2 state bits, while the code cache supports the S and I state only and therefore requires only one state bit.
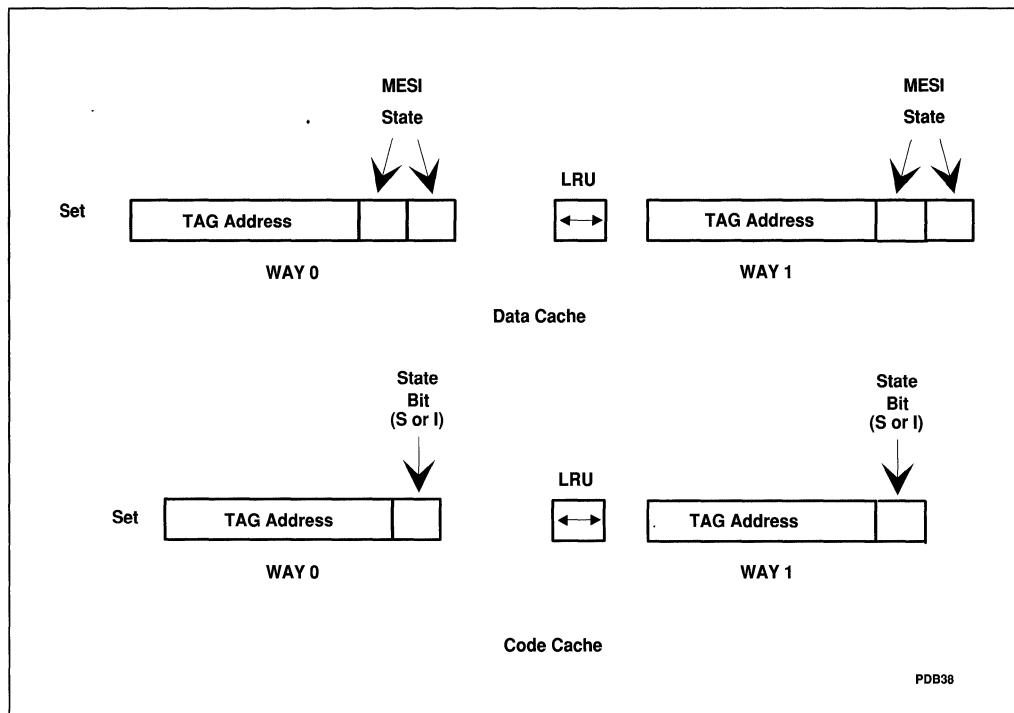
Figure 3-3. Conceptual Organization of Code and Data Caches

3.7.2. Cache Structure

The instruction and data caches can be accessed simultaneously. The instruction cache can provide up to 32 bytes of raw opcodes and the data cache can provide data for two data references all in the same clock. This capability is implemented partially through the tag structure. The tags in the data cache are triple ported. One of the ports is dedicated to snooping while the other two are used to lookup two independent addresses corresponding to data references from each of the pipelines. The instruction cache tags are also triple ported. Again, one port is dedicated to support snooping and other two ports facilitate split line accesses (simultaneously accessing upper half of one line and lower half of the next line).

The storage array in the data cache is single ported but interleaved on 4 byte boundaries to be able to provide data for two simultaneous accesses to the same cache line.

Each of the caches are parity protected. In the instruction cache, there are parity bits on a quarter line basis and there is one parity bit for each tag. The data cache contains one parity bit for each tag and a parity bit per byte of data.

Each of the caches are accessed with physical addresses and each cache has its own TLB (translation lookaside buffer) to translate linear addresses to physical addresses. The data cache has a 4-way set associative, 64-entry TLB for 4 Kbyte pages and a separate 4-way set associative, 8-entry TLB to support 4 Mbyte pages. The code cache has one 4-way set

associative, 32-entry TLB for 4 Kbyte pages and 4 Mbyte pages which are cached in 4 Kbyte increments. The TLBs associated with the instruction cache are single ported whereas the data cache TLBs are fully dual ported to be able to translate two independent linear addresses for two data references simultaneously. Replacement in the TLBs is handled by a pseudo LRU mechanism (similar to the Intel486 CPU) that requires 3 bits per set. The tag and data arrays of the TLBs are parity protected with a parity bit associated with each of the tag and data entries in the TLBs.

3.7.3. Cache Operating Modes

The operating modes of the caches are controlled by the CD (cache disable) and NW (not write through) bits in CR0. See Table 3-2 for a description of the modes. For normal operation and highest performance, these bits should both be reset to "0". The bits come out of RESET as CD = NW = 1.

intel®

Table 3-2. Cache Operating Modes

CD	NW	Description
1	1	Read hits access the cache.
		Read misses do not cause linefills.
		Write hits update the cache, but do not access memory.
		Write hits will cause Exclusive State lines to change to Modified State.
		Shared lines will remain in the Shared state after write hits.
		Write misses access memory.
		Inquire and invalidation cycles do not effect the cache state or contents.
		This is the state after reset.
1	0	Read hits access the cache.
		Read misses do not cause linefills.
		Write hits update the cache.
		Writes to Shared lines and write misses update external memory.
		Writes to Shared lines can be changed to the Exclusive State under the control of the WB/WT# pin.
		Inquire cycles (and invalidations) are allowed.
0	1	GP(0)
0	0	Read hits access the cache.
		Read misses may cause linefills.
		These lines will enter the Exclusive or Shared state under the control of the WB/WT# pin.
		Write hits update the cache.
		Only writes to shared lines and write misses appear externally.
		Writes to Shared lines can be changed to the Exclusive State under the control of the WB/WT# pin.
		Inquire cycles (and invalidations) are allowed.

To completely disable the cache, the following two steps must be performed.

1. CD and NW must be set to 1.
2. The caches must be flushed.

If the cache is not flushed, cache hits on reads will still occur and data will be read from the cache. In addition, the cache must be flushed after being disabled to prevent any inconsistencies with memory.

3.7.4. Page Cacheability

Two bits for cache control, PWT and PCD are defined in the page table and page directory entries. The state of these bits are driven out on the PWT and PCD pins during memory access cycles. The PWT bit controls write policy for the second level caches used with the Pentium processor. Setting PWT to 1 defines a write through policy for the current page, while clearing PWT to 0 defines a writeback policy for the current page.

The PCD bit controls cacheability on a page by page basis. The PCD bit is internally ANDed with the KEN# signal to control cacheability on a cycle by cycle basis. PCD = 0 enables cacheing, while PCD = 1 disables it. Cache line fills are enabled when PCD = 0 and KEN# = 0.

3.7.4.1. PCD AND PWT GENERATION

The value driven on PCD is a function of the PCD bits in CR3, the page directory pointer, the page directory entry and the page table entry, and the CD and PG bits in CR0.

The value driven on PWT is a function of the PCD bits in CR3, the page directory pointer, the page directory entry and the page table entry, and the PG bit in CR0 (CR0.CD does not affect PWT).

CR0.CD = 1

If cacheing is disabled, the PCD pin is always driven high. CR0.CD does not affect the PWT pin.

CR0.PG = 0

If paging is disabled, the PWT pin is forced low and the PCD pin reflects the CR0.CD. The PCD and PWT bits in CR3 are assumed 0 during the cacheing process.

CR0.CD = 0, PG = 1, normal operation

The PCD and PWT bits from the last entry (can be either PDE or PTE, depends on 4 Mbyte or 4 Kbyte mode) are cached in the TLB and are driven anytime the page mapped by the TLB entry is referenced.

CR0.CD = 0, PG = 1, during TLB Refresh

During TLB refresh cycles when the PDE and PTE entries are read, the PWT and PCD bits are obtained as shown in Tables 3-3 and 3-4.

Table 3-3. 32-Bits/4KB Pages

PCD/PWT taken from	During accesses to:
CR3	PDE
PDE	PTE
PTE	all other paged mem references

Table 3-4. 32-Bits/4MB Pages

PCD/PWT taken from:	During accesses to:
CR3	PDE
PDE	all other paged mem references

Figure 3-4 shows how PCD and PWT are generated.
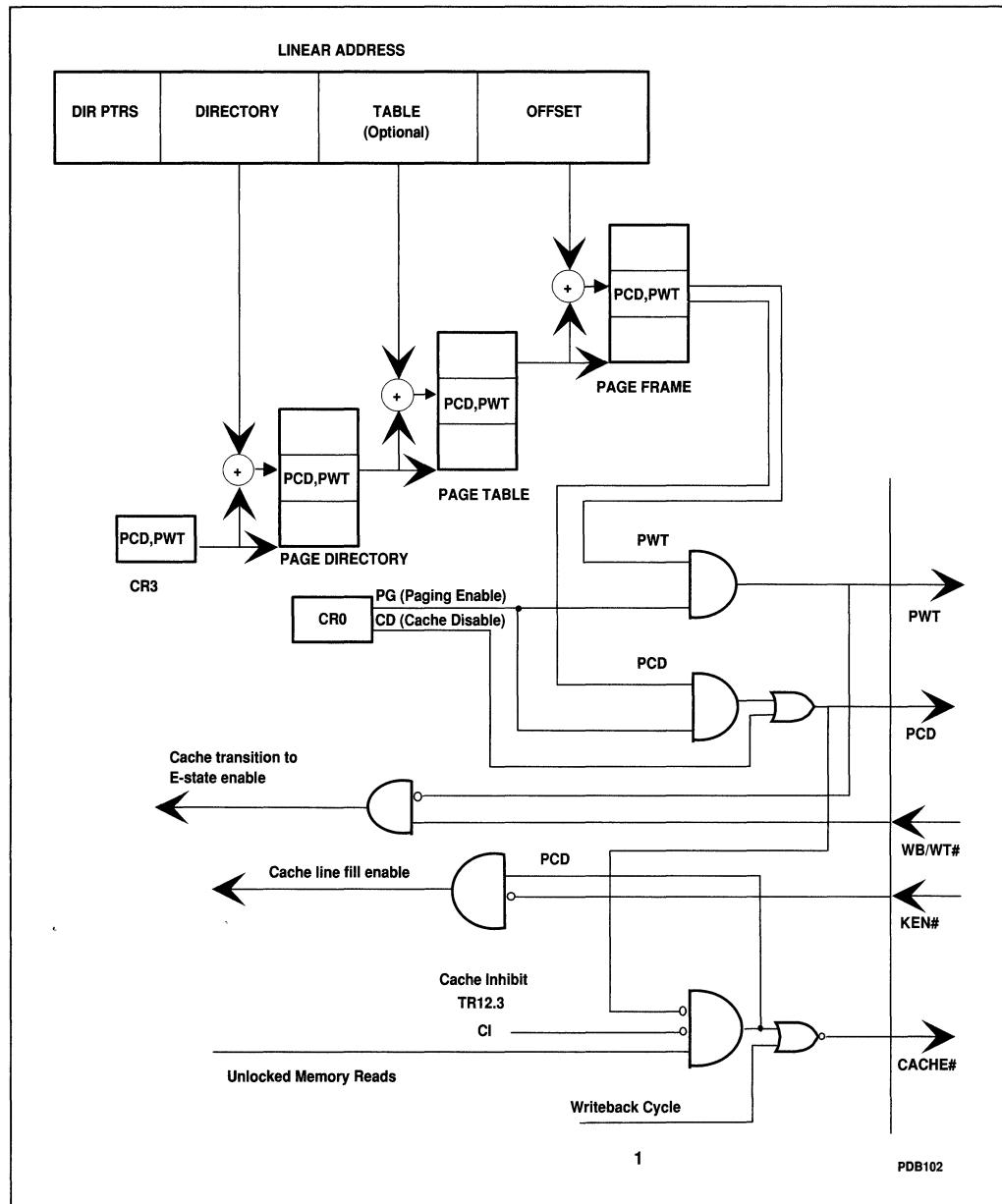
Figure 3-4. PCD and PWT Generation

3.7.5.   Inquire Cycles

Inquire cycles are initiated by the system to determine if a line is present in the code or data cache, and what its state is (This document refers to inquire cycles and snoop cycles interchangeably).

Inquire cycles are driven to the Pentium processor when a bus master other than the Pentium processor initiates a read or write bus cycle. Inquire cycles are driven to the Pentium processor when the bus master initiates a read to determine if the Pentium processor data cache contains the latest information. If the snooped line is in the Pentium processor data cache in the modified state, the Pentium processor has the most recent information and must schedule a writeback of the data. Inquire cycles are driven to the Pentium processor when the other bus master initiates a write to determine if the Pentium processor code or data cache contains the snooped line and to invalidate the line if it is present. Inquire cycles are described in detail in the "Bus Functional Description" chapter.

3.7.6.   Cache Flushing

The on chip cache can be flushed by external hardware or by software instructions.

Flushing the cache through hardware is accomplished by driving the FLUSH# pin low. This causes the cache to write back all modified lines in the data cache and mark the state bits for both caches invalid. The Flush Acknowledge special cycle is driven by the Pentium processor when all writebacks and invalidations are complete.

The INVD and WBINVD instructions cause the on-chip caches to be invalidated also. WBINVD causes the modified lines in the internal data cache to be written back, and all lines in both caches to be marked invalid. After execution of the WBINVD instruction, the Writeback and Flush special cycles are driven to indicate to any external cache that it should writeback and invalidate its contents.

INVD causes all lines in both caches to be invalidated. Modified lines in the data cache are not written back. The Flush special cycle is driven after the INVD instruction is executed to indicate to any external cache that it should invalidate its contents. Care should be taken when using the INVD instruction that cache consistency problems are not created.

Note that the implementation of the INVD and WBINVD instructions are processor dependent. Future processor generations may implement these instructions differently.

3.7.7.   Data Cache Consistency Protocol (MESI Protocol)

The Pentium processor Cache Consistency Protocol is a set of rules by which states are assigned to cached entries (lines). The rules apply for memory read/write cycles only. I/O and special cycles are not run through the data cache.

Every line in the Pentium processor data cache is assigned a state dependent on both Pentium processor generated activities and activities generated by other bus masters (snooping). The Pentium processor Data Cache Protocol consists of 4 states that define whether a line is valid

(HIT/MISS), if it is available in other caches, and if it has been MODIFIED. The four states are the M (Modified), E (Exclusive), S (Shared) and the I (Invalid) states and the protocol is referred to as the MESI protocol. A definition of the states is given below:

M - Modified: An M-state line is available in ONLY one cache and it is also MODIFIED (different from main memory). An M-state line can be accessed (read/written to) without sending a cycle out on the bus.

E - Exclusive: An E-state line is also available in ONLY one cache in the system, but the line is not MODIFIED (i.e. it is the same as main memory). An E-state line can be accessed (read/written to) without generating a bus cycle. A write to an E-state line will cause the line to become MODIFIED.

S - Shared: This state indicates that the line is potentially shared with other caches (i.e. the same line may exist in more than one cache). A read to an S-state line will not generate bus activity, but a write to a SHARED line will generate a write-through cycle on the bus. The write-through cycle may invalidate this line in other caches. A write to an S-state line will update the cache.

I - Invalid: This state indicates that the line is not available in the cache. A read to this line will be a MISS and may cause the Pentium processor to execute a LINE FILL (fetch the whole line into the cache from main memory). A write to an INVALID line will cause the Pentium processor to execute a write-through cycle on the bus.

3.7.7.1.   STATE TRANSITION TABLES

Lines cached in the Pentium processor can change state because of processor generated activity or as a result of activity on the Pentium processor bus generated by other bus masters (snooping). State transitions happen because of processor generated transactions (memory reads/writes) and by a set of external input signals and internally generated variables. The Pentium processor also drives certain pins as a consequence of the Cache Consistency Protocol.

3.7.7.1.1.   Read Cycle

Table 3-5 shows the state transitions for lines in the data cache during unlocked read cycles.

Table 3-5. Data Cache State Transitions for UNLOCKED Pentium™ Processor Initiated Read Cycles*

Present State	Pin Activity	Next State	Description
M	n/a	M	Read hit; data is provided to processor core by cache. No bus cycle is generated.
E	n/a	E	Read hit; data is provided to processor core by cache. No bus cycle is generated.
S	n/a	S	Read hit; Data is provided to the processor by the cache. No bus cycle is generated.
I	CACHE# low AND KEN# low AND WB/WT# high AND PWT low	E	Data item does not exist in cache (MISS). A bus cycle (read) will be generated by the Pentium™ processor. This state transition will happen if WB/WT# is sampled high with first BRDY# or NA#.
I	CACHE# low AND KEN# low AND (WB/WT# low OR PWT high)	S	Same as previous read miss case except that WB/WT# is sampled low with first BRDY# or NA#.
I	CACHE# high OR KEN# high	I	KEN# pin inactive; the line is not intended to be cached in the Pentium processor.

*Locked accesses to the data cache will cause the accessed line to transition to the Invalid state

Note the transition from I to E or S states (based on WB/WT#) happens only if KEN# is sampled low with the first of BRDY# or NA#, and the cycle is transformed into a LINE FILL cycle. If KEN# is sampled high, the line is not cached and remains in the I state.

3.7.7.1.2. Write Cycle

The state transitions of data cache lines during Pentium processor generated write cycles are illustrated in the next table. Writes to SHARED lines in the data cache are always sent out on the bus along with updating the cache with the write item. The status of the PWT and WB/WT# pins during these write cycles on the bus determines the state transitions in the data cache during writes to S-state lines.

A write to a SHARED line in the data cache will generate a write cycle on the Pentium processor bus to update memory and/or invalidate the contents of other caches. If the PWT pin

is driven high when the write cycle is run on the bus, the line will be updated, and will stay in the S-state regardless of the status of the WB/WT# pin that is sampled with the first BRDY# or NA#. If PWT is driven low, the status of the WB/WT# pin sampled along with the first BRDY# or NA# for the write cycle determines what state (E:S) the line transitions to.

The state transition from S to E is the only transition in which the data and the status bits are not updated at the same time. The data will be updated when the write is written to the Pentium processor write buffers. The state transition does not occur until the write has completed on the bus (BRDY# has been returned). Writes to the line after the transition to the E-state will not generate bus cycles. However, it is possible that writes to the same line that were buffered or in the pipeline before the transition to the E state will generate bus cycles after the transition to E-state.

An inactive EWBE# input will stall subsequent writes to an E or an M state line. All subsequent writes to E or M state lines are held off until EWBE# is returned active.

intel.

Table 3-6. Data Cache State Transitions for Pentium™ Processor Initiated Write Cycles

Present State	Pin Activity	Next State	Description
M	n/a	M	Write hit; update data cache. No bus cycle generated to update memory.
E	n/a	M	Write hit; update cache only. No bus cycle generated; line is now MODIFIED.
S	PWT low AND WB/WT# high	E	Write hit; data cache updated with write data item. A write-through cycle is generated on bus to update memory and/or invalidate contents of other caches. The state transition occurs after the writethrough cycle completes on the bus (with the last BRDY#).
S	PWT low AND WB/WT# low	S	Same as above case of write to S-state line except that WB/WT# is sampled low.
S	PWT high	S	Same as above cases of writes to S state lines except that this is a write hit to a line in a write through page; status of WB/WT# pin is ignored.
I	n/a	I	Write MISS; a write through cycle is generated on the bus to update external memory. No allocation done.

Note that memory writes are buffered while I/O writes are not. There is no guarantee of synchronization between completion of memory writes on the bus and instruction execution after the write. A serializing instruction needs to be executed to synchronize writes with the next instruction if necessary.

3.7.7.1.3.   Inquire Cycles (Snooping)

The purpose of inquire cycles is to check whether the address being presented is contained within the caches in the Pentium processor. Inquire cycles may be initiated with or without an INVALIDATION request (INV = 1 or 0). An inquire cycle is run through the data and code caches through a dedicated snoop port to determine if the address is in one of the Pentium processor caches. If the address is in a Pentium processor cache, the HIT# pin is asserted. If the address hits a modified line in the data cache, the HITM# pin is also asserted and the modified line is then written back onto the bus.

The state transition tables for inquire cycles are given below:

Table 3-7. Cache State Transitions During Inquire Cycles

Present State	Next State INV=1	Next State INV=0	Description
M	I	S	Snoop hit to a MODIFIED line indicated by HIT# and HITM# pins low. Pentium™ processor schedules the writing back of the modified line to memory.
E	I	S	Snoop hit indicated by HIT# pin low; no bus cycle generated.
S	I	S	Snoop hit indicated by HIT# pin low; no bus cycle generated.
I	I	I	Address not in cache; HIT# pin high.

3.7.7.2.   PENTIUM PROCESSOR CODE CACHE CONSISTENCY PROTOCOL

The Pentium processor code cache follows a subset of the MESI protocol. Accesses to the code cache are either a Hit (Shared) or a Miss (Invalid).

In the case of a read hit, the cycle is serviced internally to the Pentium processor and no bus activity is generated. In the case of a read miss, the read is sent to the external bus and may be converted to a line fill.

Lines are never overwritten in the code cache. Writes generated by the Pentium processor are snooped by the code cache. If the snoop is a hit in the code cache, the line is invalidated. If there is a miss, the code cache is not affected. Note that all code writes (D/C# = 0, W/R# = 1) are propagated to the system bus.

intel®

4

Microprocessor Initialization and Configuration

intel®

CHAPTER 4
MICROPROCESSOR INITIALIZATION AND CONFIGURATION

Before normal operation of the Pentium processor can begin, the Pentium processor must be initialized by driving the RESET pin active. The RESET pin forces the Pentium processor to begin execution in a known state. Several features are optionally invoked at the falling edge of RESET: Built in Self Test (BIST), Functional Redundancy Checking, and Tristate Test Mode.

In addition to the standard RESET pin, the Pentium processor has implemented an initialization pin (INIT) that allows the processor to begin execution in a known state without disrupting the contents of the internal caches or the floating point state.

This chapter describes the Pentium processor power up and initialization procedures as well as the test and configuration features enabled at the falling edge of RESET.

4.1. POWER UP SPECIFICATIONS

During power up, RESET must be asserted while VCC is approaching nominal operating voltage to prevent internal bus contention which could negatively affect the reliability of the processor.

In order for RESET to be recognized, the CLK input needs to be toggling. This mandates the following additional specification: During power up, CLK must be toggling while VCC is approaching nominal operating voltage (CLK does not need to meet its duty cycle, stability, and frequency specifications during this time).

RESET must remain asserted for 1 millisecond after VCC and CLK have reached their AC/DC specifications.

4.2. TEST AND CONFIGURATION FEATURES (BIST, FRC, TRISTATE TEST MODE)

The INIT, FLUSH#, and FRCMC# inputs are sampled when RESET transitions from high to low to determine if BIST will be run, or if tristate test mode or checker mode will be entered (respectively).

If RESET is driven synchronously, these signals must be at their valid level and meet setup and hold times on the clock before the falling edge of RESET. If RESET is asserted asynchronously, these signals must be at their valid level two clocks before and after RESET transitions from high to low.

4.2.1. Built in Self Test

Self test is initiated by driving the INIT pin high when RESET transitions from high to low.

No bus cycles are run by the Pentium processor during self test. The duration of self test is approximately 2^{19} clocks. Approximately 70% of the devices in the Pentium processor are tested by BIST.

The Pentium processor BIST consists of two parts: hardware self test and microcode self test.

During the hardware portion of BIST, the microcode ROM and all large PLAs are tested. All possible input combinations of the microcode ROM and PLAs are tested.

The constant ROMs, BTB, TLBs, and all caches are tested by the microcode portion of BIST. The array tests (caches, TLBs, and BTB) have two passes. On the first pass, data patterns are written to arrays, read back and checked for mismatches. The second pass writes the complement of the initial data pattern, reads it back, and checks for mismatches. The constant ROMs are tested by using the microcode to add various constants and check the result against a stored value.

Upon successful completion of BIST, the cumulative result of all tests are stored in the EAX register. If EAX contains 0h, then all checks passed; any non-zero result indicates a faulty unit. Note that if an internal parity error is detected during BIST, the processor will assert the IERR# pin and attempt to shutdown.

4.2.2. Tristate Test Mode

When the FLUSH# pin is sampled low when RESET transitions from high to low, the Pentium processor enters tristate test mode. The Pentium processor floats all of its output pins and bi-directional pins including pins which are never floated during normal operation (except TDO). Tristate test mode can be initiated in order to facilitate testing board interconnects. The Pentium processor remains in tristate test mode until the RESET pin is asserted again.

4.2.3. Functional Redundancy Checking

The functional redundancy checking master/checker configuration input is sampled when RESET is high to determine whether the Pentium processor is configured in master mode (FRCMC# high) or checker mode (FRCMC# low). The final master/checker configuration of the Pentium processor is determined the clock before the falling edge of RESET. When con-figured as a master, the Pentium processor drives its output pins as required by the bus protocol. When configured as a checker, the Pentium processor tristates all outputs (except IERR# and TDO) and samples the output pins (that would normally be driven in master mode). If the sampled value differs from the value computed internally, the Pentium processor asserts IERR# to indicate an error. Note that IERR# will not be asserted due to an FRC mismatch until two clocks after the ADS# of the first bus cycle (or in the third clock of the bus cycle).

4.3. INITIALIZATION WITH RESET, INIT AND BIST

Two pins, RESET and INIT, are used to reset the Pentium processor in different manners. A "cold" or "power on" RESET refers to the assertion of RESET while power is initially being applied to the Pentium processor. A "warm" RESET refers to the assertion of RESET or INIT while Vcc and CLK remain within specified operating limits.

Table 4-1 shows the effect of asserting RESET and/or INIT.

Table 4-1. Pentium™ Processor Reset Modes

RESET	INIT	BIST Run?	Effect on Code and Data Caches	Effect on FP Registers	Effect on BTB, TLBs and SDC
0	0	No	n/a	n/a	n/a
0	1	No	None	None	Invalidated
1	0	No	Invalidated	Initailized	Invalidated
1	1	Yes	Invalidated	Initialized	Invalidated

Toggling either the RESET pin or the INIT pin individually forces the Pentium processor to begin execution at address FFFFFFF0h. The internal instruction cache and data cache are invalidated when RESET is asserted (modified lines in the data cache are NOT written back). The instruction cache and data cache are not altered when the INIT pin is asserted without RESET. In both cases, the branch target buffer (BTB) and translation lookaside buffers (TLBs) are invalidated.

After RESET (with or without BIST) or INIT, the Pentium processor will start executing instructions at location FFFFFFF0H. When the first Intersegment Jump or Call instruction is executed, address lines A20-A31 will be driven low for CS-relative memory cycles and the Pentium processor will only execute instructions in the lower one Mbyte of physical memory. This allows the system designer to use a ROM at the top of physical memory to initialize the system.

RESET is internally hardwired and forces the Pentium processor to terminate all execution and bus cycle activity within 2 clocks. No instruction or bus activity will occur as long as RESET is active. INIT is implemented as an edge triggered interrupt and will be recognized when an instruction boundary is reached. As soon as the Pentium processor completes the INIT sequence, instruction execution and bus cycle activity will continue at address FFFFFFF0h even if the INIT pin is not deasserted.

At the conclusion of RESET (with or without self test) or INIT, the DX register will contain a component identifier. The upper byte will contain 05h and the lower byte will contain a stepping identifier.

Table 4-2 defines the processor state after RESET, INIT and RESET with BIST (built in self test).

Table 4-2. Register State after RESET, INIT and BIST
(Register States Are Given in Hexadecimal Format)

Storage element	RESET (no BIST)	RESET (BIST)	INIT
EAX	0	0 if pass	0
EDX	0500+stepping	0500+stepping	0500+stepping
ECX, EBX, ESP, EBP, ESI, EDI	0	0	0
EFLAGS	2	2	2
EIP	0FFF0	0FFF0	0FFF0
CS	selector = F000	selector = F000	selector = F000
	AR = P, R/W, A	AR = P, R/W, A	AR = P, R/W, A
	base = FFFF0000	base = FFFF0000	base = FFFF0000
	limit = FFFF	limit = FFFF	limit = FFFF
DS,ES,FE,GS,SS	selector = 0	selector = 0	selector = 0
	AR = P, R/W, A	AR = P, R/W, A	AR = P, R/W, A
	base = 0	base = 0	base = 0
	limit = FFFF	limit = FFFF	limit = FFFF
(I/G/L)DTR, TSS	selector = 0	selector = 0	selector = 0
	base = 0	base = 0	base = 0
	AR = P, R/W	AR = P, R/W	AR = P, R/W
	limit = FFFF	limit = FFFF	limit = FFFF
CR0	60000010	60000010	Note 1
CR2,3,4	0	0	0
DR3-0	0	0	0
DR6	FFFF0FF0	FFFF0FF0	FFFF0FF0
DR7	00000400	00000400	00000400
Time Stamp Counter	0	0	UNCHANGED

Table 4-2. Register State after RESET, INIT and BIST
(Register States Are Given in Hexadecimal Format) (Contd.)

Storage element	RESET (no BIST)	RESET (BIST)	INIT
Control and Event Select	0	0	UNCHANGED
TR12	0	0	UNCHANGED
all other MSR's	undefined	undefined	UNCHANGED
CW	0040	0040	UNCHANGED
SW	0	0	UNCHANGED
TW	5555	5555	UNCHANGED
FIP,FEA,FCS, FDS,FOP	0	0	UNCHANGED
FSTACK	0	0	UNCHANGED
Data & Code Cache	invalid	invalid	UNCHANGED
Code Cache TLB, Data Cache TLB, BTB, SDC	invalid	invalid	invalid

NOTE

1. CD and NW are unchanged, bit 4 is set to 1, all other bits are cleared.

4.3.1. Recognition Of Interrupts After Reset

In order to guarantee recognition of the edge sensitive interrupts (FLUSH#, NMI, R/S#, SMI#) after RESET or after RESET with BIST, the interrupt input must not be asserted until four clocks after RESET is deasserted, regardless of whether BIST is run or not.
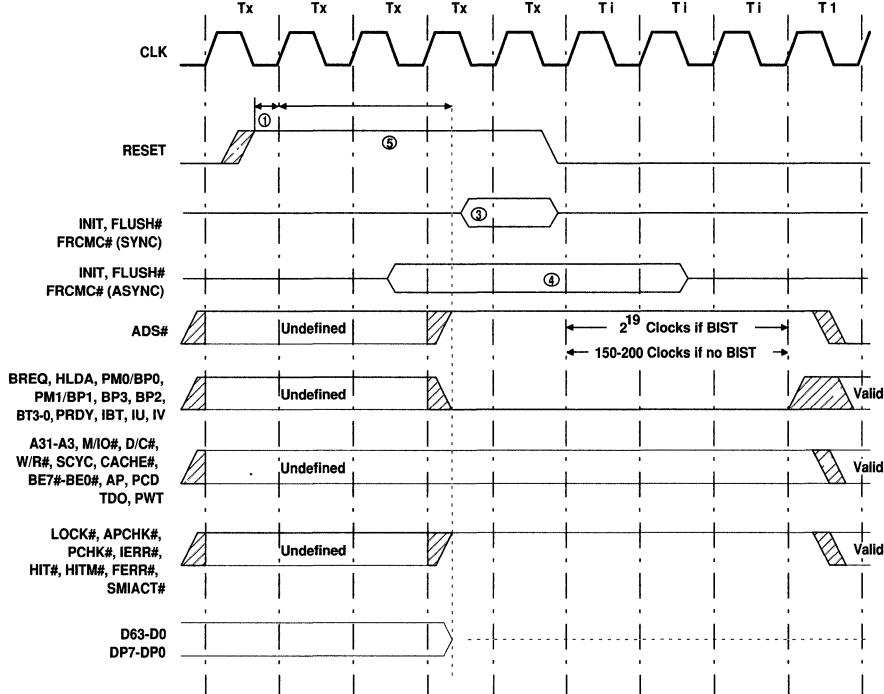
4.3.2. Pin State During/After RESET

The Pentium processor recognizes and will respond to HOLD, AHOLD and BOFF# during RESET. Figure 4-1 shows the processor state during and after a power on RESET if HOLD, AHOLD, and BOFF# are inactive. Note that the address bus (A31-A3, BE7#-BE0#) and cycle definition pins (M/IO#, D/C#, W/R#, CACHE#, SCYC, and LOCK#) are undefined from the time RESET is asserted until the start of the first bus cycle.

The following lists the state of the output pins after RESET assuming HOLD, AHOLD and BOFF# are inactive, boundary scan is not invoked, and no internal parity error is detected.

High: LOCK#, ADS#, APCHK#, PCHK#, IERR#, HIT#, HITM#, FERR#, SMIACT#

Low: HLDA, BREQ, PM0/BP0, PM1/BP1, BP3, BP2, PRDY, IBT, IU, IV, BT3-BT0

High Impedance: D63-D0, DP7-DP0

Undefined: A31-A3, AP, BE7#-BE0#, W/R#, M/IO#, D/C#, PCD, PWT, CACHE#, TDO, SCYC

1 RESET must meet setup and hold times to guarantee recognition on a specific clock edge.
   If RESET does not need to be recognized on a specific clock edge it may be asserted asynchronously.

2 At power up, RESET needs to be asserted for 1 ms after Vcc and CLK have reached
   their AC/DC specifications. For warm reset, RESET needs to be asserted for at least 15 clocks
   while Vcc and CLK remain within specified operating limits.

3 If RESET is driven synchronously, FLUSH#, FRCMC#, and INIT must be at their valid level
   and meet setup and hold times the clock before the falling edge of RESET.

4 If RESET is driven asynchronously, FLUSH#, FRCMC#, and INIT must be at their valid level
   2 clocks before and after the falling edge of RESET.

5 An assertion of RESET takes at least 2 clocks to affect the pins.

PDB26

Figure 4-1. Pin States During RESET

intel®

5

Hardware Interface

intel®

CHAPTER 5
HARDWARE INTERFACE

The Pentium processor bus is similar to the Intel486 microprocessor bus, but it has distinct differences and improvements. Several new features have been added to increase performance, to support writeback cacheing, and add functionality.

The data bus on the Pentium processor has been increased to 64-bits, allowing the larger cache line to be filled with the standard four data transfers. Burst read cycles were carried forward from the Intel486 microprocessor and writeback cycles are bursted on the Pentium processor. The Pentium processor implements bus cycle pipelining which allows two bus cycles to be outstanding on the bus simultaneously.

An initialization pin, INIT was added (in addition to the RESET pin) to provide a method to switch from protected to real mode while maintaining the contents of the caches and floating point state.

The data parity feature implemented by the Intel486 microprocessor has been extended to support the entire 64-bit data bus, and address parity and internal parity features have been added to the Pentium processor. In addition, support for functional redundancy checking and the machine check exception were also added to the Pentium processor.

The test access port (TAP) for IEEE Standard 1149.1 Boundary Scan is implemented on the Pentium processor, along with a new mode that also uses the TAP, probe mode.

System management mode, similar to that on the Intel386 SL microprocessor, is implemented on the Pentium processor. A method to track instruction execution through each of the pipelines has also been implemented.

This chapter describes the pins that interface to the system that allow these features to be implemented at the system level. The pin descriptions are arranged alphabetically for ease of reference. The pins are grouped functionally as defined in Table 1-6.

5.1. DETAILED PIN DESCRIPTIONS

Each pin name has a brief descriptive heading organized as follows:

Pin Symbol	Pin Name
	Function
	Input/Output

Each heading is followed by three sections that describe the signal function, when the signal is driven or sampled, and the relation that signal has to other signals.

The # symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage. When a # symbol is not present after the signal name, the signal is active, or asserted at the high voltage level.

5.1.1. A20M#

A20M#	Address 20 Mask
	Used to emulate the 1 Mbyte address wraparound on the 8086
	Asynchronous Input

Signal Description

When the address 20 mask input is asserted, the Pentium processor masks physical address bit 20 (A20) before performing a lookup to the internal caches or driving a memory cycle on the bus. A20M# is provided to emulate the address wraparound at one Mbyte which occurs on the 8086.

A20M# must only be asserted when the processor is in real mode. **The effect of asserting A20M# in protected mode is undefined** and may be implemented differently in future processors.

Inquire cycles and writebacks caused by inquire cycles are not affected by this input. Address bit A20 is not masked when an external address is driven into the Pentium processor for an inquire cycle. Note that if an OUT instruction is used to modify A20M# this will not affect previously prefetched instructions. A serializing instruction must be executed to guarantee recognition of A20M# before a specific instruction.

When Sampled

A20M# is sampled on every rising clock edge. A20M# is level sensitive and active low. This pin is asynchronous, but must meet setup and hold times for recognition in any specific clock. To guarantee that A20M# will be recognized before the first ADS# after RESET, A20M# must be asserted within two clocks after the falling edge of RESET.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A20	When asserted, A20M# will mask the value of address pin A20.

5.1.2.   A31-A3

A31-A3	Address Lines
	Defines the physical area of memory or I/O accessed.
	Input/Output

Signal Description

As outputs, the address lines (A31-A3) along with the byte enable signals (BE7#-BE0#) form the address bus and define the physical area of memory or I/O accessed.

The Pentium processor is capable of addressing 4-gigabytes of physical memory space and 64K bytes of I/O address space.

As inputs, the address bus lines A31-A5 are used to drive addresses back into the processor to perform inquire cycles. Since inquire cycles affect an entire 32-byte line, the logic values of A4 and A3 are not used for the hit/miss decision, however A4 and A3 must be at valid logic level and meet setup and hold times during inquire cycles.

When Sampled/Driven

When an output, the address is driven in the same clock as ADS#. The address remains valid from the clock in which ADS# is asserted until AHOLD is asserted or the clock after the earlier of NA# or the last BRDY#.

When an input, the address must be returned to the processor to meet setup and hold times in the clock EADS# is sampled asserted.

int_el ®

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A20M#	Causes address pin A20 to be masked
ADS#	A31-A3 are driven with ADS# (except when a external inquire cycle causes a writeback before AHOLD is deasserted, see the Bus Functional Description chapter)
AHOLD	A31-A3 are floated one clock after AHOLD is asserted.
AP	Even address parity is driven/sampled with the address bus on AP.
APCHK#	The status of the address parity check is driven on the APCHK# pin.
BE7#-BE0#	Completes the definition of the physical area of memory or I/O accessed.
BOFF#	A31-A3 are floated one clock after BOFF# is asserted.
EADS#	A31-A5 are sampled with EADS# during inquire cycles.
HIT#	HIT# is driven to indicate whether the inquire address driven on A31-A5 is valid in an internal cache.
HITM#	HITM# is driven to indicate whether the inquire address driven on A31-A5 is in the modified state in the data cache.
HLDA	A31-A3 are floated when HLDA is asserted.
INV	INV determines if the inquire address driven to the processor on A31-A5 should be invalidated or marked as shared if it is valid in an internal cache.

5.1.3. ADS#

ADS#	Address Strobe
	Indicates that a new valid bus cycle is currently being driven by the Pentium processor.
	Output

Signal Description

The address status output indicates that a new valid bus cycle is currently being driven by the Pentium processor. The following pins are driven to their valid level in the clock ADS# is asserted: A31-A3, AP, BE7#-0#, CACHE#, LOCK#, M/IO#, W/R#, D/C#, SCYC, PWT, PCD.

ADS# is used by external bus circuitry as the indication that the processor has started a bus cycle. The external system may sample the bus cycle definition pins on the next rising edge of the clock after ADS# is driven active.

ADS# floats during bus HOLD and BOFF#. ADS# is not driven low to begin a bus cycle while AHOLD is asserted unless the cycle is a writeback due to an external invalidation. An active (floating low) ADS# in the clock after BOFF# is asserted should be ignored by the system.

When Driven

ADS# is driven active in the first clock of a bus cycle and is driven inactive in the second and subsequent clocks of the cycle. ADS# is driven inactive when the bus is idle.

Relation to Other Signals

Pin Symbol	Relation to Other Symbols
A31-A3 AP BE7#-BE3# CACHE# D/C# LOCK# M/IO# PCD PWT SCYC W/R#	These signals are driven valid in the clock in which ADS# is asserted
AHOLD	ADS# will not be driven if AHOLD is asserted (except when a external inquire cycle causes a writeback before AHOLD is deasserted, see the Bus Functional Description chapter)
BOFF#	ADS# is floated one clock after BOFF# is asserted
BREQ	BREQ is always asserted in the clock that ADS# is driven
BT3-BT0	BT3-BT0 are driven to their valid level with the ADS# of a branch trace message special cycle.
FLUSH#	The flush special cycle is driven as a result of the assertion of FLUSH#.
HLDA	ADS# is floated when HLDA is asserted
IBT	The branch trace message special cycle is driven after an assertion of IBT if TR12.TR is set to 1.
INTR	The interrupt acknowledge cycle is driven as a result of the assertion of INTR.
NA#	If NA# is sampled asserted and an internal bus request is pending, the Pentium processor drives out the next bus cycle and asserts ADS#.

5.1.4. AHOLD

AHOLD	Address Hold
	Floats the address bus so an inquire cycle can be driven to the Pentium processor.
	Synchronous Input

Signal Description

In response to the address hold request input the Pentium processor will stop driving A31-A3, BT3-BT0, and AP in the next clock. This pin is intended to be used for running inquire cycles to the Pentium processor. AHOLD allows another bus master to drive the Pentium processor address bus with the address for an inquire cycle. Since inquire cycles affect the entire cache line, although A31-A3 are floated during AHOLD, only A31-A5 are used by the Pentium processor for inquire cycles (and parity checking). Address pins 3 and 4 are logically ignored during inquire cycles but must be at a valid logic level when sampled.

While AHOLD is active, the address bus will be floated, but the remainder of the bus can remain active. For example, data can be returned for a previously driven bus cycle when AHOLD is active. In general, the Pentium processor will not issue a bus cycle (ADS#) while AHOLD is active, the *only* exception to this is writeback cycles due to an external snoop will be driven while AHOLD is asserted.

Since the Pentium processor floats its bus immediately (in the next clock) in response to AHOLD, an address hold acknowledge is not required.

When AHOLD is deasserted, the Pentium processor will drive the address bus in the next clock. It is the responsibility of the system designer to prevent address bus contention. This can be accomplished by ensuring that other bus masters have stopped driving the address bus before AHOLD is deasserted. Note the restrictions to the deassertion of AHOLD discussed in the inquire cycle section of the Bus Functional Description chapter.

AHOLD is recognized during RESET and INIT. Note that the internal caches are flushed as a result of RESET, so invalidation cycles run during RESET are unnecessary.

When Sampled

AHOLD is sampled on every rising clock edge, including during RESET and INIT.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A31-A3	A31-A3 are floated as a result of the assertion of AHOLD
ADS#	ADS# will not be driven if AHOLD is asserted (except when a external inquire cycle causes a writeback before AHOLD is deasserted, see the Bus Functional Description chapter)
AP	AP is floated as a result of the assertion of AHOLD
BT3-BT0	The branch trace outputs are floated as a result of the assertion of AHOLD
EADS#	EADS# is recognized while AHOLD is asserted.

5.1.5. AP

AP	Address Parity
	Bi-directional address parity pin for the address lines of processor.
	Input/Output

Signal Description

This is the bi-directional address parity pin for the address lines of processor. There is one address parity pin for the address lines A31-A5. Note A4 and A3 are not included in the parity determination.

When an output, AP is driven by the Pentium processor with even parity information on all Pentium processor generated cycles in the same clock as the address driven. (Even address parity means that there are an even number of HIGH outputs on A31-A5 and the AP pins.)

When an input, even parity information must be returned to the Pentium processor on this pin during inquire cycles in the same clock that EADS# is sampled asserted to insure that the correct parity check status is driven on the APCHK# output.

The value read on the AP pin does not affect program execution. The value returned on the AP pin is used only to determine even parity and drive the APCHK# output with the proper value. It is the responsibility of the system to take appropriate actions if a parity error occurs. If parity checks are not implemented in the system, AP may be connected to VCC through a pullup resisitor and the APCHK# pin may be ignored.

When Sampled/Driven

When an output, AP is driven by the Pentium processor with even parity information on all Pentium processor generated cycles in the same clock as the address driven. The AP output remains valid from the clock in which ADS# is asserted until AHOLD is asserted or the clock after the earlier of NA# or the last BRDY#.

When an input, even parity information must be returned to the Pentium processor on this pin during inquire cycles in the same clock that EADS# is sampled asserted to guarantee that the proper value is driven on APCHK#. The AP input must be at a valid level and meet setup and hold times when sampled.

intel®

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A31-A5	The AP pin is used to create even parity with the A31-A5 pins.
ADS#	AP is driven with ADS# (except when a external inquire cycle causes a write-back before AHOLD is deasserted, see the Bus Functional Description chapter).
AHOLD	AP is floated one clock after AHOLD is asserted.
APCHK#	The status of the address parity check is driven on the APCHK# output.
BOFF#	AP is floated one clock after BOFF# is asserted.
EADS#	AP is sampled with EADS# during inquire cycles.
HLDA	AP is floated when HLDA is asserted.

5.1.6. APCHK#

APCHK#	Address Parity Check
	The status of the address parity check is driven on the APCHK# output.
	Output

Signal Description

APCHK# is asserted two clocks after EADS# is sampled active if the Pentium processor has detected a parity error on the A31-A5 during inquire cycles.

Driving APCHK# is the only effect that bad address parity has on the Pentium processor. It is the responsibility of the system to take appropriate action if a parity error occurs. If parity checks are not implemented in the system, the APCHK# pin may be ignored.

When Driven

APCHK# is valid for one clock two clocks after EADS# is sampled asserted. APCHK# will remain active for one clock each time a parity error is detected. At all other times it is inactive (HIGH). APCHK# is not floated with AHOLD, HOLD, or BOFF#. The APCHK# signal is glitch free.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
AP	Even address parity with the A31-A5 should be returned to the Pentium processor on the AP pin. If even parity is not driven, the APCHK# pin is asserted.
A31-A5	The AP pin is used to create even parity with A31-A5. If even parity is not driven to the Pentium processor, the APCHK# pin is asserted.
EADS#	APCHK# is driven to its valid level two clocks after EADS# is sampled asserted.

intel®

5.1.7. BE7#-BE0#

BE7#-BE0#	Byte Enables
	Helps define the physical area of memory or I/O accessed.
	Output

Signal Description

The byte enable outputs are used in conjunction with the address lines to provide physical memory and I/O port addresses. The byte enables are used to determine which bytes of data must be written to external memory, or which bytes were requested by the CPU for the current cycle.

- BE7# applies to D63-D56
- BE6# applies to D55-D48
- BE5# applies to D47-D40
- BE4# applies to D39-D32
- BE3# applies to D31-D24
- BE2# applies to D23-D16
- BE1# applies to D15-D8
- BE0# applies to D7-D0

In the case of cacheable reads (line fill cycles), all 8 bytes of data must be driven to the Pentium processor regardless of the state of the byte enables. If the requested read cycle is a single transfer cycle, valid data must be returned on the data lines corresponding to the active byte enables. Data lines corresponding to inactive byte enables need not be driven with valid logic levels. Even data parity is checked and driven only on the data bytes that are enabled by the byte enables.

When Driven

The byte enables are driven in the same clock as ADS#. The byte enables are driven with the same timing as the address (A31-3). The byte enables remain valid from the clock in which ADS# is asserted until the clock after the earlier of NA# or the last BRDY#. The byte enables do not float with AHOLD.

I

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A31-A3	A31-3 and BE7#-BE0# together define the physical area of memory or I/O accessed.
ADS#	BE7#-BE0# are driven with ADS#.
BOFF#	BE7#-BE0# are floated one clock after BOFF# is asserted.
D63-D0	BE7#-BE0# indicate which data bytes are being requested or driven by the Pentium processor.
DP7-DP0	Even data parity is checked/driven only on the data bytes enabled by BE7#-BE0#
HLDA	BE7#-BE0# are floated when HLDA is asserted.

5.1.8. BOFF#

BOFF#	Backoff
	The back off input is used to force the Pentium processor off the bus in the next clock.
	Synchronous Input

Signal Description

In response to BOFF#, the Pentium processor will abort all outstanding bus cycles that have not yet completed and float the Pentium processor bus in the next clock. The processor floats all pins normally floated during bus hold. Note that since the bus is floated in the clock after BOFF# is asserted, an acknowledge is not necessary (HLDA is not asserted in response to BOFF#).

The processor remains in bus hold until BOFF# is negated, at which time the Pentium processor restarts any aborted bus cycle(s) in their entirety by driving out the address and status and asserting ADS#.

This pin can be used to resolve a deadlock situation between two bus masters.

Any data with BRDY# returned to the processor while BOFF# is asserted is ignored.

BOFF# has higher priority than BRDY#. If both BOFF# and BRDY# occur in the same clock, BOFF# takes effect.

BOFF# also has precedence over BUSCHK#. If BOFF# and BUSCHK# are both asserted during a bus cycle, BOFF# causes the BUSCHK# to be forgotten.

When Sampled

BOFF# is sampled on every rising clock edge, including when RESET and INIT are asserted.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A3-A31 ADS# AP BE7#-BE3# CACHE# D/C# D63-D0 DP7-DP0 LOCK# M/IO# PCD PWT SCYC W/R#	These signals float in response to BOFF#
BRDY#	If BRDY# and BOFF# are asserted simultaneously, BOFF# takes priority and BRDY# is ignored.
EADS#	EADS# is recognized when BOFF# is asserted.
HLDA	The same pins are floated when HLDA or BOFF# is asserted.
BUSCHK#	If BUSCHK# and BOFF# are both asserted during a bus cycle, BOFF# takes priority and BUSCHK# is forgotten.
NA#	If NA# and BOFF# are asserted simultaneously, BOFF# takes priority and NA# is ignored.

intel®

5.1.9.   BP[3:2], PM/BP[1:0]

BP3-0	Breakpoint and Performance Monitoring
PM1-0	BP3-BP0 externally indicate a breakpoint match.
	Output

Signal Description

The breakpoint pins (BP3-BP0) correspond to the debug registers DR3-DR0. These pins externally indicate a breakpoint match of the corresponding debug register when the debug registers are programmed to test for breakpoint matches.

BP1 and BP0 are multiplexed with the Performance Monitoring pins (PM1 and PM0). The PB1 and PB0 bits in the Debug Mode Control Register determine if the pins are configured as breakpoint or performance monitoring pins. The pins come out of reset configured for performance monitoring (for more information see Appendix A).

When Driven

The BP[3:2], PM/BP[1:0] pins are driven in every clock and are not floated during bus HOLD, or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
PM1-PM0	BP1 and BP0 share pins with PM1 and PM0

5.1.10. BRDY#

BRDY#	Burst Ready
	Transfer complete indication.
	Synchronous Input

Signal Description

The burst ready input indicates that the external system has presented valid data on the data pins in response to a read, or that the external system has accepted the Pentium processor data in response to a write request.

Each cycle generated by the Pentium processor will either be a single transfer read or write, or a burst cache line fill or writeback. For single data transfer cycles, one BRDY# is expected to be returned to the Pentium processor. Once this BRDY# is returned, the cycle is complete. For burst transfers, four data transfers are expected by the Pentium processor. The cycle is ended when the fourth BRDY# is returned.

When Sampled

This signal is sampled in the T2, T12 and T2P bus states.

intel®

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BOFF#	If BOFF# and BRDY# are asserted simultaneously, BOFF# takes priority and BRDY# is ignored.
BUSCHK#	BUSCHK# is sampled with BRDY#.
CACHE#	In conjunction with the KEN# input, CACHE# determines whether the bus cycle will consist of 1 or 4 transfers.
D63-D0	During reads, the D63-D0 pins are sampled by the Pentium processor with BRDY#.
	During writes, BRDY# indicates that the system has accepted D63-D0.
DP7-0	During reads, the DP7-0 pins are sampled by the Pentium processor with BRDY#.
	During writes, BRDY# indicates that the system has accepted DP7-0.
EWBE#	EWBE# is sampled with each BRDY# of a write cycle.
KEN#	KEN# is sampled & latched by the Pentium processor with the earlier of the first BRDY# or NA#. Also, in conjunction with the CACHE# input, KEN# determines whether the bus cycle will consist of 1 or 4 transfers (assertions of BRDY#).
LOCK#	LOCK# is deasserted after the last BRDY# of the locked sequence.
PCHK#	PCHK# indicates the results of the parity check two clocks after BRDY# is returned for reads.
PEN#	PEN# is sampled with BRDY# for read cycles.
WB/WT#	WB/WT# is sampled & latched by the Pentium processor with the earlier of the first BRDY# or NA#.

I

5.1.11. BREQ

BREQ	Bus Request
	Indicates externally when a bus cycle is pending internally.
	Output

Signal Description

The Pentium processor asserts the BREQ output whenever a bus cycle is pending internally. BREQ is always asserted in the first clock of a bus cycle with ADS#. Furthermore, if the Pentium processor is not currently driving the bus (due to AHOLD, HOLD, or BOFF#), BREQ is asserted in the same clock that ADS# would have been asserted if the Pentium processor were driving the bus. After the first clock of the bus cycle, BREQ may change state. Every assertion of BREQ is not guaranteed to have a corresponding assertion of ADS#.

External logic can use the BREQ signal to arbitrate between multiple processors. This signal is always driven regardless of the state of AHOLD, HOLD or BOFF#.

When Driven

BREQ is always driven by the Pentium processor, and is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	BREQ is always asserted in the clock that ADS# is asserted.

5.1.12. BT3-BT0

BT3-BT0	Branch Trace
	Provide bits 0-2 of the branch target linear address and the default operand size during a Branch Trace Message Special Cycle.
	Output

Signal Description

The Branch Trace pins provide bits 2-0 of the branch target linear address and the default operand size during a Branch Trace Message Special Cycle.

BT0:   Address bit A0 of the branch target linear address
BT1:   Address bit A1 of the branch target linear address
BT2:   Address bit A2 of the branch target linear address
BT3:   Driven high if the default operand size of the current instruction is 32-bits
          Driven low if the default operand size of the current instruction is 16-bits

The Branch Trace Message Special Cycle is part of the Pentium processor execution tracing protocol. If the execution tracing enable bit (bit 1) in TR12 is set to 1, a branch trace message special cycle will be driven each time IBT is asserted, i.e. whenever a branch is taken.

When Driven

The BT3-BT0 outputs are driven to their valid level with the ADS# of a branch trace message special cycle. These outputs remain valid until AHOLD is asserted or the clock after the earlier of NA# or the last BRDY#. At all other times these outputs are undefined.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	BT3-BT0 are driven to their valid level with the ADS# of a branch trace message special cycle.
AHOLD	BT3-BT0 are floated one clock after AHOLD is asserted
IBT	If TR12.TR is set to 1, BT3-BT0 are driven along with the branch trace message special cycle for each assertion of IBT.

5.1.13. BUSCHK#

BUSCHK#	Bus Check
	Allows the system to signal an unsuccessful completion of a bus cycle.
	Synchronous Input
	Internal Pullup Resistor

Signal Description

The bus check input pin allows the system to signal an unsuccessful completion of a bus cycle. If this pin is sampled active, the Pentium processor will latch the address and control signals of the failing cycle in the machine check registers. If in addition, the MCE bit in CR4 is set, the Pentium processor will vector to the machine check exception upon completion of the current instruction.

If BUSCHK# is asserted in the middle of a cycle, the system must return all expected BRDY#s to the Pentium processor. BUSCHK# is remembered by the processor if asserted during a bus cycle. The processor decides after the last BRDY# whether to take the machine check exception or not.

BOFF# has precedence over BUSCHK#. If BOFF# and BUSCHK# are both asserted during a bus cycle, the BOFF# causes the BUSCHK# to be forgotten.

When Sampled

BUSCHK# is sampled when BRDY# is returned to the Pentium processor.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BOFF#	If BOFF# and BUSCHK# are both asserted during a bus cycle, the BOFF# signal causes the BUSCHK# to be forgotten.
BRDY#	BUSCHK# is sampled with BRDY#

intel®

5.1.14. CACHE#

CACHE#	Cacheability
	External indication of internal cacheability.
	Output

Signal Description

The cacheability output is a cycle definition pin. For Pentium processor initiated cycles this pin indicates internal cacheability of the cycle (if a read), and indicates a burst writeback (if a write). CACHE# is asserted for cycles coming from the cache (writebacks) and for cycles that will go into the cache if KEN# is asserted (linefills). More specifically, CACHE# is asserted for cacheable reads, cacheable code fetches, and writebacks. It is driven inactive for non-cacheable reads, TLB replacements, locked cycles (except writeback cycles from an external snoop that interrupt a locked read/modify/write sequence), I/O cycles, special cycles and writethroughs.

For read cycles, the CACHE# pin indicates whether the Pentium processor will allow the cycle to be cached. If CACHE# is asserted for a read cycle, the cycle will be turned into a cache line fill if KEN# is returned active to the Pentium processor. If this pin is driven inactive during a read cycle, Pentium processor will not cache the returned data, regardless of the state of the KEN#.

If this pin is asserted for a write cycle, it indicates that the cycle is a burst writeback cycle. Writethroughs cause a non-burst write cycle to be driven to the bus. The Pentium processor does not support write allocations (cache line fills as a result of a write miss).

When Driven

CACHE# is driven to its valid level in the same clock as the assertion of ADS# and remains valid until the clock after the earlier of NA# or the last BRDY#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	CACHE# is driven to its valid level with ADS#.
BOFF#	CACHE# floats one clock after BOFF# is asserted.
BRDY#	In conjunction with the KEN# input, CACHE# determines whether the bus cycle will consist of 1 or 4 transfers (assertions of BRDY#).
HLDA	CACHE# floats when HLDA is asserted.
KEN#	KEN# and CACHE# are used together to determine if a read will be turned into a linefill

5.1.15. CLK

CLK	Clock
	Fundamental Timing for the Pentium processor.
	Input

Signal Description

The clock input provides the fundamental timing for the Pentium processor. Its frequency is the internal operating frequency of the Pentium processor and requires TTL levels. All external timing parameters except TDI, TDO, TMS, and TRST# are specified with respect to the rising edge of CLK.

When Sampled

CLK is a clock signal and is used as a reference for sampling other signals. During power up, CLK must toggle (but not necessarily meet specifications) while VCC is approaching nominal operating voltage. VCC specifications and clock duty cycle, stability and frequency specifications must be met for 1 millisecond before the negation of RESET. If at any time during normal operation one of these specifications is violated, the power on RESET sequence must be repeated. This requirement is to insure proper operation of the phase locked loop circuitry on the clock input.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
All except TCK, TDI, TDO, TMS, TRST#	External timing parameters are measured from the rising edge of CLK for all signals except TDI, TDO, TMS, TCK, and TRST#.

5.1.16. D/C#

D/C#	Data/Code
	Distinguishes a data access from a code access.
	Output

Signal Description

The Data/Code signal is one of the primary bus cycle definition pins. D/C# distinguishes between data (D/C# = 1) and code/special cycles (D/C# = 0).

When Driven

The D/C# pin is driven valid in the same clock as ADS# and the cycle address. It remains valid from the clock in which ADS# is asserted until the clock after the earlier of NA# or the last BRDY#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	D/C# is driven to valid state with ADS#.
BOFF#	D/C# floats one clock after BOFF# is asserted.
HLDA	D/C# floats when HLDA is asserted.

5.1.17. D63-D0

D63-D0	Data Lines
	Forms the 64-bit data bus.
.	Input/Output

Signal Description

The bi-directional lines, D63-D0 form the 64 data bus lines for the Pentium processor. Lines D7-D0 define the least significant byte of the data bus; lines D63-D56 define the most significant byte of the data bus.

When Sampled/Driven

When the CPU is driving the data lines (during writes), they are driven during the T2, T12, or T2P clocks for that cycle.

During reads, the CPU samples the data bus when BRDY# is returned.

D63-D0 are floated during T1, TD, and Ti states.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BE7#-BE0#	BE7#-BE0# indicate which data bytes are being requested or driven by the Pentium processor.
BOFF#	D63-D0 float one clock after BOFF# is asserted.
BRDY#	BRDY# indicates that the data bus transfer is complete.
DP7-DP0	Even data parity is driven/sampled with the data bus on DP7-DP0.
HLDA	D63-D0 float when HLDA is asserted.
PCHK#	The status of the data bus parity check is driven on PCHK#.
PEN#	Even data parity with D63-D0 should be returned on to the Pentium processor on the DP pin. If a data parity error occurs, and PEN# is enabled, the cycle will be latched and a machine check exception will be taken if CR4.MCE = 1

5.1.18. DP7-DP0

DP7-DP0	Data Parity
	Bi-directional data parity pins for the data bus.
	Input/Output

Signal Description

These are the bi-directional data parity pins for the processor. There is one parity pin for each byte of the data bus. DP7 applies to D63-D56, DP0 applies to D7-D0.

As outputs, the data parity pins are driven by the Pentium processor with even parity information for writes in the same clock as write data. Even parity means that there are an even number of HIGH logic values on the eight corresponding data bus pins and the parity pin.

As inputs, even parity information must be driven back to the Pentium processor on these pins in the same clock as the data to ensure that the correct parity check status is indicated by the Pentium processor.

The value read on the data parity pins does not affect program execution unless PEN# is also asserted. If PEN# is not asserted, the value returned on the DP pins is used only to determine even parity and drive the PCHK# output with the proper value. If PEN# is asserted when a parity error occurs the cycle address and type will be latched in the MCA and MCT registers. If in addition, the MCE bit in CR4 is set, a machine check exception will be taken.

It is the responsibility of the system to take appropriate actions if a parity error occurs. If parity checks are not implemented in the system, the DP/PEN# pins should be tied to Vcc through a pullup resistor and the PCHK# pin may be ignored.

When Sampled/Driven

As outputs, the data parity pins are driven by the Pentium processor with even parity information in the same clock as write data. The parity remains valid until sampled by the assertion of BRDY# by the system.

As inputs, even parity information must be driven back to the Pentium processor on these pins in the same clock as the data to ensure that the correct parity check status is indicated by the Pentium processor. The data parity pins must be at a valid logic level and meet setup and hold times when sampled.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BE7#-BE0#	Even data parity is checked/driven only on the data bytes enabled by BE7#-BE0#.
BOFF#	DP7-DP0 are floated one clock after BOFF# is asserted.
BRDY#	DP7-DP0 are sampled with BRDY# for reads.
D63-D0	The DP7-0 pins are used to create even parity with D63-D0 on a byte by byte basis. DP7-DP0 are driven with D63-D0 for writes.
HLDA	DP7-DP0 are floated when HLDA is asserted.
PCHK#	The status of the data parity check is driven on the PCHK# output.
PEN#	The DP7-DP0 pins are used to create even parity with D63-D0. If even parity is not detected, and PEN# is enabled, the cycle address and type will be latched. If in addition CR4.MCE = 1, the machine check exception will be taken .

5.1.19. EADS#

EADS#	External Address Strobe
	Signals the Pentium processor to run an inquire cycle with the address on the bus.
	Synchronous Input

Signal Description

The EADS# input indicates that a valid external address has been driven onto the Pentium processor address pins to be used for an inquire cycle. The address driven to the Pentium processor when EADS# is sampled asserted will be checked with the current cache contents. The HIT# and HITM# signals will be driven to indicate the result of the comparison. If the INV pin is returned active (high) to the Pentium processor in the same clock as EADS# is sampled asserted, an inquire hit will result in that line being invalidated. If the INV pin is returned inactive (low), an inquire hit will result in that line being marked Shared (S).

When Sampled

EADS# is recognized two clocks after an assertion of AHOLD or BOFF#, or one clock after an assertion of HLDA. In addition, the Pentium processor will ignore an assertion of EADS# if the processor is driving the address bus, or if HITM# is active, or in the clock after ADS# or EADS# is asserted.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A31-A5	The inquire cycle address must be valid on A31-A5 when EADS# is sampled asserted.
A4-A3	These signals must be at a valid logic level when EADS# is sampled asserted.
AHOLD	EADS# is recognized while AHOLD is asserted.
AP	AP is sampled when EADS# is sampled asserted.
APCHK#	APCHK# is driven to its valid level two clocks after EADS# is sampled asserted.
BOFF#	EADS# is recognized while BOFF# is asserted.
HIT#	HIT# is driven to its valid level two clocks after EADS# is sampled asserted.
HITM#	HITM# is driven to its valid level two clocks after EADS# is sampled asserted.
HLDA	EADS# is recognized while HLDA is asserted.
INV	INV is sampled with EADS# to determine the final state of the cache line in the case of an inquire hit.

5.1.20. EWBE#

EWBE#	External Write Buffer Empty
	Provides the option of strong write ordering to the memory system.
	Synchronous Input

Signal Description

The external write buffer empty input, when inactive (high), indicates that a write through cycle is pending in the external system. When the Pentium processor generates a write (memory or I/O), and EWBE# is sampled inactive, the Pentium processor will hold off all subsequent writes to all E or M-state lines until all write through cycles have completed, as indicated by EWBE# being active. In addition, if the Pentium processor has a write pending in a write buffer, the Pentium processor will also hold off all subsequent writes to E or M-state lines. This insures that writes are visible from outside the Pentium processor in the same order as they were generated by software.

When the Pentium processor serializes instruction execution through the use of a serializing instruction, it waits for the EWBE# pin to go active before fetching and executing the next instruction.

After the OUT or OUTS instructions are executed, the Pentium processor ensures that EWBE# has been sampled active before beginning to execute the next instruction. Note that the instruction may be prefetched if EWBE# is not active, but it will not execute until EWBE# is sampled active.

When Sampled

EWBE# is sampled with each BRDY# of a write cycle. If sampled deasserted, the Pentium processor repeatedly samples EWBE# in each clock until it is asserted. Once sampled asserted, the Pentium processor ignores EWBE# until the next BRDY# of a write cycle.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BRDY#	EWBE# is sampled with each BRDY# of a write cycle.
SMIACT#	SMIACT# is not asserted until EWBE# is asserted.

5.1.21. FERR#

FERR#	Floating Point Error
	The floating point error output is driven active when an unmasked floating point error occurs.
	Output

Signal Description

The floating point error output is driven active when an unmasked floating point error occurs. FERR# is similar to the ERROR# pin on the Intel387™ math coprocessor. FERR# is included for compatibility with systems using DOS type floating point error reporting.

In some cases, FERR# is asserted when the next floating point instruction is encountered and in other cases it is asserted before the next floating point instruction is encountered depending upon the execution state of the instruction causing the exception.

The following class of floating point exceptions drive FERR# at the time the exception occurs (i.e. before encountering the next floating point instruction):

1. Stack fault, all invalid operation exceptions and and denormal exceptions on: all-transcendental instructions, FSCALE, FXTRACT, FPREM, FPREM(1), FBLD, FLD_extended, FRNDINT, and stack fault and invalid operation exceptions on Floating Point arithmetic instructions with an integer operand (FIADD/FIMUL/FISUB/FIDIV, etc.).

2. All real stores (FST/FSTP), Floating Point integer stores (FIST/FISTP) and BCD store (FBSTP) (true for all exception on stores except Precision Exception).

The following class of floating point exceptions drive FERR# only after encountering the next floating point instruction.

1. Numeric underflow, overflow and precision exception on: Transcendental instructions, FSCALE, FXTRACT, FPREM, FPREM(1), FRNDINT, and Precision Exception on all types of stores to memory.

2. All exception on basic arithmetic instructions (FADD/FSUB/FMUL/FDIV/FSQRT/FCOM/FUCOM...)

FERR# is deasserted when the FCLEX, FINIT, FSTENV, or FSAVE instructions are executed.

When Driven

FERR# is driven in every clock and is not floated during bus HOLD or BOFF#. The FERR# signal is glitch free.

Relation to Other Signals

None

5.1.22. FLUSH#

FLUSH#	Cache Flush
	Writes all modified lines in the data cache back and flushes the code and data caches.
	Asynchronous Input

Signal Description

When asserted, the cache flush input forces the Pentium processor to writeback all modified lines in the data cache and invalidate both internal caches. A Flush Acknowledge special cycle will be generated by the Pentium processor indicating completion of the invalidation and writeback.

FLUSH# is implemented in the Pentium processor as an interrupt, so it is recognized on instruction boundaries. External interrupts are ignored while FLUSH# is being serviced. Once FLUSH# is sampled active it is ignored until the flush acknowledge special cycle is driven.

If FLUSH# is sampled low when RESET transitions from high to low, tristate test mode is entered.

When Sampled

FLUSH# is sampled on every rising clock edge. FLUSH# is falling edge sensitive and recognized on instruction boundaries. Recognition of FLUSH# is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. If it meets setup and hold times, FLUSH# need only be asserted for one clock. To guarantee recognition if FLUSH# is asserted asynchronously, it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor and remain asserted for a minimum pulse width of two clocks.

If the processor is in the HALT or Shutdown state, FLUSH# is still recognized. The processor will return to the HALT or Shutdown state after servicing the FLUSH#.

If FLUSH# is sampled low when RESET transitions from high to low, tristate test mode is entered. If RESET is negated synchronously, FLUSH# must be at its valid level and meet setup and hold times on the clock before the falling edge of RESET. If RESET is negated asynchronously, FLUSH# must be at its valid level two clocks before and after RESET transitions from high to low.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS# and cycle definition pins.	Writeback cycles are driven as a result of FLUSH# assertion.
	The Flush Special Cycle is driven as a result of FLUSH# assertion.
RESET	If FLUSH# is sampled low when RESET transitions from high to low, tristate test mode is entered.

5.1.23. FRCMC#

FRCMC#	Functional Redundancy Checking Master/Checker Configuration
	Determines whether the Pentium processor is configured as a Master or Checker.
	Asynchronous Input

Signal Description

The functional redundancy checking master/checker configuration input is sampled in every clock that RESET is asserted to determine whether the Pentium processor is configured in master mode (FRCMC# high) or checker mode (FRCMC# low). When configured as a master, the Pentium processor drives its output pins as required by the bus protocol. When configured as a checker, the Pentium processor tristates all outputs (except IERR# and TDO) and samples the output pins that would normally be driven in master mode. If the sampled value differs from the value computed internally, the Checker Pentium processor asserts IERR# to indicate an error.

Note that the final configuration as a master or checker is set after RESET and may not be changed other than by a subsequent RESET. FRCMC# is sampled in every clock that RESET is asserted to prevent bus contention before the final mode of the processor is determined.

When Sampled

This pin is sampled in any clock in which RESET is asserted. FRCMC# is sampled in the clock before RESET transitions from high to low to determine the final mode of the processor. If RESET is negated synchronously, FRCMC# must be at its valid level and meet setup and hold times on the clock before the falling edge of RESET. If RESET is negated asynchronously, FRCMC# must be at its valid level two clocks before and after RESET transitions from high to low.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
IERR#	IERR# is asserted by the Checker Pentium processor in the event of an FRC error.
RESET	FRCMC# is sampled when RESET is asserted to determine if the Pentium processor is in Master or Checker mode.

5.1.24. HIT#

HIT#	Inquire Cycle Hit/Miss Indication
	Externally indicates whether an inquire cycle resulted in a hit or miss.
	Output

Signal Description

The HIT# output is driven to reflect the outcome of an inquire cycle. If an inquire cycle hits a valid line (M, E, or S) in either the Pentium processor data or instruction cache, HIT# is asserted two clocks after EADS# has been sampled asserted by the processor. If the inquire cycle misses Pentium processor cache, HIT# is negated two clocks after EADS# is sampled asserted. This pin changes its value only as a result of an inquire cycle and retains its value between cycles.

When Driven

HIT# reflects the hit or miss outcome of the inquire cycle 2 clocks after EADS# is sampled asserted. After RESET, this pin is driven high. It changes it value only as a result of an inquire cycle. This pin is always driven. It is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relative to Other Signals
A31-A5	HIT# is driven to indicate whether the inquire address driven on A31-A5 is valid in an internal cache.
EADS#	HIT# is driven two clocks after EADS# is sampled asserted to indicate the out-come of the inquire cycle.
HITM#	HITM# is never asserted without HIT# also being asserted.

5.1.25. HITM#

HITM#	Hit/Miss to a Modified Line
	Externally indicates whether an inquire cycle hit a modified line in the data cache.
	Output

Signal Description

The HITM# output is driven to reflect the outcome of an inquire cycle. If an inquire cycle hits a modified line in the Pentium processor data cache, HITM# is asserted two clocks after EADS# has been sampled asserted by the processor and a writeback cycle is scheduled to be driven to the bus. If the inquire cycle misses Pentium processor cache, HITM# is negated two clocks after EADS# is sampled asserted.

HITM# can be used to inhibit another bus master from accessing the data until the line is completely written back.

HITM# is asserted two clocks after an inquire cycle hits a modified line in the Pentium processor cache. ADS# for the writeback cycle will be asserted no earlier than two clocks after the assertion of HITM#. ADS# for the writeback cycle will be driven even if AHOLD for the inquire cycle is not yet deasserted. ADS# for a writeback of an external snoop cycle is the only ADS# that will be driven while AHOLD is asserted.

When Driven

HITM# is driven two clocks after EADS# is sampled asserted to reflect the outcome of the inquire cycle. HITM# remains asserted until two clocks after the last BRDY# of writeback is returned. This pin is always driven. It is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A31-A5	HITM# is driven to indicate whether the inquire address driven on A31-A5 is in the modified state in the data cache.
EADS#	HITM# is driven two clocks after EADS# is sampled asserted.
HIT#	HITM# is never asserted without HIT# also being asserted.

5.1.26. HLDA

HLDA	Bus Hold Acknowledge
	External indication that the Pentium processor outputs are floated.
	Output

Signal Description

The bus hold acknowledge output goes active in response to a hold request presented on the HOLD pin. HLDA indicates that the Pentium processor has given the bus to another local bus master. Internal instruction execution will continue from the internal caches during bus HOLD/HLDA.

When leaving bus hold, HLDA will be driven inactive and the Pentium processor will resume driving the bus. If the Pentium processor has bus cycle pending, it will be driven in the same clock that HLDA is deasserted.

The operation of HLDA is not affected by the assertion of BOFF#. If HOLD is asserted while BOFF# is asserted, HLDA will be asserted two clocks later. If HOLD goes inactive while BOFF# is asserted, HLDA is deasserted two clocks later.

When Driven

When the Pentium processor bus is idle, HLDA is driven high two clocks after HOLD is asserted, otherwise, HLDA is driven high two clocks after the last BRDY# of the current cycle is returned. It is driven active in the same clock that the Pentium processor floats its bus. When leaving bus hold, HLDA will be driven inactive 2 clocks after HOLD is deasserted and the Pentium processor will resume driving the bus. If the Pentium processor has bus cycle pending, it will be driven in the same clock that HLDA is deasserted. The HLDA signal is glitch free.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A3-A31 ADS# AP BE7#-BE3# CACHE# D/C# D63-D0 DP7-DP0 LOCK# M/IO# PCD PWT SCYC W/R#	These signals float in response to HLDA
BOFF#	The same pins are floated when HLDA or BOFF# is asserted.
EADS#	EADS# is recognized while HLDA is asserted.
HOLD	The assertion of HOLD causes HLDA to be asserted when all outstanding cycles are complete.

5.1.27. HOLD

HOLD	Bus Hold
	The bus hold request input allows another bus master complete control of the Pentium processor bus.
	Synchronous Input

Signal Description

The bus hold request input allows another bus master complete control of the Pentium processor bus. In response to HOLD, after completing all outstanding bus cycles the Pentium processor will float most of its output and input/output pins and assert HLDA. The Pentium processor will maintain its bus in this state until HOLD is deasserted. Cycles that are locked together will not be interrupted by bus HOLD. HOLD is recognized during RESET.

When Sampled

HOLD is sampled on every rising clock edge including during RESET and INIT.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A3-A31	These are the signals floated in response to HOLD
ADS#	
AP	
BE7#-BE3#	
CACHE#	
D/C#	
D63-D0	
DP7-DP0	
LOCK#	
M/IO#	
PCD	
PWT	
SCYC	
W/R#	
HLDA	HLDA is asserted when the Pentium processor relinquishes the bus in response to the HOLD request.

intel.

5.1.28. IBT

IBT	Instruction Branch Taken
	Externally indicates that a branch was taken.
	Output

Signal Description

The instruction branch taken output is driven active (high) for one clock to indicate that a branch was taken. If the execution tracing enable bit in TR12 is set to a branch trace message special cycle will be driven subsequent to the assertion of IBT.

When Driven

This output is always driven by the Pentium processor. It is driven high for 1 clock each time a branch is taken. It is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS# and cycle definition pins	The branch trace message special cycle is driven after an assertion of IBT if the TR12.TR bit is set to 1.
BT3-BT0	If TR12.TR is set to 1, BT3-BT0 are driven along with the branch trace message special cycle for each assertion of IBT.
IU IV	IBT is not asserted without IU and possibly IV being asserted also.

5.1.29. IERR#

IERR#	Internal or Functional Redundancy Check Error
	Alerts the system of internal parity errors and functional redundancy errors
	Output

Signal Description

The internal error output is used to alert the system of two types of errors, internal parity errors and functional redundancy errors.

If a parity error occurs on a read from an internal array (reads during normal instruction execution, reads during a flush operation, reads during BIST and testability cycles, and reads during inquire cycles), the Pentium processor will assert the IERR# pin for one clock and then shutdown. Shutdown will occur provided the processor is not prevented from doing so by the error.

If the Pentium processor is configured as a checker (by FRCMC# being sampled low while RESET is asserted) and a mismatch occurs between the value sampled on the pins and the value computed internally, the Pentium processor will assert IERR# two clocks after the mismatched value is returned. Shutdown is not entered as a result of a function redundancy error.

It is the responsibility of the system to take appropriate action if an internal parity or FRC error occurs.

When Driven

IERR# is driven in every clock. While RESET is active IERR# is driven high. After RESET is deasserted, IERR# will not be asserted due to an FRC mismatch until after the first clock of the first bus cycle. Note however that IERR# may be asserted due to an internal parity error before the first bus cycle. IERR# is asserted for 1 clock for each detected FRC or internal parity error two clocks after the error is detected. IERR# is asserted for each detected mismatch, so IERR# may be asserted for more than one consecutive clock.

IERR# is not floated with HOLD or BOFF#. IERR# is a glitch free signal.

Relation to Other Signals

Pin Symbol	Relative to Other Signals
FRCMC#	If the Pentium processor is configured as a Checker, IERR# will be asserted in the event of an FRC error.

5.1.30. IGNNE#

IGNNE#	Ignore Numeric Exception
	Determines whether or not numeric exceptions should be ignored.
	Asynchronous Input

Signal Description

This is the ignore numerics exception input. This pin has no effect when the NE bit in CR0 is set to 1. When the CR0. NE bit is 0, this pin is functional as follows:

When the IGNNE# pin is asserted, the Pentium processor will ignore any pending unmasked numeric exception and continue executing floating point instructions for the entire duration that this pin is asserted.

When IGNNE# is not asserted and a pending unmasked numeric exception exists, (SW.ES = 1), the Pentium processor will behave as follows:

On encountering a floating point instruction that is one of FINIT, FCLEX, FSTENV, FSAVE, FSTSW, FSTCW, FENI, FDISI, or FSETPM, the Pentium processor will execute the instruction in spite of the pending exception.   .

On encountering any floating point instruction other than FINIT, FCLEX, FSTENV, FSAVE, FSTSW, FSTCW, FENI, FDISI, or FSETPM, the Pentium processor will stop execution and wait for an external interrupt.

When Sampled

IGNNE# is sampled on every rising clock edge. Recognition of IGNNE# is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. To guarantee recognition if IGNNE# is asserted asynchronously, it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor and remain asserted for a minimum pulse width of two clocks.

Relation to Other Signals

None

5.1.31. INIT

INIT	Initialization
	Forces the Pentium processor to begin execution in a known state without flushing the caches or affecting floating point state.
	Asynchronous Input

Signal Description

The initialization input forces the Pentium processor to begin execution in a known state. The processor state after INIT is the same as the state after RESET except that the internal caches, write buffers, model specific registers, and floating point registers retain the values they had prior to INIT. The Pentium processor starts execution at physical address FFFFFFF0H.

INIT can be used to help performance for DOS extenders written for the 80286. INIT provides a method to switch from protected to real mode while maintaining the contents of the internal caches and floating point state. INIT may not be used instead of RESET after power-up.

Once INIT is sampled active, the INIT sequence will begin on the next instruction boundary (unless a higher priority interrupt is requested before the next instruction boundary). The INIT sequence will continue to completion and then normal processor execution will resume, independent of the deassertion of INIT. ADS# will be asserted to drive bus cycles even if INIT is not deasserted.

If INIT is sampled high when RESET transitions from high to low the Pentium processor will perform built-in self test prior to the start of program execution.

When Sampled

INIT is sampled on every rising clock edge. INIT is an edge sensitive interrupt. Recognition of INIT is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. To guarantee recognition if INIT is asserted asynchronously, it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor and remain asserted for a minimum pulse width of two clocks. INIT must remain active for three clocks prior to the BRDY# of an I/O write cycle to guarantee that the Pentium processor recognizes and processes INIT right after an I/O write instruction.

If INIT is sampled high when RESET transitions from high to low the Pentium processor will perform built-in self test. If RESET is driven synchronously, INIT must be at its valid level the clock before the falling edge of RESET. If RESET is driven asynchronously, INIT must be at its valid level two clocks before and after RESET transitions from high to low.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
RESET	If INIT is sampled high when RESET transitions from high to low, BIST will be performed.

5.1.32. INTR

INTR	External Interrupt
	The INTR input indicates that an external interrupt has been generated.
	Asynchronous Input

Signal Description

The INTR input indicates that an external interrupt has been generated. The interrupt is maskable by the IF bit in the EFLAGS register. If the IF bit is set, the Pentium processor will vector to an interrupt handler after the current instruction execution is completed. Upon recognizing the interrupt request, the Pentium processor will generate two locked interrupt acknowledge bus cycles in response to the INTR pin going active. INTR must remain active until the first interrupt acknowledge cycle is generated to assure that the interrupt is recognized.

When Sampled

INTR is sampled on every rising clock edge. INTR is an asynchronous input, but recognition of INTR is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. To guarantee recognition if INTR is asserted asynchronously it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS# and cycle definition pins	An interrupt acknowledge cycle is driven as a result of the INTR pin assertion.
LOCK#	LOCK# is asserted for interrupt acknowledge cycles.

5.1.33. INV

INV	Invalidation Request
	Determines final state of a cache line as a result of an inquire hit.
	Synchronous Input

Signal Description

The INV input is driven to the Pentium processor during an inquire cycle to determine the final cache line state (S or I) in case of an inquire cycle hit. If INV is returned active (high) to the Pentium processor in the same clock as EADS# is sampled asserted, an inquire hit will result in that line being invalidated. If the INV pin is returned inactive (low), an inquire hit will result in that line being marked Shared (S). If the inquire cycle is a miss in the cache, the INV input has no effect.

If an inquire cycle hits a modified line in the data cache, the line will be written back regardless of the state of INV.

When Sampled

The INV input is sampled with the EADS# of the inquire cycle.

Relation to Other Signals

Pin Symbol	Relative to Other Signals
A31-A5	INV determines if the inquire address driven to the processor on A31-A5 should be invalidated or marked as shared if it is valid in an internal cache.
EADS#	INV is sampled with EADS#.

5.1.34. IU

IU	U-Pipe Instruction Complete
	Externally indicates that an instruction in the u-pipeline has completed execution.
	Output

Signal Description

The IU output is driven active (high) for one clock to indicate that an instruction in the u-pipeline has completed execution.

When Driven

This pin is always driven by the Pentium processor. It is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relative to Other Signals
IBT	IBT is not asserted without IU being asserted also.
IV	IV is not asserted without IU being asserted also.

5.1.35. IV

IV	V-Pipe Instruction Complete
	Externally indicates that an instruction in the v-pipeline has completed execution.
	Output

Signal Description

The IV output is driven active (high) for one clock to indicate that an instruction in the v-pipeline has completed execution.

When Driven

This pin is always driven by the Pentium processor. It is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relative to Other Signals
IBT	IBT is not asserted without IU being asserted also.
IU	IV is not asserted without IU being asserted also.

intel®

5.1.36. KEN#

KEN#	Cache Enable
	Indicates to the Pentium processor whether or not the system can support a cache line fill for the current cycle.
	Synchronous Input

Signal Description

KEN# is the cache enable input. It is used to determine whether the current cycle is cacheable or not and consequently is used to determine cycle length.

When the Pentium processor generates a read cycle that can be cached (CACHE# asserted) and KEN# is active, the cycle will be transformed into a burst cache line fill. During a cache line fill the byte enable outputs should be ignored and valid data must be returned on all 64 data lines. The Pentium processor will expect 32-bytes of valid data to be returned in four BRDY# transfers.

If KEN# is not sampled active, a line fill will not be performed (regardless of the state of CACHE#) and the cycle will be a single transfer read.

Once KEN# is sampled active for a cycle, the cacheability cannot be changed. If a cycle is backed off (BOFF#) after the cacheability of the cycle has been determined, the same cacheability attribute on KEN# must be returned to the processor when the cycle is redriven.

When Sampled

KEN# is sampled once in a cycle to determine cacheability. It is sampled and latched with the first BRDY# or NA# of a cycle, however it must meet setup and hold times on every clock edge.

Relation to Other Signals

Pin Symbol	Relative to Other Signals
BRDY#	KEN# is sampled with the first of the first BRDY# or NA# for that cycle. Also, in conjunction with the CACHE# input, KEN# determines whether the bus cycle will consist of 1 or 4 transfers (assertions of BRDY#).
CACHE#	KEN# determines cacheability only if the CACHE# pin is asserted.
NA#	KEN# is sampled with the first of the first BRDY# or NA# for that cycle.
W/R#	KEN# determines cacheability only if W/R# indicates a read.

5.1.37. LOCK#

LOCK#	Bus Lock
	Indicates to the system that the current sequence of bus cycles should not be interrupted.
	Output

Signal Description

The bus lock output indicates that the Pentium processor is running a read-modify-write cycle where the external bus must not be relinquished between the read and write cycles. Read-modify-write cycles of this type are used to implement memory based semaphores. Interrupt Acknowledge cycles are also locked.

If a cycle is split due to a misaligned memory operand, two reads followed by two writes may be locked together. When LOCK# is asserted, the current bus master should be allowed exclusive access to the system bus.

The Pentium processor will not allow a bus hold when LOCK# is asserted, but address holds (AHOLD) and BOFF# are allowed. LOCK# is floated during bus hold.

All locked cycles will be driven to the external bus. If a locked address hits a valid location in one of the internal caches, the cache location is invalidated (if the line is in the modified state, it is written back before it is invalidated). Locked read cycles will not be transformed into cache line fill cycles regardless of the state of KEN#.

LOCK# is guaranteed to be deasserted for at least one clock between back to back locked cycles.

When Driven

LOCK# goes active with the ADS# of the first locked bus cycle and goes inactive after the BRDY# is returned for the last locked bus cycle. The LOCK# signal is glitch free.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	LOCK# is driven with the ADS# of the first locked cycle.
BOFF#	LOCK# floats one clock after BOFF# is asserted
BRDY#	LOCK# is deasserted after the last BRDY# of the locked sequence.
HLDA	LOCK# floats when HLDA is asserted.
NA#	ADS# is not asserted to pipeline an additional cycle if LOCK# is asserted, regardless of the state of NA#.
INTR	LOCK# is asserted for interrupt acknowledge cycles.
SCYC	SCYC is driven active if the locked cycle is misaligned.

intel®

5.1.38. M/IO#

M/IO#	Memory/Input-Output
	Distinguishes a memory access from an I/O access.
	Output

Signal Description

The Memory/Input-Output signal is one of the primary bus cycle definition pins. M/IO# distinguishes between memory (M/IO# =1) and I/O (M/IO# =0) cycles.

When Driven

M/IO# is driven valid in the same clock as ADS# and the cycle address. It remains valid from the clock in which ADS# is asserted until the clock after the earlier of NA# or the last BRDY#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	M/IO# is driven to its valid state with ADS#.
BOFF#	M/IO# floats one clock after BOFF# is asserted.
HLDA	M/IO# floats when HLDA is asserted.

5.1.39. NA#

NA#	Next Address
	Indicates that external memory is prepared for a pipelined cycle.
	Synchronous Input

Signal Description

The Next Address input, when active, indicates that external memory is ready to accept a new bus cycle although all data transfers for the current cycle have not yet completed. This is referred to as bus cycle pipelining.

The Pentium processor will drive out a pending cycle in response to NA# no sooner than two clocks after NA# is asserted. The Pentium processor supports up to 2 outstanding bus cycles. ADS# is not asserted to pipeline an additional cycle if LOCK# is asserted, or during a writeback cycle. In addition, ADS# will not be asserted to pipeline a locked cycle or a writeback cycle into the current cycle.

NA# is latched internally, so once it is sampled active during a cycle, it need not be held active to be recognized. The KEN#, and WB/WT# inputs for the current cycle are sampled with the first NA#, if NA# is asserted before the first BRDY# of the current cycle.

When Sampled

NA# is sampled in all T2, TD and T2P clocks.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	If NA# is sampled asserted and an internal bus request is pending, the Pentium processor drives out the next bus cycle and asserts ADS#.
KEN#	KEN# is sampled with the first of the first BRDY# or NA# for that cycle.
WB/WT#	WB/WT# is sampled with the first of the first BRDY# or NA# for that cycle.
LOCK#	ADS# is not asserted to pipeline an additional cycle if LOCK# is asserted, regardless of the state of NA#.
BOFF#	If NA# and BOFF# are asserted simultaneously, BOFF# takes priority and NA# is ignored.

5.1.40. NMI

NMI	Non Maskable Interrupt
	Indicates that an external non-maskable interrupt has been generated.
	Asynchronous Input

Signal Description

The non-maskable interrupt request input indicates that an external non-maskable interrupt has been generated. Asserting NMI causes an interrupt with an internally supplied vector value of 2. External interrupt acknowledge cycles are not generated.

If NMI is asserted during the execution of the NMI service routine it will remain pending and will be recognized after the IRET is executed by the NMI service routine. At most, one assertion of NMI will be held pending.

When Sampled

NMI is sampled on every rising clock edge. NMI is rising edge sensitive. Recognition of NMI is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. To guarantee recognition if NMI is asserted asynchronously, it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor and remain asserted for a minimum pulse width of two clocks.

Relation to Other Signals

None

5.1.41. PCD

PCD	Page Cacheability Disable
	Externally reflects the cacheability paging attribute bit in CR3, PDE, or PTE.
	Output

Signal Description

PCD is driven to externally reflect the cache disable paging attribute bit for the current cycle. PCD corresponds to bit 4 of CR3, the Page Directory Entry, or the Page Table Entry. For cycles that are not paged when paging is enabled (for example I/O cycles) PCD corresponds to bit 4 in CR3. In real mode or when paging is disabled, the PCD pin reflects the cache disable bit in control register 0 (CR0.CD).

PCD is masked by the CD (cache disable) bit in CR0. When CD =1, the Pentium processor forces PCD HIGH. When CD =0, PCD is driven with the value of the page table entry/directory.

The purpose of PCD is to provide an external cacheability indication on a page by page basis.

When Driven

The PCD pin is driven valid in the same clock as ADS# and the cycle address. It remains valid from the clock in which ADS# is asserted until the clock after the earlier of NA# or the last BRDY#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	PCD is driven valid with ADS#
BOFF#	PCD floats one clock after BOFF# is asserted.
HLDA	PCD floats when HLDA is asserted.

5.1.42. PCHK#

PCHK#	Data Parity Check
	Indicates the result of a parity check on a data read.
	Output

Signal Description

The data parity check pin indicates the result of a parity check on a data read. Data parity is checked during code reads, memory reads, and I/O reads. Data parity is not checked during the first Interrupt Acknowledge cycle. PCHK# indicates the parity status only for the bytes on which valid data is expected. Parity is checked for all data bytes for which a byte enable is asserted. In addition, during a cache line fill, parity is checked on the entire data bus regardless of the state of the byte enables.

PCHK# is driven low two clocks after BRDY# is returned if incorrect parity was returned.

Driving PCHK# is the only effect that bad data parity has on the Pentium processor unless PEN# is also asserted. The data returned to the processor is not discarded.

If PEN# is asserted when a parity error occurs, the cycle address and type will be latched in the MCA and MCT registers. If in addition, the MCE bit in CR4 is set, a machine check exception will be taken.

It is the responsibility of the system to take appropriate actions if a parity error occurs. If parity checks are not implemented in the system, the PCHK# pin may be ignored, and PEN# pulled high (or CR4.MCE cleared).

When Driven

PCHK# is driven low two clocks after BRDY# is returned if incorrect parity was returned. PCHK# remains low one clock for each clock in which a parity error was detected. At all other times PCHK# is inactive (HIGH). PCHK# is not floated during bus HOLD or BOFF#. PCHK# is a glitch free signal.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BRDY#	PCHK# is driven to its valid level two clocks after the assertion of BRDY#.
D63-D0	The DP7-DP0 pins are used to create even parity with D63-D0. If even parity is not returned, the PCHK# pin is asserted.
DP7-DP0	Even data parity with D63-D0 should be returned on to the Pentium processor on the DP pin. If even parity is not returned, the PCHK# pin is asserted.

5.1.43. PEN#

PEN#	Parity Enable
	Indicates to the Pentium processor that the correct data parity is being returned by the system. Determines if a Machine Check Exception should be taken if a data parity error is detected.
	Synchronous Input

Signal Description

The PEN# input (along with CR4.MCE) determines whether a machine check exception will be taken as a result of a data parity error on a read cycle. If this pin is sampled active in the clock a data parity error is detected, the Pentium processor will latch the address and control signals of the cycle with the parity error in the machine check registers. If in addition the machine check enable bit in CR4 is set to "1", the Pentium processor will vector to the machine check exception before the beginning of the next instruction.

This pin may be tied to Vss.

When Sampled

This signal is sampled when BRDY# is asserted for memory and I/O read cycles and the second interrupt acknowledge cycle.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BRDY#	PEN# is sampled with BRDY# for read cycles.
D63-D0	The DP7-DP0 pins are used to create even parity with D63-D0. If even parity is not returned, and PEN# is enabled, the cycle will be latched and a MCE will be taken if CR4.MCE = 1.
DP7-DP0	Even data parity with D63-D0 should be returned to the Pentium processor on the DP pins. If even parity is not returned, and PEN# is enabled, the cycle will be latched and a MCE will be taken if CR4.MCE = 1

5.1.44. PM/BP[1:0]

PM/BP1-0	Performance Monitoring and Breakpoint
	See Appendix A.
	Output

Signal Description

For more information on the performance monitoring pins, see Appendix A.

BP1 and BP0 are multiplexed with the Performance Monitoring pins (PM1 and PM0). The PB1 and PB0 bits in the Debug Mode Control Register determine if the pins are configured as breakpoint or performance monitoring pins. The pins come out of reset configured for performance monitoring (for more information see Appendix A).

When Driven

The BP[3:2], PM/BP[1:0] pins are driven in every clock and are not floated during bus HOLD, or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BP1-BP0	PM1 and PM0 are share pins with BP1 and BP0

5.1.45. PRDY

PRDY	PRDY
	For use with the Intel debug port
	Output

Signal Description

The PRDY output pin indicates that the processor has stopped normal execution in response to the R/S# pin going active, or Probe Mode being entered. See Appendix A for more information regarding Probe Mode.

The PRDY pin is provided for use with the Intel debug port described in the "Debugging" chapter.

When Driven

This output is always driven by the Pentium processor. It is not floated during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
R/S#	R/S# is also used with the Intel debug port. Deassertion of R/S# to resume normal operation should only occur while PRDY is asserted.

5.1.46. PWT

PWT	Page Write Through
	Externally reflects the writethrough paging attribute bit in CR3, PDE, or PTE.
	Output

Signal Description

PWT is driven to externally reflect the cache write through paging attribute bit for the current cycle. PWT corresponds to bit 3 of CR3, the Page Directory Entry, or the Page Table Entry. For cycles that are not paged when paging is enabled (for example I/O cycles), PWT correspond to bit 3 in CR3. In real mode or when paging is disabled, the Pentium processor drives PWT low.

PWT can override the affect of the WB/WT# pin. If PWT is asserted for either reads or writes, the line is saved in, or remains in, the Shared (S) state.

When Driven

The PWT pin is driven valid in the same clock as ADS# and the cycle address. It remains valid from the clock in which ADS# is asserted until the clock after the earlier of NA# or the last BRDY#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	PWT is driven valid with ADS#
BOFF#	PWT floats one clock after BOFF# is asserted.
HLDA	PWT floats when HLDA is asserted.
WB/WT#	PWT is used in conjunction with the WB/WT# pin to determine the MESI state of cache lines.

5.1.47. R/S#

R/S#	R/S#
	For use with the Intel debug port.
	Asynchronous Input
	Internal Pullup Resistor

Signal Description

The R/S# pin is provided for use with the Intel debug port described in the "Debugging" chapter.

The R/S# input is an asynchronous, edge sensitive interrupt used to stop the normal execution of the processor and place it into an idle state. The R/S# pin is implemented as an interrupt. A high to low transition on the R/S# pin will interrupt the processor and cause it to stop execution at the next instruction boundary. While in this mode, the processor does not recognize any external interrupts. External interrupts that are latched are held pending and are serviced when the processor resumes normal operation. Those interrupts that are not latched must be held pending until they are recognized by the processor after R/S# is deasserted.

The R/S# pin works in conjunction with the PRDY output from the processor. Waiting for the PRDY output to go active ensures that the processor has stopped all execution. A low to high transition on the R/S# pin to resume normal operation must not occur until the PRDY output from the processor is sampled asserted.

Since the R/S# pin functions as an interrupt, the frequency at which it may toggle and its recognition by the processor can affect normal instruction execution. In order to guarantee execution of at least one instruction between back to back assertion of the R/S# pin, the system can qualify every subsequent assertion of R/S# with two assertions of the IU output from the Pentium processor. After R/S# is deasserted, the processor activates the IU pin for one clock to indicate that it has successfully returned from the interrupt (resumed normal operation). This is the first assertion of the IU pin. The Pentium processor then generates a prefetch cycle to re-fill the instruction pipeline and continue code execution. As soon as one instruction has completed execution, the processor activates the IU pin again for one clock. This second assertion of IU confirms that at least one instruction has completed execution before R/S# is asserted again.

When Sampled

This pin should not be driven except in conjunction with the Intel debug port.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
PRDY	PRDY is also used with the Intel debug port. A low to high transition on the R/S# pin to resume normal operation must not occur until the PRDY output is asserted.

5.1.48. RESET

RESET	Reset	-
	Forces the Pentium processor to begin execution at a known state.	
	Asynchronous Input	

Signal Description

The RESET input forces the Pentium processor to begin execution at a known state. All the Pentium processor internal caches (code and data caches, the translation lookaside buffers, branch target buffer and segment descriptor cache) will be invalidated upon the RESET. Modified lines in the data cache are not written back. When RESET is asserted, the Pentium processor will immediately abort all bus activity and perform the RESET sequence. The Pentium processor starts execution at FFFFFFF0H.

When RESET transitions from high to low, FLUSH# is sampled to determine if tristate test mode will be entered, FRCMC# is sampled to determine if the Pentium processor will be configured as a master or a checker, and INIT is sampled to determine if BIST will be run.

When Sampled

RESET is sampled on every rising clock edge. RESET must remain asserted for a minimum of 1 millisecond after VCC and CLK have reached their AC/DC specifications for the "cold" or "power on" reset. During power up, RESET should be asserted while VCC is approaching nominal operating voltage (the simplest way to insure this is to place a pullup resistor on RESET). RESET must remain active for at least 15 clocks while Vcc and CLK are within their operating limits for a "warm reset". Recognition of RESET is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. To guarantee recognition if RESET is asserted asynchronously, it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor.

FLUSH#, FRCMC# and INIT are sampled when RESET transitions from high to low to determine if tristate test mode or checker mode will be entered, or if BIST will be run. If RESET is driven synchronously, these signals must be at their valid level and meet setup and hold times on the clock before the falling edge of RESET. If RESET is driven asynchronously, these signals must be at their valid level two clocks before and after RESET transitions from high to low.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
FLUSH#	If FLUSH# is sampled low when RESET transitions from high to low, tristate test mode will be entered.
FRCMC#	FRCMC# is sampled when RESET transitions from high to low to determine if the Pentium processor is in Master or Checker mode.
INIT	If INIT is sampled high when RESET transitions from high to low, BIST will be performed.

5.1.49. SCYC

SCYC	Split Cycle Indication
	Indicates that a misaligned locked transfer is on the bus.
	Output

Signal Description

The split cycle output is activated during misaligned locked transfers. It is asserted to indicate that more than two cycles will be locked together. This signal is defined for locked cycles only. It is undefined for cycles which are not locked.

The Pentium processor defines misaligned transfers as a 16-bit or 32-bit transfer which crosses a 4-byte boundary, or a 64-bit transfer which crosses an 8-byte boundary.

When Driven

SCYC is asserted with the first ADS# of a misaligned lock cycle and remains valid until the clock after the earlier of NA# or the last BRDY# of the last locked cycle.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	SCYC is driven valid in the same clock as ADS#.
BOFF#	SCYC is floated one clock after BOFF# is asserted.
HLDA	SCYC is floated when HLDA is asserted.
LOCK#	SCYC is defined for locked cycles only.

5.1.50. SMI#

SMI#	System Management Interrupt
	Latches a System Management Interrupt request.
	Asynchronous Input
	Internal Pullup Resistor

Signal Description

The system management interrupt input latches a System Management Interrupt request. After SMI# is recognized on an instruction boundary, the Pentium processor waits for all writes to complete and EWBE# to be asserted, then asserts the SMIACT# output. The processor will then save its register state to SMRAM space and begin to execute the SMM handler. The RSM instruction restores the registers and returns to the user program.

Subsequent SMI# requests are not acknowledged while the processor is in system management mode (SMM) and are held pending until the processor completes an RSM instruction.

When Sampled

SMI# is sampled on every rising clock edge. SMI# is a falling edge sensitive input. Recognition of SMI# is guaranteed in a specific clock if it is asserted synchronously and meets the setup and hold times. To guarantee recognition if SMI# is asserted asynchronously, it must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor and remain asserted for a minimum pulse width of two clocks.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
SMIACT#	When the SMI# input is recognized, the Pentium processor asserts SMIACT#.

intel ®

5.1.51. SMIACT#

SMIACT#	System Management Interrupt Active
	Indicates that the processor is operating in SMM.
	Output

Signal Description

The system management interrupt active output is asserted in response to the assertion of SMI#. It indicates that the processor is operating in System Management Mode (SMM). It will remain active (low) until the processor executes the RSM instruction to leave SMM.

When Driven

SMIACT# is driven active in response to the assertion of SMI# after all internally pending writes are complete and the EWBE# pin is active (low). It will remain active (low) until the processor executes the RSM instruction to leave SMM. This signal is always driven. It does not float during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
EWBE#	SMIACT# is not asserted until EWBE# is active
SMI#	SMIACT# is asserted when the SMI# is recognized.

5.1.52. TCK

TCK	Test Clock Input
	Provides Boundary Scan clocking function.
	Input
	Internal Pullup Resistor

Signal Description

This is the testability clock input that provides the clocking function for the Pentium processor boundary scan in accordance with the boundary scan interface (IEEE Std 1149.1). It is used to clock state information and data into and out of the Pentium processor during boundary scan or probe mode operation. State select information and data are clocked into the Pentium processor on the rising edge of TCK on TMS and TDI inputs respectively. Data is clocked out of the Pentium processor on the falling edge of TCK on TDO.

When TCK is stopped in a low state the boundary scan latches retain their state indefinitely. When boundary scan is not used, TCK should be tied high or left as a no-connect.

When Sampled

TCK is a clock signal and is used as a reference for sampling other boundary scan signals.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
TDI	Serial data is clocked into the Pentium processor on the rising edge of TCK.
TDO	Serial data is clocked out of the Pentium processor on the falling edge of TCK.
TMS	TAP controller state transitions occur on the rising edge of TCK.

5.1.53. TDI

TDI	Test Data Input
	Input to receive serial test data and instructions.
	Synchronous Input to TCK
	Internal Pullup Resistor

Signal Description

This is the serial input for the Boundary Scan and Probe Mode test logic. TAP instructions and data are shifted into the Pentium processor on the TDI pin on the rising edge of TCK when the TAP controller is in the SHIFT-IR and SHIFT-DR states. During all other states, TDI is a "don't care".

An internal pull-up resistor is provided on TDI to ensure a known logic state if an open circuit occurs on the TDI path. Note that when "1" is continuously shifted into the instruction register, the BYPASS instruction is selected.

When Sampled

TDI is sampled on the rising edge of TCK during the SHIFT-IR and SHIFT-DR states. During all other states, TDI is a "don't care".

Relation to Other Signals

Pin Symbol	Relation to Other Signals
TCK	TDI is sampled on the rising edge of TCK.
TDO	In the SHIFT-IR and SHIFT-DR TAP controller states, TDO contains the output data of the register being shifted and TDI provides the input.
TMS	TDI is sampled only in the SHIFT-IR and SHIFT DR states (controlled by TMS).

5.1.54. TDO

TDO	Test Data Output
	Outputs serial test data and instructions.
	Output

Signal Description

This is the serial output of the Boundary Scan and Probe Mode test logic. TAP and Probe Mode instructions and data are shifted out of the Pentium processor on the TDO pin on the falling edge of TCK when the TAP controller is in the SHIFT-IR and SHIFT-DR states. During all other states, the TDO pin is driven to the high impedance state to allow connecting TDO of different devices in parallel.

When Driven

TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT-DR TAP controller states. At all other times, TDO is driven to the high impedance state. TDO does not float during bus HOLD or BOFF#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
TCK	TDO is driven on the falling edge of TCK.
TDI	In the SHIFT-IR and SHIFT-DR TAP controller states, TDI provides the input data to the register being shifted and TDO provides the output.
TMS	TDO is driven only in the SHIFT-IR and SHIFT DR states (controlled by TMS).

intel.

5.1.55. TMS

TMS	Test Mode Select
	Controls TAP controller state transitions.
	Synchronous Input to TCK
	Internal Pullup Resistor

Signal Description

This a Boundary Scan test logic control input. The value of this input signal sampled at the rising edge of TCK controls the sequence of TAP controller state changes.

To ensure deterministic behavior of the TAP controller, TMS is provided with an internal pullup resistor. If boundary scan is not used, TMS may be tied to VCC or left unconnected.

When Sampled

TMS is sampled on every rising edge of TCK.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
TCK	TMS is sampled on every rising edge of TCK.
TDI	TDI is sampled only in the SHIFT-IR and SHIFT DR states (controlled by TMS).
TDO	TDO is driven only in the SHIFT-IR and SHIFT DR states (controlled by TMS).

5.1.56. TRST#

TRST#	Test Reset
	Allows the TAP controller to be asynchronously initialized.
	Asynchronous Input
	Internal Pullup Resistor

Signal Description

This is a Boundary Scan test logic reset or initialization pin. When asserted, it allows the TAP controller to be asynchronously initialized. When asserted TRST# will force the TAP controller into the Test Logic Reset State. When in this state, the test logic is disabled so that normal operation of the device can continue unhindered. During initialization the Pentium processor initializes the instruction register with the IDCODE instruction.

An alternate method of initializing the TAP controller is to Drive TMS high for at least 5 TCK cycles. In addition, the Pentium processor implements a power on TAP controller reset function. When the Pentium processor is put through its normal power on/RESET function, the TAP controller is automatically reset by the processor. The user does not have to assert the TRST# pin or drive TMS high after the falling edge of RESET.

When Sampled

TRST# is an asynchronous input.

Relation to Other Signals

None

5.1.57. W/R#

W/R#	Write/Read
	Distinguishes a read cycle from a write cycle.
	Output

Signal Description

The Write/Read signal is one of the primary bus cycle definition pins. W/R# distinguishes between write (W/R# = 1) and read cycles (W/R# = 0).

When Driven

W/R# is driven valid in the same clock as ADS# and the cycle address. It remains valid from the clock in which ADS# is asserted until the clock after the earlier of NA# or the last BRDY#.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	W/R# is driven to its valid state with ADS#.
BOFF#	W/R# floats one clock after BOFF# is asserted.
HLDA	W/R# floats when HLDA is asserted.
KEN#	KEN# determines cacheability only if W/R# indicates a read.

5.1.58. WB/WT#

WB/WT#	Writeback/Writethrough
	This pin allows a cache line to be defined as write back or write through on a line by line basis.
	Synchronous Input

Signal Description

This pin allows a cache line to be defined as write back or write through on a line by line basis. As a result, in conjunction with the PWT pin, it controls the MESI state that the line is saved in.

If WB/WT# is sampled high during a memory read cycle and the PWT pin is low, the line is saved in the Exclusive (E) state in the cache. If WB/WT# is sampled low during a memory read cycle the line is saved in the Shared (S) state in the cache.

If WB/WT# is sampled high during a write to a shared line in the cache and the PWT pin is low, the line transitions to the E state. If WB/WT# is sampled low during a write to a shared line in the cache, the line remains in the S state.

If for either reads or writes the PWT pin is high the line is saved in, or remains in, the Shared (S) state.

When Sampled

This pin is sampled with KEN# on the clock in which the first BRDY# or NA# is returned, however it must meet setup and hold times on every clock edge.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BRDY# NA#	WB/WT# is sampled with the first of the first BRDY# or NA for that cycle.
PWT	If PWT is high, WB/WT# is a "don't care".

intel®

6

Bus Functional
Description

intel®

CHAPTER 6
BUS FUNCTIONAL DESCRIPTION

The Pentium processor bus is designed to support a 528 Mbyte/sec data transfer rate at 66 MHz. All data transfers occur as a result of one or more bus cycles. This chapter describes the Pentium processor bus cycles and the Pentium processor data transfer mechanism.

6.1. PHYSICAL MEMORY AND I/O INTERFACE

Pentium processor memory is accessible in 8, 16, 32, and 64-bit quantities. Pentium processor I/O is accessible in 8, 16, and 32 bit quantities. The Pentium processor can directly address up to 4 Gbytes of physical memory, and up to 64 Kbytes of I/O.

In hardware, memory space is organized as a sequence of 64-bit quantities. Each 64-bit location has eight individually addressable bytes at consecutive memory addresses (see Figure 6-1).
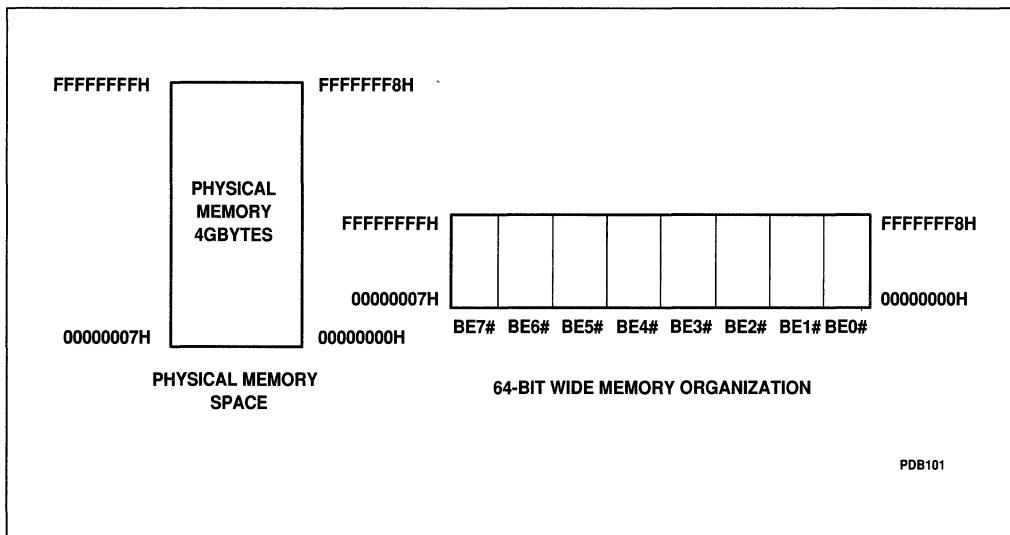
Figure 6-1. Memory Organization

I/O space is organized as a sequence of 32-bit quantities. Each 32-bit quantity has four individually addressable bytes at consecutive memory addresses. See Figure 6-2 for a conceptual diagram of the I/O space.

intel®
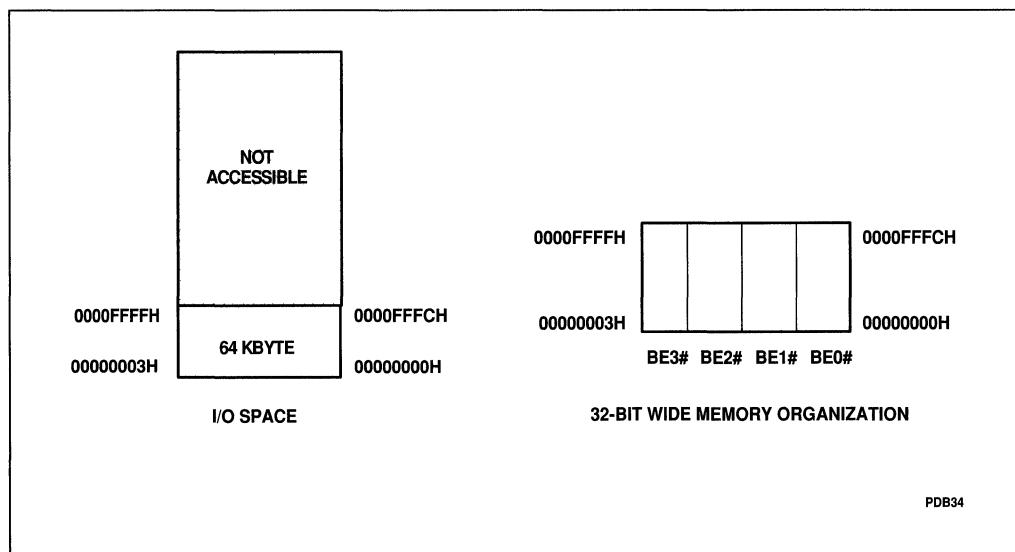
Figure 6-2. I/O Space Organization

64-bit memories are organized as arrays of physical quadwords (8-byte words). Physical quadwords begin at addresses evenly divisible by 8. The quadwords are addressable by physical address lines A31-A3.

32-bit memories are organized as arrays of physical dwords (4-byte words). Physical dwords begin at addresses evenly divisible by 4. The dwords are addressable by physical address lines A31-A3, and A2. A2 can be decoded from the byte enables according to Table 6-2.

16-bit memories are organized as arrays of physical words (2-byte words). Physical words begin at addresses evenly divisible by 2. The words are addressable by physical address lines A31-A3, A2-A1, and BHE#, BLE#. A2 and A1 can be decoded from the byte enables according to Table 6-2, BHE# and BLE# can be decoded from the byte enables according to Table 6-3 and Table 6-4.

To address 8-bit memories, the lower 3 address lines (A2-A0) must be decoded from the byte enables as indicated in Table 6-2.

6.2. DATA TRANSFER MECHANISM

All data transfers occur as a result of one or more bus cycles. Logical data operands of byte, word, dword, and quadword lengths may be transferred. Data may be accessed at any byte boundary, but two cycles may be required for misaligned data transfers. The Pentium processor considers a 2-byte or 4-byte operand that crosses a 4-byte boundary to be misaligned. In addition, an 8-byte operand that crosses and 8-byte boundary is misaligned.

Like the Intel486 CPU, the Pentium processor address signals are split into two components. High-order address bits are provided by the address lines A31-A3. The byte enables BE7#-BE0# form the low-order address and select the appropriate byte of the 8-byte data bus.

The byte enable outputs are asserted when their associated data bus bytes are involved with the present bus cycle as shown in Table 6-1. Non-contiguous byte enable patterns will never occur.

Table 6-1. Pentium™ Processor Byte Enables and Associated Data Bytes

Byte Enable Signal	Associated Data Bus Signals
BE0#	D0-D7 (byte 0 - least significant)
BE1#	D8-D15 (byte 1)
BE2#	D16-D23 (byte 2)
BE3#	D24-D31 (byte 3)
BE4#	D32-D39 (byte 4)
BE5#	D40-D47 (byte 5)
BE6#	D48-D55 (byte 6)
BE7#	D56-D63 (byte 7- most significant)

Address bits A2-A0 of the physical address can be decoded from the byte enables according to Table 6-2. The byte enables can also be decoded to generate BLE# (byte low enable) and BHE# (byte high enable) to address 16-bit memory systems (see Table 6-3 and Table 6-4).

Table 6-2. Generating A2-A0 from BE7-0#

A2	A1	A0	BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#
0	0	0	X	X	X	X	X	X	X	LOW
0	0	1	X	X	X	X	X	X	LOW	HIGH
0	1	0	X	X	X	X	X	LOW	HIGH	HIGH
0	1	1	X	X	X	X	LOW	HIGH	HIGH	HIGH
1	0	0	X	X	X	LOW	HIGH	HIGH	HIGH	HIGH
1	0	1	X	X	LOW	HIGH	HIGH	HIGH	HIGH	HIGH
1	1	0	X	LOW	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH
1	1	1	LOW	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH

Table 6-3. When BLE# is Active

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	BLE#
X	X	X	X	X	X	X	LOW	LOW
X	X	X	X	X	LOW	HIGH	HIGH	LOW
X	X	X	LOW	HIGH	HIGH	HIGH	HIGH	LOW
X	LOW	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	LOW

Table 6-4. When BHE# is Active

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	BHE#
X	X	X	X	X	X	LOW	X	LOW
X	X	X	X	LOW	X	HIGH	HIGH	LOW
X	X	LOW	X	HIGH	HIGH	HIGH	HIGH	LOW
LOW	X	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	LOW

Because the data bus is 64-bits, special considerations need to be made for interfacing to 32-bit memory systems. Address bit 2 along with the appropriate byte enable signals need to be generated by external hardware. Address bit 2 is generated as shown in Table 6-2. New byte enable signals BE3'#- BE0'# are generated as shown in Tables 6-5 through 6-6.

Table 6-5. When BE3'# is Active

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	BE3'#
LOW	X	X	X	LOW	X	X	X	LOW

Table 6-6. When BE2'# is Active

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	BE2'#
X	LOW	X	X	X	LOW	X	X	LOW

Table 6-7. When BE1'# is Active

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	BE1'#
X	X	LOW	X	X	X	LOW	X	LOW

Table 6-8. When BE0'# is Active

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	BE0'#
X	X	X	LOW	X	X	X	LOW	LOW

6.2.1. Interfacing With 8, 16, 32, and 64 bit Memories

In 64-bit physical memories such as Figure 6-3, each 8-byte qword begins at a byte address that is a multiple of eight. A31-A3 are used as a 8-byte qword select and BE7#-BE0# select individual bytes within the word.
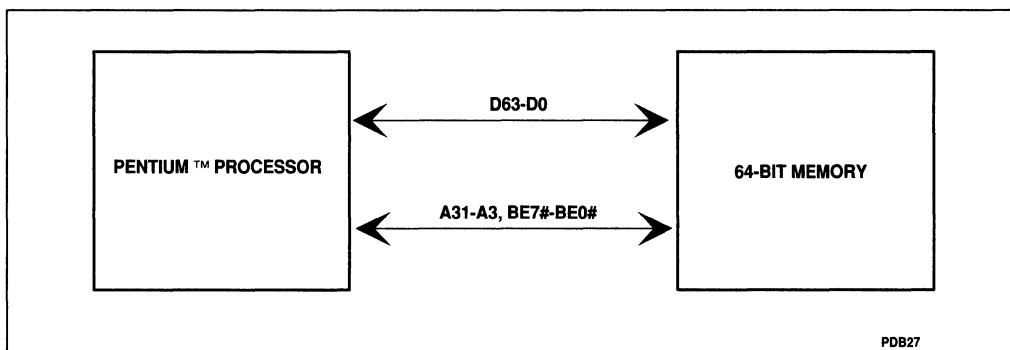
Figure 6-3. Pentium™ Processor With 64-Bit Memory

Memories that are 32-bits wide require external logic for generating A2, and BE3'#-BE0'#. Memories that are 16-bits wide require external logic for generating A2, A1, BHE# and BLE#. Memories that are 8-bits wide require external logic for generating A2, A1, and A0. All memory systems that are less than 64-bits wide require external byte swapping logic for routing data to the appropriate data lines.

The Pentium processor expects all the data requested by the byte enables to be returned as one transfer (with one BRDY#), so byte assembly logic is required to return all requested bytes to the Pentium processor at one time. Note that the Pentium processor does not support BS8# or BS16# (or BS32#), so this logic must be implemented externally if necessary.

Figure 6-4 shows the Pentium processor address bus interface to 64, 32, 16 and 8-bit memories. Address bits A2, A1, and A0 and BHE#, BLE#, and BE3'#-BE0'# are decoded as shown in Table 6-2 through Table 6-8.
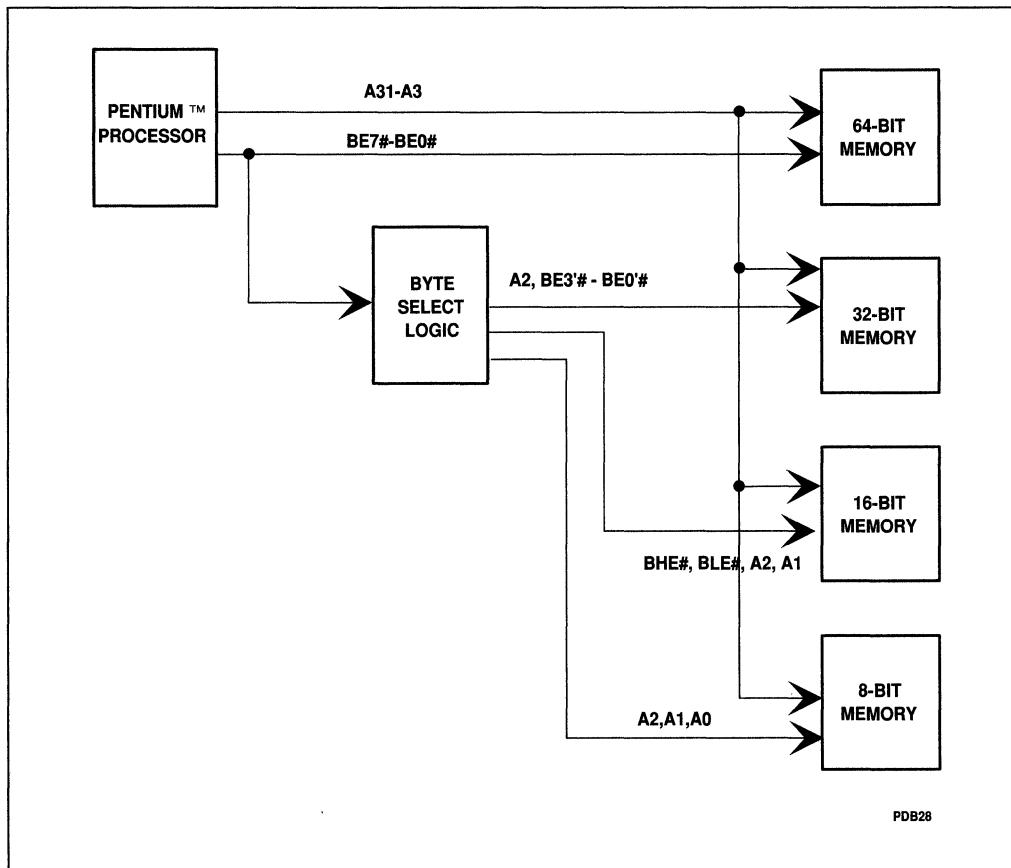
Figure 6-4. Addressing 32, 16, and 8-Bit Memories

Figure 6-5 shows the Pentium processor data bus interface to 32, 16 and 8-bit wide memories. External byte swapping logic is needed on the data lines so that data is supplied to and received from the Pentium processor on the correct data pins (see Table 6-1). For memory widths smaller than 64-bits, byte assembly logic is needed to return all bytes of data requested by the Pentium processor in one cycle.
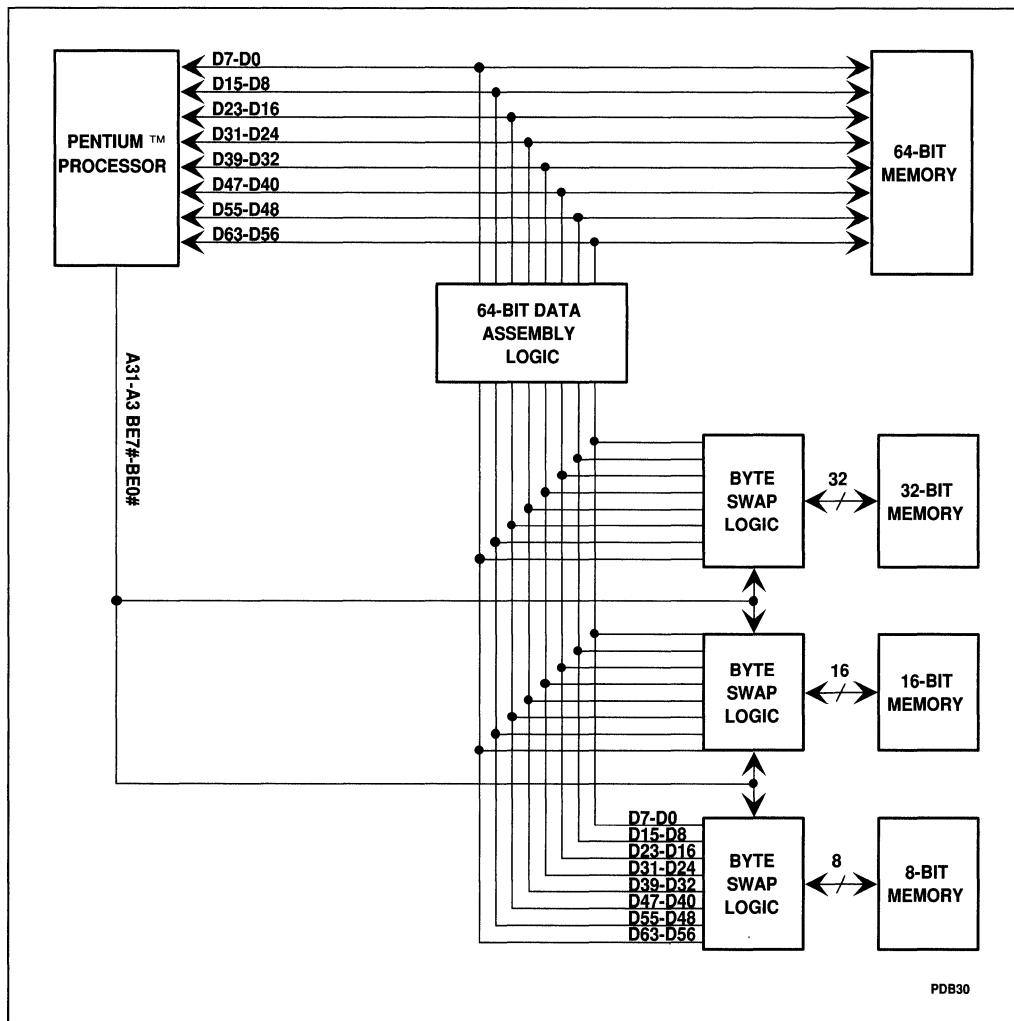
Figure 6-5. Data Bus Interface to 32, 16, and 8-bit Memories

Operand alignment and size dictate when two cycles are required for a data transfer. Table 6-9 shows the transfer cycles generated by the Pentium processor for all combinations of logical operand lengths and alignment. Table 6-9 applies to both locked and unlocked transfers. When multiple cycles are required to transfer a multi-byte logical operand, the highest order bytes are transferred first.

Table 6-9. Transfer Bus Cycles for Bytes, Words, Dwords, Quadwords

Length of transfer	1 byte	2 bytes							
Low order address	xxx	000	001	010	011	100	101	110	111
1st Transfer	b	w	w	w	hb	w	w	w	hb
byte enables driven	0	BE0-1#	BE1-2#	BE2-3#	BE4#	BE4-5#	BE5-6#	BE6-7#	BE0#
value driven on A3		0	0	0	0	0	0	0	1
2nd Transfer (if needed)					lb				lb
byte enables driven					BE3#				BE7#
value driven on A3					0				0

Length of transfer	4 bytes							
Low order address	000	001	010	011	100	101	110	111
1st Transfer	d	hb	hw	h3	d	hb	hw	h3
byte enables driven	BE0-3#	BE4#	BE4-5#	BE4-6#	BE4-7#	BE0#	BE0-1#	BE0-2#
low order address	0	0	0	0	0	1	1	1
2nd Transfer (if needed)		l3	lw	lb		l3	lw	lb
byte enables driven		BE1-3#	BE2-3#	BE3#		BE5-7#	BE6-7#	BE7#
value driven on A3		0	0	0		0	0	0

Length of transfer	8 bytes							
Low order address	000	001	010	011	100	101	110	111
1st Transfer	q	hb	hw	h3	hd	h5	h6	h7
byte enables driven	BE0-7#	BE0#	BE0-1#	BE0-2#	BE0-3#	BE0-4#	BE0-5#	BE0-6#
value driven on A3	0	1	1	1	1	1	1	1
2nd Transfer (if needed)		l7	l6	l5	ld	l3	lw	lb
byte enables driven		BE1-7#	BE2-7#	BE3-7#	BE4-7#	BE5-7#	BE6-7#	BE7#
value driven on A3		0	0	0	0	0	0	0

Key:

b = byte transfer   w = 2-byte transfer 3 = 3-byte transfer   d = 4-byte transfer

5 = 5-byte transfer   6 = 6-byte transfer   7 = 7-byte transfer   q = 8-byte transfer

h = high order   l = low order

8-byte operand:

high order byte	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	low order byte

↑   ↑

byte with highest address   byte with lowest address

6.3. BUS CYCLES

The following terminology is used in this document to describe the Pentium processor bus functions. The Pentium processor requests data transfer cycles, bus cycles, and bus operations. A *data transfer cycle* is one data item, up to 8 bytes in width, being returned to the Pentium processor or accepted from the Pentium processor with BRDY# asserted. A *bus cycle* begins with the Pentium processor driving an address and status and asserting ADS#, and ends when the last BRDY# is returned. A bus cycle may have 1 or 4 data transfers. A *burst cycle* is a bus cycle with 4 data transfers. A *bus operation* is a sequence of bus cycles to carry out a specific function, such as a locked read-modify-write or an interrupt acknowledge.

The section titled "Bus State Definition" describes each of the bus states, and shows the bus state diagram.

Table 6-10 and Table 6-11 list all of the bus cycles that will be generated by the Pentium processor. Note that inquire cycles (initiated by EADS#) may be generated from the system to the Pentium processor.

intel®

Table 6-10. Pentium™ Processor Initiated Bus Cycles

M/IO#	D/C#	W/R#	CACHE#*	KEN#	Cycle Description	# of Transfers
0	0	0	1	x	Interrupt Acknowledge (2 locked cycles)	1 transfer each cycle
0	0	1	1	x	Special Cycle (Table 6-11)	1
0	1	0	1	x	I/O Read, 32-bits or less, Non-cacheable	1
0	1	1	1	x	I/O Write, 32-bits or less Non-cacheable	1
1	0	0	1	x	Code Read, 64-bits, Non-cacheable	1
1	0	0	x	1	Code Read, 64-bits, Non-cacheable	1
1	0	0	0	0	Code Read, 256-bit burst line fill	4
1	0	1	x	x	Intel Reserved (will not be driven by the Pentium™ processor)	n/a
1	1	0	1	x	Memory Read, 64-bits or less, Non-cacheable	1
1	1	0	x	1	Memory Read, 64-bits or less, Non-cacheable	1
1	1	0	0	0	Memory Read, 256-bit burst line fill	4
1	1	1	1	x	Memory Write, 64-bits or less Non-cacheable	1
1	1	1	0	x	256-bit Burst Writeback	4

* CACHE# will not be asserted for any cycle in which M/IO# is driven low, or for any cycle in which PCD is driven high.

Table 6-11. Special Bus Cycles Encoding

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	Special Bus Cycle
1	1	1	1	1	1	1	0	Shutdown
1	1	1	1	1	1	0	1	Flush (INVD,WBINVD instr)
1	1	1	1	1	0	1	1	Halt
1	1	1	1	0	1	1	1	Write Back (WBINVD instruction)
1	1	1	0	1	1	1	1	Flush Acknowledge (FLUSH# assertion)
1	1	0	1	1	1	1	1	Branch Trace Message

Note that all burst reads are cacheable, and all cacheable read cycles are bursted. There are no non-cacheable burst reads, or non-burst cacheable reads.

The remainder of this chapter describes all of the above bus cycles in detail. In addition, locked operations and bus cycle pipelining will be discussed.

6.3.1.   Single-Transfer Cycle

The Pentium processor supports a number of different types of bus cycles. The simplest type of bus cycle is a single-transfer non-cacheable 64-bit cycle, either with or without wait states. Non-pipelined read and write cycles with 0 waitstates are shown in Figure 6-6.

The Pentium processor initiates a cycle by asserting the address status signal (ADS#) in the first clock. The clock in which ADS# is asserted is by definition the first clock in the bus cycle. The ADS# output indicates that a valid bus cycle definition and address is available on the cycle definition pins and the address bus. The CACHE# output is deasserted (high) to indicate that the cycle will be a single transfer cycle.

For a zero wait-state transfer, BRDY# is returned by the external system in the second clock of the bus cycle. BRDY# indicates that the external system has presented valid data on the data pins in response to a read or the external system has accepted data in response to a write. The Pentium processor samples the BRDY# input in the second and subsequent clocks of a bus cycle (the T2, T12 and T2P bus states, see the Bus State Definition section of this chapter for more information).

The timing of the data parity input, DP, and the parity check output, PCHK#, is also shown in Figure 6-6. DP is driven by the Pentium processor and returned to the Pentium processor in the same clock as the data. PCHK# is driven two clocks after BRDY# is returned for reads with the results of the parity check.

intel®
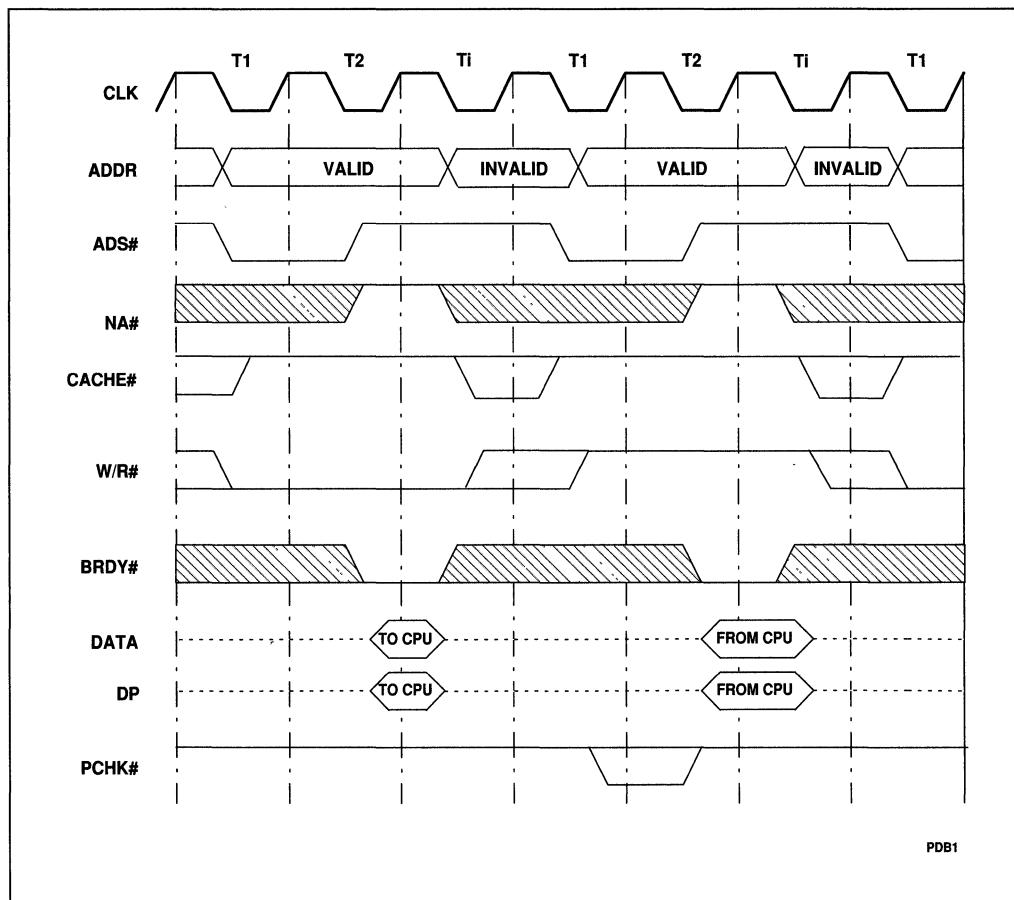
Figure 6-6. Non Pipelined Read and Write

If the system is not ready to drive or accept data, wait states can be added to these cycles by not returning BRDY# to the processor at the end of the second clock. Cycles of this type, with one and two wait states added are shown in Figure 6-7. Note that BRDY# must be driven inactive at the end of the second clock. Any number of wait states can be added to Pentium processor bus cycles by maintaining BRDY# inactive.

I
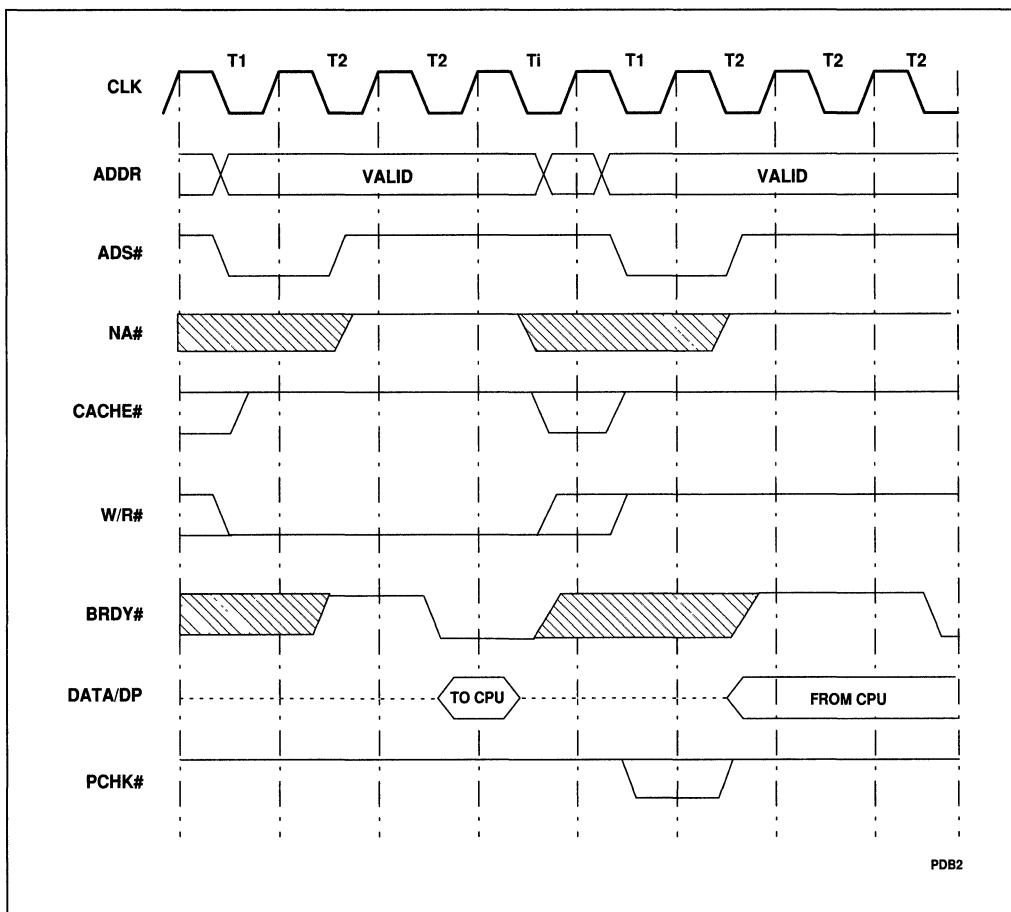
Figure 6-7. Non Pipelined Read and Write With Wait States

6.3.2. Burst Cycles

For bus cycles that require more than a single data transfer (cacheable cycles and writeback cycles), the Pentium processor uses the burst data transfer. In burst transfers, a new data item can be sampled or driven by the Pentium processor in consecutive clocks. In addition the addresses of the data items in burst cycles all fall within the same 32-byte aligned area (corresponding to an internal Pentium processor cache line).

The implementation of burst cycles is via the BRDY# pin. While running a bus cycle of more than one data transfer, the Pentium processor requires that the memory system perform a burst transfer and follow the burst order (see Table 6-12). Given the first address in the burst sequence, the address of subsequent transfers must be calculated by external hardware. This requirement exists because the Pentium processor address and byte-enables are asserted for the first transfer and are not re-driven for each transfer. The burst sequence is optimized for two

intel®

bank memory subsystems and is shown in Table 6-12. The addresses are in hexadecimal form.

Table 6-12. Pentium™ Processor Burst Order

1st Address	2nd Address	3rd Address	4th Address
0	8	10	18
8	0	18	10
10	18	0	8
18	10	8	0

The cycle length is driven by the Pentium processor together with cycle specification (see Table 6-10), and the system should latch this information and terminate the cycle on time with the appropriate number of transfers. The fastest burst cycle possible requires 2 clocks for the first data item to be returned/driven with subsequent data items returned/driven every clock.

6.3.2.1.   BURST READ CYCLES

When initiating any read, the Pentium processor will present the address and byte enables for the data item requested. When the cycle is converted into a cache line fill, the first data item returned should correspond to the address sent out by the Pentium processor, however, the byte enables should be ignored, and valid data must be returned on all 64 data lines. In addition, the address of the subsequent transfers in the burst sequence must be calculated by external hardware since the address and byte enables are not re-driven for each transfer.

Figure 6-8 shows a cacheable burst read cycle. Note that in this case the initial cycle generated by the Pentium processor might have been satisfied by a single data transfer, but was transformed into a multiple-transfer cache fill by KEN# being returned active on the clock that the first BRDY# is returned. In this case KEN# has such an effect because the cycle is internally cacheable in the Pentium processor (CACHE# pin is driven active). KEN# is only sampled once during a cycle to determine cacheability.

PCHK# is driven with the parity check status two clocks after each BRDY#.
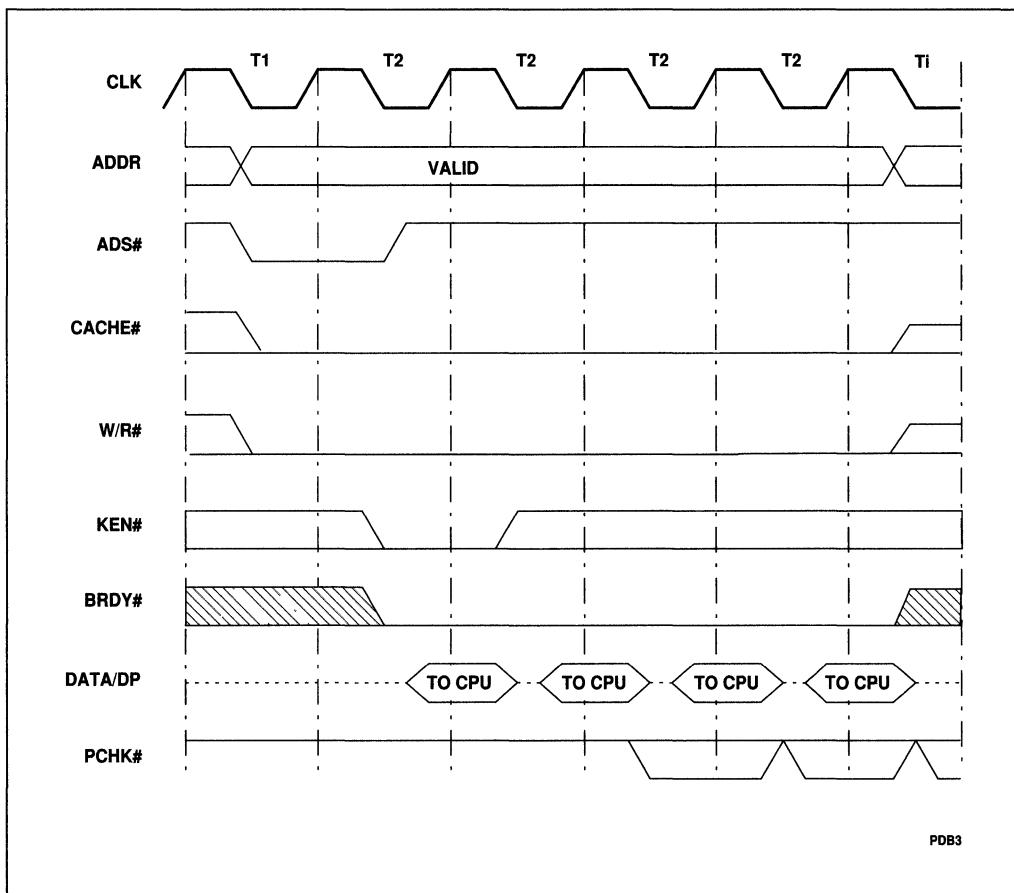
Figure 6-8. Basic Burst Read Cycle

Data will be sampled only in the clock that BRDY# is returned, which means that data need not be sent to Pentium processor every clock in the burst cycle. An example burst cycle where two clocks are required for every burst item is shown in Figure 6-9.
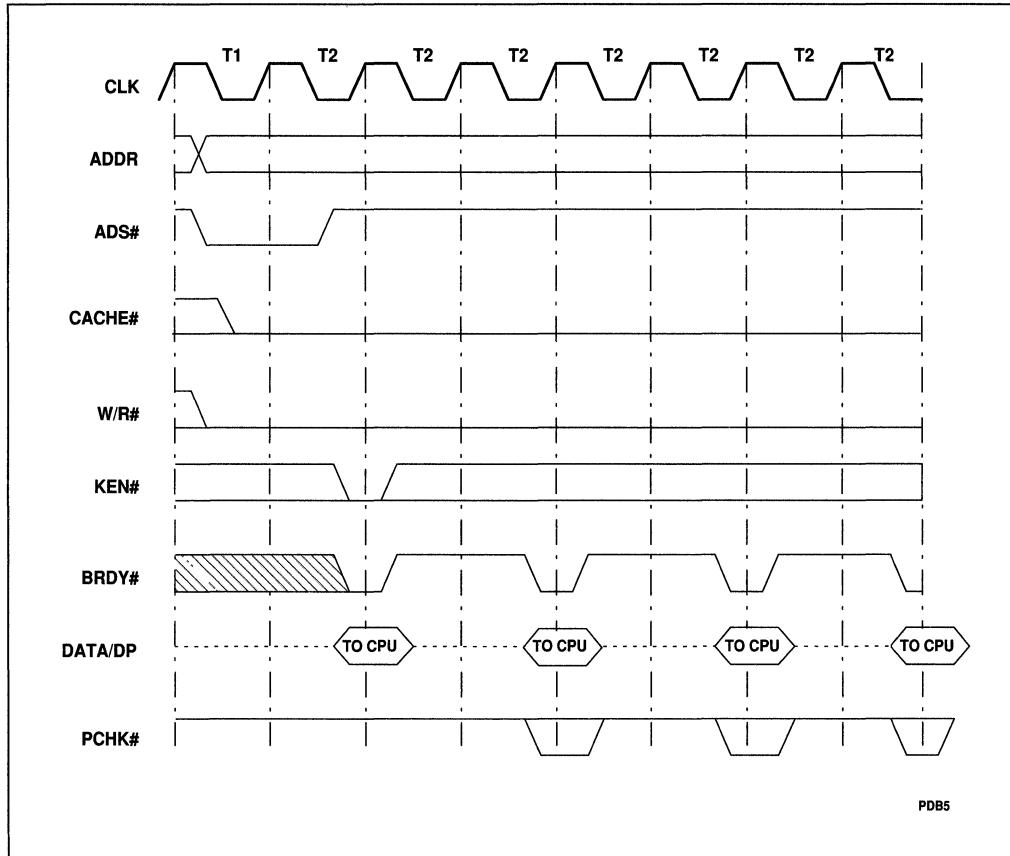
Figure 6-9. Slow Burst Read Cycle

6.3.2.2.   BURST WRITE CYCLES

Figure 6-10 shows the timing diagram of basic burst write cycle. KEN# is ignored in burst write cycle. If the CACHE# pin is active (low) during a write cycle, it indicates that the cycle will be a burst writeback cycle. Burst write cycles are always writebacks of modified lines in the data cache. Writeback cycles have several causes:

1.   Writeback due to replacement of a modified line in the data cache,

2.   Writeback due to an inquire cycle that hits a modified line in the data cache,

3.   Writeback due to an internal snoop that hits a modified line in the data cache,

4. Writebacks caused by asserting the FLUSH# pin,

5. Writebacks caused by executing the WBINVD instruction.

Writeback cycles are described in more detail in the Inquire Cycle section of this chapter.

The only write cycles that are burstable by the Pentium processor are writeback cycles. All other write cycles will be 64-bit or less, single transfer bus cycles.
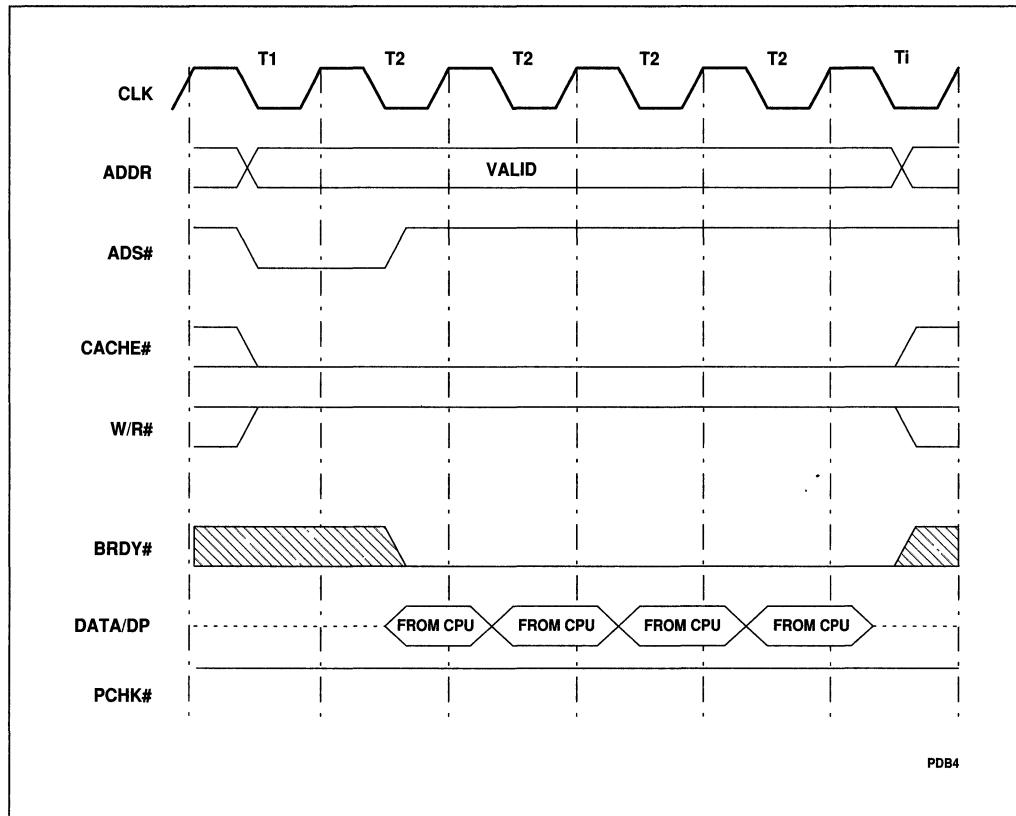
Figure 6-10. Basic Burst Write Cycle

For writeback cycles, the lower 5 bits of the first burst address always starts at 0 therefore the burst order becomes 0, 8h, 10h, and 18h. Again, note that the address of the subsequent transfers in the burst sequence must be calculated by external hardware since the Pentium processor does not drive the address and byte enables for each transfer.

6.3.3.   Locked Operations

The Pentium processor architecture provides a facility to perform atomic accesses of memory. For example, a programmer can change the contents of a memory-based variable and be assured that the variable was not accessed by another bus master between the read of the

variable and the update of that variable. This functionality is provided for select instructions using a LOCK prefix, and also for instructions which implicitly perform locked read modify write cycles such as the XCHG (exchange) instruction when one of its operands is memory based. Locked cycles are also generated when a segment descriptor or page table entry is updated and during interrupt acknowledge cycles.

In hardware, the LOCK functionality is implemented through the LOCK# pin, which indicates to the outside world that the Pentium processor is performing a read-modify-write sequence of cycles, and that the Pentium processor should be allowed atomic access for the location that was accessed with the first locked cycle. Locked operations begin with a read cycle and end with a write cycle. Note that the data width read is not necessarily the data width written. For example, for descriptor access bit updates the Pentium processor fetches 8 bytes and writes one byte.

A locked operation is a combination of one or multiple read cycles followed by one or multiple write cycles. Programmer generated locked cycles and locked page table/directory accesses are treated differently and are described in the following sections.

6.3.3.1.   PROGRAMMER GENERATED LOCKS AND SEGMENT DESCRIPTOR UPDATES

For programmer generated locked operations and for segment descriptor updates, the sequence of events is determined by whether or not the accessed line is in the internal cache and what state that line is in.

6.3.3.1.1.   Cached Lines in the Modified (M) State

Before a programmer initiated locked cycle or a segment descriptor update is generated, the Pentium processor first checks if the line is in the Modified (M) state. If it is, the Pentium processor drives an unlocked writeback first (leaving the line in the Invalid, I, state) and then runs the locked read on the external bus. Since the operand may be misaligned, it is possible that the Pentium processor may do two writeback cycles before starting the first locked read. In the misaligned scenario the sequence of bus cycles is: writeback, writeback, locked read, locked read, locked write, then the last locked write. Note that although a total of six cycles are generated, the LOCK# pin is active only during the last four cycles. In addition, the SCYC pin is asserted during the last four cycles to indicate that a misaligned lock cycle is occurring. In the aligned scenario the sequence of cycles is writeback, locked read, locked write. The LOCK# pin is asserted for the last two cycles (SCYC is not asserted and indicates that the locked cycle is aligned). The cache line is left in the Invalid state after the locked operation.

6.3.3.1.2.   Non-cached (I-State), S-State and E-State Lines

A programmer initiated locked cycle or a segment descriptor update to a S, E or I-state line is always forced out to the bus and the line is transitioned to the Invalid state. Since the line is not in the M-State, no writeback is necessary. Because the line is transitioned to the Invalid state, the locked write is forced out to the bus also. The cache line is left in the Invalid state after the locked operation.

6.3.3.2. PAGE TABLE/DIRECTORY LOCKED CYCLES

In addition to programmer generated locked operations, the Pentium processor performs locked operations to set the dirty and accessed bits in page tables/page directories. The Pentium processor runs the following sequence of bus cycles to set the dirty/accessed bit.

6.3.3.2.1. Cached Lines in the Modified (M) State

If there is a TLB miss, the Pentium processor issues an (unlocked) read cycle to determine if the dirty or accessed bits need to be set. If the line is modified in the internal data cache, the line is written back to memory (lock not asserted). If the dirty or accessed bits need to be set, the Pentium processor then issues a locked read modify write operation. The sequence of bus cycles to set the dirty or accessed bits in a page table/directory when the line is in the M-state is: unlocked read, unlocked writeback, locked read, then locked write. The line is left in the Invalid state after the locked operation. Note that accesses to the page tables/directories will not be misaligned.

6.3.3.2.2. Non-cached (I-State), S-State and E-State Lines

If the line is in the E,S or I state, the locked cycle is always forced out to the bus and the line is transitioned to the Invalid state. The sequence of bus cycles for an internally generated locked operation is locked read, locked write. The line is left in the Invalid state. Note that accesses to the page tables/directories will not be misaligned.

6.3.3.3. LOCK# OPERATION DURING AHOLD/HOLD/BOFF#

LOCK# is not deasserted if AHOLD is asserted in the middle of a locked cycle.

LOCK# is floated during bus HOLD, but if HOLD is asserted during a sequence of locked cycles, HLDA will not be asserted until the locked sequence is complete.

LOCK# will float if BOFF# is asserted in the middle of a locked cycle, and is driven low again when the cycle is restarted. If BOFF# is asserted during the read cycle of a locked read-modify write, the locked cycle is redriven from the read when BOFF# is deasserted. If BOFF# is asserted during the write cycle of a locked read-modify write, only the write cycle is redriven when BOFF# is deasserted. The system is responsible for ensuring that other bus masters do not access the operand being locked if BOFF# is asserted during a LOCKed cycle.

6.3.3.4. INQUIRE CYCLES DURING LOCK#

This section describes the Pentium processor bus cycles that will occur if an inquire cycle is driven while LOCK# is asserted. Note that inquire cycles are only recognized if AHOLD, BOFF# or HLDA is asserted and the external system returns an external snoop address to the Pentium processor. If AHOLD, BOFF# or HLDA is not asserted when EADS# is driven, EADS# is ignored. Note also that an inquire cycle can not hit the "locked line" because the LOCK cycle invalidated it.

Because HOLD is not acknowledged when LOCK# is asserted, inquire cycles run in conjunction with the assertion of HOLD can not be driven until LOCK# is deasserted and

HLDA is asserted.

BOFF# takes priority over LOCK#. Inquire cycles are permitted while BOFF# is asserted. If an inquire cycle hits a modified line in the data cache, the writeback due to the snoop hit will be driven before the locked cycle is re-driven. LOCK# will be asserted for the writeback.

An inquire cycle with AHOLD may be run concurrently with a locked cycle. If the inquire cycle hits a modified line in the data cache, the writeback may be driven between the locked read and the locked write. If the writeback is driven between the locked read and write, LOCK# will be asserted for the writeback.

NOTE

> Only writebacks due to an external snoop hit to a modified line may be driven between the locked read and the locked write of a LOCKed sequence. No other writebacks (due to an internal snoop hit or data cache replacement) are allowed to invade a LOCKed sequence.

6.3.3.5.   LOCK# TIMING AND LATENCY

The timing of LOCK# is shown in Figure 6-11. Note that ADS# is asserted with the ADS# of the read cycle and remains active until the BRDY# of the write cycle is returned. Figure 6-12 shows an example of two consecutive locked operations. Note that the Pentium processor automatically inserts at least one idle clock between two *consecutive* locked operations to allow the LOCK# pin to be sampled inactive by external hardware. Figure 6-13 shows an example of a misaligned locked operation with SCYC asserted.

The maximum number of Pentium processor initiated cycles that will be locked together is four. Four cycles are locked together when data is misaligned for programmer generated locks (read, read, write, write). SCYC will be asserted for misaligned locked cycles. Note that accesses to the page tables/directories will not be misaligned.

![intel®]

BUS FUNCTIONAL DESCRIPTION
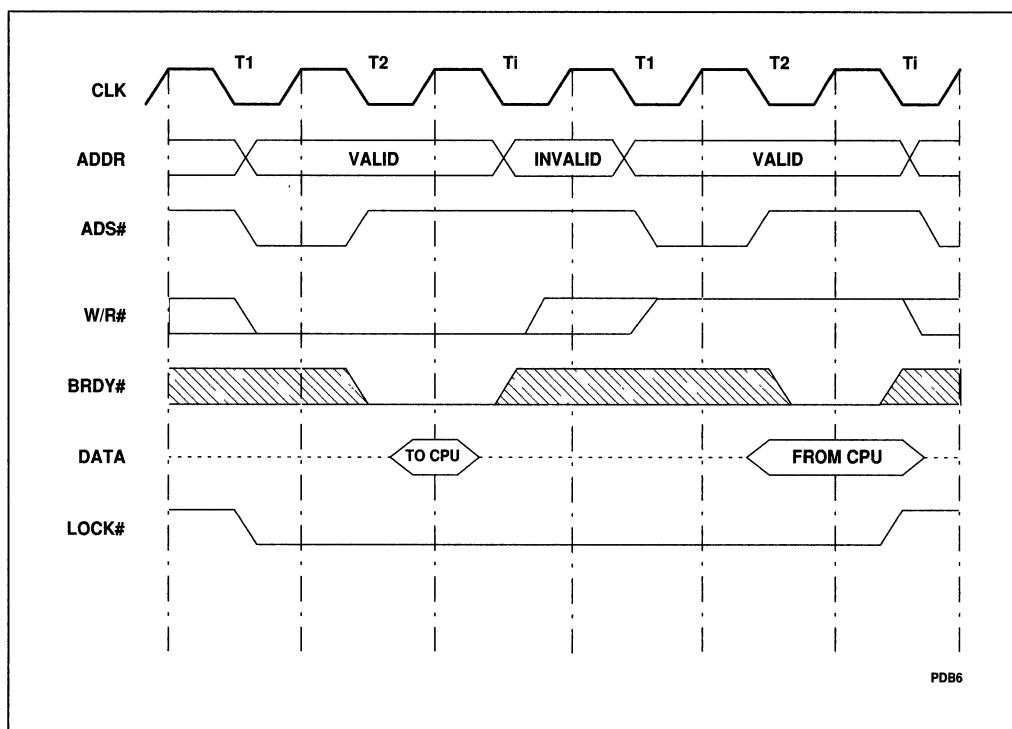
Figure 6-11. LOCK# Timing

6-21
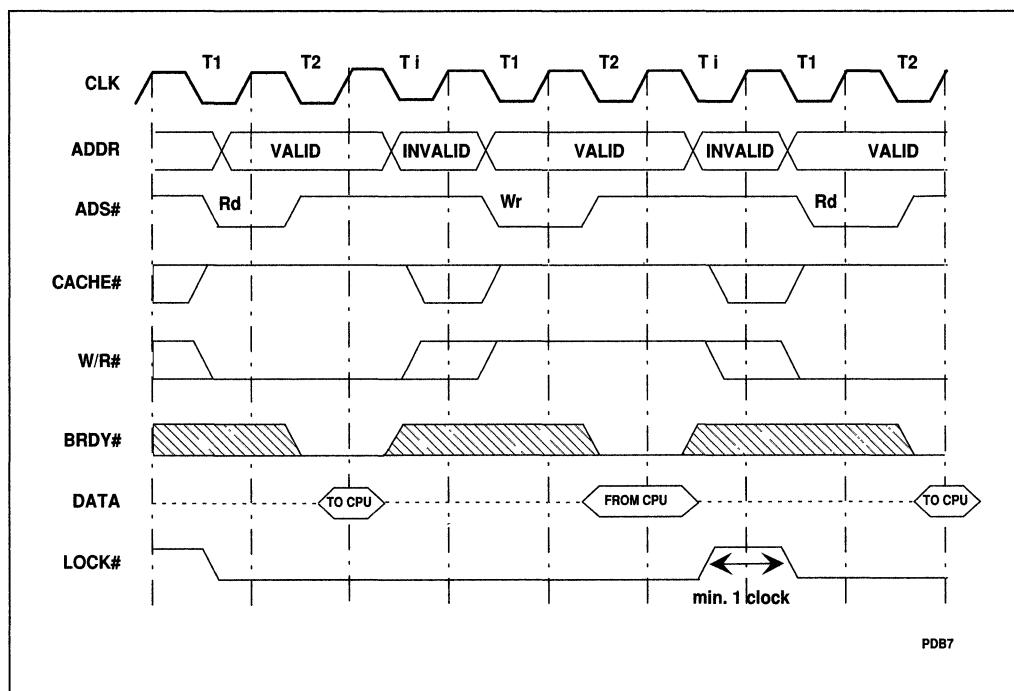
Figure 6-12. Two Consecutive Locked Operations
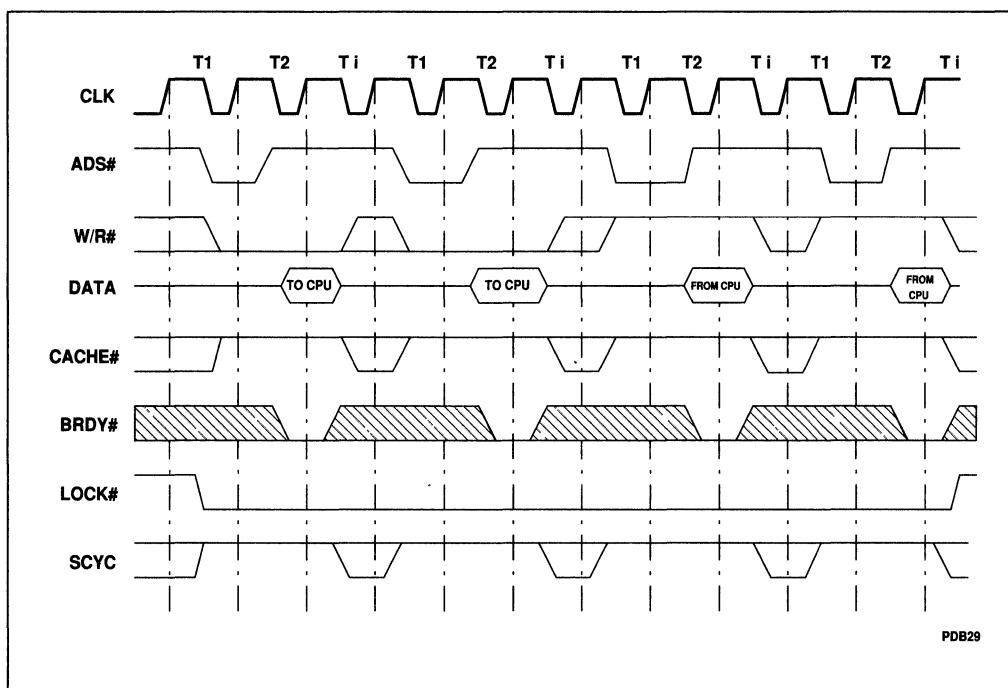
Figure 6-13. Misaligned Locked Cycles

6.3.4. BOFF#

In a multi-master system, another bus master may require the use of the bus to enable the Pentium processor to complete its current cycle. The BOFF# pin is provided to prevent this deadlock situation. If BOFF# is asserted, the Pentium processor will immediately (in the next clock) float the bus (see Figure 6-14). Any bus cycles in progress are aborted and any data returned to the processor in the clock BOFF# is asserted is ignored. In response to BOFF#, the Pentium processor floats the same pins as HOLD, but HLDA is not asserted. BOFF# overrides BRDY#, so if both are sampled active in the same clock, BRDY# is ignored. The Pentium processor samples the BOFF# pin every clock.
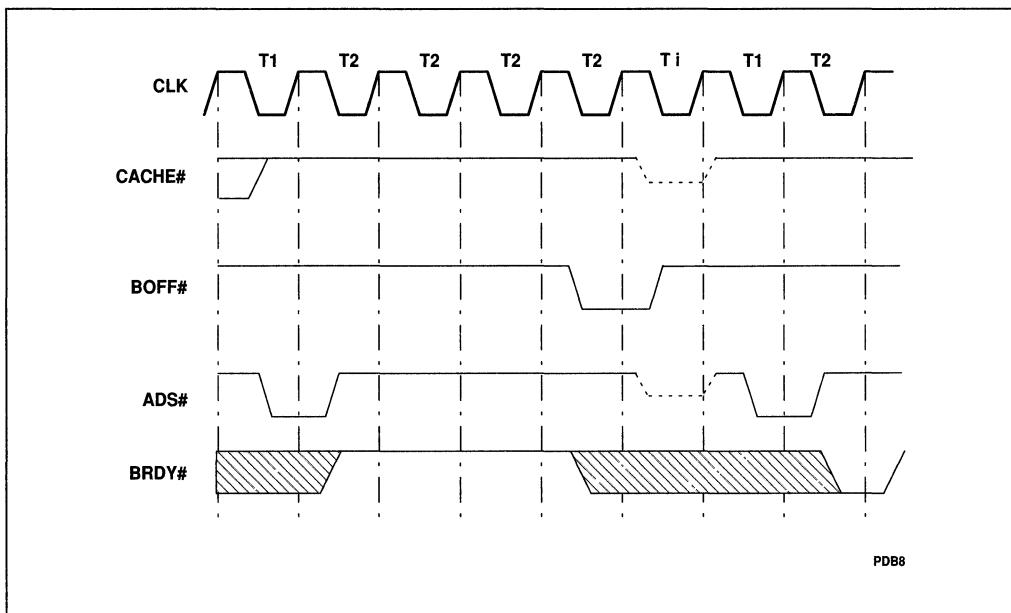
Figure 6-14. Back Off Timing

The device that asserts BOFF# to the Pentium processor is free to run any bus cycle while the Pentium processor is in the high impedance state. If BOFF# is asserted after the Pentium processor has started a cycle, the new master should wait for memory to return BRDY# before driving a cycle. Waiting for BRDY# provides a handshake to insure that the memory system is ready to accept a new cycle. If the bus is idle when BOFF# is asserted, the new master can start its cycle two clocks after issuing BOFF#. The system must wait two clocks after the assertion of BOFF# to begin its cycle to prevent address bus contention.

The bus remains in the high impedance state until BOFF# is negated. At that time, the Pentium processor restarts all aborted bus cycles from the beginning by driving out the address and status and asserting ADS#. Any data returned before BOFF# was asserted is used to continue internal execution, however that data is not placed in an internal cache. Any aborted bus cycles will be restarted from the beginning.

External hardware should assure that if the cycle attribute KEN# was returned to the processor (with first BRDY# or NA#) before the cycle was aborted, it must be returned with the same value after the cycle is restarted. In other words, backoff cannot be used to change the cacheability property of the cycle. The WB/WT# attribute may be changed when the cycle is restarted.

If more than one cycle is outstanding when BOFF# is asserted, the Pentium processor will restart both outstanding cycles in their original order. The cycles will not be pipelined unless NA# is asserted appropriately.

A pending writeback cycle due to an external snoop hit will be reordered in front of any cycles aborted due to BOFF#. For example, if a snoop cycle is run concurrently with a line fill, and the snoop hits an M state line and then BOFF# is asserted, the writeback cycle due to the snoop will be driven from the Pentium processor before the cache line fill cycle is restarted.

The system must not rely on the original cycle, that was aborted due to BOFF#, from restarting immediately after BOFF# is deasserted. In addition to reordering writebacks due to external snoop hit in front of cycles that encounter a BOFF#, the processor may also reorder bus cycles in the following situations:

1. A pending writeback cycle due to an internal snoop hit will be reordered in front of any cycles aborted due to BOFF#. If a read cycle is running on the bus, and an internal snoop of that read cycle hits a modified line in the data cache, and the system asserts BOFF#, the Pentium processor will drive out a writeback cycle resulting from the internal snoop hit. After completion of the writeback cycle, the processor will then restart the original read cycle. This circumstance can occur during accesses to the page tables/directories, and during prefetch cycles, since these accesses cause a bus cycle to be generated before the internal snoop to the data cache is performed.

2. If BOFF# is asserted during a data cache replacement writeback cycle, the writeback cycle will be aborted and then restarted once BOFF# is deasserted. However, during the BOFF#, if the processor encounters a request to access the page table/directory in memory, this request will be reordered in front of the replacement writeback cycle that was aborted due to BOFF#. The Pentium processor will first run the sequence of bus cycles to service the page table/directory access and then restart the original replacement writeback cycle.

Asserting BOFF# in the same clock as ADS# may cause the Pentium processor to leave the ADS# signal floating low. Since ADS# is floating low, a peripheral device may think that a new bus cycle has begun even though the cycle was aborted. There are several ways to approach this situation:

1. Design the system's state machines/logic such that ADS# is not recognized the clock after ADS# is sampled active.

2. Recognize a cycle as ADS# asserted and BOFF# negated in the previous clock.

3. Assert AHOLD one clock before asserting BOFF#.

6.3.5. Bus Hold

The Pentium processor provides a bus hold, hold acknowledge protocol using the HOLD and HLDA pins. HOLD is used to indicate to the Pentium processor that another bus master wants control of the bus. When the Pentium processor completes all outstanding bus cycles, it will release the bus by floating its external bus, and drive HLDA active. An example HOLD/HLDA transaction is shown in Figure 6-15.

intel®
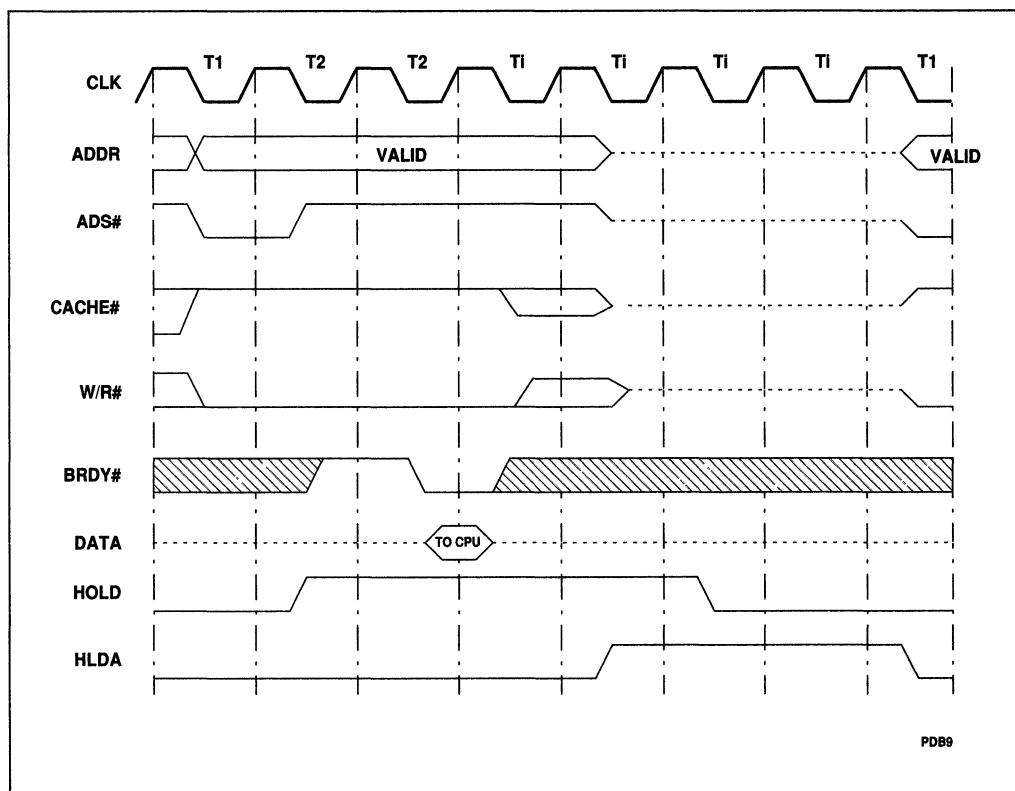
Figure 6-15. HOLD/HLDA Cycles

The Pentium processor recognizes HOLD while RESET is asserted, when BOFF# is asserted, during Probe Mode, and during BIST (built in self test). HOLD is not recognized when LOCK# is asserted. Once HOLD is recognized, HLDA will be asserted two clocks after the later of the last BRDY# or HOLD assertion. Because of this, it is possible that a cycle may begin after HOLD is asserted, but before HLDA is driven. The maximum number of cycles that will be driven after HOLD is asserted is one. BOFF# may be used if it is necessary to force the Pentium processor to float its bus in the next clock. Figure 6-15 shows the latest HOLD may be asserted relative to ADS# to guarantee that HLDA will be asserted before another cycle is begun.

The operation of HLDA is not affected by the assertion of BOFF#. If HOLD is asserted while BOFF# is asserted, HLDA will be asserted two clocks later. If HOLD goes inactive while BOFF# is asserted, HLDA is deasserted two clocks later.

Note that HOLD may be acknowledged between two bus cycles in a misaligned access.

All outputs are floated when HLDA is asserted except: APCHK#, BREQ, FERR#, HIT#, HITM#, HLDA, IERR#, PCHK#, PRDY, BP3-2, PM1/BP1, PM0/BP0, SMIACT#, IU, IV, IBT and TDO.

6.3.6. Interrupt Acknowledge

The Pentium processor generates interrupt acknowledge cycles in response to maskable interrupt requests generated on the interrupt request input (INTR) pin (if interrupts are enabled). Interrupt acknowledge cycles have a unique cycle type generated on the cycle type pins.

An example interrupt acknowledge transaction is shown in Figure 6-16. Interrupt acknowledge cycles are generated in locked pairs. Data returned during the first cycle is ignored, however the specified data setup and hold times must be met. The interrupt vector is returned during the second cycle on the lower 8 bits of the data bus. The Pentium processor has 256 possible interrupt vectors.

The state of address bit 2 (as decoded from the byte enables) distinguishes the first and second interrupt acknowledge cycles. The byte address driven during the first interrupt acknowledge cycle is 4: (A31-A3) low, BE4# low, BE7# - BE5# high, and BE3# - BE0# high. The address driven during the second interrupt acknowledge cycle is 0 (A31-A3 low, BE0# low, and BE7# - BE1# high).

Interrupt acknowledge cycles are terminated when the external system returns BRDY#. Wait states can be added by withholding BRDY#. The Pentium processor automatically generates at least one idle clock between the first and second cycles, however the external system is responsible for interrupt controller (8259A) recovery.
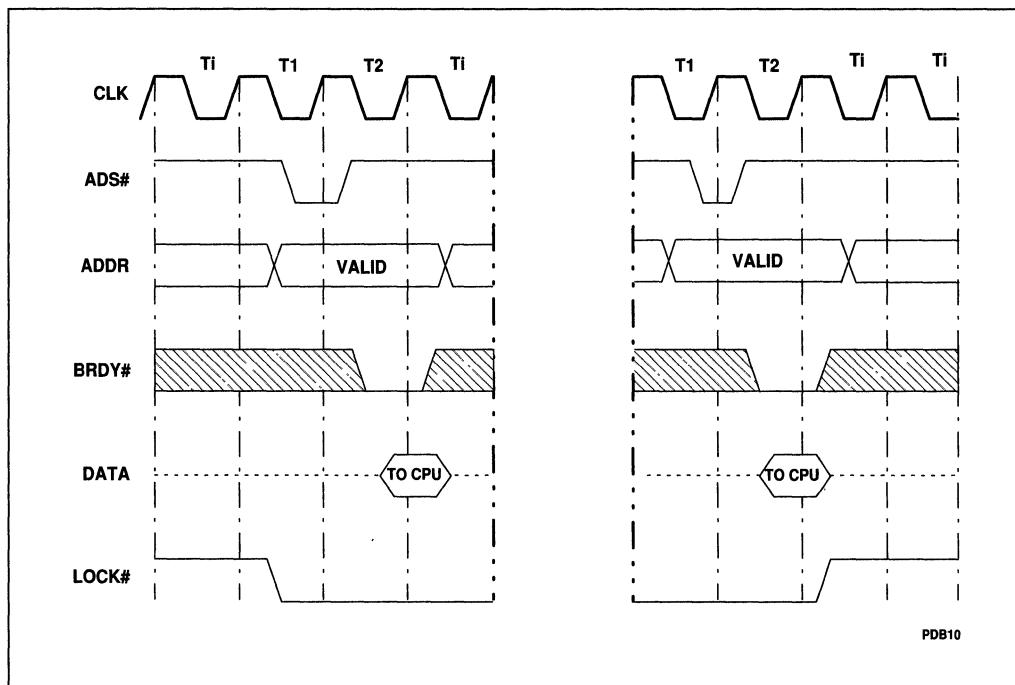
Figure 6-16. Interrupt Acknowledge Cycles

6.3.7. Flush Operations

The FLUSH# input is implemented in the Pentium processor as an asynchronous interrupt, similar to NMI. Therefore, unlike the Intel486 microprocessor, FLUSH# is recognized on instruction boundaries only. FLUSH# is latched internally, so once setup, hold and pulse width times have been met, FLUSH# may be deasserted, even if a bus cycle is in progress.

To execute a flush operation, the Pentium processor first writes back all modified lines to external memory. The lines in the internal caches are invalidated as they are written back. After the write-back and invalidation operations are complete, a special cycle, flush acknowledge, is generated by the Pentium processor to inform the external system.

6.3.8. Special Bus Cycles

The Pentium processor provides six special bus cycles to indicate that certain instructions have been executed, or certain conditions have occurred internally. The special bus cycles in Table 6-13 are defined when the bus cycle definition pins are in the following state: M/IO# = 0, D/C# = 0 and W/R# = 1. During the special cycles the data bus is undefined and the address lines A31-A3 are driven to "0". The external hardware must acknowledge all special bus cycles by returning BRDY#.

Table 6-13. Special Bus Cycles Encoding

BE7#	BE6#	BE5#	BE4#	BE3#	BE2#	BE1#	BE0#	Special Bus Cycle
1	1	1	1	1	1	1	0	Shutdown
1	1	1	1	1	1	0	1	Flush (INVD,WBINVD instr)
1	1	1	1	1	0	1	1	Halt
1	1	1	1	0	1	1	1	Write Back (WBINVD instruction)
1	1	1	0	1	1	1	1	Flush Acknowledge (FLUSH# assertion)
1	1	0	1	1	1	1	1	Branch Trace Message

Shutdown can be generated due to the following reasons: 1) if any other exception occurs while the Pentium processor is attempting to invoke the double-fault handler, or 2) an internal parity error is detected. Prior to going into shutdown, the Pentium processor will not write back the M-state lines. During shutdown, the internal caches remain in the same state unless an inquire cycle is run or the cache is flushed. The FLUSH#, SMI#, and R/S# pins are recognized while the Pentium processor is in a shutdown state. The Pentium processor will remain in shutdown until NMI, INIT, or RESET is asserted.

The Flush Special Cycle is driven after the INVD (invalidate cache) or WBINVD (write back invalidate cache) instructions are executed. The Flush Special Cycle is driven to indicate to the external system that the internal caches were invalidated and that external caches should also be invalidated. Note that the Flush Special Cycle does not imply that the modified lines in the data cache were written back.

The Halt Special Cycle is driven when a HLT instruction is executed. Externally, halt differs from shutdown in only two ways: 1) in the resulting byte enables that are asserted, and 2) the Pentium processor will exit the HLT state if INTR is asserted and maskable interrupts are enabled in addition to the assertion of NMI, INIT or RESET.

The Write Back Special Cycle is driven after the WBINVD instruction is executed. It indicates that modified lines in the Pentium processor data cache were written back to memory or a second level cache. The Write Back Special Cycle also indicates that modified lines in external caches should be written back. After the WBINVD instruction is executed, writeback special cycle is generated, followed by the flush special cycle. Note that INTR is not recognized while the WBINVD instruction is being executed.

When the FLUSH# pin is asserted to the Pentium processor, all modified lines in the data cache are written back and all lines in the code and data caches are invalidated. The Flush Acknowledge Special Cycle is driven after the write back and invalidations are complete. The Flush Acknowledge special cycle is driven only in response to the FLUSH# pin being activated. Note that the Flush Acknowledge special cycle indicates that all modified lines were written back and all cache lines were invalidated while the Flush special cycle only indicates that all cache lines were invalidated.

The Branch Trace Message Special Cycle is part of the Pentium processor execution tracing protocol. If the execution tracing enable bit (bit 1) in TR12 is set to 1, a Branch Trace message

special cycle will be driven each time IBT is asserted, i.e. whenever a branch is taken. The Branch Trace message special cycle is the only special cycle that does not drive "0's" on the address bus, however like the other special cycles, the data bus is undefined. When the branch trace message is driven, the following is driven on the address bus:

A31-A3:   Bits 31-3 of the branch target linear address

BT2-BT0:   Bits 2-0 of the branch target linear address
           (the byte enables should not be decoded for A2-A0)

BT3:   High if the default operand size is 32-bits,
           Low if the default operand size is 16-bits

6.3.9.   Bus Error Support

Pentium processor provides basic support for bus error handling through data and address parity check. Even data parity will be generated by the processor for every enabled byte in write cycles and will be checked for all valid bytes in read cycles. The PCHK# output signals if a data parity error is encountered for reads.

Even address parity will be generated for A31-A5 during write and read cycles, and checked during inquire cycles. The APCHK# output signals if an address parity error is encountered during inquire cycles.

External hardware is free to take whatever actions are appropriate after a parity error. For example, external hardware may signal an interrupt if PCHK# or APCHK# is asserted. Please refer to the Error Detection chapter for the details.

6.3.10.   Pipelined Cycles

The NA# input indicates to the Pentium processor that it may drive another cycle before the current one is completed. Cacheability (KEN#) and cache policy (WB/WT#) indicators for the current cycle are sampled in the same clock NA# is sampled active (or the first BRDY# for that cycle, whichever comes first). Note that the WB/WT#, and KEN# inputs are sampled with the first of BRDY# or NA# even if NA# does not cause a pipelined cycle to be driven because there was no pending cycle internally or two cycles are already outstanding.

The NA# input is latched internally, so even if a cycle is not pending internally in the clock that NA# is sampled active, but becomes pending before the current cycle is complete, the pending cycle will be driven to the bus even if NA# was subsequently deasserted.

LOCK# and writeback cycles are not pipelined into other cycles and other cycles are not pipelined into them (regardless of the state of NA#). Special cycles and I/O cycles may be pipelined.

An example of burst pipelined back to back reads is shown in Figure 6-17. The assertion of NA# causes a pending cycle to be driven 2 clocks later. Note KEN# timing.

int_el ®
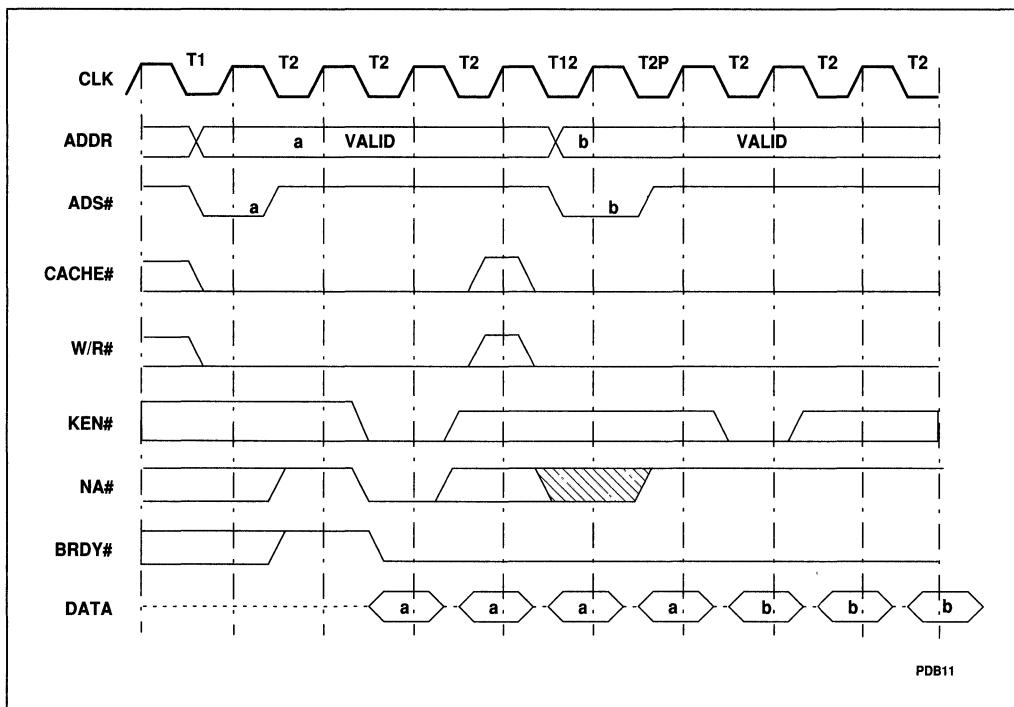
Figure 6-17. Two Pipelined Cache Line Fills

Write cycles can be pipelined into read cycles and read cycles can be pipelined into write cycles, but one dead clock will be inserted between read and write cycles to allow bus turnover (see the bus state diagram in the Bus State Definition section of this chapter). Pipelined back to back read/write cycles are shown in Figure 6-18.

intel®
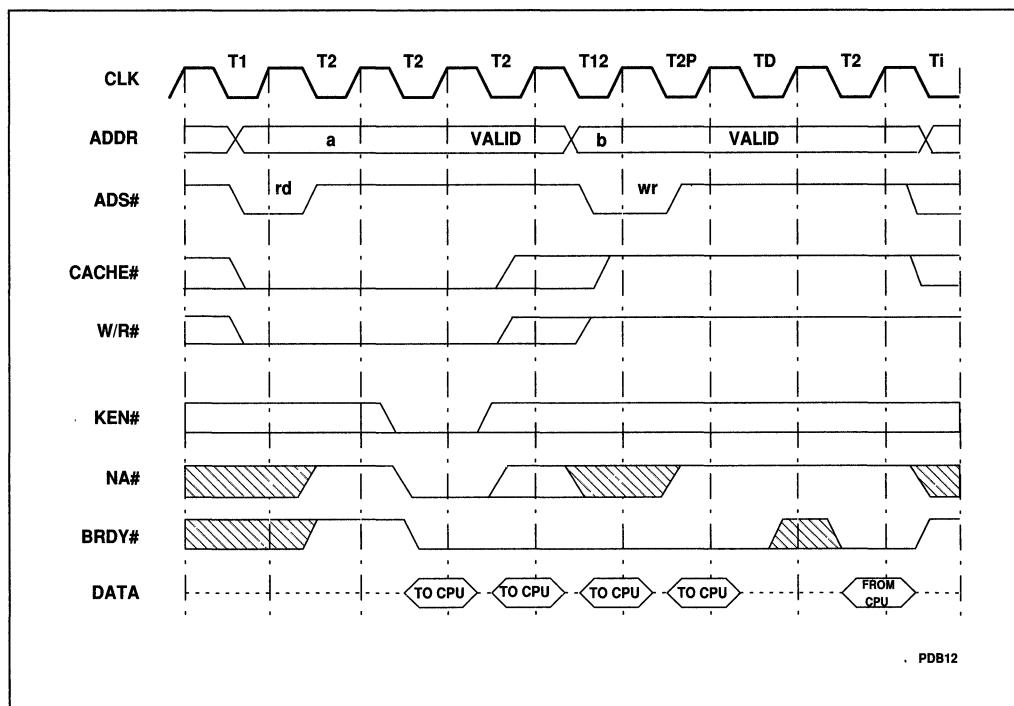
Figure 6-18. Pipelined Back-to-Back Read/Write Cycles

6.3.10.1.   KEN# AND WB/WT# SAMPLING FOR PIPELINED CYCLES

KEN# and WB/WT# are sampled with NA# or BRDY# for that cycle, whichever comes first.
Figure 6-19 and Figure 6-20 clarify this specification.
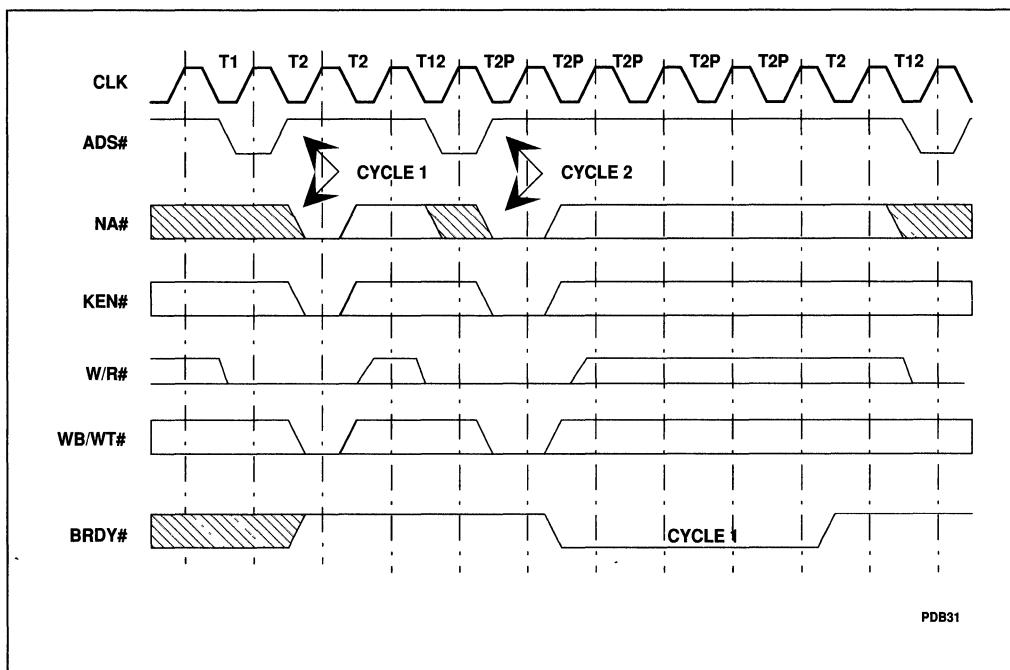
Figure 6-19. KEN# and WB/WT# Sampling with NA#

Figure 6-19 shows that even though 2 cycles have been driven, the NA# for the second cycle still causes KEN# and WB/WT# to be sampled for the second cycle. A third ADS# will not be driven until all the BRDY#s for cycle 1 have been returned to the Pentium processor.

intel®
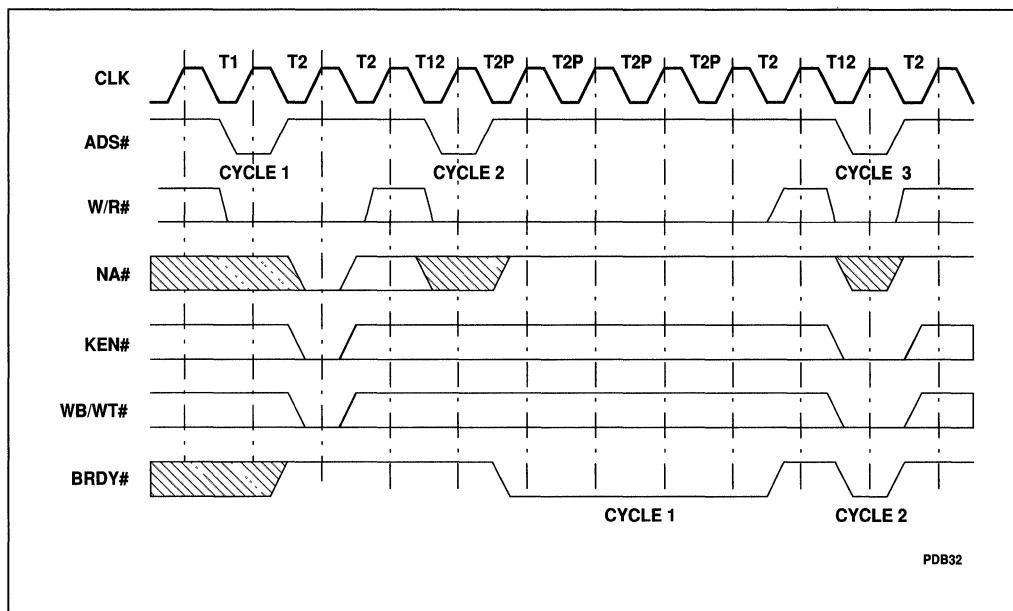
Figure 6-20. KEN# and WB/WT# Sampling with BRDY#

Figure 6-20 shows that two cycles are outstanding on the Pentium processor bus. The assertion of NA# caused the sampling of KEN# and WB/WT# for the first cycle. The assertion of the four BRDY#s for the first cycle DO NOT cause the KEN# and WB/WT# for the second cycle to be sampled. KEN# and WB/WT# for the second cycle are sampled with the first BRDY# for the second cycle (in this example).

6.4. CACHE CONSISTENCY CYCLES (INQUIRE CYCLES)

The purpose of an inquire cycle is to check whether a particular address is cached in a Pentium processor internal cache and optionally invalidate it. After an inquire cycle is complete, the system has information on whether or not a particular address location is cached and what state it is in.

An inquire cycle is typically performed by first asserting AHOLD to force the Pentium processor to float its address bus, waiting two clocks, and then driving the inquire address and INV and asserting EADS#. Inquire cycles may also be executed while the Pentium processor is forced off the bus due to HLDA, or BOFF#. Because the entire cache line is affected by an inquire cycle, only A31-A5 need to be driven with the valid inquire address. Although the value of A4-A3 is ignored, these inputs should be driven to a valid logic level during inquire cycles for circuit reasons. The INV pin is driven along with the inquire address to indicate whether the line should be invalidated (INV high) or marked as shared (INV low) in the event of an inquire hit.

After the Pentium processor determines if the inquire cycle hit a line in either internal cache, it drives the HIT# pin. HIT# is asserted (low) two clocks after EADS# is sampled asserted[1] if the inquire cycle hit a line in the code or data cache. HIT# is deasserted (high) two clocks after

EADS# is sampled asserted if the inquire cycle missed in both internal caches. The HIT# output changes its value only as a result of an inquire cycle. It retains its value between inquire cycles. In addition, the HITM# pin is asserted two clocks after EADS# if the inquire cycle hit a modified line in the data cache. HITM# is asserted to indicate to the external system that the Pentium processor contains the most current copy of the data and any device needing to read that data should wait for the Pentium processor to write it back. The HITM# output remains asserted until two clocks after the last BRDY# of the writeback cycle is asserted.

The external system must inhibit inquire cycles during BIST (initiated by INIT being sampled high on the falling edge of RESET), and during the Boundary Scan Instruction RUNBIST. When the model specific registers (test registers) are used to read or write lines directly to or from the cache it is important that external snoops (inquire cycles) are inhibited to guarantee predictable results when testing. This can be accomplished by inhibiting the snoops externally or by putting the processor in SRAM mode (CR0.CD=CR0.NW=1).

The EADS# input is ignored during external snoop writeback cycles (HITM# asserted), or during the clock after ADS# or EADS# is active. EADS# is also ignored when the processor is in SRAM mode, or when the processor is driving the address bus.

Note that the Pentium processor may drive the address bus in the clock after AHOLD is deasserted. It is the responsibility of the system designer to ensure that address bus contention does not occur. This can be accomplished by not deasserting AHOLD to the Pentium processor until all other bus masters have stopped driving the address bus.

Figure 6-21 shows an inquire cycle that misses both internal caches. Note that both the HIT# and HITM# signals are deasserted two clocks after EADS# is sampled asserted.

Figure 6-22 shows an inquire cycle that invalidates a non-modified line. Note that INV is asserted (high) in the clock that EADS# is returned. Note that two clocks after EADS# is sampled asserted, HIT# is asserted and HITM# is deasserted.

Figure 6-21 and Figure 6-22 both show that the AP pin is sampled/driven along with the address bus, and that the APCHK# pin is driven with the address parity status two clocks after EADS# is sampled asserted.

An inquire cycle that hits a M-state line is shown in Figure 6-23. Both the HIT# and HITM# outputs are asserted two clocks after EADS# is sampled asserted. ADS# for the writeback cycle will occur no earlier than two clocks after the assertion of HITM#.
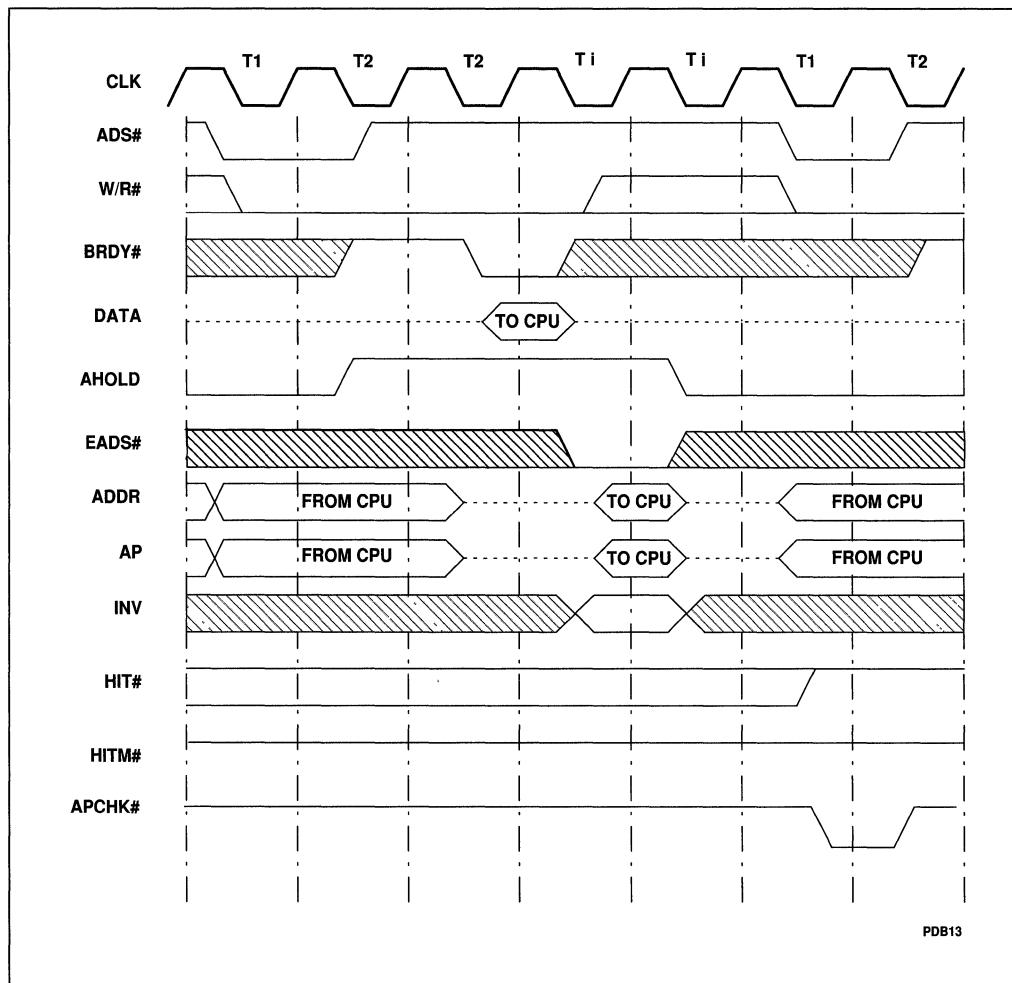
Figure 6-21. Inquire Cycle that Misses Pentium™ Processor Cache

intel®
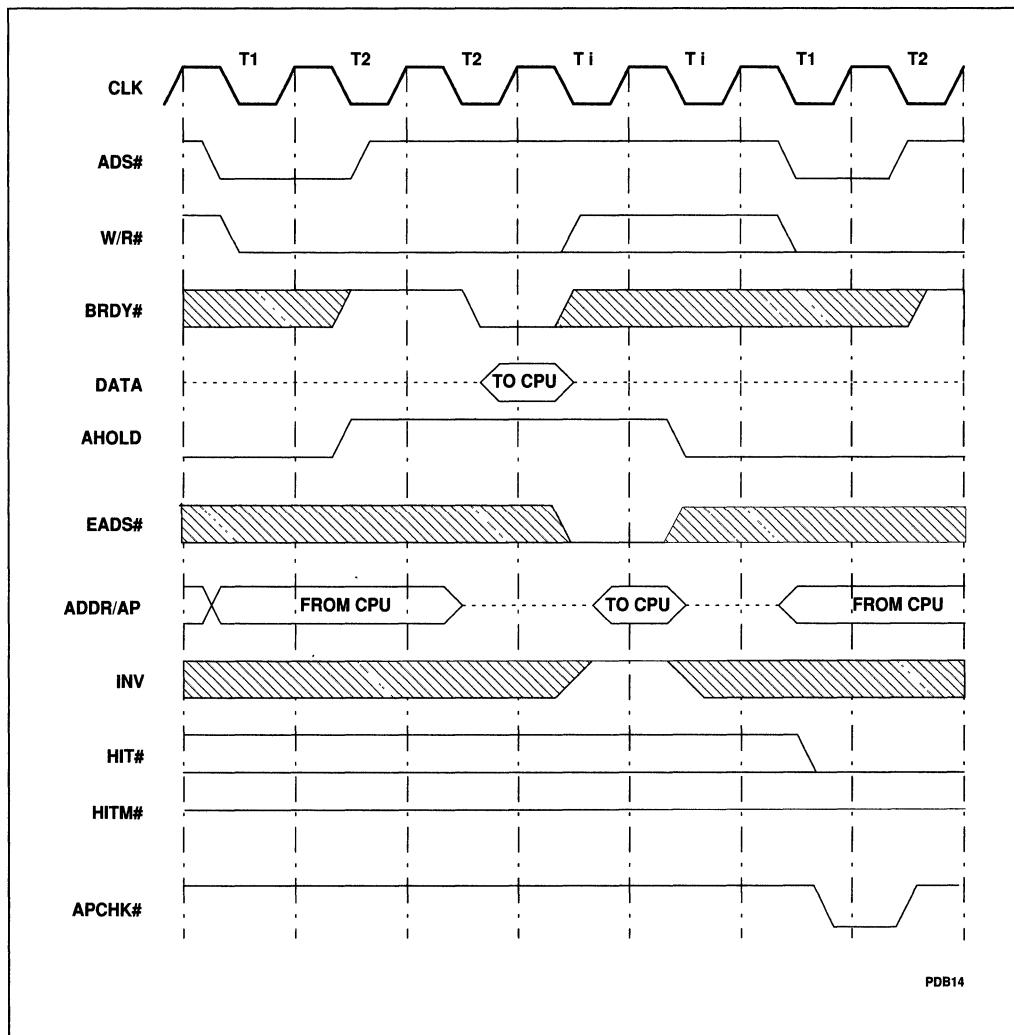
Figure 6-22. Inquire Cycle that Invalidates non-M-State Line

intel®
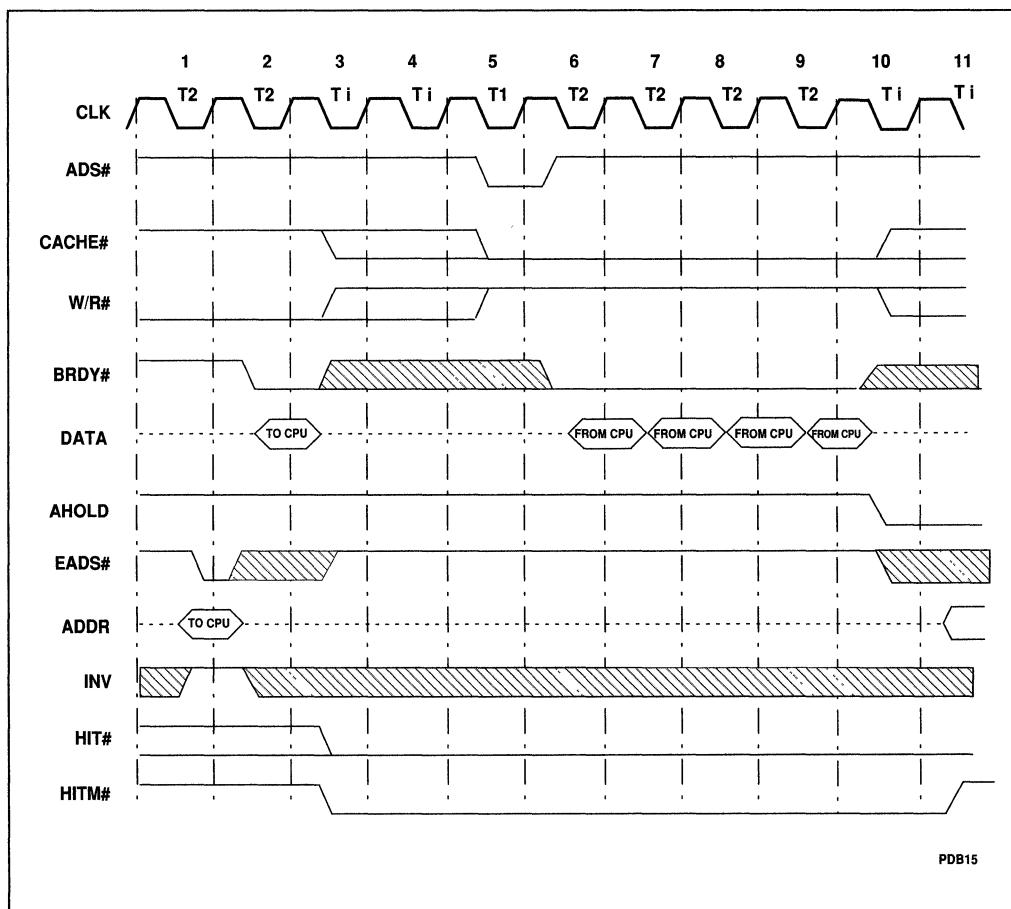
Figure 6-23. Inquire Cycle that Invalidates M-State Line

HITM# is asserted only if an inquire cycle (external snoop) hits a modified line in the Pentium processor data cache. HITM# is not asserted for internal snoop writeback cycles or cache relacement writeback cycles. HITM# informs the external system that the inquire cycle hit a modified line in the data cache and that line will be written back. Any ADS# driven by the Pentium processor while HITM# is asserted will be the ADS# of the writeback cycle. The HITM# signal will stay active until last BRDY# is returned for the corresponding inquire cycle. Writeback cycles start at burst address 0.

Note that ADS# is asserted despite the AHOLD signal being active. This ADS# initiates a writeback cycle corresponding to the inquire hit. Such a cycle can be initiated while address lines are floating to support multiple inquires within a single AHOLD session. This functionality can be used during secondary cache replacement processing if its line is larger than the Pentium processor cache line (32-bytes). Although the cycle specification is driven properly by the processor, address pins are not driven because AHOLD forces the Pentium processor off the address bus. If AHOLD is cleared before Pentium processor drives out the inquire writeback cycle, the Pentium processor will drive the correct address for inquire write-

back in the next clock. The ADS# to initiate a writeback cycle as a result of an inquire hit is the only time ADS# will be asserted while AHOLD is also asserted.

Note that in the event of an address parity error during inquire cycles, the snoop cycle will not be inhibited. If the inquire hits a modified line in this situation and an active AHOLD prevents the Pentium processor from driving the addrss bus, the Pentium processor will potentially write back a line at an address other than the one intended. If the Pentium processor is not driving the address bus during the writeback cycle, it is possible that memory will be corrupted.

If BOFF# or HLDA were asserted to perform the inquire cycle, the writeback cycle would wait until BOFF# or HLDA were deasserted.

State machines should not depend on a writeback cycle to follow an assertion of HITM#. HITM# may be negated without a corresponding writeback cycle being run. This may occur as a result of the internal caches being invalidated due to the INVD instruction or by testability accesses. Note as indicated earlier in this section that inquire cycles occurring during testability accesses will generate unpredictable results. In addition, a second writeback cycle will not be generated for an inquire cycle which hits a line that is already being written back, see Figure 6-28. This can happen if an inquire cycle hits a line in one of the Pentium processor writeback buffers.

6.4.1.   Restrictions on Deassertion of AHOLD

To prevent the address and data buses from switching simultaneously, the following restrictions are placed on the negation of AHOLD: (i) AHOLD must not be negated in the same clock as the assertion of BRDY# during a write cycle; (ii) AHOLD must not be negated in the dead clock between write cycles pipelined into read cycles; and (iii) AHOLD must not be negated in the same clock as the assertion of ADS# while HITM# is asserted. Note that there are two clocks between EADS# being sampled asserted and HITM# being asserted, and a further minimum of two clocks between an assertion of HITM# and ADS#.

These restrictions on the deassertion of AHOLD are the only considerations the system designer needs to make to prevent the simultaneous switching of the address and data buses. All other considerations are handled internally.

Figure 6-23 can be used to illustrate restrictions (i) and (iii). AHOLD may be deasserted in clock 2, 3, or 4, but not in clock 5, 6, 7, 8 or 9.

Figure 6-24 and Figure 6-25 depict restrictions (i) and (ii) respectively. Note that there are no restrictions on the assertion of AHOLD.

intel®
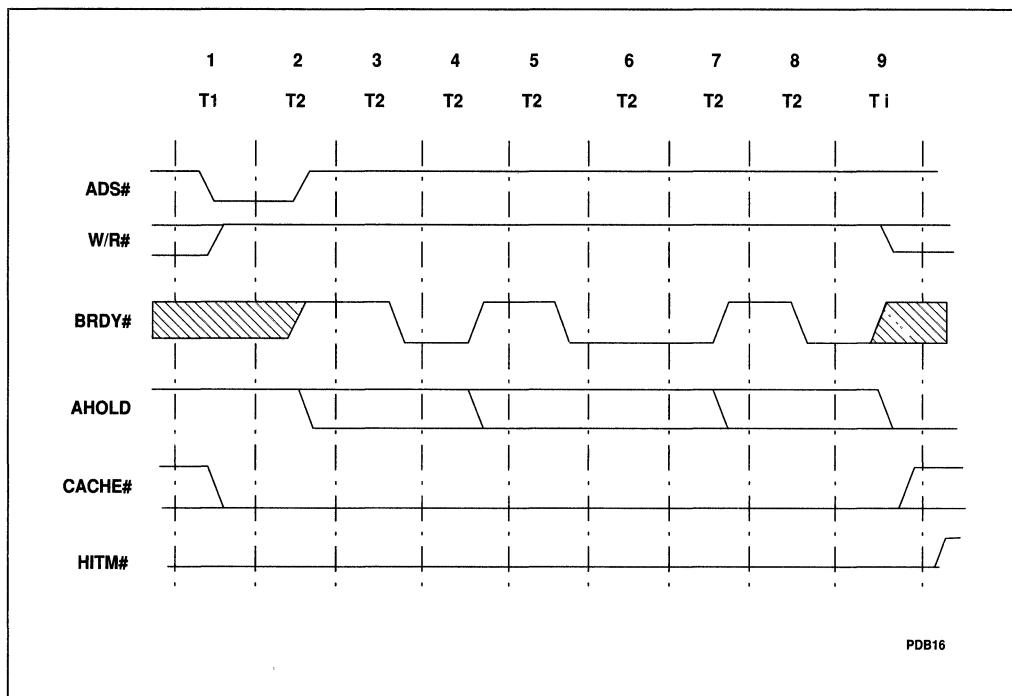
Figure 6-24. AHOLD Restriction During Write Cycles

Figure 6-24 shows a writeback (due to a previous snoop that is not shown). ADS# for the writeback is asserted even though AHOLD is asserted. Note that AHOLD can be deasserted in clock 2, 4, 7, or 9. AHOLD can not be deasserted in clock 1, 3, 5, 6, or 8.
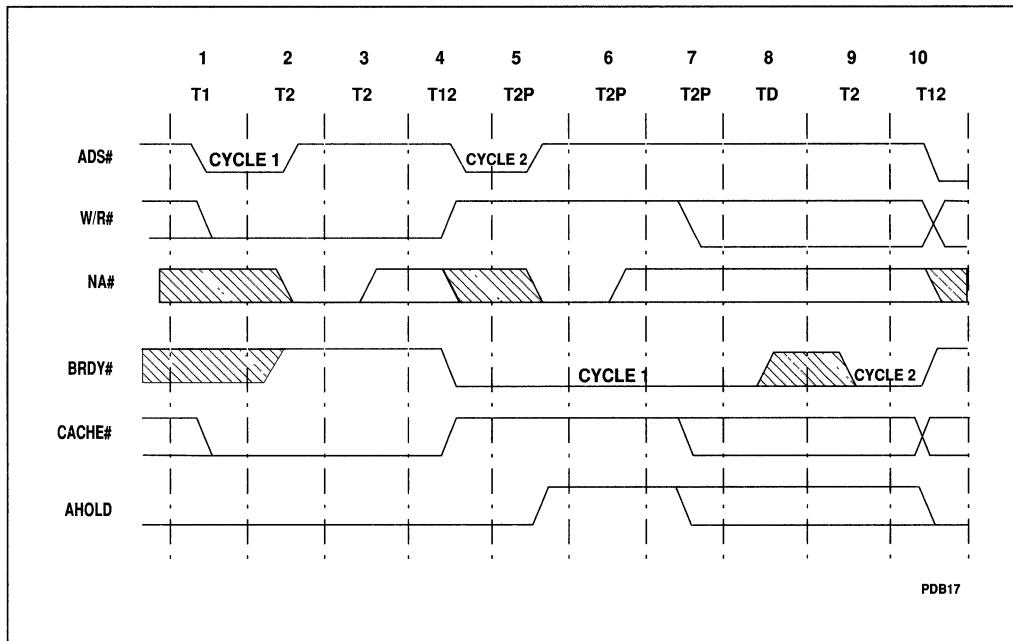
Figure 6-25. AHOLD Restriction During TD

Figure 6-25 shows a write cycle being pipelined into a read cycle. Note that if AHOLD is asserted in clock 5, it can be deasserted in clock 7 before the TD, or in clock 10 after the TD, but it can not be deasserted in clock 8 (the TD clock). AHOLD can not be deasserted in clock 9 because BRDY# for the write cycle is being returned.

6.4.2. Rate of Inquire Cycles

Pentium processor can accept inquire cycles at a maximum rate of one every other clock. However, if an inquire cycle hits an M-state line of the Pentium processor, subsequent inquire cycles will be ignored until the line is written back and HITM# is deasserted. EADS# is also ignored the clock after ADS# is asserted.

6.4.3. Internal Snooping

"Internal snoop" is the term used to describe the snooping of the internal code or data caches that is not initiated by the assertion of EADS# by the external system. Internal snooping occurs in the three cases described below. Note that neither HIT# nor HITM# are asserted as a result of an internal snoop.

1. An internal snoop occurs if an access is made to the code cache, and that access is a miss. In this case, if the accessed line is in the S or E-state in the data cache, the line is invalidated. If the accessed line is in the M-state in the data cache, the line is written back then invalidated.

2. An internal snoop occurs if an access is made to the data cache, and that access is a miss or a writethrough. In this case, if the accessed line is valid in the code cache, the line is invalidated.

3. An internal snoop occurs if a there is a write to the accessed and/or dirty bits in the page table/directory entires. In this case, if the accessed line is valid in either the code or data cache, the line is invalidated. If the accessed line is in the M-state in the data cache, the line is written back then invalidated.

6.4.4. Snooping Responsibility

In systems with external second level caches allowing concurrent activity of the memory bus and Pentium processor bus, it is desirable to run invalidate cycles concurrently with other Pentium processor bus activity. Writes on the memory bus can cause invalidations in the secondary cache at the same time that the Pentium processor fetches data from the secondary cache. Such cases can occur at any time relative to each other, and therefore the order in which the invalidation is requested, and data is returned to the Pentium processor becomes important.

The Pentium processor always snoops the instruction and data caches when it accepts an inquire cycle. If a snoop comes in during a line fill, the Pentium processor also snoops the line currently being filled. If more than one cacheable cycle is outstanding (through pipelining), the addresses of both outstanding cycles are snooped.

For example, during line fills, the Pentium processor starts snooping the address(es) associated with the line(s) being filled after KEN# has been sampled active for the line(s). Each line is snooped until it is put in the cache. If a snoop hits a line being currently filled, the Pentium processor will assert HIT# and the line will end up in the cache in the S or I state depending on the value of the INV pin sampled during the inquire cycle. The Pentium processor will

however use the data returned for that line as a memory operand for the instruction that caused the data cache miss/line fill or execute an instruction contained in a code cache miss/line fill.

Figure 6-26 and Figure 6-27 illustrate the snoop responsibility pickup. Figure 6-26 shows a non-pipelined cycle, while Figure 6-27 illustrates a pipelined cycle. The figures show the earliest EADS# assertion that will cause snooping of the line being cached relative to first BRDY# or NA#.
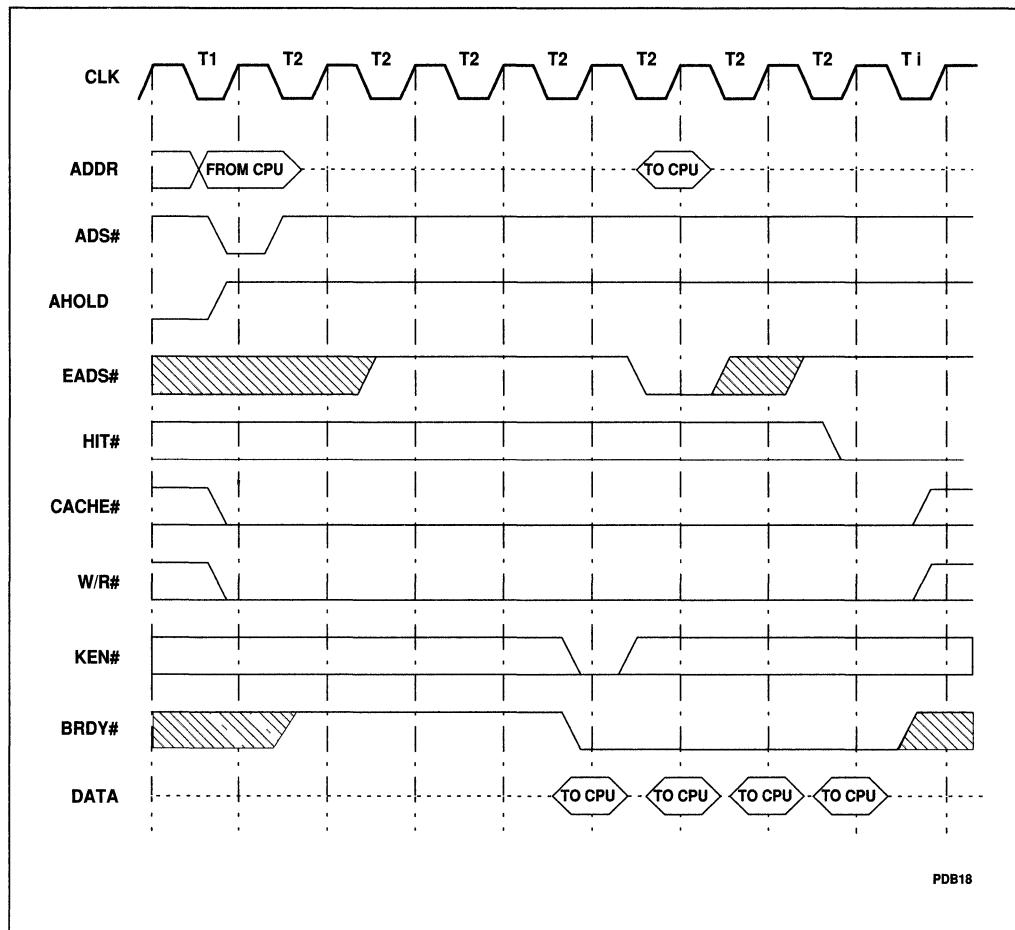
Figure 6-26. Snoop Responsibility Pickup — Non-Pipelined Cycles

intel®
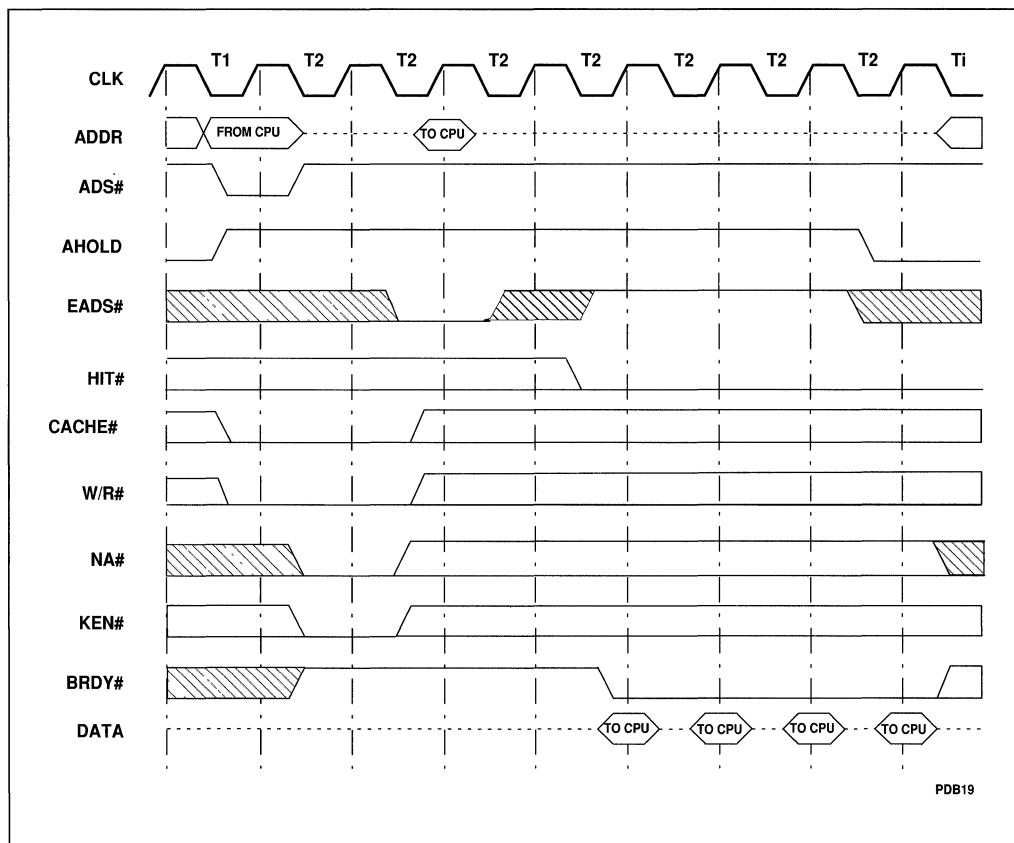
Figure 6-27. Snoop Responsibility Pickup — Pipelined Cycle

The Pentium processor also snoops M state lines in the writeback buffers until the writeback of the M state lines are complete. If a snoop hits an M state line in a writeback buffer, both HIT# and HITM# are asserted. Figure 6-28 illustrates snooping (snoop responsibility drop) of an M state line that is being written back because it has been replaced with a "new" line in the data cache. It shows the latest EADS# assertion, relative to the last BRDY# of the writeback cycle that will result in a snoop hit to the line being written back. HITM# stays asserted until the writeback is complete. Note that no additional ADS# is asserted during the writeback cycle.

The HIT# signal is a super set of the HITM# signal; it is always asserted with HITM#.
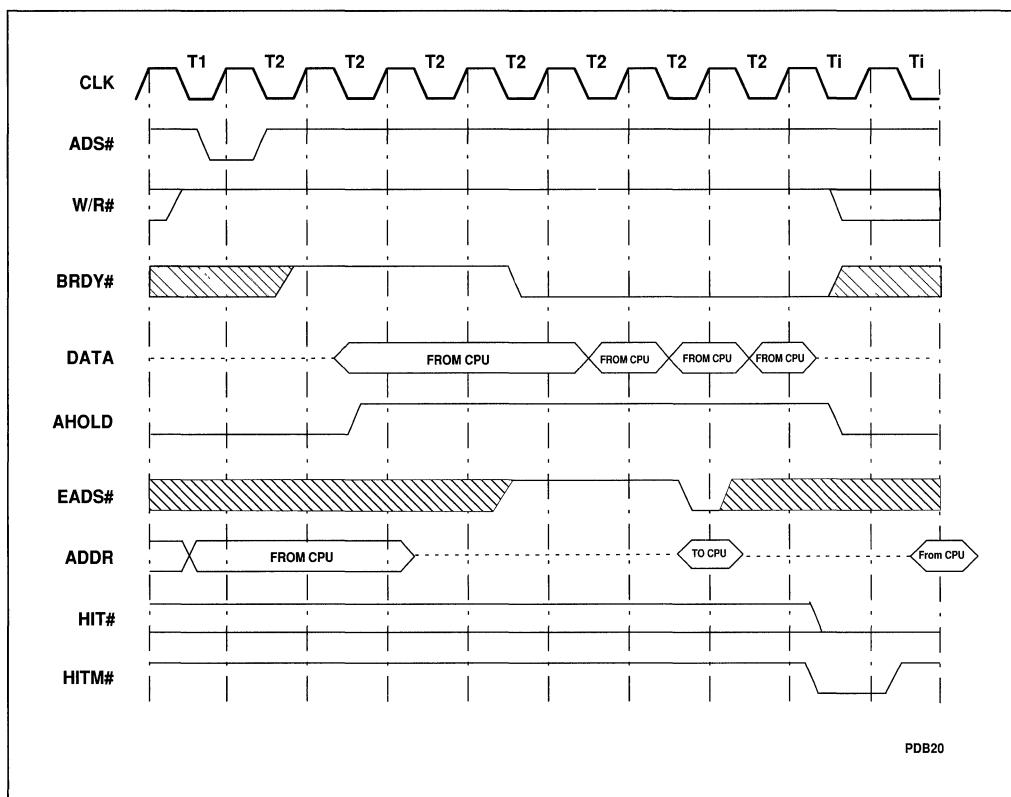
Figure 6-28. Latest Snooping of Writeback Buffer

6.5.   BUS DIFFERENCES BETWEEN THE Intel486 MICROPROCESSOR AND THE PENTIUM PROCESSOR

The Pentium processor bus is designed to be similar to the Intel486 CPU bus for ease of use. In addition, enhancements have been made to achieve higher performance and provide better support for multi-processing systems.

This section is provided as a quick reference for those designers familiar with the Intel486 microprocessor.

The following are differences between the Pentium processor and Intel486 CPU buses:

● The Pentium processor has 64-bit data bus, while the Intel486 CPU supports 32-bit data bus. The Pentium processor has more byte enables (BE7#-BE0#) and more data parity pins (DP7-0) than the Intel486 CPU to support this larger data bus size.

● The Pentium processor supports address pipelining through the NA# input to provide the capability of driving up to two cycles to the bus concurrently.

- The Pentium processor samples the cacheability input KEN# with the earlier of NA# or the first BRDY#. KEN# is sampled only once. The Intel486 CPU samples KEN# twice, the clock before the first and last RDY#/BRDY# of the cache line fill cycle.

- Burst length information is driven by the Pentium processor via the CACHE# pin together with the address. The Intel486 CPU controls burst length with the BLAST# pin.

- The Pentium processor generates 8 byte writes as one bus cycle, and therefore does not have PLOCK# pin.

- The Pentium processor does not change lower-order bits of address and byte enables during the burst.

- The Pentium processor requires write-backs and line fills to be run as burst cycles, and the burst cannot be terminated in the middle (no RDY# or BLAST# pins).

- Non-cacheable burst cycles are not supported by the Pentium processor. Non-burst cacheable cycles are not supported by the Pentium processor. On the Pentium processor, cacheable implies burst-able.

- The Pentium processor supports a write back cache protocol with the following new pins: CACHE#, HIT#, HITM#, INV and WB/WT#.

- The Pentium processor does not support the dynamic bus sizing implemented with BS8# and BS16#.

- The Pentium processor does not allow invalidations every clock, or invalidations while the Pentium processor is driving the address bus.

- The Pentium processor guarantees an idle clock between consecutive LOCKed cycles.

- The Pentium processor provides the SCYC pin which indicates a split cycle during locked operations.

- Non-cacheable code prefetches are 8 bytes for the Pentium processor, not 16 bytes.

- The Pentium processor has an INIT pin to perform the reset function while maintaining the state of the internal caches and the floating point machine state.

- The Pentium processor supports strong store ordering between the Pentium processor and the external system through the EWBE# pin.

- The Pentium processor supports internal parity error checking, enhanced data parity checking, and address parity error checking. The following new pins were added to implement these new features: APCHK#, BUSCHK#, PEN#, IERR# and AP.

- The Pentium processor includes boundary scan with the following pins: TDI, TDO, TMS, TRST#, and TCK.

- The Pentium processor has IU, IV, and IBT pins and a branch trace message special cycle to support execution tracing.

- The Pentium processor supports Functional Redundancy Checking (FRC) with the FRCMC# and IERR# pins.

- The Pentium processor supports performance monitoring and external breakpoint indications with the following pins: BP3, BP2, PM1/BP1, and PM0/BP0.

- The Pentium processor implements system management mode using the SMI# input and the SMIACT# output.

- On the Pentium processor, after a bus cycle is aborted with BOFF#, the bus cycle is restarted from the beginning. Data returned previous to BOFF# is not saved. The Intel486 CPU stores the data that was returned previous to the BOFF# assertion and restarts the cycle at the point it was aborted.

- FLUSH# is an edge triggered input. It is recognized once for every falling edge. It is implemented as an interrupt, and therefore recognized only at instruction boundaries.

6.6. BUS STATE DEFINITION

This section describes the Pentium processor bus states in detail. See Figure 6-29 for the bus state diagram.

Ti: This is the bus idle state. In this state, no bus cycles are being run. The Pentium processor may or may not be driving the address and status pins, depending on the state of the HLDA, AHOLD, and BOFF# inputs. An asserted BOFF# or RESET will always force the state machine back to this state. HLDA will only be driven in this state.

T1: This is the first clock of a bus cycle. Valid address and status are driven out and ADS# is asserted. There is one outstanding bus cycle.

T2: This is the second and subsequent clock of the first outstanding bus cycle. In state T2, data is driven out (if the cycle is a write), or data is expected (if the cycle is a read), and the BRDY# pin is sampled. There is one outstanding bus cycle.

T12: This state indicates there are two outstanding bus cycles, and that the Pentium processor is starting the second bus cycle at the same time that data is being transferred for the first. In T12, the Pentium processor drives the address and status and asserts ADS# for the second outstanding bus cycle, while data is transferred and BRDY# is sampled for the first outstanding cycle.

T2P: This state indicates there are two outstanding bus cycles, and that both are in their second and subsequent clocks. In T2P, data is being transferred and BRDY# is sampled for the first outstanding cycle. The address, status and ADS# for the second outstanding cycle were driven sometime in the past (in state T12).

TD: This state indicates there is one outstanding bus cycle, that its address, status and ADS# have already been driven sometime in the past (in state T12), and that the data and BRDY# pins are not being sampled because the data bus requires one dead clock to turn around between consecutive reads and writes, or writes and reads. The Pentium processor enters TD if in the previous clock there were two outstanding cycles, the last BRDY# was returned, and a dead clock is needed. The timing diagrams in the next section give examples when a dead clock is needed.

Table 6-14 gives a brief summary of bus activity during each bus state. Figure 6-29 shows the Pentium processor bus state diagram.

intel®

Table 6-14. Pentium™ Processor Bus Activity

Bus State	Cycles Outstanding	ADS# Asserted New Address Driven	BRDY# Sampled Data Transferred
Ti	0	No	No
T1	1	Yes	No
T2	1	No	Yes
T12	2	Yes	Yes
T2P	2	No	Yes
TD	1	No	No
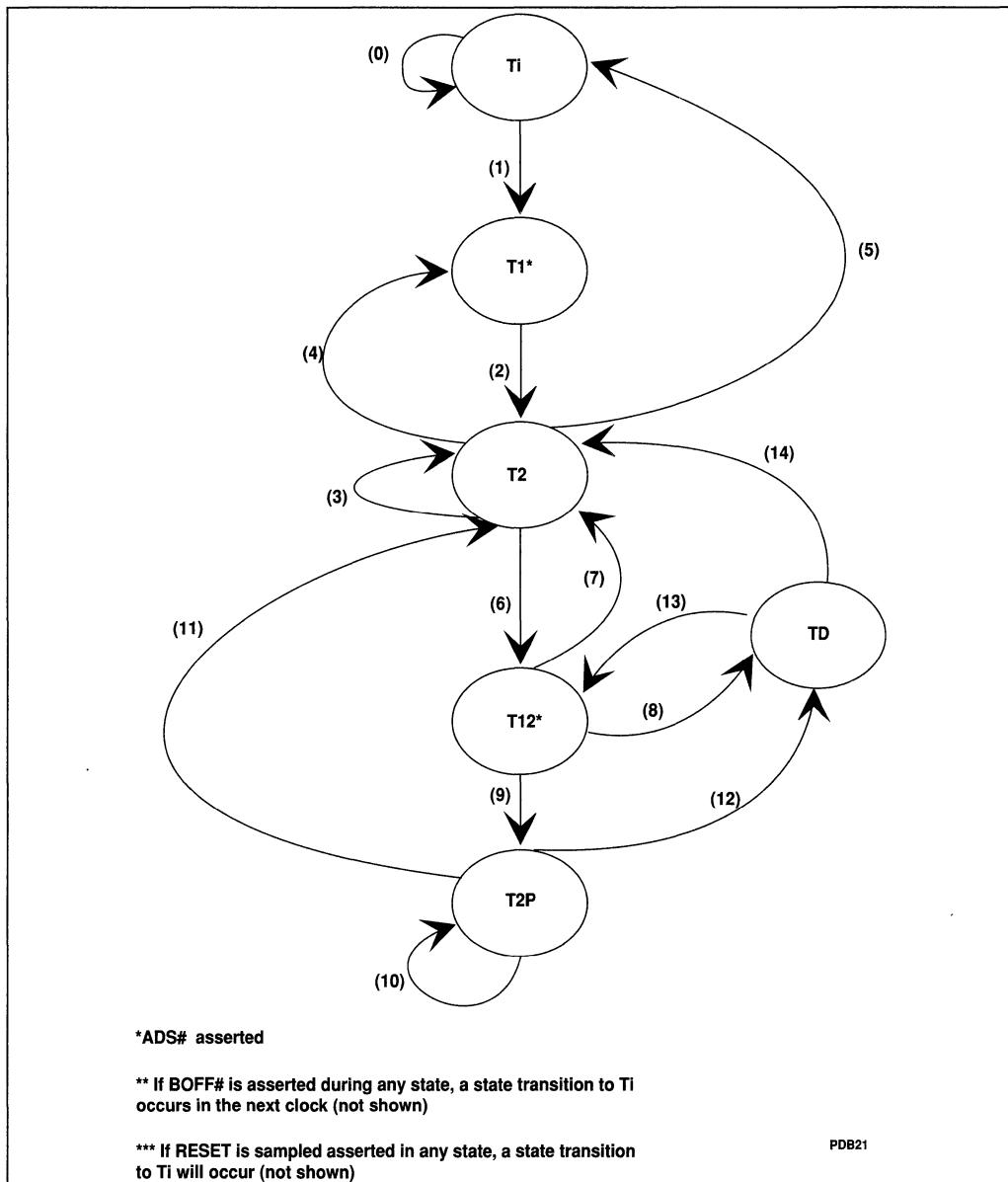
Figure 6-29. Pentium™ Processor Bus Control State Machine

6.6.1. State Transitions

The state transition equations with descriptions are listed below. In the equations, "&" means logical AND, "+" means logical OR, and "#" placed after label means active low. The NA#

used here is actually a delayed version of the external NA# pin (delayed by one clock). The definition of request pending is:

The Pentium processor has generated a new bus cycle internally & HOLD (delayed by one clock) negated & BOFF# negated & (AHOLD negated + HITM# asserted);

Note that once NA# is sampled asserted the Pentium processor latches NA# and will pipeline a cycle when one becomes pending even if NA# is subsequently deasserted.

(0) No Request Pending

(1) Request Pending;

The Pentium processor starts a new bus cycle & ADS# is asserted in the T1 state.

(2) Always;

With BOFF# negated, and a cycle outstanding the Pentium processor always moves to T2 to process the data transfer.

(3) Not Last BRDY# & (No Request Pending + NA# Negated);

The Pentium processor stays in T2 until the transfer is over if no new request becomes pending or if NA# is not asserted.

(4) Last BRDY# & Request Pending & NA# Sampled Asserted;

If there is a new request pending when the current cycle is complete, and if NA# was sampled asserted, the Pentium processor begins from T1.

(5) Last BRDY# & (No Request Pending + NA# Negated);

If no cycle is pending when the Pentium processor finishes the current cycle or NA# is not asserted, the Pentium processor goes back to the idle state.

(6) Not Last BRDY# & Request Pending & NA# Sampled Asserted;

While the Pentium processor is processing the current cycle (one outstanding cycle), if another cycle becomes pending and NA# is asserted, the Pentium processor moves to T12 indicating that the Pentium processor now has two outstanding cycles. ADS# is asserted for the second cycle.

(7) Last BRDY# & No dead clock;

When the Pentium processor finishes the current cycle, and no dead clock is needed, it goes to the T2 state.

(8) Last BRDY# & Need a dead clock;

When the Pentium processor finishes the current cycle, and a dead clock is needed, it goes to the TD state.

(9) Not Last BRDY#;

With BOFF# negated, and the current cycle not complete, the Pentium processor always moves to T2P to process the data transfer.

(10) Not Last BRDY#;

The Pentium processor stays in T2P until the first cycle transfer is over.

(11) Last BRDY# & No dead clock;

When the Pentium processor finishes the first cycle, and no dead clock is needed, it goes to T2 state.

(12) Last BRDY# & Need a dead clock;

When the first cycle is complete, and a dead clock is needed, it goes to TD state.

(13) Request Pending & NA# sampled asserted;

If NA# was sampled asserted and there is a new request pending , it goes to T12 state.

(14) No Request Pending + NA# Negated;

If there is no new request pending, or NA# was not asserted, it goes to T2 state.

6.6.2. Dead Clock Timing Diagrams

The timing diagrams in Figure 6-30 and Figure 6-31 show bus cycles with and without a dead clock.
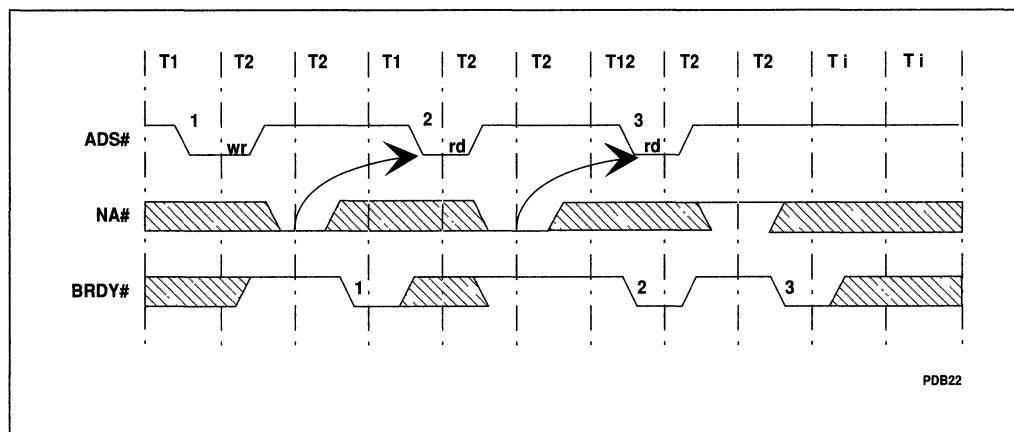
Figure 6-30. Bus Cycles Without Dead Clock

In Figure 6-30, cycles 1 and 2 can be either read or write cycles and no dead clock would be needed because only one cycle is outstanding when those cycles are driven. To prevent a dead clock from being necessary after cycle 3 is driven it must be "opposite" of cycle 2. That is if cycle 2 is a read cycle, cycle 3 must be a write cycle in order to prevent a dead clock. If cycle 2 is a write cycle, cycles 3 must be a read cycle to prevent a dead clock.

intel®
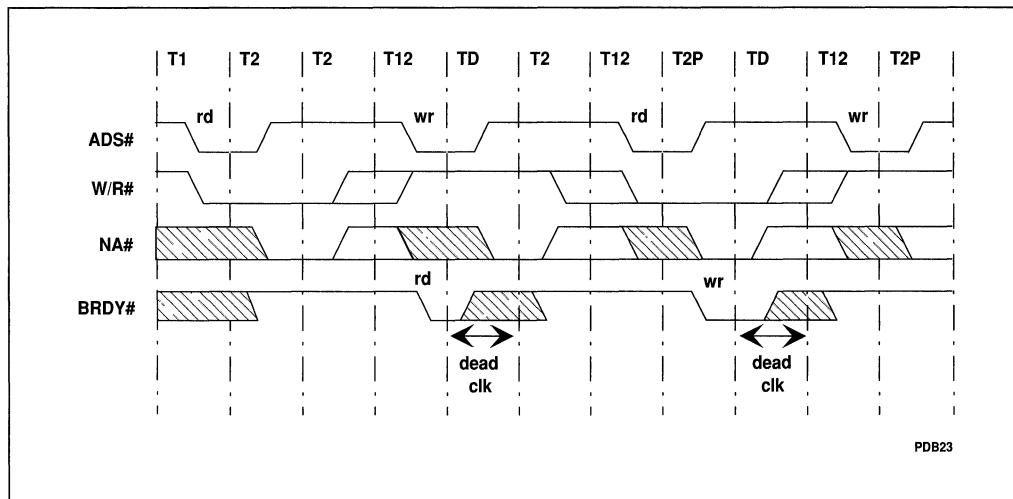
Figure 6-31. Bus Cycles with TD Dead Clock

---

[1]Since the EADS# input is ignored by the processor in certain clocks, the two clocks reference is from the clock in which EADS# is asserted and actually sampled by the processor at the end of this clock (i.e. rising edge of next clock) as shown in Figure 6-22.

intel ®

7

Electrical
Specifications

.

intel ®

CHAPTER 7
ELECTRICAL SPECIFICATIONS

7.1. POWER AND GROUND

For clean on-chip power distribution, the Pentium processor has 50 Vcc (power) and 49 Vss (ground) inputs. Power and ground connections must be made to all external Vcc and Vss pins of the Pentium processor. On the circuit board, all Vcc pins must be connected to a Vcc plane. All Vss pins must be connected to a Vss plane.

7.2. DECOUPLING RECOMMENDATIONS

Liberal decoupling capacitance should be placed near the Pentium processor. The Pentium processor driving its large address and data buses at high frequencies can cause transient power surges, particularly when driving large capacitive loads.

Low inductance capacitors (i.e. surface mount capacitors) and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by connecting capacitors directly to the Vcc and Vss planes, with minimal trace length between the component pads and vias to the plane. Capacitors specifically for PGA packages are also commercially available.

These capacitors should be evenly distributed among each component. Capacitor values should be chosen to ensure they eliminate both low and high frequency noise components.

7.3. CONNECTION SPECIFICATIONS

All NC pins must remain unconnected.

For reliable operation, always connect unused inputs to an appropriate signal level. Unused active low inputs should be connected to Vcc. Unused active high inputs should be connected to ground.

7.4. MAXIMUM RATINGS

Table 7-1 is a stress rating only. Functional operation at the maximums is not guaranteed. Functional operating conditions are given in the A.C. and D.C. specification tables.

Extended exposure to the maximum ratings may affect device reliability. Furthermore, although the Pentium processor contains protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.

Table 7-1. Absolute Maximum Ratings

Case temperature under bias.	-65°C to 110°C
Storage temperature.	-65°C to 150°C
Voltage on any pin with respect to ground	-0.5 Vcc to Vcc $+ 0.5$ (V)
Supply voltage with respect to Vss	-0.5V to $+6.5$V

7.5. D.C. SPECIFICATIONS

Table 7-2 lists the D.C. specifications associated with the Pentium processor.

I

Table 7-2. Pentium™ Processor D.C. Specifications

$V_{CC} = 5V \pm 5\%$, $T_{CASE} = 0$ to $+85$ °C					
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Notes**
V_{IL}	Input Low Voltage	-0.3	+0.8	V	TTL Level
V_{IH}	Input High Voltage	2.0	Vcc+0.3	V	TTL Level
V_{OL}	Output Low Voltage		0.45	V	TTL Level (1)
V_{OH}	Output High Voltage	2.4		V	TTL Level (2)
I_{CC}	Power Supply Current		3200 2910	mA mA	66 MHz, (7), (8) 60 MHz, (7), (9)
I_{LI}	Input Leakage Current		±15	uA	$0 \le V_{IN} \le V_{CC}$, (4)
I_{LO}	Output Leakage Current		±15	uA	$0 \le V_{OUT} \le V_{CC}$ Tristate , (4)
I_{IL}	Input Leakage Current		-400	uA	$V_{IN} = 0.45V$, (5)
I_{IH}	Input Leakage Current		200	uA	$V_{IN} = 2.4V$, (6)
C_{IN}	Input Capacitance		15	pF	
C_O	Output Capacitance		20	pF	
$C_{I/O}$	I/O Capacitance		25	pF	
C_{CLK}	CLK Input Capacitance		8	pF	
C_{TIN}	Test Input Capacitance		15	pf	
C_{TOUT}	Test Output Capacitance		20	pf	
C_{TCK}	Test Clock Capacitance		8	pf	

NOTES:
(1) Parameter measured at 4mA load.
(2) Parameter measured at 1mA load.
(4) This parameter is for input without pullup or pulldown.
(5) This parameter is for input with pullup.
(6) This parameter is for input with pulldown.
(7) Worst case average Icc for a mix of test patterns.
(8) (16 W max.) Typical Pentium™ processor supply current is 2600 mA (13 W) at 66 MHz.
(9) (14.6 W max.) Typical Pentium processor supply current is 2370 mA (11.9 W) at 60 MHz.

7.6.   A.C. SPECIFICATIONS

The 66 MHz and 60 MHz A.C. specifications given in Tables 7-3 and 7-4 consist of output delays, input setup requirements and input hold requirements. All A.C. specifications (with the exception of those for the TAP signals) are relative to the rising edge of the CLK input.

All timings are referenced to 1.5 volts for both "0" and "1" logic levels unless otherwise specified. Within the sampling window, a synchronous input must be stable for correct Pentium processor operation.

Care should be taken to read all notes associated with a particular timing parameter. In addition, the following list of notes apply to the timing specification tables in general and are not associated with any one timing. They are 2, 5, 6, and 14.

Table 7-3. 66 MHz Pentium™ Processor A.C. Specifications

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	$V_{CC}=5V\pm5\%$; $T_{case}=0°C$ to $85°C$; $C_L=0$ pF					
	Frequency	33.33	66.66	MHz		1x CLK
t_1	CLK Period	15		nS	7.1	
t_{1a}	CLK Period Stability		+/-250	pS		(18), (19), (20), (21)
t_2	CLK High Time	4		nS	7.1	@2V, (1)
t_3	CLK Low Time	4		nS	7.1	@0.8V, (1)
t_4	CLK Fall Time	0.15	1.5	nS	7.1	(2.0V-0.8V), (1)
t_5	CLK Rise Time	0.15	1.5	nS	7.1	(0.8V-2.0V), (1)
t_6	ADS#, A3-A31, BT0-3, PWT, PCD,BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Valid Delay	1.5	8.0	nS	7.2	
t_{6a}	AP Valid Delay	1.5	9.5	nS	7.2	
t_7	ADS#, AP, A3-A31, BT0-3, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10	nS	7.3	(1)
t_8	PCHK#, APCHK#, IERR#, FERR# Valid Delay	1.5	8.3	nS	7.2	(4)
t_9	BREQ,HLDA, SMIACT# Valid Delay	1.5	8.0	nS	7.2	(4)
t_{10}	HIT#,HITM# Valid Delay	1.5	8.0	nS	7.2	
t_{11}	PM0-1, BP0-3, IU, IV, IBT Valid Delay	1.5	10	nS	7.2	
t_{11a}	PRDY Valid Delay	1.5	8.0	nS	7.2	
t_{12}	D0-D63,DP0-7 Write Data Valid Delay	1.5	9	nS	7.2	
t_{13}	D0-D63,DP0-7 Write Data Float Delay		10	nS	7.3	(1)
t_{14}	A5-A31 Setup Time	6.5		nS	7.4	
t_{15}	A5-A31 Hold Time	1.5		nS	7.4	
t_{16}	EADS#, INV, AP Setup Time	5		nS	7.4	
t_{17}	EADS#, INV, AP Hold Time	1.5		nS	7.4	
t_{18}	KEN#, WB/WT# Setup Time	5		nS	7.4	
t_{18a}	NA# Setup Time	4.5		nS	7.4	
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.5		nS	7.4	
t_{20}	BRDY# Setup Time	5		nS	7.4	
t_{21}	BRDY# Hold Time	1.5		nS	7.4	

intel®

Table 7-3. 66 MHz Pentium™ Processor A.C. Specifications (Contd.)

$V_{CC}=5V\pm5\%$; $T_{case}=0^\circ C$ to $85^\circ C$; $C_L = 0$ pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{22}	AHOLD, BOFF# Setup Time	5.5		nS	7.4	
t_{23}	AHOLD, BOFF# Hold Time	1.5		nS	7.4	
t_{24}	BUSCHK#, EWBE#, HOLD, PEN# Setup Time	5		nS	7.4	
t_{25}	BUSCHK#, EWBE#, HOLD, PEN# Hold Time	1.5		nS	7.4	
t_{26}	A20M#, INTR, Setup Time	5		nS	7.4	(12), (16)
t_{27}	A20M#, INTR, Hold Time	1.5		nS	7.4	(13)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5		nS	7.4	(16), (17)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.5		nS	7.4	
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width, Async	2		Clks		(15), (17)
t_{31}	R/S# Setup Time	5		nS	7.4	(12), (16), (17)
t_{32}	R/S# Hold Time	1.5		nS	7.4	(13)
t_{33}	R/S# Pulse Width, Async.	2		CLKs		(15), (17)
t_{34}	D0-D63 Read Data Setup Time	3.8		nS	7.4	
t_{34a}	DP0-7 Read Data Setup Time	4.0		nS	7.4	
t_{35}	D0-D63,DP0-7 Read Data Hold Time	2		nS	7.4	
t_{36}	RESET Setup Time	5		nS	7.5	(11), (12), (16)
t_{37}	RESET Hold Time	1.5		nS	7.5	(11), (13)
t38	RESET Pulse Width, Vcc & CLK Stable	15		Clks	7.5	(11)
t39	RESET Active After Vcc & CLK Stable	1		mS	7.5	power up, (11)
t_{40}	Pentium™ processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time	5		nS	7.5	(12), (16), (17)
t_{41}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time	1.5		nS	7.5	(13)
t_{42}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time, Async.	2		CLKs	7.5	(16)

Table 7-3. 66 MHz Pentium™ Processor A.C. Specifications (Contd.)

V_{CC}=5V±5%; T_{case}=0°C to 85°C; C_L= 0 pF						
Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{43}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time, Async.	2		CLKs	7.5	
t_{44}	TCK Frequency	--	16	MHz		
t_{45}	TCK Period	62.5		nS	7.1	
t_{46}	TCK High Time	25		nS	7.1	@2V, (1)
t_{47}	TCK Low Time	25		nS	7.1	@0.8V, (1)
t_{48}	TCK Fall Time		5	nS	7.1	(2.0V-0.8V), (1), (8), (9)
t_{49}	TCK Rise Time		5	nS	7.1	(0.8V-2.0V), (1), (8), (9)
t_{50}	TRST# Pulse Width	40		nS	7.7	(1), Asynchronous
t_{51}	TDI, TMS Setup Time	5		nS	7.6	(7)
t_{52}	TDI, TMS Hold Time	13		nS	7.6	(7)
t_{53}	TDO Valid Delay	3	20	nS	7.6	(8)
t_{54}	TDO Float Delay		25	nS	7.6	(1), (8)
t_{55}	All Non-Test Outputs Valid Delay	3	20	nS	7.6	(3), (8), (10)
t_{56}	All Non-Test Outputs Float Delay		25	nS	7.6	(1), (3), (8), (10)
t_{57}	All Non-Test Inputs Setup Time	5		nS	7.6	(3), (7), (10)
t_{58}	All Non-Test Inputs Hold Time	13		nS	7.6	(3), (7), (10)

NOTES:

1. Not 100% tested. Guaranteed by design/characterization.

2. TTL input test waveforms are assumed to be 0 to 3 Volt transitions with 1Volt/ns rise and fall times.

3. Non-Test Outputs and Inputs are the normal output or input signals (besides TCK, TRST#, TDI, TDO, and TMS). These timings correspond to the response of these signals due to boundary scan operations.

4. APCHK#, FERR#, HLDA, IERR#, LOCK#, and PCHK# are glitch free outputs. Glitch free signals monotonically transition without false transitions (i.e. glitches).

5. 0.8 V/ns <= CLK input rise/fall time <= 8 V/ns.

6. 0.3 V/ns <= Input rise/fall time <= 5 V/ns.

7. Referenced to TCK rising edge.

8. Referenced to TCK falling edge.

9. 1ns can be added to the maximum TCK rise and fall times for every 10 MHz of frequency below 16 MHz.

10. During probe mode operation, use the normal specified timings. Do not use the boundary scan timings (t_{55-58}).

11. FRCMC# should be tied to Vcc (high) to ensure proper operation of the Pentium processor as a master Pentium processor.

12. Setup time is required to guarantee recognition on a specific clock.

13. Hold time is required to guarantee recognition on a specific clock.

14. All TTL timings are referenced from 1.5 V.

15. To guarantee proper asynchronous recognition, the signal must have been deasserted (inactive) for a minmum of 2 clocks before being returned active and must meet the minimum pulse width.

16. This input may be driven asynchronously.

17. When driven asynchronously, NMI, FLUSH#, R/S#, INIT, and SMI must be deasserted (inactive) for a minimum of 2 clocks before being returned active.

18. Functionality is guaranteed by design/characterization.

19. Measured on rising edge of adjacent CLKs at 1.5V.

20. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal and external clocks, the jitter frequency spectrum should not have any power spectrum peaking between 500 KHz and 1/3 of the CLK operating frequency.

21. The amount of jitter present must be accounted for as a component of CLK skew between devices.

intel®

Table 7-4. 60 MHz Pentium™ Processor A.C. Specifications

V_{CC}=5V±5%; T_{case}=0°C to 85°C; C_L= 0 pF						
Symbol	Parameter	Min	Max	Unit	Figure	Notes
	Frequency	33.33	60	MHz		1x CLK
t_1	CLK Period	16.67		nS	7.1	
t_{1a}	CLK Period Stability		+/-250	pS		(18), (19), (20), (21)
t_2	CLK High Time	4		nS	7.1	@2V, (1)
t_3	CLK Low Time	4		nS	7.1	@0.8V, (1)
t_4	CLK Fall Time	0.15	1.5	nS	7.1	(2.0V-0.8V), (1)
t_5	CLK Rise Time	0.15	1.5	nS	7.1	(0.8V-2.0V), (1)
t_6	ADS#, A3-A31, BT0-3, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Valid Delay	1.5	9.0	nS	7.2	
t_{6a}	AP Valid Delay	1.5	10.5	nS	7.2	
t_7	ADS#, AP, A3-A31, BT0-3, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		11	nS	7.3	(1)
t_8	PCHK#, APCHK#, IERR#, FERR# Valid Delay	1.5	9.3	nS	7.2	(4)
t_9	BREQ,HLDA, SMIACT# Valid Delay	1.5	9.0	nS	7.2	(4)
t_{10}	HIT#,HITM# Valid Delay	1.5	9.0	nS	7.2	
t_{11}	PM0-1, BP0-3, IU, IV, IBT Valid Delay	1.5	11	nS	7.2	
t_{11a}	PRDY Valid Delay	1.5	9.0	nS	7.2	
t_{12}	D0-D63,DP0-7 Write Data Valid Delay	1.5	10	nS	7.2	
t_{13}	D0-D63,DP0-7 Write Data Float Delay		11	nS	7.3	(1)
t_{14}	A5-A31 Setup Time	7		nS	7.4	
t_{15}	A5-A31 Hold Time	1.5		nS	7.4	
t_{16}	EADS#, INV, AP Setup Time	5.5		nS	7.4	
t_{17}	EADS#, INV, AP Hold Time	1.5		nS	7.4	
t_{18}	KEN#, WB/WT# Setup Time	5.5		nS	7.4	
t_{18a}	NA# Setup Time	5.0		nS	7.4	
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.5		nS	7.4	
t_{20}	BRDY# Setup Time	5.5		nS	7.4	
t_{21}	BRDY# Hold Time	1.5		nS	7.4	

Table 7-4. 60 MHz Pentium™ Processor A.C. Specifications (Contd.)

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	V_{CC}=5V±5%; T_{case}=0°C to 85°C; C_L= 0 pF					
t_{22}	AHOLD, BOFF# Setup Time	6		nS	7.4	
t_{23}	AHOLD, BOFF# Hold Time	1.5		nS	7.4	
t_{24}	BUSCHK#, EWBE#, HOLD, PEN# Setup Time	5.5		nS	7.4	
t_{25}	BUSCHK#, EWBE#, HOLD, PEN# Hold Time	1.5		nS	7.4	
t_{26}	A20M#, INTR, Setup Time	5.5		nS	7.4	(12), (16)
t_{27}	A20M#, INTR, Hold Time	1.5		nS	7.4	(13)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5.5		nS	7.4	(16), (17)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.5		nS	7.4	
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width, Async	2		Clks		(15), (17)
t_{31}	R/S# Setup Time	5.5		nS	7.4	(12), (16), (17)
t_{32}	R/S# Hold Time	1.5		nS	7.4	(13)
t_{33}	R/S# Pulse Width, Async.	2		CLKs		(15), (17)
t_{34}	D0-D63 Read Data Setup Time	4.3		nS	7.4	
t_{34a}	DP0-7 Read Data Setup Time	4.5		nS	7.4	
t_{35}	D0-D63, DP0-7 Read Data Hold Time	2		nS	7.4	
t_{36}	RESET Setup Time	5.5		nS	7.5	(11), (12), (16)
t_{37}	RESET Hold Time	1.5		nS	7.5	(11), (13)
t38	RESET Pulse Width, Vcc & CLK Stable	15		Clks	7.5	(11)
t39	RESET Active After Vcc & CLK Stable	1		mS	7.5	power up, (11)
t_{40}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time	5.5		nS	7.5	(12), (16), (17)
t_{41}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time	1.5		nS	7.5	(13)
t_{42}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time, Async.	2		CLKs	7.5	(16)

Table 7-4. 60 MHz Pentium™ Processor A.C. Specifications (Contd.)

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	$V_{CC}=5V\pm5\%$; $T_{case}=0^{\circ}$C to 85°C; $C_{L}= 0$ pF					
t_{43}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time, Async.	2		CLKs	7.5	
t_{44}	TCK Frequency	--	16	MHz		
t_{45}	TCK Period	62.5		nS	7.1	
t_{46}	TCK High Time	25		nS	7.1	@2V, (1)
t_{47}	TCK Low Time	25		nS	7.1	@0.8V, (1)
t_{48}	TCK Fall Time		5	nS	7.1	(2.0V-0.8V), (1), (8), (9)
t_{49}	TCK Rise Time		5	nS	7.1	(0.8V-2.0V), (1), (8), (9)
t_{50}	TRST# Pulse Width	40		nS	7.7	(1), Async
t_{51}	TDI, TMS Setup Time	5		nS	7.6	(7)
t_{52}	TDI, TMS Hold Time	13		nS	7.6	(7)
t_{53}	TDO Valid Delay	3	20	nS	7.6	(8)
t_{54}	TDO Float Delay		25	nS	7.6	(1), (8)
t_{55}	All Non-Test Outputs Valid Delay	3	20	nS	7.6	(3), (8), (10)
t_{56}	All Non-Test Outputs Float Delay		25	nS	7.6	(1), (3), (8), (10)
t_{57}	All Non-Test Inputs Setup Time	5		nS	7.6	(3), (7), (10)
t_{58}	All Non-Test Inputs Hold Time	13		nS	7.6	(3), (7), (10)

NOTES:

1. Not 100% tested. Guaranteed by design/characterization.

2. TTL input test waveforms are assumed to be 0 to 3 Volt transitions with 1Volt/ns rise and fall times.

3. Non-Test Outputs and Inputs are the normal output or input signals (besides TCK, TRST#, TDI, TDO, and TMS). These timings correspond to the response of these signals due to boundary scan operations.

4. APCHK#, FERR#, HLDA, IERR#, LOCK#, and PCHK# are glitch free outputs. Glitch free signals monotonically transition without false transitions (i.e. glitches).

5. 0.8 V/ns <= CLK input rise/fall time <= 8 V/ns.

6. 0.3 V/ns <= Input rise/fall time <= 5 V/ns.

7. Referenced to TCK rising edge.

8. Referenced to TCK falling edge.

9. 1ns can be added to the maximum TCK rise and fall times for every 10 MHz of frequency below 16 MHz.

10. During probe mode operation, use the normal specified timings. Do not use the boundary scan timings (t_{55-58}).

11. FRCMC# should be tied to Vcc (high) to ensure proper operation of the Pentium processor as a master Pentium processor.

12. Setup time is required to guarantee recognition on a specific clock.

13. Hold time is required to guarantee recognition on a specific clock.

14. All TTL timings are referenced from 1.5 V.

15. To guarantee proper asynchronous recognition, the signal must have been deasserted (inactive) for a minmum of 2 clocks before being returned active and must meet the minimum pulse width.

16. This input may be driven asynchronously.

17. When driven asynchronously, NMI, FLUSH#, R/S#, INIT, and SMI must be deasserted (inactive) for a minimum of 2 clocks before being returned active.

18. Functionality is guaranteed by design/characterization.

19. Measured on rising edge of adjacent CLKs at 1.5V.

20. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal and external clocks, the jitter frequency spectrum should not have any power spectrum peaking between 500 KHz and 1/3 of the CLK operating frequency.

21. The amount of jitter present must be accounted for as a component of CLK skew between devices.

Each valid delay is specified for a 0 pF load. The system designer should use I/O buffer modeling to account for signal flight time delays.

Tv = t5, t49
Tw = t4, t48
Tx = t3, t47
Ty = t1, t45
Tz = t2, t46
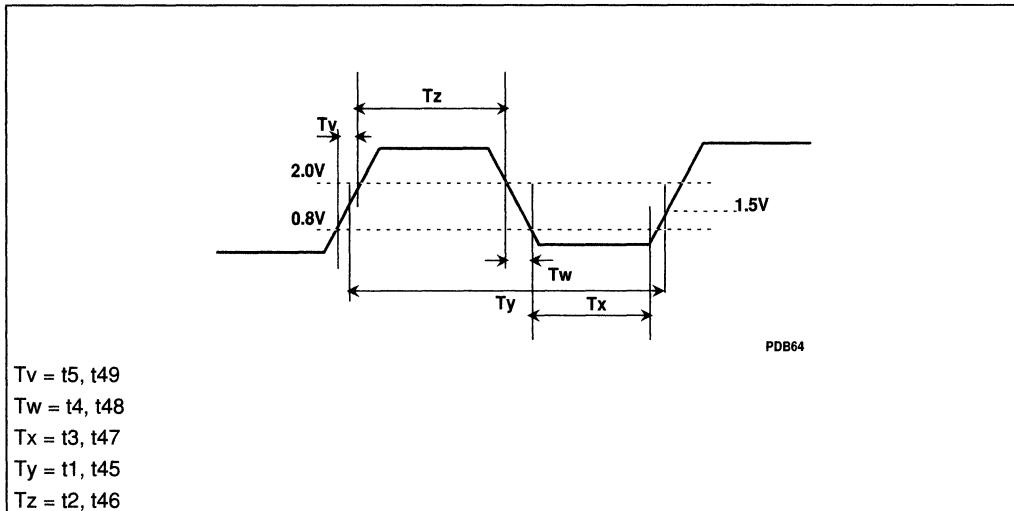
Figure 7-1. Clock Waveform

Tx = t6, t6a, t8, t9, t10, t11, t11a, t12
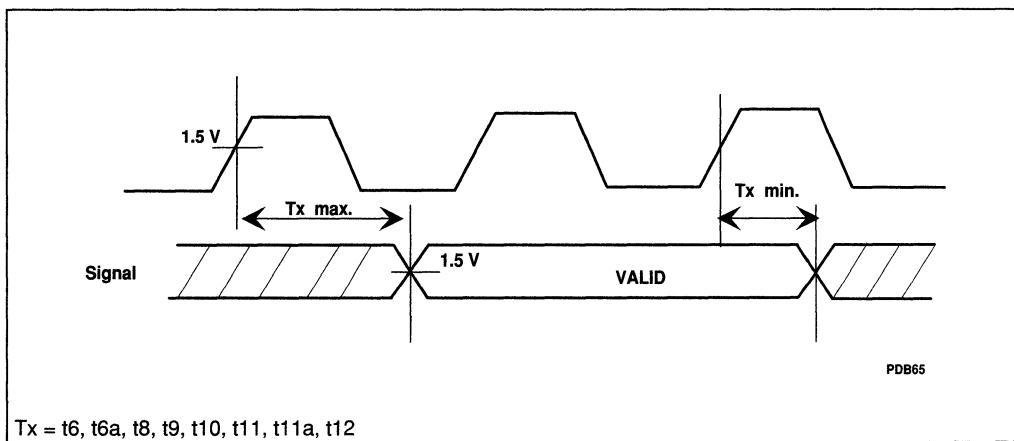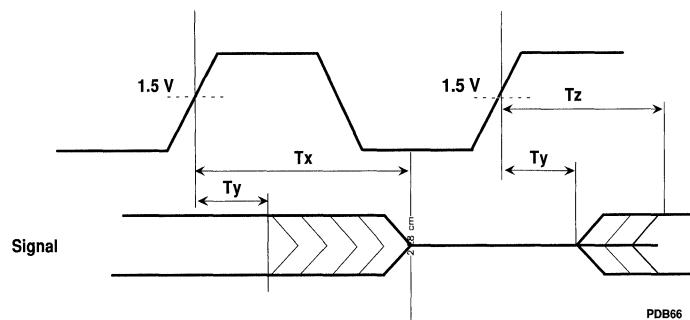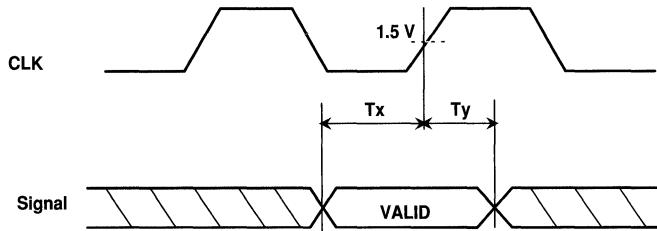
Figure 7-2. Valid Delay Timings

intel®

Tx = t7, t13
Ty = t6min, t12min

Figure 7-3. Float Delay Timings

Tx = t14, t16, t18, t18a, t20, t22, t24, t26, t28, t31, t34, t34a
Ty = t15, t17, t19, t21, t23, t25, t27, t29, t32, t35

Figure 7-4. Setup and Hold Timings
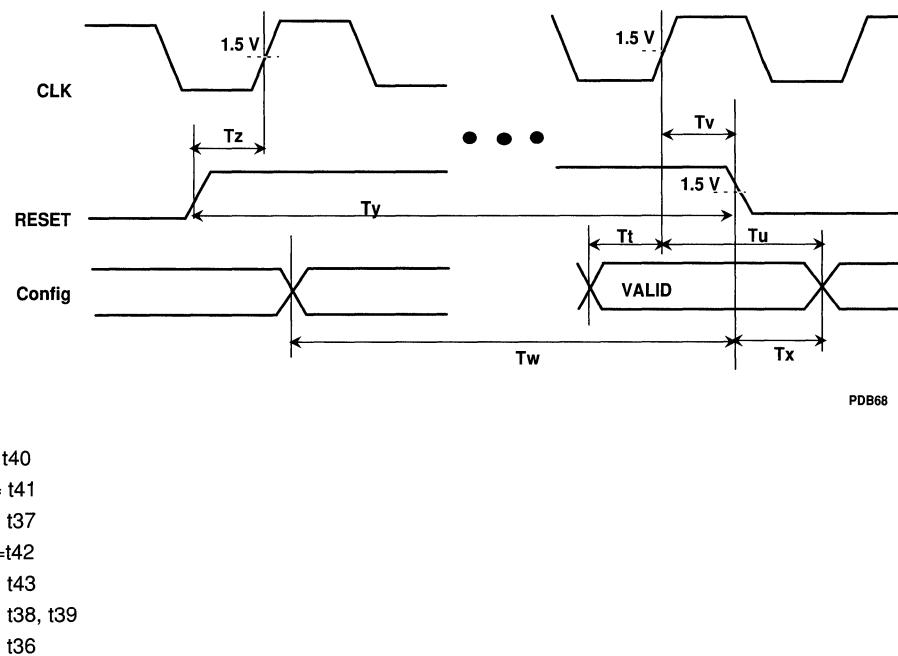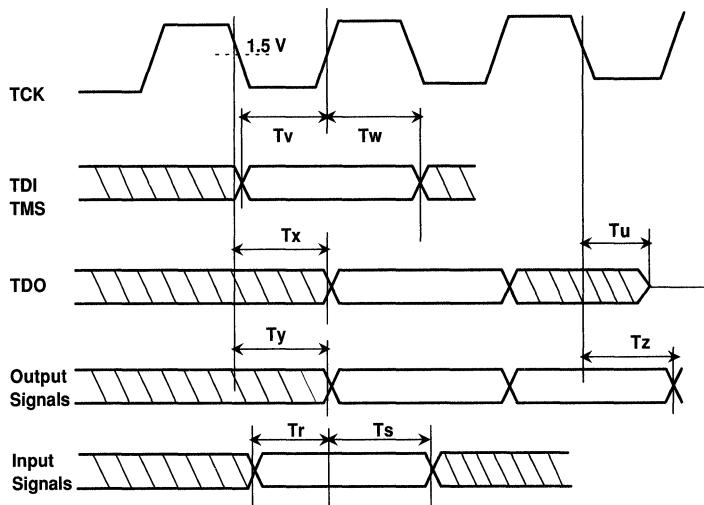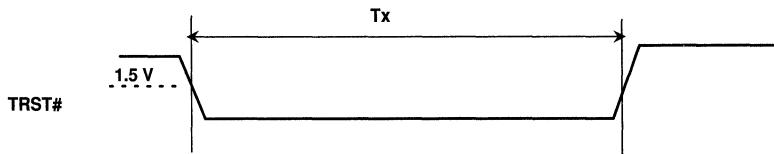
Figure 7-5. Reset and Configuration Timings

intel ®

Tr = t57
Ts = t58
Tu = t54
Tv = t51
Tw = t52
Tx = t53
Ty = t55
Tz = t56

Figure 7-6. Test Timings

Tx = t50

Figure 7-7. Test Reset Timings

Each valid delay is specified for a 0 pF load. The system designer should use I/O buffer modeling to account for signal delays due to loading. Table 7-5 lists the buffer type to be used for each signal in the external interface.

Table 7-5. External Interface Signal Buffer Assignment

Device	Signals	Type	Driver Buffer Type	Receiver Buffer Type
Pentium™ processor	A20M#, FLUSH#, FRCMC#, HOLD, IGNNE#, INIT, INTR, NMI, PEN#, R/S#, RESET, SMI#, TDI, TMS,	I	N/A	ER1
	AHOLD, BOFF#, EADS#, EWBE#, KEN#, NA#, WB/WT#	I	N/A	ER3
	INV	I	N/A	ER3a
	BRDY#, BUSCHK#, TRST#	I	N/A	ER2
	CLK	I	N/A	ER8
	TCK	I	N/A	ER9
	A3-20	I/O	ED7	ER7
	A21-31, BT0-3	I/O	ED4	ER6
	D0-63, DP0-7	I/O	ED3	ER5
	AP	I/O	ED5	ER4
	ADS#, HITM#, W/R#	O	ED6	N/A
	BE0-7#, CACHE#, SCYC, LOCK#, PWT, PCD, M/IO#, D/C#, BREQ, HIT#	O	ED2	N/A
	APCHK#, BP3-0#, PM1, PM0, FERR#, HLDA, IBT, IERR#, IU, IV, PCHK#, PRDY, SMIACT#, TDO	O	ED1	N/A

7.7. OVERSHOOT/UNDERSHOOT GUIDELINES

The overshoot/undershoot guideline is provided to limit signals transitioning beyond Vcc or Vss due to the fast signal switching at these frequencies. Excessive ringback is the dominant harmful effect resulting from overshoot/undershoot.

Overshoot (Undershoot) is the absolute value of the maximum voltage above Vcc (below Vss). The guideline assumes the absence of diodes on the input. This guideline should be used in simulations, without the diodes present, to ensure overshoot (undershoot) is within the acceptable range.

Maximum Overshoot/Undershoot on Inputs = 1.6 Volts

(without diodes)

int**ط**₀

Ringback is the absolute value of the maximum voltage at the receiving pin below Vcc (or above Vss) relative to Vcc (or Vss) level after the signal has reached its maximum voltage level. The input diodes are assumed present. This guideline is provided to allow system designers to verify, in an actual system, the decisions made based on simulation using the overshoot (undershoot) guideline. Ringback only applies if the signal crossed above Vcc (below Vss).

Maximum Ringback on Inputs = 0.8 Volts
(with diodes)
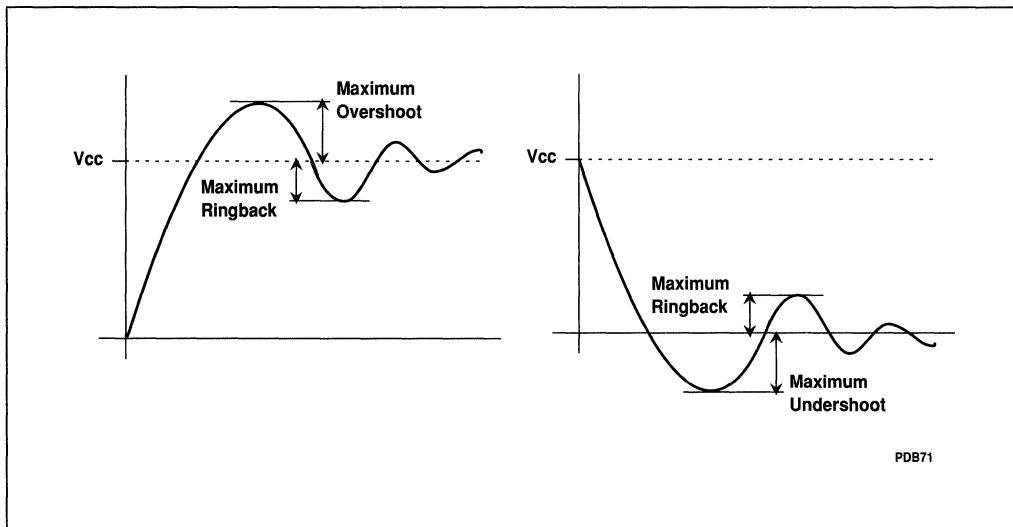
Figure 7-8. Overshoot/Undershoot and Ringback Guidelines

intel®

8

I/O Buffer Models

intel®

CHAPTER 8
I/O BUFFER MODELS

The first order I/O buffer model is a simplified representation of the complex input and output buffers used in the Pentium processor. Figure 8-1 shows the structure of the input buffer model and Figure 8-2 shows the output buffer model. Table 8-1 and Table 8-2 show the parameters used to specify these models.
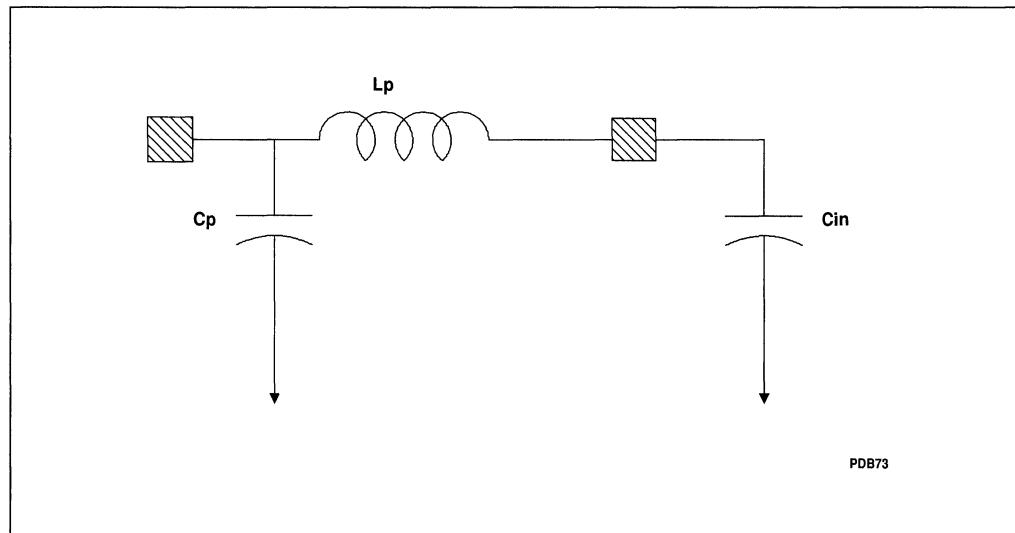
Figure 8-1. First Order Input Buffer

Table 8-1. Parameters Used in the Specification of the First Order Input Buffer Model

Parameter	Description
Cin	Minimum and Maximum value of the capacitance of the input buffer model.
Lp	Minimum and Maximum value of the package inductance.
Cp	Minimum and Maximum value of the package capacitance.
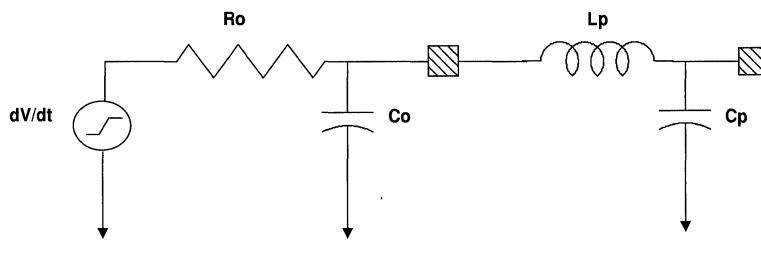
Figure 8-2. First Order Output Buffer

Table 8-2. Parameters Used in the Specification of the First Order Output Buffer Model

Parameter	Description
dV/dt	Minimum and maximum value of the rate of change of the open circuit voltage source used in the output buffer model.
Ro	Minimum and maximum value of the output impedance of the output buffer model.
Co	Minimum and Maximum value of the capacitance of the output buffer model.
Lp	Minimum and Maximum value of the package inductance.
Cp	Minimum and Maximum value of the package capacitance.

Table 8-3 and Table 8-4 list the minimum and maximum parameters for each buffer type within the Pentium processor. These parameters supply the information to use in the circuits shown in Figure 8-1 and Figure 8-2 to model the processors behavior in a given environment.

intel®

Table 8-3. Specification of Input External Buffer Model Parameters

Buffer Type	Cp (pF)		Lp (nH)		Cin (pF)	
	min	max	min	max	min	max
ER1	0.8	10.2	5.2	20.6	1.1	1.5
ER2	1.4	6.8	6.7	16.5	1.7	2.3
ER3	1.1	1.8	6.2	11.3	2.6	3.5
ER3a	7.3	9.9	14.9	20.1	2.6	3.5
ER4	0.5	6.6	5.3	15.2	3.6	4.8
ER5	0.7	7.8	5.4	17.0	3.7	4.9
ER6	0.5	6.6	5.3	15.2	4.2	5.6
ER7	1.3	5.6	6.5	13.5	12.7	17.1
ER8	1.6	2.2	6.2	8.4	1.7	2.3
ER9	2.2	2.9	7.2	9.7	1.9	2.5

intel®

Table 8-4. Specification of Output External Interface Buffer Model Parameters

Buffer Type	Transition	Component	dV/dt (V/nsec)		Ro (Ohms)		Co (pF)		Lp (nH)		Cp (pF)	
			min	max	min	max	min	max	min	max	min	max
ED1	Rising	Pentium™ processor	4.5/3.6	5.5/1.1	21	59	3.6	4.8	5.6	19.9	0.7	9.7
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	3.6	4.8	5.6	19.9	0.7	9.7
ED2	Rising	Pentium processor	4.5/3.6	5.5/1.1	21	59	3.6	4.8	6.8	18.9	1.4	9.1
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	3.6	4.8	6.8	18.9	1.4	9.1
ED3	Rising	Pentium processor	4.5/3.6	5.5/1.1	21	59	3.7	4.9	5.4	17.0	0.7	7.8
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	3.7	4.9	5.4	17.0	0.7	7.8
ED4	Rising	Pentium processor	4.5/3.6	5.5/1.1	21	59	4.2	5.6	5.3	15.2	0.5	6.6
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	4.2	5.6	5.3	15.2	0.5	6.6
ED5	Rising	Pentium processor	4.5/3.6	5.5/1.1	21	59	3.6	4.8	5.3	15.2	0.5	6.6
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	3.6	4.8	5.3	15.2	0.5	6.6
ED6	Rising	Pentium processor	4.5/3.6	5.5/1.1	21	59	12.1	16.3	6.3	10.2	1.4	2.6
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	12.1	16.3	6.3	10.2	1.4	2.6
ED7	Rising	Pentium processor	4.5/3.6	5.5/1.1	21	59	12.7	17.1	6.5	13.5	1.3	5.6
	Falling	Pentium processor	4.5/2.6	5.5/1.1	18	54	12.7	17.1	6.5	13.5	1.3	5.6

8.1.   INPUT DIODE MODELS

In addition to the input and output buffer parameters, input protection diode models are provided for the external interface I/O buffer models. These diodes have been optimized to provide ESD protection and provide some level of clamping. Note however, the signal quality specifications for both the optimized and external interfaces are defined assuming the diodes are not present in the simulation. It is important that these specifications are met because there is a limit to the amount of clamping the diode can attain. The diode model is provided because it may be useful in modeling the behavior of other devices driving transmission lines with the Pentium processor as the receiving device.

Figure 8-3 shows the components of the diode model. It consists of two diodes, one connected to Vcc, D2, and one to Vss, D1. Each diode is modeled by the combination of an ideal diode in series with a resistance.
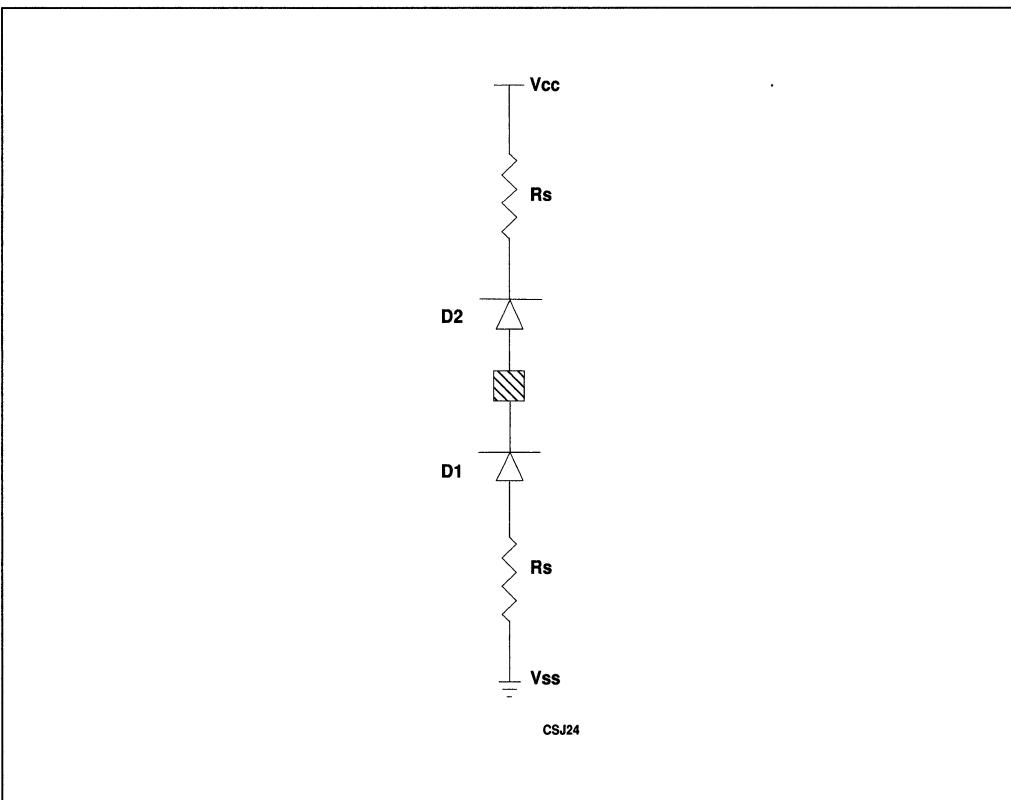
Figure 8-3. Input Diode Model

The diode model should be added to the input model for both inputs and I/O signals when desired. Figure 8-4 shows the complete input model with the diodes added.

intel®
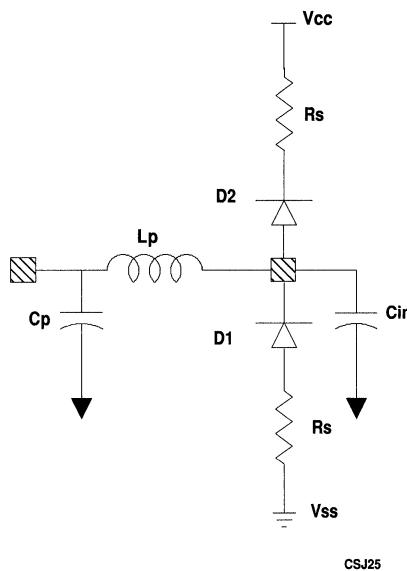
Figure 8-4. Complete Input Model Including Diode

The specific parameters associated with each diode are listed below. Table 8-5 lists the buffer types with their corresponding diode I-V curve and series resistance. Table 8-6 provides the diode I-V curve data for both D1 and D2 for each buffer type.

Table 8-5. Diode Parameter List

Input Model Type	Buffer Type	Driver Mode	Diode	Diode I-V Curve Type	Rs (Ohms)
ER4, ER5, ER6	I/O	std	D1	IV1	6.5
			D2	IV2	6.5
ER7	I/O	xlg	D1	IV3	6.5
			D2	IV4	6.5
ER1, ER2, ER3, ER3a, ER8, ER9	I	N/A	D1	IV5	6.5
			D2	IV6	6.5

Table 8-6. Data for Diode I-V Curves

Curve Type					Curve Type			
IV1		IV2			IV1		IV2	
Vd	Id	Vd	Id		Vd	Id	Vd	Id
0v	0a	0v	0a		0.699v	24.81ua	0.698v	46.58ua
25mv	.0053pa	25mv	.0037pa		0.722v	58.06ua	0.719v	0.12ma
50mv	.0062pa	50mv	.0038pa		0.744v	0.13ma	0.737v	0.26ma
75mv	.0083pa	75mv	.0039pa		0.762v	0.25ma	0.751v	0.47ma
100mv	.0134pa	100mv	.0041pa		0.778v	0.44ma	0.762v	0.76ma
0.125v	.0260pa	0.125v	.0046pa		0.79v	0.69ma	0.772v	0.76ma
0.15v	.0572pa	0.15v	.0062pa		0.8v	1ma	0.779v	1.07ma
0.175v	0.13pa	0.175v	.0107pa		0.809v	0.99ma	0.784v	1.49ma
0.2v	0.33pa	0.2v	.0237pa		0.817v	1.32ma	0.788v	1.91ma
0.225v	0.8pa	0.225v	.0621pa		0.822v	1.73ma	0.792v	2.3ma
0.25v	1.98pa	0.25v	0.18pa		0.826v	2.13ma	0.795v	2.7ma
0.275v	4.91pa	0.275v	0.51pa		0.83v	2.52ma	0.798v	3.13ma
0.3v	12.18pa	0.3v	1.49pa		0.834v	2.93ma	0.801v	3.57ma
0.325v	30.22pa	0.325v	4.4pa		0.837v	3.35ma	0.803v	4.01ma
0.35v	74.98pa	0.35v	12.96pa		0.84v	3.79ma	0.806v	4.46ma
0.375v	0.19na	0.375v	38.22pa		0.843v	4.22ma	0.808v	4.91ma
0.4v	0.46na	0.4v	0.11na		0.845v	4.67ma	0.809v	5.37ma
0.425v	1.15na	0.425v	0.33na		0.848v	5.11ma	0.811v	5.83ma
0.45v	2.84na	0.45v	0.98na		0.85v	5.57ma	0.813v	6.3ma
0.475v	7.26na	0.475v	2.89na		0.852v	6.02ma	0.814v	6.76ma
0.5v	18.02na	0.5v	8.95na		0.854v	6.48ma	0.816v	7.23ma
0.525v	44.72na	0.525v	26.4na		0.856v	6.94ma	0.817v	7.7ma
0.55v	0.11ua	0.55v	77.85na		0.857v	7.4ma	0.819v	8.17ma
0.575v	0.28ua	0.575v	0.23ua		0.859v	7.87ma	0.82v	8.64ma
0.6v	0.68ua	0.6v	0.68ua		0.86v	8.34ma	0.821v	9.12ma
0.625v	1.69ua	0.625v	1.99ua		0.862v	8.81ma	0.822v	9.59ma
0.65v	4.18ua	0.65v	5.82ua		0.863v	9.28ma	0.823v	10.07ma
0.674v	10.26ua	0.674v	16.79ua		0.864v	9.75ma	0.824v	10.55ma

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type				Curve Type			
IV1		IV2		IV1		IV2	
Vd	Id	Vd	Id	Vd	Id	Vd	Id
0.866v	10.22ma	0.825v	11.03ma	0.888v	23.26ma	0.843v	24.16ma
0.867v	10.7ma	0.826v	11.51ma	0.888v	23.75ma	0.843v	24.65ma
0.868v	11.17ma	0.827v	11.99ma	0.889v	24.24ma	0.844v	25.14ma
0.869v	11.65ma	0.828v	12.47ma	0.889v	24.73ma	0.844v	25.63ma
0.87v	12.13ma	0.829v	12.95ma	0.89v	25.22ma	0.845v	26.12ma
0.871v	12.61ma	0.83v	13.43ma	0.89v	25.71ma	0.845v	26.61ma
0.872v	13.09ma	0.83v	13.92ma	0.891v	26.2ma	0.845v	27.11ma
0.873v	13.56ma	0.831v	14.4ma	0.891v	26.69ma	0.846v	27.6ma
0.874v	14.05ma	0.832v	14.89ma	0.892v	27.18ma	0.846v	28.09ma
0.875v	14.53ma	0.833v	15.37ma	0.892v	27.67ma	0.847v	28.58ma
0.876v	15.01ma	0.833v	15.86ma	0.893v	28.16ma	0.847v	29.07ma
0.877v	15.49ma	0.834v	16.34ma	0.893v	28.65ma	0.847v	29.57ma
0.878v	15.97ma	0.835v	16.83ma	0.894v	29.14ma	0.848v	30.06ma
0.878v	16.46ma	0.835v	17.32ma	0.894v	29.63ma	0.848v	30.55ma
0.879v	16.94ma	0.836v	17.8ma	0.895v	30.12ma	0.848v	31.04ma
0.88v	17.43ma	0.836v	18.29ma	0.895v	30.61ma	0.849v	31.53ma
0.881v	17.91ma	0.837v	18.78ma	0.895v	31.1ma	0.849v	32.03ma
0.881v	18.39ma	0.838v	19.27ma	0.896v	31.59ma	0.849v	32.52ma
0.882v	18.88ma	0.838v	19.75ma	0.896v	32.09ma	0.85v	33.01ma
0.883v	19.37ma	0.839v	20.24ma	0.897v	32.58ma	0.85v	33.51ma
0.883v	19.85ma	0.839v	20.73ma	0.897v	33.07ma	0.85v	34ma
0.884v	20.34ma	0.84v	21.22ma	0.898v	33.56ma	0.851v	34.49ma
0.885v	20.83ma	0.84v	21.71ma	0.898v	34.05ma	0.851v	34.99ma
0.885v	21.31ma	0.841v	22.2ma	0.898v	34.54ma	0.851v	35.48ma
0.886v	21.8ma	0.841v	22.69ma	0.899v	35.04ma	0.852v	35.97ma
0.886v	22.29ma	0.842v	23.18ma	0.899v	35.53ma	0.852v	36.47ma
0.887v	22.78ma	0.842v	23.67ma	0.899v	36.02ma	0.852v	36.96ma

intel

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type				Curve Type			
IV1		IV2		IV1		IV2	
Vd	Id	Vd	Id	Vd	Id	Vd	Id
0.9v	36.51ma	0.853v	37.45ma	0.908v	49.84ma	0.86v	50.81ma
0.9v	37.01ma	0.853v	37.95ma	0.909v	50.34ma	0.86v	51.31ma
0.901v	37.5ma	0.853v	38.44ma	0.909v	50.83ma	0.86v	51.8ma
0.901v	37.99ma	0.854v	38.94ma	0.909v	51.33ma	0.86v	52.3ma
0.901v	38.49ma	0.854v	39.43ma	0.909v	51.82ma	0.861v	52.79ma
0.902v	38.98ma	0.854v	39.92ma	0.91v	52.32ma	0.861v	53.29ma
0.902v	39.47ma	0.854v	40.42ma	0.91v	52.81ma	0.861v	53.79ma
0.902v	39.96ma	0.855v	40.91ma	0.91v	53.3ma	0.861v	54.28ma
0.903v	40.46ma	0.855v	41.41ma	0.91v	53.8ma	0.861v	54.78ma
0.903v	40.95ma	0.855v	41.9ma	0.911v	54.29ma	0.862v	55.27ma
0.903v	41.44ma	0.856v	42.4ma	0.911v	54.79ma	0.862v	55.77ma
0.904v	41.94ma	0.856v	42.89ma	0.911v	55.28ma	0.862v	56.27ma
0.904v	42.43ma	0.856v	43.39ma	0.911v	55.78ma	0.862v	56.76ma
0.904v	42.92ma	0.856v	43.88ma	0.912v	56.27ma	0.862v	57.26ma
0.904v	43.42ma	0.857v	44.38ma	0.912v	56.77ma	0.863v	57.75ma
0.905v	43.91ma	0.857v	44.87ma	0.912v	57.27ma	0.863v	58.25ma
0.905v	44.41ma	0.857v	45.37ma	0.912v	57.76ma	0.863v	58.74ma
0.905v	44.9ma	0.857v	45.86ma	0.913v	58.26ma	0.863v	59.24ma
0.906v	45.39ma	0.858v	46.36ma	0.913v	58.75ma	0.863v	59.74ma
0.906v	45.89ma	0.858v	46.85ma	0.913v	59.25ma	0.864v	60.23ma
0.906v	46.38ma	0.858v	47.35ma	0.913v	59.74ma	0.864v	60.73ma
0.907v	46.88ma	0.858v	47.84ma	0.913v	60.24ma	0.864v	61.23ma
0.907v	47.37ma	0.859v	48.34ma	0.914v	60.73ma	0.864v	61.72ma
0.907v	47.86ma	0.859v	48.83ma	0.914v	61.23ma	0.864v	62.22ma
0.907v	48.36ma	0.859v	49.33ma	0.914v	61.72ma	0.864v	62.71ma
0.908v	48.85ma	0.859v	49.82ma	0.914v	62.22ma	0.865v	63.21ma
0.908v	49.35ma	0.859v	50.32ma	0.915v	62.71ma	0.865v	63.71ma

intel®

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type				Curve Type			
IV1		IV2		IV1		IV2	
Vd	Id	Vd	Id	Vd	Id	Vd	Id
0.915v	63.21ma	0.865v	64.2ma	0.919v	72.63ma	0.868v	73.64ma
0.915v	63.71ma	0.865v	64.7ma	0.919v	73.13ma	0.868v	74.14ma
0.915v	64.2ma	0.865v	65.2ma	0.919v	73.63ma	0.868v	74.63ma
0.915v	64.7ma	0.866v	65.69ma	0.919v	74.12ma	0.869v	75.13ma
0.916v	65.19ma	0.866v	66.19ma	0.919v	74.62ma	0.869v	75.63ma
0.916v	65.69ma	0.866v	66.69ma	0.919v	75.11ma	0.869v	76.12ma
0.916v	66.18ma	0.866v	67.18ma	0.92v	75.61ma	0.869v	76.62ma
0.916v	66.68ma	0.866v	67.68ma	0.92v	76.11ma	0.869v	77.12ma
0.916v	67.18ma	0.866v	68.18ma	0.92v	76.6ma	0.869v	77.62ma
0.917v	67.67ma	0.867v	68.67ma	0.92v	77.1ma	0.87v	78.11ma
0.917v	68.17ma	0.867v	69.17ma	0.92v	77.6ma	0.87v	78.61ma
0.917v	68.66ma	0.867v	69.67ma	0.921v	78.09ma	0.87v	79.11ma
0.917v	69.16ma	0.867v	70.16ma	0.921v	78.59ma	0.87v	79.6ma
0.917v	69.66ma	0.867v	70.66ma	0.921v	79.09ma	0.87v	80.1ma
0.918v	70.15ma	0.867v	71.16ma	0.921v	79.58ma	0.87v	80.6ma
0.918v	70.65ma	0.868v	71.65ma	0.921v	80.08ma	0.87v	81.09ma
0.918v	71.14ma	0.868v	72.15ma	0.921v	80.58ma	0.871v	81.59ma
0.918v	71.64ma	0.868v	72.65ma	0.922v	81.07ma	0.871v	82.09ma
0.918v	72.14ma	0.868v	73.14ma				

intel®

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type					Curve Type			
IV3		IV4			IV3		IV4	
Vd	Id	Vd	Id		Vd	Id	Vd	Id
0v	0a	0v	0a		0.696v	83ua	0.693v	0.14ma
25mv	.021pa	25mv	.013pa		0.716v	0.17ma	0.711v	0.29ma
50mv	.024pa	50mv	.013pa		0.734v	0.33ma	0.724v	0.52ma
75mv	.032pa	75mv	.014pa		0.748v	0.55ma	0.735v	0.81ma
100mv	.051pa	100mv	.014pa		0.759v	0.82ma	0.744v	0.81ma
0.125v	.098pa	0.125v	.016pa		0.768v	1.14ma	0.751v	1.13ma
0.15v	0.21pa	0.15v	.021pa		0.776v	1.14ma	0.756v	1.55ma
0.175v	0.5pa	0.175v	.037pa		0.783v	1.48ma	0.76v	1.97ma
0.2v	1.21pa	0.2v	.083pa		0.788v	1.9ma	0.763v	2.36ma
0.225v	2.98pa	0.225v	0.22pa		0.792v	2.31ma	0.767v	2.77ma
0.25v	7.35pa	0.25v	0.62pa		0.796v	2.7ma	0.77v	3.2ma
0.275v	18.22pa	0.275v	1.79pa		0.799v	3.12ma	0.772v	3.64ma
0.3v	45.17pa	0.3v	5.26pa		0.803v	3.54ma	0.775v	4.09ma
0.325v	0.11na	0.325v	15.47pa		0.805v	3.98ma	0.777v	4.54ma
0.35v	0.28na	0.35v	45.6pa		0.808v	4.42ma	0.779v	4.99ma
0.375v	0.69na	0.375v	0.13na		0.81v	4.86ma	0.781v	5.45ma
0.4v	1.71na	0.4v	0.4na		0.813v	5.31ma	0.782v	5.91ma
0.425v	4.25na	0.425v	1.17na		0.815v	5.77ma	0.784v	6.37ma
0.45v	10.85na	0.45v	3.45na		0.817v	6.22ma	0.786v	6.84ma
0.475v	26.93na	0.475v	10.68na		0.819v	6.68ma	0.787v	7.31ma
0.5v	66.83na	0.5v	31.49na		0.82v	7.15ma	0.788v	7.78ma
0.525v	0.17ua	0.525v	92.86na		0.822v	7.61ma	0.79v	8.25ma
0.55v	0.41ua	0.55v	0.27ua		0.823v	8.08ma	0.791v	8.72ma
0.575v	1.02ua	0.575v	0.81ua		0.825v	8.55ma	0.792v	9.2ma
0.6v	2.52ua	0.6v	2.37ua		0.826v	9.01ma	0.793v	9.67ma
0.625v	6.22ua	0.625v	6.93ua		0.828v	9.49ma	0.794v	10.15ma
0.649v	15.2ua	0.649v	19.9ua		0.829v	9.96ma	0.795v	10.63ma
0.673v	36.33ua	0.672v	54.64ua		0.83v	10.43ma	0.796v	11.11ma

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type				Curve Type			
IV3		IV4		IV3		IV4	
Vd	Id	Vd	Id	Vd	Id	Vd	Id
0.831v	10.91ma	0.797v	11.59ma	0.853v	24.46ma	0.815v	25.22ma
0.832v	11.38ma	0.798v	12.07ma	0.853v	24.95ma	0.815v	25.71ma
0.834v	11.86ma	0.799v	12.55ma	0.854v	25.44ma	0.815v	26.2ma
0.835v	12.34ma	0.8v	13.03ma	0.854v	25.93ma	0.816v	26.69ma
0.836v	12.82ma	0.801v	13.51ma	0.855v	26.41ma	0.816v	27.19ma
0.837v	13.3ma	0.801v	14ma	0.855v	26.9ma	0.817v	27.68ma
0.838v	13.78ma	0.802v	14.48ma	0.856v	27.39ma	0.817v	28.17ma
0.838v	14.26ma	0.803v	14.97ma	0.856v	27.88ma	0.818v	28.66ma
0.839v	14.74ma	0.804v	15.45ma	0.857v	28.38ma	0.818v	29.15ma
0.84v	15.22ma	0.804v	15.94ma	0.857v	28.87ma	0.818v	29.65ma
0.841v	15.7ma	0.805v	16.42ma	0.858v	29.36ma	0.819v	30.14ma
0.842v	16.19ma	0.806v	16.91ma	0.858v	29.85ma	0.819v	30.63ma
0.843v	16.67ma	0.806v	17.39ma	0.859v	30.34ma	0.819v	31.12ma
0.843v	17.15ma	0.807v	17.88ma	0.859v	30.83ma	0.82v	31.62ma
0.844v	17.64ma	0.807v	18.37ma	0.86v	31.32ma	0.82v	32.11ma
0.845v	18.12ma	0.808v	18.86ma	0.86v	31.81ma	0.82v	32.6ma
0.846v	18.61ma	0.809v	19.34ma	0.86v	32.3ma	0.821v	33.09ma
0.846v	19.1ma	0.809v	19.83ma	0.861v	32.79ma	0.821v	33.59ma
0.847v	19.58ma	0.81v	20.32ma	0.861v	33.29ma	0.821v	34.08ma
0.848v	20.07ma	0.81v	20.81ma	0.862v	33.78ma	0.822v	34.57ma
0.848v	20.55ma	0.811v	21.3ma	0.862v	34.27ma	0.822v	35.07ma
0.849v	21.04ma	0.811v	21.79ma	0.862v	34.76ma	0.822v	35.56ma
0.849v	21.53ma	0.812v	22.28ma	0.863v	35.25ma	0.823v	36.05ma
0.85v	22.02ma	0.812v	22.77ma	0.863v	35.75ma	0.823v	36.55ma
0.851v	22.5ma	0.813v	23.26ma	0.864v	36.24ma	0.823v	37.04ma
0.851v	22.99ma	0.813v	23.75ma	0.864v	36.73ma	0.824v	37.54ma
0.852v	23.48ma	0.814v	24.24ma	0.864v	37.22ma	0.824v	38.03ma
0.852v	23.97ma	0.814v	24.73ma	0.865v	37.72ma	0.824v	38.52ma

intel®

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type				Curve Type			
IV3		IV4		IV3		IV4	
Vd	Id	Vd	Id	Vd	Id	Vd	Id
0.865v	38.21ma	0.825v	39.02ma	0.873v	51.54ma	0.831v	52.38ma
0.865v	38.7ma	0.825v	39.51ma	0.873v	52.04ma	0.832v	52.88ma
0.866v	39.2ma	0.825v	40.01ma	0.874v	52.53ma	0.832v	53.37ma
0.866v	39.69ma	0.825v	40.5ma	0.874v	53.03ma	0.832v	53.87ma
0.866v	40.18ma	0.826v	40.99ma	0.874v	53.52ma	0.832v	54.36ma
0.867v	40.68ma	0.826v	41.49ma	0.874v	54.02ma	0.832v	54.86ma
0.867v	41.17ma	0.826v	41.98ma	0.875v	54.51ma	0.833v	55.35ma
0.867v	41.66ma	0.826v	42.48ma	0.875v	55.01ma	0.833v	55.85ma
0.868v	42.16ma	0.827v	42.97ma	0.875v	55.5ma	0.833v	56.35ma
0.868v	42.65ma	0.827v	43.47ma	0.875v	56ma	0.833v	56.84ma
0.868v	43.14ma	0.827v	43.96ma	0.876v	56.49ma	0.833v	57.34ma
0.869v	43.64ma	0.828v	44.46ma	0.876v	56.99ma	0.834v	57.83ma
0.869v	44.13ma	0.828v	44.95ma	0.876v	57.48ma	0.834v	58.33ma
0.869v	44.62ma	0.828v	45.45ma	0.876v	57.98ma	0.834v	58.83ma
0.869v	45.12ma	0.828v	45.94ma	0.877v	58.47ma	0.834v	59.32ma
0.87v	45.61ma	0.829v	46.44ma	0.877v	58.97ma	0.834v	59.82ma
0.87v	46.11ma	0.829v	46.93ma	0.877v	59.47ma	0.835v	60.31ma
0.87v	46.6ma	0.829v	47.43ma	0.877v	59.96ma	0.835v	60.81ma
0.871v	47.09ma	0.829v	47.92ma	0.877v	60.46ma	0.835v	61.31ma
0.871v	47.59ma	0.829v	48.42ma	0.878v	60.95ma	0.835v	61.8ma
0.871v	48.08ma	0.83v	48.91ma	0.878v	61.45ma	0.835v	62.3ma
0.872v	48.58ma	0.83v	49.41ma	0.878v	61.94ma	0.835v	62.8ma
0.872v	49.07ma	0.83v	49.9ma	0.878v	62.44ma	0.836v	63.29ma
0.872v	49.57ma	0.83v	50.4ma	0.879v	62.93ma	0.836v	63.79ma
0.872v	50.06ma	0.831v	50.89ma	0.879v	63.43ma	0.836v	64.28ma
0.873v	50.55ma	0.831v	51.39ma	0.879v	63.93ma	0.836v	64.78ma
0.873v	51.05ma	0.831v	51.88ma	0.879v	64.42ma	0.836v	65.28ma
				0.879v	64.92ma	0.837v	65.77ma

int_{el} ®

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type					Curve Type			
IV3		IV4			IV3		IV4	
Vd	Id	Vd	Id		Vd	Id	Vd	Id
0.88v	65.41ma	0.837v	66.27ma		0.883v	73.85ma	0.839v	74.71ma
0.88v	65.91ma	0.837v	66.77ma		0.883v	74.34ma	0.84v	75.21ma
0.88v	66.4ma	0.837v	67.26ma		0.883v	74.84ma	0.84v	75.71ma
0.88v	66.9ma	0.837v	67.76ma		0.883v	75.33ma	0.84v	76.21ma
0.88v	67.4ma	0.837v	68.26ma		0.884v	75.83ma	0.84v	76.7ma
0.881v	67.89ma	0.838v	68.75ma		0.884v	76.33ma	0.84v	77.2ma
0.881v	68.39ma	0.838v	69.25ma		0.884v	76.82ma	0.84v	77.7ma
0.881v	68.88ma	0.838v	69.75ma		0.884v	77.32ma	0.84v	78.19ma
0.881v	69.38ma	0.838v	70.24ma		0.884v	77.82ma	0.841v	78.69ma
0.881v	69.88ma	0.838v	70.74ma		0.885v	78.31ma	0.841v	79.19ma
0.882v	70.37ma	0.838v	71.24ma		0.885v	78.81ma	0.841v	79.68ma
0.882v	70.87ma	0.839v	71.73ma		0.885v	79.31ma	0.841v	80.18ma
0.882v	71.36ma	0.839v	72.23ma		0.885v	79.8ma	0.841v	80.68ma
0.882v	71.86ma	0.839v	72.73ma		0.885v	80.3ma	0.841v	81.18ma
0.882v	72.36ma	0.839v	73.22ma		0.885v	80.8ma	0.841v	81.67ma
0.883v	72.85ma	0.839v	73.72ma		0.886v	81.29ma	0.842v	82.17ma
0.883v	73.35ma	0.839v	74.22ma		0.886v	81.79ma	0.842v	82.67ma

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type					Curve Type			
IV5		IV6			IV5		IV6	
Vd	Id	Vd	Id		Vd	Id	Vd	Id
0v	0a	0v	0a		0.675v	4.63ua	0.675v	3.43ua
25mv	.0009pa	25mv	.0008pa		0.699v	11.35ua	0.7v	9.97ua
50mv	.0012pa	50mv	.0008pa		0.724v	27.37ua	0.724v	28.32ua
75mv	.0022pa	75mv	.0008pa		0.747v	63.69ua	0.746v	75.71ua
100mv	.0044pa	100mv	.0009pa		0.768v	0.14ma	0.766v	0.18ma
0.125v	.0100pa	0.125v	.001pa		0.787v	0.27ma	0.782v	0.36ma
0.15v	.0239pa	0.15v	.0013pa		0.802v	0.47ma	0.795v	0.61ma
0.175v	.0584pa	0.175v	.0022pa		0.814v	0.73ma	0.804v	0.92ma
0.2v	0.14pa	0.2v	.0047pa		0.823v	1.03ma	0.812v	0.92ma
0.225v	0.36pa	0.225v	.0124pa		0.832v	1.03ma	0.819v	1.25ma
0.25v	0.88pa	0.25v	.0348pa		0.839v	1.36ma	0.824v	1.68ma
0.275v	2.19pa	0.275v	0.1pa		0.845v	1.77ma	0.828v	2.1ma
0.3v	5.44pa	0.3v	0.3pa		0.849v	2.18ma	0.831v	2.5ma
0.325v	13.49pa	0.325v	0.87pa		0.853v	2.57ma	0.834v	2.92ma
0.35v	33.48pa	0.35v	2.57pa		0.857v	2.98ma	0.837v	3.35ma
0.375v	83.09pa	0.375v	7.59pa		0.86v	3.4ma	0.84v	3.79ma
0.4v	0.21na	0.4v	22.37pa		0.863v	3.84ma	0.842v	4.24ma
0.425v	0.51na	0.425v	65.99pa		0.865v	4.27ma	0.844v	4.69ma
0.45v	1.27na	0.45v	0.19na		0.868v	4.72ma	0.846v	5.14ma
0.475v	3.15na	0.475v	0.57na		0.87v	5.17ma	0.848v	5.6ma
0.5v	8.05na	0.5v	1.69na		0.872v	5.62ma	0.849v	6.07ma
0.525v	19.97na	0.525v	5.24na		0.874v	6.07ma	0.851v	6.53ma
0.55v	49.56na	0.55v	15.46na		0.876v	6.53ma	0.853v	7ma
0.575v	0.12ua	0.575v	45.58na		0.878v	6.99ma	0.854v	7.47ma
0.6v	0.31ua	0.6v	0.13ua		0.88v	7.46ma	0.855v	7.94ma
0.625v	0.76ua	0.625v	0.4ua		0.881v	7.92ma	0.857v	8.41ma
0.65v	1.87ua	0.65v	1.17ua		0.883v	8.39ma	0.858v	8.88ma

intel®

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type					Curve Type			
IV5		IV6			IV5		IV6	
Vd	Id	Vd	Id		Vd	Id	Vd	Id
0.884v	8.86ma	0.859v	9.36ma		0.908v	21.86ma	0.878v	22.45ma
0.885v	9.33ma	0.86v	9.83ma		0.909v	22.34ma	0.879v	22.94ma
0.887v	9.8ma	0.861v	10.31ma		0.909v	22.83ma	0.879v	23.43ma
0.888v	10.28ma	0.862v	10.79ma		0.91v	23.32ma	0.88v	23.92ma
0.889v	10.75ma	0.863v	11.27ma		0.91v	23.81ma	0.88v	24.41ma
0.89v	11.23ma	0.864v	11.75ma		0.911v	24.3ma	0.881v	24.9ma
0.891v	11.7ma	0.865v	12.23ma		0.911v	24.79ma	0.881v	25.39ma
0.892v	12.18ma	0.866v	12.71ma		0.912v	25.27ma	0.882v	25.88ma
0.894v	12.66ma	0.867v	13.2ma		0.913v	25.76ma	0.882v	26.37ma
0.894v	13.14ma	0.867v	13.68ma		0.913v	26.25ma	0.883v	26.86ma
0.895v	13.62ma	0.868v	14.16ma		0.914v	26.74ma	0.883v	27.35ma
0.896v	14.1ma	0.869v	14.65ma		0.914v	27.23ma	0.883v	27.85ma
0.897v	14.58ma	0.87v	15.13ma		0.915v	27.72ma	0.884v	28.34ma
0.898v	15.06ma	0.87v	15.62ma		0.915v	28.21ma	0.884v	28.83ma
0.899v	15.55ma	0.871v	16.1ma		0.915v	28.7ma	0.885v	29.32ma
0.9v	16.03ma	0.872v	16.59ma		0.916v	29.19ma	0.885v	29.81ma
0.901v	16.51ma	0.872v	17.07ma		0.916v	29.69ma	0.885v	30.31ma
0.901v	17ma	0.873v	17.56ma		0.917v	30.18ma	0.886v	30.8ma
0.902v	17.48ma	0.874v	18.05ma		0.917v	30.67ma	0.886v	31.29ma
0.903v	17.96ma	0.874v	18.54ma		0.918v	31.16ma	0.886v	31.78ma
0.904v	18.45ma	0.875v	19.02ma		0.918v	31.65ma	0.887v	32.28ma
0.904v	18.94ma	0.875v	19.51ma		0.919v	32.14ma	0.887v	32.77ma
0.905v	19.42ma	0.876v	20ma		0.919v	32.63ma	0.887v	33.26ma
0.906v	19.91ma	0.876v	20.49ma		0.919v	33.12ma	0.888v	33.76ma
0.906v	20.39ma	0.877v	20.98ma		0.92v	33.62ma	0.888v	34.25ma
0.907v	20.88ma	0.877v	21.47ma		0.92v	34.11ma	0.888v	34.74ma
0.907v	21.37ma	0.878v	21.96ma		0.921v	34.6ma	0.889v	35.24ma

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type				Curve Type			
IV5		IV6		IV5		IV6	
Vd	Id	Vd	Id	Vd	Id	Vd	Id
0.921v	35.09ma	0.889v	35.73ma	0.93v	48.41ma	0.896v	49.08ma
0.921v	35.58ma	0.889v	36.22ma	0.93v	48.91ma	0.896v	49.58ma
0.922v	36.08ma	0.89v	36.72ma	0.93v	49.4ma	0.897v	50.07ma
0.922v	36.57ma	0.89v	37.21ma	0.93v	49.9ma	0.897v	50.57ma
0.922v	37.06ma	0.89v	37.7ma	0.931v	50.39ma	0.897v	51.06ma
0.923v	37.56ma	0.891v	38.2ma	0.931v	50.89ma	0.897v	51.56ma
0.923v	38.05ma	0.891v	38.69ma	0.931v	51.38ma	0.898v	52.05ma
0.923v	38.54ma	0.891v	39.19ma	0.932v	51.88ma	0.898v	52.55ma
0.924v	39.03ma	0.891v	39.68ma	0.932v	52.37ma	0.898v	53.04ma
0.924v	39.53ma	0.892v	40.17ma	0.932v	52.87ma	0.898v	53.54ma
0.924v	40.02ma	0.892v	40.67ma	0.932v	53.36ma	0.898v	54.04ma
0.925v	40.51ma	0.892v	41.16ma	0.933v	53.86ma	0.899v	54.53ma
0.925v	41.01ma	0.893v	41.66ma	0.933v	54.35ma	0.899v	55.03ma
0.925v	41.5ma	0.893v	42.15ma	0.933v	54.85ma	0.899v	55.52ma
0.926v	41.99ma	0.893v	42.65ma	0.933v	55.34ma	0.899v	56.02ma
0.926v	42.49ma	0.893v	43.14ma	0.934v	55.84ma	0.899v	56.52ma
0.926v	42.98ma	0.894v	43.64ma	0.934v	56.33ma	0.9v	57.01ma
0.927v	43.47ma	0.894v	44.13ma	0.934v	56.83ma	0.9v	57.51ma
0.927v	43.97ma	0.894v	44.63ma	0.934v	57.32ma	0.9v	58ma
0.927v	44.46ma	0.894v	45.12ma	0.934v	57.82ma	0.9v	58.5ma
0.928v	44.96ma	0.895v	45.62ma	0.935v	58.31ma	0.9v	59ma
0.928v	45.45ma	0.895v	46.11ma	0.935v	58.81ma	0.901v	59.49ma
0.928v	45.94ma	0.895v	46.61ma	0.935v	59.3ma	0.901v	59.99ma
0.929v	46.44ma	0.895v	47.1ma	0.935v	59.8ma	0.901v	60.48ma
0.929v	46.93ma	0.896v	47.6ma	0.936v	60.29ma	0.901v	60.98ma
0.929v	47.43ma	0.896v	48.09ma	0.936v	60.79ma	0.901v	61.48ma
0.929v	47.92ma	0.896v	48.59ma	0.936v	61.28ma	0.902v	61.97ma

intel®

Table 8-6. Data for Diode I-V Curves (Contd.)

Curve Type					Curve Type			
IV5		IV6			IV5		IV6	
Vd	Id	Vd	Id		Vd	Id	Vd	Id
0.936v	61.78ma	0.902v	62.47ma		0.94v	71.7ma	0.905v	72.4ma
0.937v	62.28ma	0.902v	62.97ma		0.941v	72.19ma	0.905v	72.9ma
0.937v	62.77ma	0.902v	63.46ma		0.941v	72.69ma	0.905v	73.39ma
0.937v	63.27ma	0.902v	63.96ma		0.941v	73.19ma	0.906v	73.89ma
0.937v	63.76ma	0.902v	64.45ma		0.941v	73.68ma	0.906v	74.39ma
0.937v	64.26ma	0.903v	64.95ma		0.941v	74.18ma	0.906v	74.88ma
0.938v	64.75ma	0.903v	65.45ma		0.941v	74.67ma	0.906v	75.38ma
0.938v	65.25ma	0.903v	65.94ma		0.942v	75.17ma	0.906v	75.88ma
0.938v	65.75ma	0.903v	66.44ma		0.942v	75.67ma	0.906v	76.38ma
0.938v	66.24ma	0.903v	66.94ma		0.942v	76.16ma	0.907v	76.87ma
0.938v	66.74ma	0.904v	67.43ma		0.942v	76.66ma	0.907v	77.37ma
0.939v	67.23ma	0.904v	67.93ma		0.942v	77.16ma	0.907v	77.87ma
0.939v	67.73ma	0.904v	68.43ma		0.943v	77.65ma	0.907v	78.36ma
0.939v	68.22ma	0.904v	68.92ma		0.943v	78.15ma	0.907v	78.86ma
0.939v	68.72ma	0.904v	69.42ma		0.943v	78.65ma	0.907v	79.36ma
0.939v	69.22ma	0.904v	69.92ma		0.943v	79.14ma	0.907v	79.85ma
0.94v	69.71ma	0.905v	70.41ma		0.943v	79.64ma	0.908v	80.35ma
0.94v	70.21ma	0.905v	70.91ma		0.943v	80.14ma	0.908v	80.85ma
0.94v	70.7ma	0.905v	71.41ma		0.944v	80.63ma	0.908v	81.35ma
0.94v	71.2ma	0.905v	71.9ma					

intel ®

9

Mechanical
Specifications

intel®

CHAPTER 9
MECHANICAL SPECIFICATIONS

The Pentium processor is packaged in a 273 pin ceramic pin grid array (PGA). The pins are arranged in a 21 by 21 matrix and the package dimensions are 2.16" X 2.16" (Table 9-1).

Table 9-1. Pentium™ Processor Package Information Summary

	Package Type	Total Pins	Pin Array	Package Size	Estimated Wattage
Pentium™ Processor	PGA	273	21 x 21	2.16" X 2.16" 5.49cm X 5.49 cm	16

NOTE: See D.C. Specifications for more detailed power specifications.

Figure 9-1 shows the package dimensions for the Pentium processor. The mechanical specifications are provided in Table 9-2.
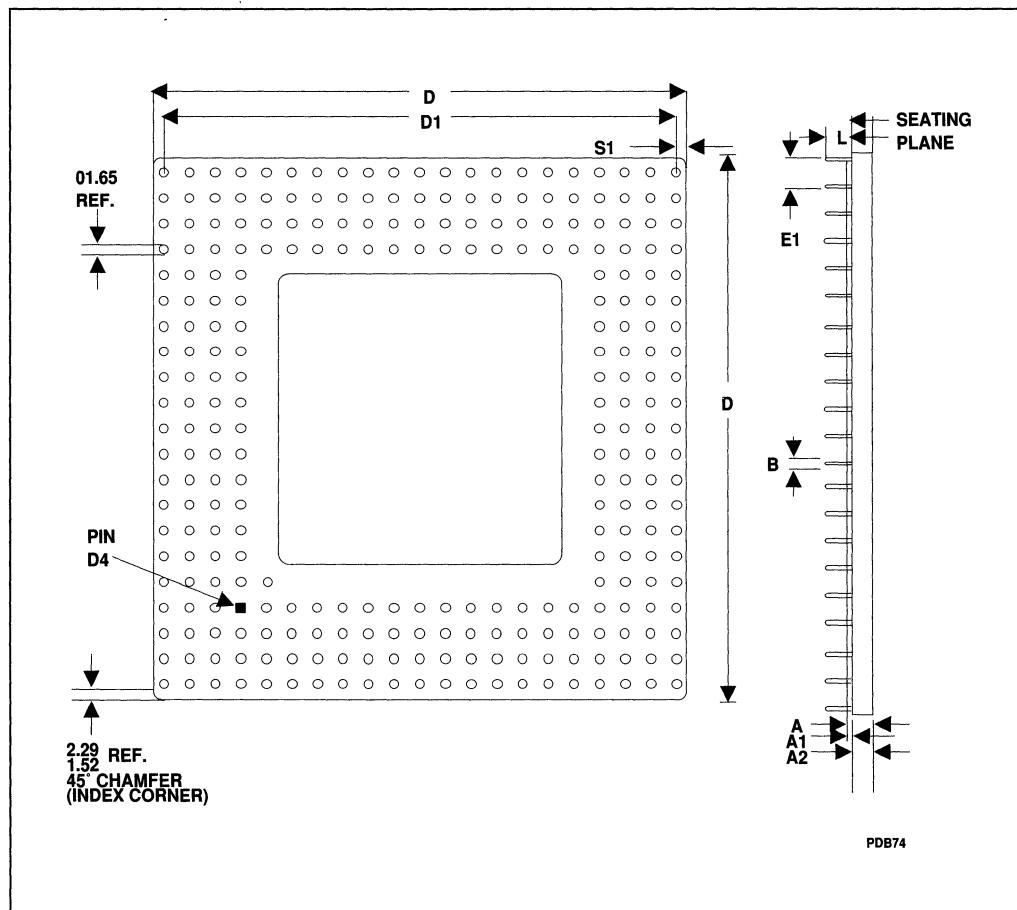
Figure 9-1. Pentium™ Processor Package Dimensions

Table 9-2. Pentium™ Processor Mechanical Specificatons

Family: Ceramic Pin Grid Array Package						
	Millimeters			**Inches**		
Symbol	**Min**	**Max**	**Notes**	**Min**	**Max**	**Notes**
A	2.84	3.51	Solid Lid	0.112	0.138	Solid Lid
A1	0.33	0.43	Solid Lid	0.013	0.017	Solid Lid
A2	2.51	3.07		0.099	0.121	
B	0.43	0.51		0.017	0.020	
D	54.61	55.11		2.150	2.170	
D1	50.67	50.93		1.995	2.005	
e1	2.29	2.79		0.090	0.110	
L	3.05	3.30		0.120	0.130	
N	273			273		
S1	1.65	2.16		0.065	0.085	

intel®

10

Thermal
Specifications

intel®

CHAPTER 10
THERMAL SPECIFICATIONS

The Pentium processor is specified for proper operation when T_C (case temperature) is within the specified range of $0^\circ C$ to $85^\circ C$. To verify that the proper T_C is maintained, it should be measured at the center of the top surface (opposite of the pins) of the device in question. To minimize the measurement errors, it is recommended to use the following approach:

- Use 36 gauge or finer diameter k, t, or j type thermocouples. The laboratory testing was done using a thermocouple made by Omega (part number: 5TC-TTK-36-36).

- Attach the thermocouple bead or junction to the center of the package top surface using high thermal conductivity cements. The laboratory testing was done by using Omega Bond (part number: OB-100).

- The thermocouple should be attached at a 90 degrees angle as shown in Figure 10-1. The measurement is made in the same way with or without a heatsink attached. When a heat sink is attached a hole should be drilled through the heat sink to allow probing the center of the package.

- If the case temperature is measured with a heat sink attached to the package, provide a shallow grove on the contact surface of the heat sink to route the thermocouple wire out.

CSJ23A
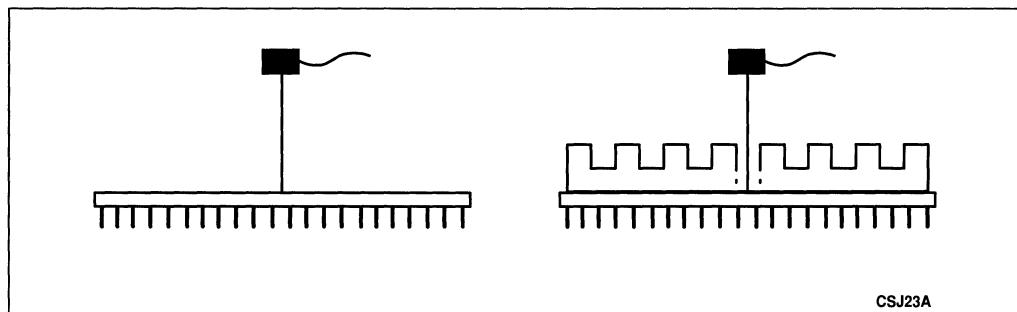
Figure 10-1. Technique for Measuring Tcase

An ambient temperature T_A is not specified directly. The only restriction is that T_C is met. To determine the allowable T_A values, the following equations may be used:

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

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

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

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

where, T_J, T_A, and T_C = Junction, Ambient and Case Temperature, respectively.

Θ_{JC}, Θ_{JA}, and Θ_{CA} = Junction-to-Case, Junction-to-Ambient, and Case-to-Ambient Thermal Resistance, respectively.

P = Maximum Power Consumption

Table 10-1 lists the Θ_{JC} and Θ_{CA} values for the Pentium processor.

Table 10-1. Junction-to-Case and Case-to-Ambient Thermal Resistances for the Pentium™ Processor (With and Without a Heat Sink)

	Θ_{JC}	Θ_{CA} vs Airflow (ft/min)					
		0	**200**	**400**	**600**	**800**	**1000**
With 0.25" Heat Sink	0.6	8.3	5.8	3.9	3.0	2.5	2.2
With 0.35" Heat Sink	0.6	7.9	5.0	3.4	2.8	2.2	2.0
With 0.65" Heat Sink	0.6	6.4	3.4	2.3	1.8	1.5	1.3
Without Heat Sink	1.2	11.6	9.4	6.7	5.4	4.6	4.2

Heat Sink: 2.05 sq. in. omni-directional pin Al heat sink with 0.050 in. pin width, 0.143 in pin-to-pin center spacing and 0.150 in. base thickness.

intel®

11

Testability

intel ®

This chapter describes the features which are included in the Pentium processor or the purpose of enhancing the testability of the Pentium processor. The capability of the Intel486 CPU test hooks are included in the Pentium processor, however some are implemented differently. In addition, new test features were added to assure timely testing and production of the system product.

Internal component testing through the Built In Self Test (BIST) feature of the Pentium processor provides 100% single stuck at fault coverage of the microcode ROM and large PLAs. Some testing of the instruction cache, data cache, Translation Lookaside Buffers (TLBs), and Branch Target Buffer (BTB) is also performed. In addition, the constant ROMs are checked.

Tristate test mode and the IEEE 1149.1 "Test Access Port and Boundary Scan" mechanism are included to facilitate testing of board connections.

See Appendix A for more information regarding the testing of the on chip caches, translation lookaside buffers, branch target buffer, second level caches, the superscalar architecture, and internal parity checking through the test registers.

Built in self test, tristate test mode, Boundary Scan, and TR12 are discussed in this chapter.

11.1. BUILT IN SELF TEST (BIST)

Self test is initiated by driving the INIT pin high when RESET transitions from high to low.

No bus cycles are run by the Pentium processor during self test. The duration of self test is approximately 2^{19} clocks. Approximately 70% of the devices in the Pentium processor are tested by BIST.

The Pentium processor BIST consists of two parts: hardware self test and microcode self test.

During the hardware portion of BIST, the microcode and all large PLAs are tested. All possible input combinations of the microcode ROM and PLAs are tested.

The constant ROMs, BTB, TLBs, and all caches are tested by the microcode portion of BIST. The array tests (caches, TLBs, and BTB) have two passes. On the first pass, data patterns are written to arrays, read back and checked for mismatches. The second pass writes the complement of the initial data pattern, reads it back, and checks for mismatches. The constant ROMs are tested by using the microcode to add various constants and check the result against a stored value.

Upon completion of BIST, the cumulative result of all tests are stored in the EAX register. If EAX contains 0h, then all checks passed; any non-zero result indicates a faulty unit. Note that if an internal parity error is detected during BIST, the processor will assert the IERR# pin and attempt to shutdown.

intel ®

11.2. TRISTATE TEST MODE

When the FLUSH# pin is sampled low in the clock prior to the RESET pin going from high to low, the Pentium processor enters tristate test mode. The Pentium processor floats all of its output pins and bi-directional pins including pins which are never floated during normal operation (except TDO). Tristate test mode can be initiated in order to facilitate testing of board connections. The Pentium processor remains in tristate test mode until the RESET pin is toggled again.

11.3. IEEE 1149.1 TEST ACCESS PORT AND BOUNDARY SCAN MECHANISM

The IEEE Standard Test Access Port and Boundary Scan Architecture (Standard 1149.1) is implemented in the Pentium processor. This feature allows board manufacturers to test board interconnects by using "boundary scan", and to test the Pentium processor itself through BIST. All output pins are tristateable through the IEEE 1149.1 mechanism. The test access port mechanism is also used in the new debug mode implemented in the Pentium processor, Probe Mode. See the Probe Mode chapter for details.

11.3.1. Pentium Processor Test Access Port (TAP)

The Pentium processor Test Access Port (TAP) contains a TAP controller, a Boundary Scan Register, a Probe Data Register, a Probe Instruction Register, 4 input pins (TDI, TCK, TMS, and TRST#), and one output pin (TDO). The TAP controller consists of an Instruction Register, a Device ID Register, a Bypass Register, a Runbist Register, and control logic. See Figure 11-1 for the TAP Block Diagram.
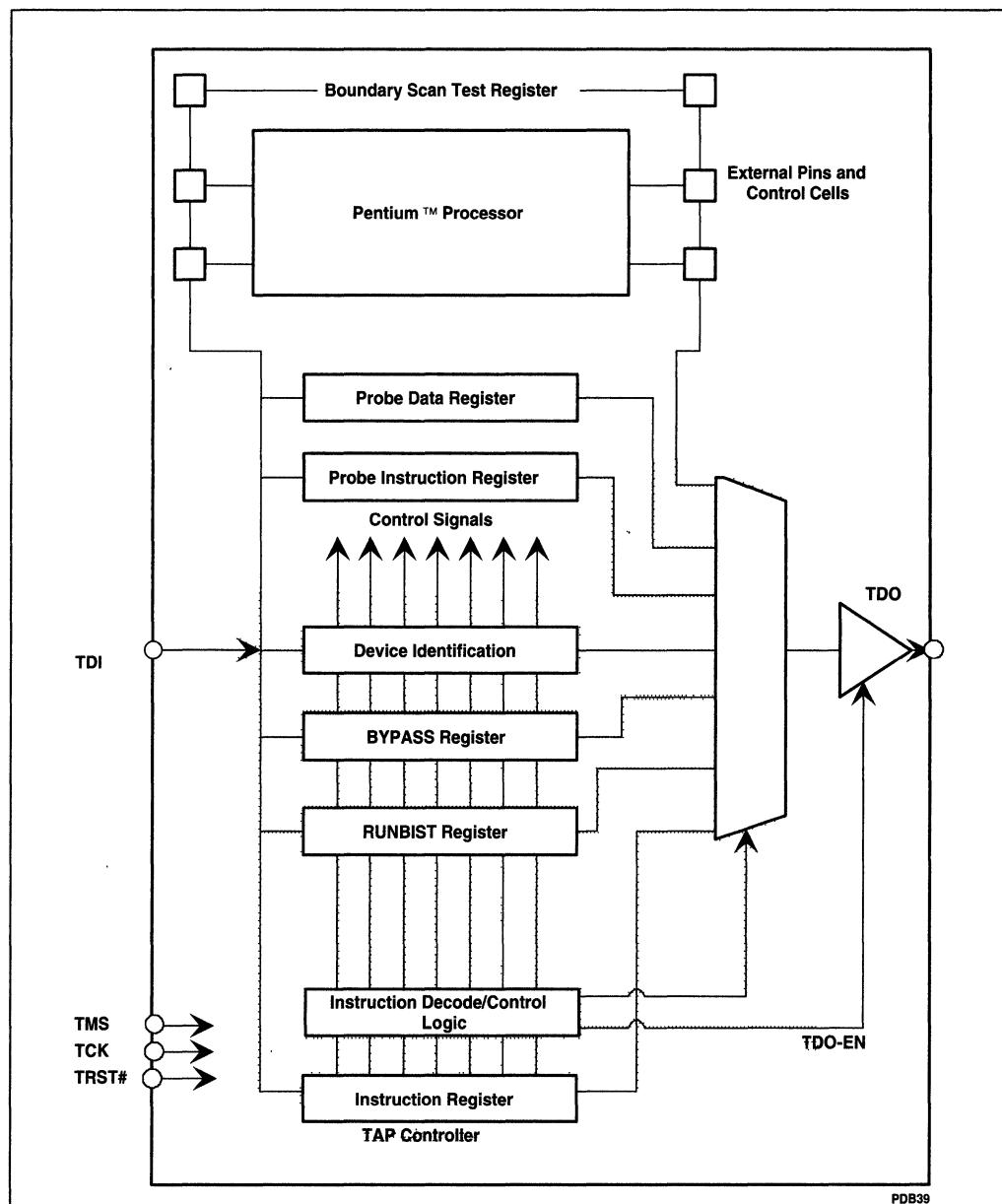
I

Figure 11-1. Test Access Port Block Diagram

11.3.1.1.   TAP PINS

As mentioned in the previous section, the TAP includes 4 input pins and one output pin. TDI (test data in) is used to shift data or instructions into the TAP in a serial manner. TDO (test data

out) shifts out the response data. TMS (test mode select) is used to control the state of the TAP controller. TCK is the test clock. The TDI and TMS inputs are sampled on the rising edge of this TCK. Asserting TRST# will force the TAP controller into the Test Logic Reset State (see the TAP controller state diagram, Figure 11-4). The input pins (TDI, TMS, TCK, and TRST#) have pullup resistors.

11.3.1.2. TAP REGISTERS

Boundary Scan Register

The IEEE standard requires that an extra single bit shift register be inserted at each pin on the device (Pentium processor). These single bit shift registers are connected into a long shift register, the Boundary Scan Register. Therefore, the Boundary Scan Register is a single shift register path containing the boundary scan cells that are connected to all input and output pins of the Pentium processor. Figure 11-2 shows the logical structure of the Boundary Scan Register. While output cells determine the value of the signal driven on the corresponding pin, input cells only capture data; they do not affect the normal operation of the device (the INTEST instruction is not supported by the Pentium processor). Data is transferred without inversion from TDI to TDO through the Boundary Scan Register during scanning. The Boundary Scan Register can be operated by the EXTEST and SAMPLE/PRELOAD instructions. The Boundary Scan Register order is defined later in this chapter.
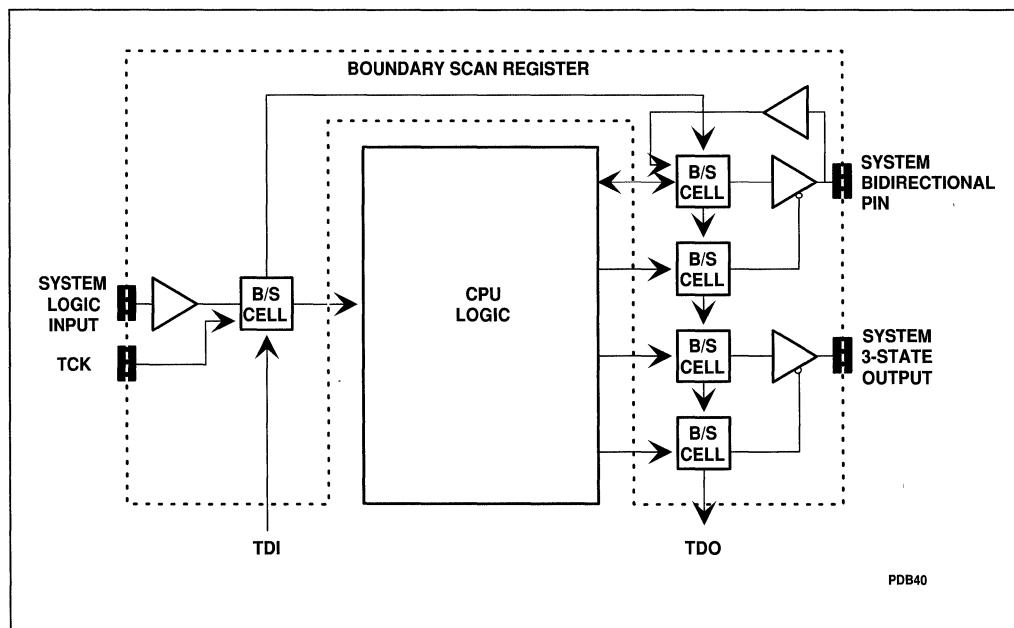

Figure 11-2. Boundary Scan Register

BYPASS Register

The Bypass Register is a one-bit shift register that provides the minimal length path between TDI and TDO. This path can be selected when no test operation is being performed by the component to allow rapid movement of test data to and from other components on the board. While the bypass register is selected data is transferred from TDI to TDO without inversion. The Bypass Register loads a logic 0 at the start of a scan cycle.

Device ID Register

The Device Identification Register contains the manufacturer's identification code, part number code, and version code in the format shown in Figure 11-3.
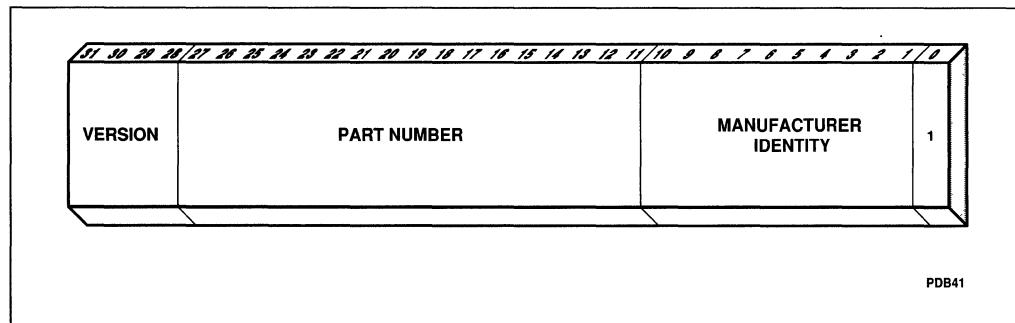
Figure 11-3. Format of the Device ID Register

The Pentium processor has divided up the 16-bit part number into 3 fields. The upper 7 bits are used to define the product type (examples: Cache, CPU architecture). The middle 4-bits are used to represent the generation or family (examples: Intel486 CPU, Pentium processor). The lower 5 bits are used to represent the model (examples: SX, DX). Using this definition, the Pentium processor ID code is shown in Table 11-1.

The version field is used to indicate the stepping ID.

Table 11-1. Device ID Register Values

| Stepping | Version | Part Number | | | Manufacturing ID | "1" | Entire Code |
		Product Type	Generation	Model			
x	xh	01h	05h	01h	09h	1	x02A1013h

Runbist Register

The Runbist Register is a one bit register used to report the results of the Pentium processor BIST when it is initiated by the RUNBIST instruction. This register is loaded with "0" upon successful completion of BIST.

Instruction Register

This register is 13-bits wide. The command field (the lower 4-bits of instruction) is used to indicate one of the following instructions: EXTEST, IDCODE, RUNBIST, SAMPLE/PRELOAD, and BYPASS. The upper 9-bits are reserved by Intel.

The most significant bit of the Instruction Register is connected to TDI, the least significant to TDO.

11.3.1.3.   TAP CONTROLLER STATE DIAGRAM

Figure 11-4 shows the 16-state TAP controller state diagram. A description of each state follows. Note that the state machine contains two main branches to access either data or instruction registers.
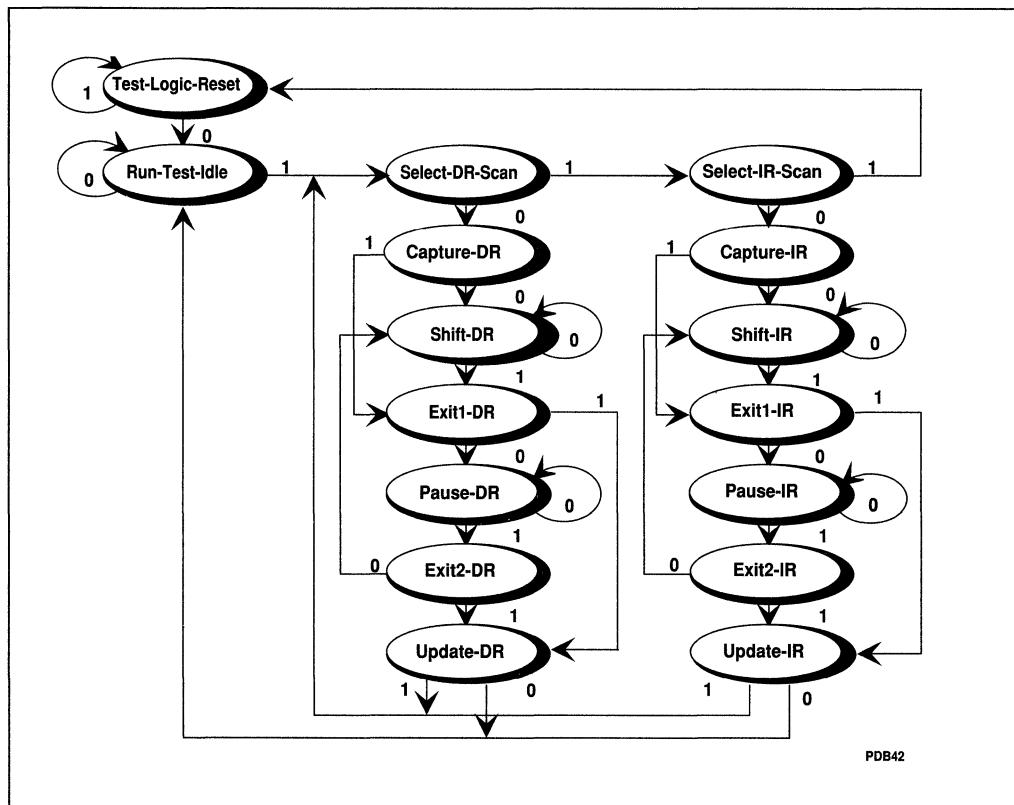
Figure 11-4. TAP Controller State Diagram

Test-Logic-Reset State

In this state, the test logic is disabled so that normal operation of the device can continue

unhindered. During initialization, the Pentium processor initializes the instruction register such that the IDCODE instruction is loaded.

No matter what the original state of the controller, the controller enters Test-Logic-Reset state when the TMS input is held high (logic 1) for at least five rising edges of TCK. The controller remains in this state while TMS is high. The TAP controller is forced to enter this state when the TRST# pin is asserted (with TCK toggling or TCK at a high logic value). The Pentium processor automatically enters this state at power-up.

Run-Test/Idle State

This is a controller state between scan operations. Once in this state, the controller remains in this state as long as TMS is held low. In devices supporting the RUNBIST instruction, the BIST is performed during this state and the result is reported in the Runbist Register. For instructions not causing functions to execute during this state, no activity occurs in the test logic. The instruction register and all test data registers retain their previous state. When TMS is high and a rising edge is applied to TCK, the controller moves to the Select-DR state.

Select-DR-Scan State

This is a temporary controller state. The test data register selected by the current instruction retains its previous state. If TMS is held low and a rising edge is applied to TCK when in this state, the controller moves into the Capture-DR state, and a scan sequence for the selected test data register is initiated. If TMS is held high and a rising edge is applied to TCK, the controller moves to the Select-IR-Scan state.

The instruction does not change in this state.

Capture-DR State

In this state, the Boundary Scan Register captures input pin data if the current instruction is EXTEST or SAMPLE/PRELOAD. The other test data registers, which do not have parallel input, are not changed.

The instruction does not change in this state.

When the TAP controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-DR state if TMS is high or the Shift-DR state if TMS is low.

Shift-DR State

In this controller state, the test data register connected between TDI and TDO as a result of the current instruction shifts data one stage toward its serial output on each rising edge of TCK.

The instruction does not change in this state.

When the TAP controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-DR state if TMS is high or remains in the Shift-DR state if TMS is low.

Exit1-DR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-DR state, which terminates the scanning process. If

TMS is held low and a rising edge is applied to TCK, the controller enters the Pause-DR state.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

Pause-DR State

The pause state allows the test controller to temporarily halt the shifting of data through the test data register in the serial path between TDI and TDO. An example use of this state could be to allow a tester to reload its pin memory from disk during application of a long test sequence.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

The controller remains in this state as long as TMS is low. When TMS goes high and a rising edge is applied to TCK, the controller moves to the Exit2-DR state.

Exit2-DR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-DR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Shift-DR state.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

Update-DR State

The Boundary Scan Register is provided with a latched parallel output to prevent changes at the parallel output while data is shifted in response to the EXTEST and SAMPLE/PRELOAD instructions. When the TAP controller is in this state and the Boundary Scan Register is selected, data is latched onto the parallel output of this register from the shift-register path on the falling edge of TCK. The data held at the latched parallel output does not change other than in this state.

All shift-register stages in the test data register selected by the current instruction retains their previous value during this state. The instruction does not change in this state.

Select-IR-Scan State

This is a temporary controller state. The test data register selected by the current instruction retains its previous state. If TMS is held low and a rising edge is applied to TCK when in this state, the controller moves into the Capture-IR state, and a scan sequence for the instruction register is initiated. If TMS is held high and a rising edge is applied to TCK, the controller moves to the Test-Logic-Reset state. The instruction does not change in this state.

Capture-IR State

In this controller state the shift register contained in the instruction register loads a fixed value on the rising edge of TCK.

The test data register selected by the current instruction retains its previous value during this

state. The instruction does not change in this state.

When the controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-IR state if TMS is held high, or the Shift-IR state if TMS is held low.

Shift-IR State

In this state the shift register contained in the instruction register is connected between TDI and TDO and shifts data one stage towards its serial output on each rising edge of TCK.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

When the controller is in this state and a rising edge is applied to TCK, the controller enters the Exit1-IR state if TMS is held high, or remains in the Shift-IR state if TMS is held low.

Exit1-IR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-IR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Pause-IR state.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

Pause-IR State

The pause state allows the test controller to temporarily halt the shifting of data through the instruction register.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

The controller remains in this state as long as TMS is low. When TMS goes high and a rising edge is applied to TCK, the controller moves to the Exit2-IR state.

Exit2-IR State

This is a temporary state. While in this state, if TMS is held high, a rising edge applied to TCK causes the controller to enter the Update-IR state, which terminates the scanning process. If TMS is held low and a rising edge is applied to TCK, the controller enters the Shift-IR state.

The test data register selected by the current instruction retains its previous value during this state. The instruction does not change in this state.

Update-IR State

The instruction shifted into the instruction register is latched onto the parallel output from the shift-register path on the falling edge of TCK. Once the new instruction has been latched, it becomes the current instruction.

Test data registers selected by the current instruction retain their previous value.

11.3.2. Boundary Scan

The IEEE Standard 1149.1 Boundary Scan is implemented using the Test Access Port and TAP Controller as described above. The Pentium processor implements all of the required boundary scan features as well as some additional features. The required pins are: TDI, TDO, TCK and TMS. The required registers are: Boundary Scan, Bypass, and the Instruction Register. Required instructions include: BYPASS, SAMPLE/PRELOAD and EXTEST. The additional pin, registers, and instructions are implemented to add additional test features and to support Probe Mode.

On the board level, the TAP provides a simple serial interface that makes it possible to test all signal traces with only a few probes. The testing is controlled through the TAP Controller State machine that can be implemented with automatic test equipment or a PLD.

On power up the TAP controller is automatically initialized to the test logic reset state (test logic disabled), so normal Pentium processor behavior is the default. The Test Logic Reset State is also entered when TRST# is asserted, or when TMS is high for 5 or more consecutive TCK clocks.

To implement boundary scan, the TDO of one device is connected to TDI of the next in a daisy chain fashion. This allows all of the I/O of the devices on this chain to be accessed through a long shift register. TMS and TCK are common to all devices.

The Boundary Scan Register for the Pentium processor contains a cell for each pin. The following is the bit order of the Pentium processor Boundary Scan Register (left to right, top to bottom):

TDI -> Reserved, Reserved, Reserved, RESET, FRCMC#, PEN#, R/S#, NMI, INTR, IGNNE#, SMI#, INIT, Reserved, CLK, Reserved, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, A23, A24, A25, A26, A27, A28, A29, A30, A31, BT0, Disabus*, BT1, BT2, BT3, BE7#, BE6#, BE5#, BE4#, BE3#, BE2#, BE1#, BE0#, SCYC, D/C#, PWT, PCD, W/R#, ADS#, ADSC#, PRDY, AP, LOCK#, HLDA, APCHK#, PCHK#, HIT#, HITM#, Disbus*, BREQ, SMIACT#, A20M#, FLUSH#, HOLD, WB/WT#, EWBE#, EADS#, BUSCHK#, AHOLD, BRDYC#, BRDY#, KEN#, NA#, INV, BOFF#, IU, IV, CACHE#, M/IO#, BP3, BP2, PM1/BP1, PM0/BP0, Dismisc*, FERR#, IERR#, Disfrc*, DP0, D0, D1, D2, D3, D4, D5, D6, D7, DP1, D8, D9, D10, D11, D12, D13, D14, D15, DP2, D16, D17, D18, D19, D20, D21, D22, D23, DP3, D24, D25, D26, D27, D28, D29, D30, D31, DP4, D32, D33, D34, D35, D36, Diswr*, D37, D38, D39, DP5, D40, D41, D42, D43, D44, D45, D46, D47, DP6, D48, D49, D50, D51, D52, D53, D54, D55, DP7, D56, D57, D58, D59, D60, D61, D62, D63, IBT -> TDO

"Reserved" includes the no connect "NC" signals on the Pentium processor.

The ADSC# and BRDYC# pins are part of the optimized interface between the Pentium processor and the 82496 Cache Controller/82491 Cache SRAM (Refer to the *82496 Cache Controller/82491 Cache SRAM Data Book for Use with the Pentium™ Processor* for further information).

The cells marked with * are control cells that are used to select the direction of bi-directional pins or tristate the output pins. If "1" is loaded into the control cell, the associated pin(s) are tristated or selected as input. The following lists the control cells and their corresponding pins:

Disabus: A31-A3, AP, BT3-BT0

Disbus: ADS#, BE7-0#, CACHE#, SCYC, M/IO#, D/C#, W/R#, PWT, PCD, LOCK#

Dismisc: BREQ, APCHK#, SMIACT#, PRDY, IU, IV, IBT, BP3, BP2, PM1/BP1, PM0/BP0,   FERR#, HITM#, HIT#, PCHK#, HLDA

Disfrc: IERR#

Diswr: D63-D0, DP7-0

11.3.2.1.   PENTIUM PROCESSOR BOUNDARY SCAN (TAP) INSTRUCTION SET

Table 11-2 shows the Pentium processor Boundary Scan TAP instructions and their instruction register encoding. A description of each instruction follows. The IDCODE and BYPASS instructions may also be executed concurrent with processor execution. The following instructions are not affected by the assertion of RESET: EXTEST, SAMPLE PRELOAD, BYPASS, and ID CODE.

The instructions should be scanned in to the TAP port least significant bit first (bit 0 of the TAP Command field is the first bit to be scanned in).

Table 11-2. TAP Instruction Set and Instruction Register Encoding

Instruction Name	Instruction Register bits 12:4	TAP Command Field [bits 3:0]
EXTEST	XXXXXXXXX	0000
Sample/Preload	XXXXXXXXX	0001
IDCODE	XXXXXXXXX	0010
Private Instruction	XXXXXXXXX	0011
Private Instruction	XXXXXXXXX	0100
Private Instruction	XXXXXXXXX	0101
Private Instruction	XXXXXXXXX	0110
RUNBIST	XXXXXXXXX	0111
Private Instruction	XXXXXXXXX	1000
Private Instruction	XXXXXXXXX	1001
Private Instruction	XXXXXXXXX	1010
Private Instruction	XXXXXXXXX	1011
Private Instruction	XXXXXXXXX	1100
BYPASS	XXXXXXXXX	1111

The TAP Command field encodings not listed in Table 11-2 (1101, 1110) are unimplemented and will be interpreted as Bypass instructions.

EXTEST	The EXTEST instruction allows testing of circuitry external to the component package, typically board interconnects. It does so by driving the values loaded into the Pentium processor's Boundary Scan Register out on the output pins corresponding to each boundary scan cell and capturing the values on the Pentium processor input pins to be loaded into their corresponding Boundary Scan Register locations. I/O pins are selected as input or output, depending on the value loaded into their control setting locations in the Boundary Scan Register. Values shifted into input latches in the Boundary Scan Register are never used by the internal logic of the Pentium processor. Note: after using the EXTEST instruction, the Pentium processor must be reset before normal (non-boundary scan) use.
SAMPLE/PRELOAD	The SAMPLE/PRELOAD performs two functions. When the TAP controller is in the Capture-DR state, the SAMPLE/PRELOAD instruction allows a "snap-shot" of the normal operation of the component without interfering with that normal operation. The instruction causes Boundary Scan Register cells associated with outputs to sample the value being driven by the Pentium processor. It causes the cells associated with inputs to sample the value being driven into the Pentium processor. On both outputs and inputs the sampling occurs on the rising edge of TCK. When the TAP controller is in the Update-DR state, the SAMPLE/PRELOAD instruction preloads data to the device pins to be driven to the board by executing the EXTEST instruction. Data is preloaded to the pins from the Boundary Scan Register on the falling edge of TCK.
IDCODE	The IDCODE instruction selects the device identification register to be connected to TDI and TDO. This allows the device identification code to be shifted out of the device on TDO.
BYPASS	The BYPASS instruction selects the Bypass Register to be connected to TDI and TDO. This effectively bypasses the test logic on the Pentium processor by reducing the shift length of the device to one bit. Note that an open circuit fault in the board level test data path will cause the Bypass Register to be selected following an instruction scan cycle due to a pull-up resistor on the TDI input. This was implemented to prevent any unwanted interference with the proper operation of the system logic.
RUNBIST	The RUNBIST instruction selects the one (1) bit Runbist Register, loads a value of "1" into the Runbist Register, and connects it to TDO. It also initiates the built-in self test (BIST) feature of the Pentium processor. After loading the RUNBIST instruction code in the instruction register, the TAP controller must be placed in the Run-Test/Idle state. BIST begins on the first rising edge of TCK after entering the Run-Test/Idle state. The TAP controller must remain in the Run-Test/Idle state until BIST is completed. It requires 2^{19} (CLK) cycles to complete BIST and report the result to the Runbist Register. After completing BIST, the value in the Runbist Register should be shifted out on TDO during the Shift-

DR state. A value of "0" being shifted out on TDO indicates BIST successfully completed. A value of "1" indicates a failure occurred. The CLK clock must be running in order to execute RUNBIST. After executing the RUNBIST instruction, the Pentium processor must be reset prior to normal (non-boundary scan) operation.

intel®

12

Error Detection

intel®

CHAPTER 12
ERROR DETECTION

The Pentium processor incorporates a number of data integrity features that are focused on the detection and limited recovery of errors. The data integrity features in the Pentium processor provide capabilities for error detection of the internal devices and the external interface. The Pentium processor also provides the capability to obtain maximum levels of error detection by incorporating Functional Redundancy Checking (FRC) support. Error detecting circuits in the Pentium processor do not limit the operating frequency of the chip.

The data integrity features in the Pentium processor can be categorized as (1) internal error detection, (2) error detection at the bus interface, and (3) FRC support.

12.1. INTERNAL ERROR DETECTION

Detection of errors of a majority of the devices in the Pentium processor is accomplished by employing parity checking in the large memory arrays of the chip. The data and instruction caches (both storage and tag arrays), translation lookaside buffers, and microcode ROM are all parity protected. The following describes the parity checking employed in the major memory arrays in the Pentium processor (MESI status bits are not parity protected):

- Parity bit per byte in the data cache storage array.
- Parity bit per entry in the data cache tag array.
- Parity bit per quarter line in the instruction cache storage array.
- Parity bit per entry in the instruction cache tag array.
- Parity bit per entry in both the data and instruction TLBs storage arrays.
- Parity bit per entry in both the data and instruction TLBs tag arrays.
- Parity bit per entry in the microcode ROM.

Parity checking as described above provides error detection coverage of 53% of the on chip devices. This error detection coverage number also includes the devices in the branch target buffer since branch predictions are always verified.

If a parity error has occurred internally, then the Pentium processor operation can no longer be trusted. Therefore, a parity error on a read from an internal array will cause the Pentium processor to assert the IERR# pin and then shutdown. (Shutdown will be entered assuming it is not prevented from doing so by the error.) Parity errors on reads during normal instruction execution, reads during a flush operation, reads during BIST and testability cycles, and reads during inquire cycles will cause IERR# to be asserted. The IERR# pin will be asserted for one clock for each clock a parity error is detected and may be latched by the system. The IERR# pin is a glitch free signal, so no spurious assertions of IERR# will occur.

In general, internal timing constraints of the Pentium processor do not allow the inhibition of writeback cycles caused by inquire cycles, FLUSH# assertion or the WBINVD instruction when a parity error is encountered. In those cases where an internal parity error occurred

during the generation of a writeback cycle, and that cycle was not able to be inhibited, the IERR# pin can be used to recognize that the writeback should be ignored. If an internal parity error occurs during a flush operation, the Pentium processor will assert the IERR# pin as stated above, and the internal caches will be left in a partially flushed state. No special cycles (flush, flush acknowledge, or writeback) will be run.

12.2. ERROR DETECTION AT PENTIUM PROCESSOR INTERFACE

The Pentium processor provides parity checking on the external address and data buses. There is one parity bit for each byte of the data bus and one parity bit for bits A31-A5 of the address bus.

12.2.1. Address Parity

A separate and independent mechanism is used for parity checking on the address bus during inquire cycles. Even address parity is driven along with the address bus during all Pentium processor initiated bus cycles and checked during inquire cycles. When the Pentium processor is driving the address bus, even parity is driven on the AP pin. When the address bus is being driven into the Pentium processor during an inquire cycle, this pin is sampled in any clock in which EADS# is sampled asserted. APCHK# is driven with the parity status two clocks after EADS# is sampled active. The APCHK# output (when active) indicates that a parity error has occurred on the address bus during an inquire. Figure 12-1 depicts an address parity error during an inquire cycle. For additional timing diagrams which show address parity, see the Bus Functional Description chapter. The APCHK# pin will be asserted for one clock for each clock a parity error is detected and may be latched by the system. The APCHK# pin is a glitch free signal, so no spurious assertions of APCHK# will occur.

In the event of an address parity error during inquire cycles, the internal snoop will not be inhibited. If the inquire hits a modified line in this situation and an active AHOLD prevents the Pentium processor from driving the address bus, the Pentium processor will potentially write back a line at an address other than the one intended. If the Pentium processor is not driving the address bus during the writeback cycle, it is possible that memory will be corrupted.
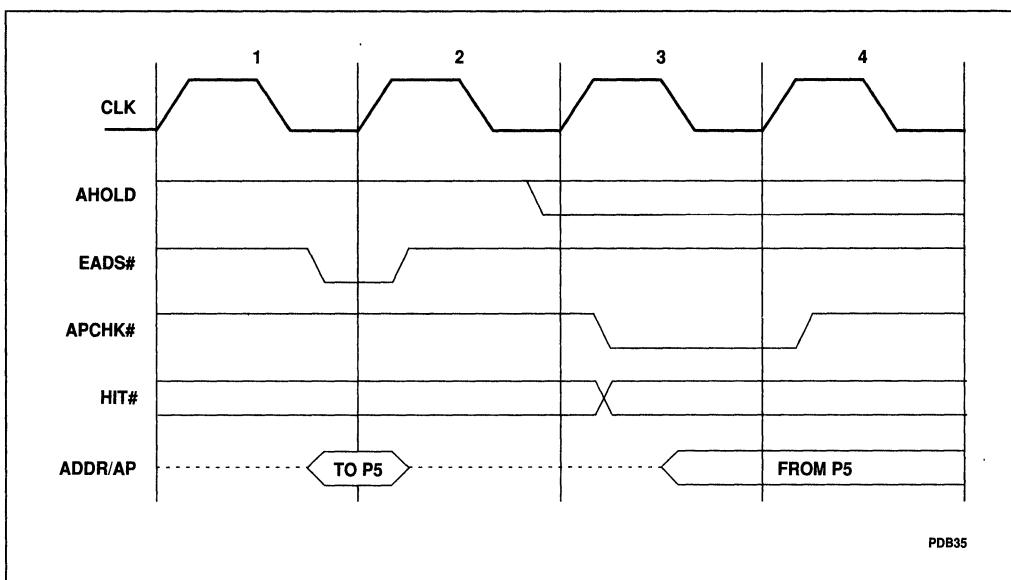
Figure 12-1. Inquire Cycle Address Parity Checking

Driving APCHK# is the only effect that bad address parity has on the Pentium processor. It is the responsibility of the system to take appropriate action if a parity error occurs. If parity checks are not implemented in the system, the APCHK# pin may be ignored.

12.2.2. Data Parity

Even data parity is driven on the DP7-DP0 pins in the same clock as the data bus during all Pentium processor initiated data write cycles. During reads, even parity information may be driven back to the Pentium processor on the data parity pins along with the data being returned. Parity status for data sampled is driven on the PCHK# pin two clocks after the data is returned. PCHK# is driven low if a data parity error was detected, otherwise it is driven high. The PCHK# pin will be asserted for one clock for each clock a parity error is detected and may be latched by the system. The PCHK# pin is a glitch free signal, so no spurious assertions of PCHK# will occur. Figure 12-2 shows when the data parity (DP) pins are driven/sampled and when the PCHK# pin is driven. For additional timing diagrams that show data parity, see the Bus Functional Description chapter.
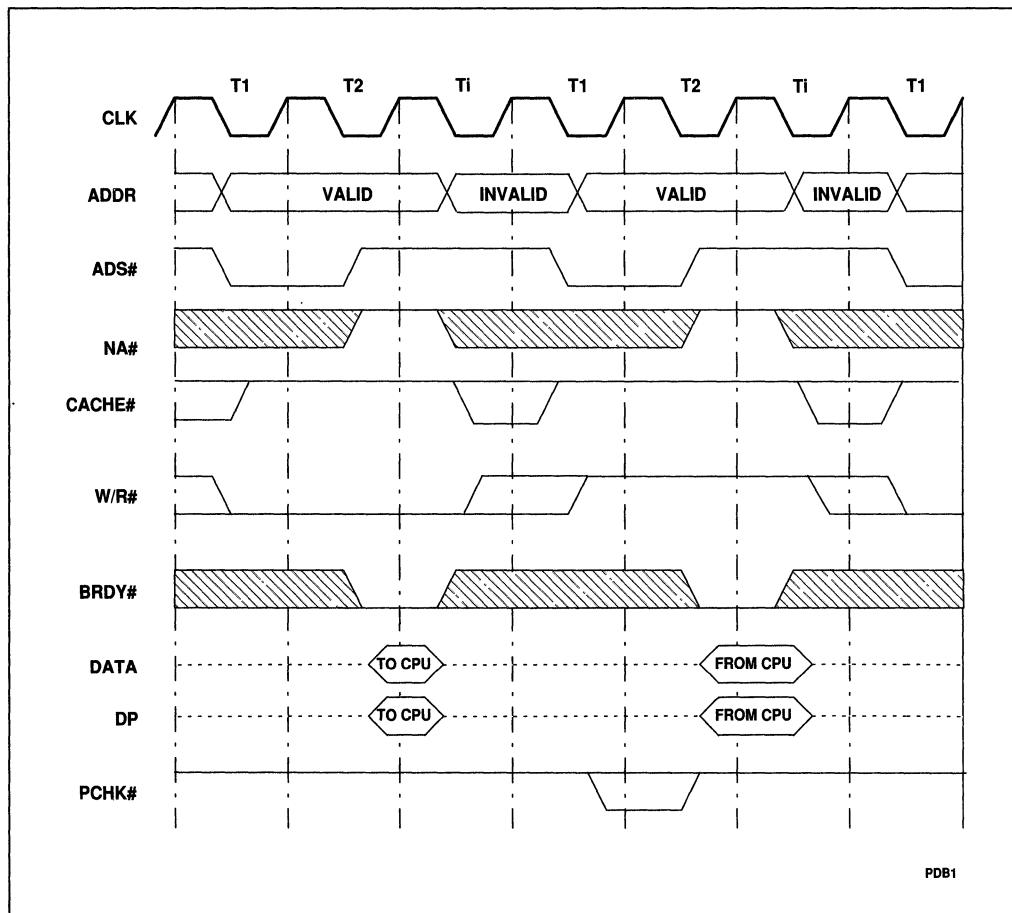
intel®

Figure 12-2. Data Parity During a Read and Write Cycle

Driving PCHK# is the only effect that bad data parity has on the Pentium processor. It is the responsibility of the system to take appropriate action if a parity error occurs. If parity checks are not implemented in the system, the PCHK# pin may be ignored.

12.2.2.1. MACHINE CHECK EXCEPTION AS A RESULT OF A DATA PARITY ERROR

The PEN# input determines whether a machine check interrupt will be taken as a result of a data parity error. If a data parity error occurs on a read for which PEN# was asserted, the physical address and cycle information of the cycle causing the parity error will be saved in the Machine Check Address Register and the Machine Check Type Register. If in addition, the CR4.MCE is set to 1, the machine check exception is taken. The "Machine Check Exception" section provides more information on the machine check exception.

The parity check pin, PCHK#, is driven as a result of read cycles regardless of the state of the PEN# input.

12.2.3. Bus Error

The BUSCHK# input provides the system a means to signal an unsuccessful completion of a bus cycle. This signal is sampled on any edge in which BRDY# is sampled, for reads and writes. If this signal is sampled active, then the cycle address and type will be latched into the Machine Check Address and Machine Check Type registers. If in addition, the CR4.MCE bit is set to 1, the processor will be vectored to the machine check exception.

Even if BUSCHK# is asserted in the middle of a cycle, BRDY# must be asserted the appropriate number of clocks required to complete the bus cycle. The purpose of BUSCHK# is to act as an indication of an error that is synchronous to bus cycles. If the machine check interrupt is not enabled, i.e. the MCE bit in the CR4 register is zero, then an assertion of BUSCHK# will not cause the processor to vector to the machine check exception.

12.2.4. Machine Check Exception

As mentioned in earlier sections, a new exception has been added to the Pentium processor. This is the machine check exception which resides at interrupt vector 18 (decimal). In processors previous to the Pentium processor, interrupt vector 18 was reserved and, therefore, there should be no interrupt routine located at vector 18. For reasons of compatibility, the MCE bit of the CR4 register will act as the machine check enable bit. When set to "1", this bit will enable the generation of the machine check exception. When reset to "0", the processor will inhibit generation of the machine check exception. CR4.MCE will be cleared on processor reset. In the event that a system is using the machine check interrupt vector for another purpose and the Machine Check Exception is enabled, the interrupt routine at vector 18 must examine the state of the CHK bit in the Machine Check Type register to determine the cause of its activation (see Figure 6-2). Note that at the time the system software sets CR4.MCE to 1, it must read the Machine Check Type register in order to clear the CHK bit.

The Machine Check Exception is an abort, that is, it is not possible to reliably restart the instruction stream or identify the instruction causing the exception. In addition, the exception does not allow the restart of the program that caused the exception. The Pentium processor does not generate an error code for this exception. Since the machine check exception is synchronous to a bus cycle and not an instruction, the IP pushed on to the stack may not be pointing to the instruction which caused the failing bus cycle.

The Machine Check Exception can be caused by one of two events: 1) Detection of data parity error during a read when the PEN# input is active, or 2) The BUSHCK# input being sampled active. When either of these events occur, the cycle address and type will be latched into the Machine Check Address (MCA) and Machine Check Type (MCT) registers (independent of the state of the CR4.MCE bit). If in addition, the CR4.MCE is "1", a machine check exception will occur. When the MCA and MCT registers are latched, the MCT.CHK bit is set to "1" indicating that their contents are valid (Figure 12-3).

The Machine Check Address register, and the Machine Check Type register are model specific, read only registers. The Machine Check Address register is a 64-bit register containing the physical address for the cycle causing the error. The Machine Check Type register is a 64-bit register containing the cycle specification information, as defined in Figure 12-3. These registers are accessed using the RDMSR instruction. When the MCT.CHK is zero,

the contents of the MCT and MCA registers are undefined. When the MCT register is read (using the RDMSR instruction), the CHK bit is reset to zero. Therefore, software must read the MCA register before reading the MCT register.
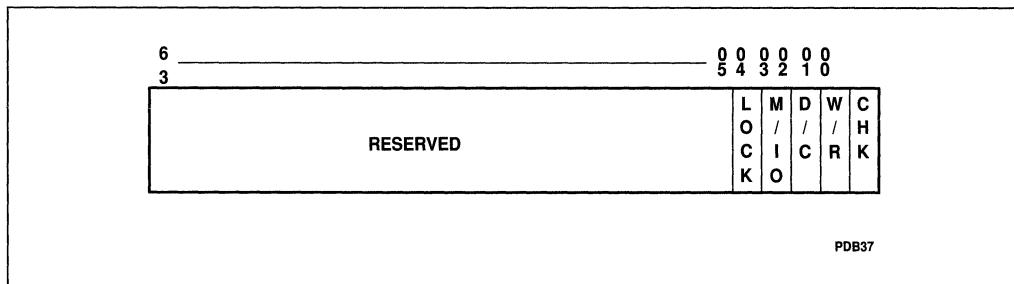

Figure 12-3. Machine Check Type Register

The bits in the Machine Check Type Register are defined as follows:

CHK: This bit is set to 1 when the Machine Check Type register is latched and is reset to 0 after the Machine Check Type register is read via the RDMSR instruction. In the event that the Machine Check Type register is latched in the same clock in which it is read, the CHK bit will be set. The CHK bit is reset to 0 on assertion of RESET. When the CHK bit is "0", the contents of the MCT and MCA registers are undefined.

M/IO#, D/C#, WR#:   These cycle definition pins can be decoded to determine if the cycle in error was a memory or I/O cycle, a data or code fetch, and a read or a write cycle.

LOCK: Set to "1" if LOCK# is asserted for the cycle

12.2.5. Functional Redundancy Checking

Functional Redundancy Checking (FRC) in the Pentium processor will provide maximum error detection (>99%) of on chip devices and the processor's interface. A "checker" Pentium processor that executes in lock step with the "master" Pentium processor is used to compare output signals every clock.

Two Pentium Processors are required to support FRC. Both the master and checker must be of the same stepping. The Pentium processor configured as a master operates according to bus protocol described in this document. The outputs of the checker Pentium processor are tristated (except IERR# and TDO) so the outputs of the master can be sampled. If the sampled value differs from the value computed internally by the checker, the checker asserts the IERR# output to indicate an error. A master-checker pair should have all pins except FRCMC#, IERR# and TDO tied together.

The Pentium processors are configured either as a master or a checker by driving the FRCMC# input to the appropriate level while RESET is asserted. If sampled low during reset, the Pentium processor enters checker mode and tristates all outputs except IERR# and TDO (IERR# is driven inactive during reset). This feature is provided to prevent bus contention

before reset is completed. The final master/checker configuration is determined when RESET transitions from high to low. The final master/checker configuration may not be changed other than by a subsequent RESET.

The IERR# pin reflects the result of the master-checker comparison. It is asserted for one clock, two clocks after the mismatch. It is asserted for each detected mismatch, so IERR# may be low for more than one consecutive clock. During the assertion of RESET, IERR# will be driven inactive. After RESET is deasserted, IERR# will not be asserted due to a mismatch until two clocks after the ADS# of the first bus cycle (i.e. in the third clock of the first bus cycle). IERR# will reflect pin comparisons thereafter. Note that IERR# may be asserted due to an internal parity error prior to the first bus cycle. It is possible for FRC mismatches to occur in the event that undefined processor state is driven off-chip, therefore no processor state should be stored without having been previously initialized.

In order for the master-checker pair to operate correctly, the system must be designed such that the master and the checker sample identical input states in the same clock. All asynchronous inputs should change state in such a manner that both the master and checker sample them in the same state in the same clock. The simplest way to do this is to design all asynchronous inputs to be synchronously controlled.

The TDO pin is not tested by FRC since it operates on a separate clock. Note that it is possible to use boundary scan to verify the connection between the master and checker by scanning into one, latching the outputs of the other and then scanning out.

The comparators at each output compare the value of the package pin with the value being driven from the core to that pin, not the value driven by boundary scan to that pin. Therefore, during the use of boundary scan, FRC mismatches (IERR# assertion) can be expected to occur.

For additional information on Functional Redundancy Checking, see Appendix A.

intel®

13

Execution Tracing

intel®

CHAPTER 13
EXECUTION TRACING

The Pentium processor includes dedicated pins and a special bus cycle to support execution tracing. This feature allows external hardware to track the flow of instructions as they execute in the processor.

Specifically, the Pentium processor dedicates three pins, IU, IV and IBT and the Branch Trace Message Special Cycle to track the flow of instructions within the processor. The IU and IV pins track the sequential flow of instructions. The IU pin is asserted to indicate that an instruction completed execution in the u-pipe. The IV pin is asserted to indicate that an instruction completed execution in the v-pipe. IBT is asserted when a taken branch instruction has completed execution. If enabled through Test Register 12 (see section 13.1), the Branch Trace Message special cycle is driven subsequent to each assertion of IBT.

Table 13-1 indicates the proper interpretation of the IU, IV, and IBT pins.

Table 13-1. Interpretation of IU, IV and IBT Pins

IU	IV	IBT	Meaning
0	0	0	No Instruction Completed
0	0	1	Does Not Occur
0	1	0	Does Not Occur
0	1	1	Does Not Occur
1	0	0	An instruction other than a taken branch has completed in the u pipe.
1	0	1	A branch was taken by an instruction in the u pipe.
1	1	0	Instructions completed in the u pipe and the v pipe. Neither was a taken branch.
1	1	1	Instructions completed in both pipes. The instruction in the v pipe was a taken branch.

The IU, IV and IBT pins are always driven, however the Branch Trace Message Special Cycle is optionally driven. If the execution tracing enable bit (bit 1) in TR12 is set to 1, a branch trace message special cycle will be driven every time IBT is asserted, i.e. every time a branch is taken. The branch trace message special cycle may be delayed by 0 or more clocks after the one in which the IBT is asserted, depending on bus activity. These cycles are buffered and do not normally stall the processor. At most two additional IBTs may be signaled before the first branch trace message is driven to the bus. If the bus is busy, the processor will stall.

When the branch trace message cycle is driven, the address bus is driven with the following information:

A31-A3:   Bits 31-3 of the branch target linear address

BT2-BT0:   Bits 2-0 of the branch target linear address (the byte enables should not be decoded for A2-A0)

BT3:   High if the default operand size is 32-bits, Low if the default operand size is 16-bits

In addition to taken conditional branches, jumps, calls, returns, software interrupts, and interrupt returns, the Pentium processor treats the following operations as causing taken branches: serializing instructions, some segment descriptor loads, hardware interrupts (including FLUSH#), and programmatic exceptions that invoke a trap or fault handler. Note that the conditions which cause the VERR, VERW, LAR and LSL instructions to clear the ZF bit in EFLAGS will also cause these instructions to be treated as taken branches. These operations will cause the IBT, IU and possibly the IV pins to be asserted. If execution tracing is enabled, then these operations will also cause a corresponding Branch Trace Message Cycle to be driven. Note that if an instruction faults, it does not complete execution but instead is flushed from the pipeline and an exception handler is invoked. The Pentium processor treats this operation as an instruction that takes a branch, thus causing the IU and IBT pins to be asserted.

13.1. TEST REGISTER 12

Test Register 12 (Figure 13-1) allows the branch trace message special cycle to be enabled or disabled.
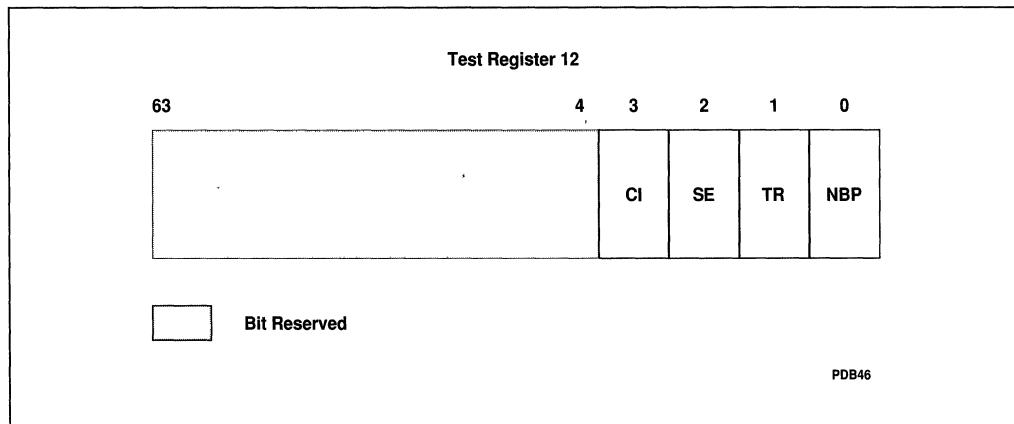

Figure 13-1. Test Register TR12

The TR12.TR bit (Tracing) controls the Branch Trace Message Special Cycle. When the TR12.TR bit is set to 1, a branch trace message special cycle is generated whenever a taken branch is executed (whenever IBT is asserted). If the TR12.TR bit is not set, IBT will still be asserted, however the branch trace message special cycle is not driven by the Pentium processor.

TR12.TR is initialized to zero on RESET. This register is write only and the reserved bits should be written with zeroes. The test registers should be written to for testability accesses only. Writing to the test registers during normal operation causes unpredictable behavior.

For information related to the TR12.NBP, TR12.SE and TR12.CI bits, see Appendix A.

intel®

14

System Management Mode

intel®

CHAPTER 14
SYSTEM MANAGEMENT MODE

The Pentium processor implements Intel's System Management Mode (SMM) architecture. This chapter describes the hardware interface to SMM. For the architectural description, refer to the System Management Mode chapter in the *Preliminary Pentium™ Processor Architecture and Programming Manual*.

14.1. SMM OVERVIEW

System Management Mode is invoked via an external interrupt. When the assertion of the SMI# input is recognized on an instruction execution boundary, the processor waits for all stores to complete, waits for the assertion of EWBE#, and asserts the SMIACT# pin. The processor then saves its register state to SMRAM space and begins to execute the SMM handler. The RSM instruction restores the registers and returns to the user program. This chapter will describe the hardware interface of this SMM implementation.

14.2. SMM HARDWARE INTERFACE

14.2.1. SMM Pins

The Pentium processor provides two pins for use in SMM systems.

SMI#: The System Management Interrupt is a falling edge sensitive input that latches a System Management Interrupt request. Subsequent SMI requests are not acknowledged while the processor is in SMM, but are held pending until the processor exits SMM through the execution of the RSM instruction.

SMIACT#: The System Management Interrupt Active output indicates that the processor is operating in System Management Mode. It remains active (low) until the processor executes the RSM instruction to leave SMM.

14.2.2. The SMI Interrupt

When an SMI is recognized on an instruction execution boundary, the processor waits for all stores to complete, and asserts the SMIACT# pin assuming no higher priority interrupt was pending. Among external interrupts, the SMI has a priority as shown below:

- FLUSH#
- SMI#

- INIT
- NMI
- INTR

FLUSH# is recognized while the processor is in SMM, however NMI and INIT are not. NMI and INIT are latched and executed when the processor exits SMM through the completion of the RSM instruction. Upon entry to SMM, the IF bit in the EFLAGs register is cleared to disable external interrupts. This is necessary because while in SMM, the Pentium processor is running in a separate memory space and possibly in a different mode. Consequently, the vectors stored previously in the interrupt descriptor table are no longer applicable. If interrupt and exception handling are going to be used while in SMM, the SMM program must set up new interrupt and exception vectors.

14.2.3. Optional Cache Flush on Entering SMM

The Pentium processor does not unconditionally write back and invalidate its cache before entering SMM. However, if SMRAM is in a location that is "shadowed" in memory that is visible to the application or operating system (default), then it is necessary for the system to flush the cache upon entering SMM. This may be accomplished by asserting the FLUSH# pin at the same time as the request to enter SMM. The priorities of the FLUSH# pin and the SMI# pin are such that the FLUSH# will be serviced first. To guarantee this behavior the following constraints on the interaction of SMI# and FLUSH# must be obeyed.

In a system where the FLUSH# and SMI# pins are synchronous and setup and hold times are met, then the FLUSH# and SMI# pins may be asserted in the same clock. In asynchronous systems, the FLUSH# pin must be asserted at least one clock before the SMI# pin to guarantee that the FLUSH# pin is serviced first. Note that in systems that use the FLUSH# pin to write back and invalidate cache contents before entering SMM, the Pentium processor will prefetch at least one cache line in between the time the Flush Acknowledge special cycle is run and the recognition of SMI# and the assertion of SMIACT#. It is the obligation of the system to ensure that these lines are not cached by returning KEN# inactive.

The Pentium processor does not writeback or invalidate its internal caches upon leaving SMM. For this reason, references to the SMRAM area must not be cached. It is the obligation of the system to ensure that the KEN# pin is sampled inactive during all references to the SMRAM area.

intel ®

15

Debugging

intel ®

<div align="right">

CHAPTER 15
DEBUGGING

</div>

The Pentium processor implements a new mode known as Probe Mode for the purpose of system debug. For more information on probe mode, see Appendix A.

15.1. DESIGNING IN A DEBUG PORT

A Debug Port, when designed into a Pentium processor-based system, allows a debugger to interface to the processor's debug hooks. An example pinout for Debug Port signals is provided in Table 15-1. Please contact your debugging tool vendor before designing in a debug port to ensure compatibility.

15.1.1. Debug Connector Description

Following are two recommended connectors to mate with the cable from the debugger. Install either of the connectors on the Pentium processor-based system board:

- AMP 104068-1 20 pos shrouded vertical header
- AMP 104069-1 20 pos shrouded right-angle header

Figure 15-1 shows the pinout of the connector footprint as viewed from the connector side of the circuit board:
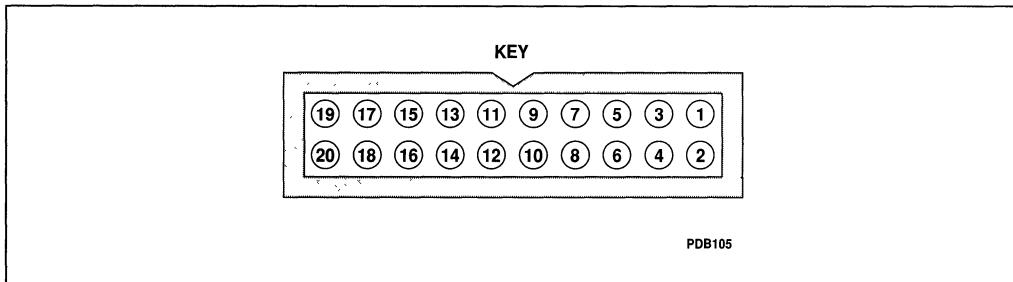

Figure 15-1. Debug Port Connector

intel ®

15.1.2. Signal Descriptions

Following are the debug port signals. Direction is given as follows:

* O = output from the Pentium processor board to the debugger;
* I = input to the Pentium processor board from the debugger.

Please contact your debugging tool vendor before designing in a debug port to ensure compatibility. For more information on the signal descriptions, see Appendix A.

Table 15-1. Debug Port Signals

Signal Name	Dir	Pin
TDO	O	13
TDI	I	12
TMS	I	14
TCLK	I	16
TRST#	I	18
BSEN#	I	20
R/S#	I	7
PRDY	O	11
INIT	O	1
RESET	O	3
DBRESET	I	2
SYSR/S#	O	9
DBINST#	I	19
VCC		6
GND		4,8,10,15,17
SMIACT#	O	5

15.1.3. Signal Quality Notes

Since the debugger connects to the Pentium processor system via a cable of significant length, care must be taken in the Pentium processor system design with regard to the signals going to the Debug Port. System outputs to the Debug Port (TDO, PRDY, INIT, RESET, SMIACT# and SYSR/S#) should have dedicated drivers to the Debug Port if the signals are used elsewhere in the system (to isolate them from the reflections from the end of the debugger cable). Series termination is recommended at the driver output. If the Pentium processor boundary scan signals are used elsewhere in the system, then the TDI, TMS, TCLK, and TRST# signals from the Debug Port should be isolated from the system signals with multiplexers.

15.1.4. Implementation Examples

Figure 15-2 shows a schematic of a minimal Debug Port implementation in which the R/S# and boundary scan pins of the Pentium processor are not used in the system.

Figure 15-3 shows a schematic of a maximal Debug Port implementation in which the R/S# and boundary scan pins of the Pentium processor are used in the system. Note that the DBINST# signal is used to multiplex the R/S# signal and that the BSEN# signal is used to multiplex the boundary scan signals.
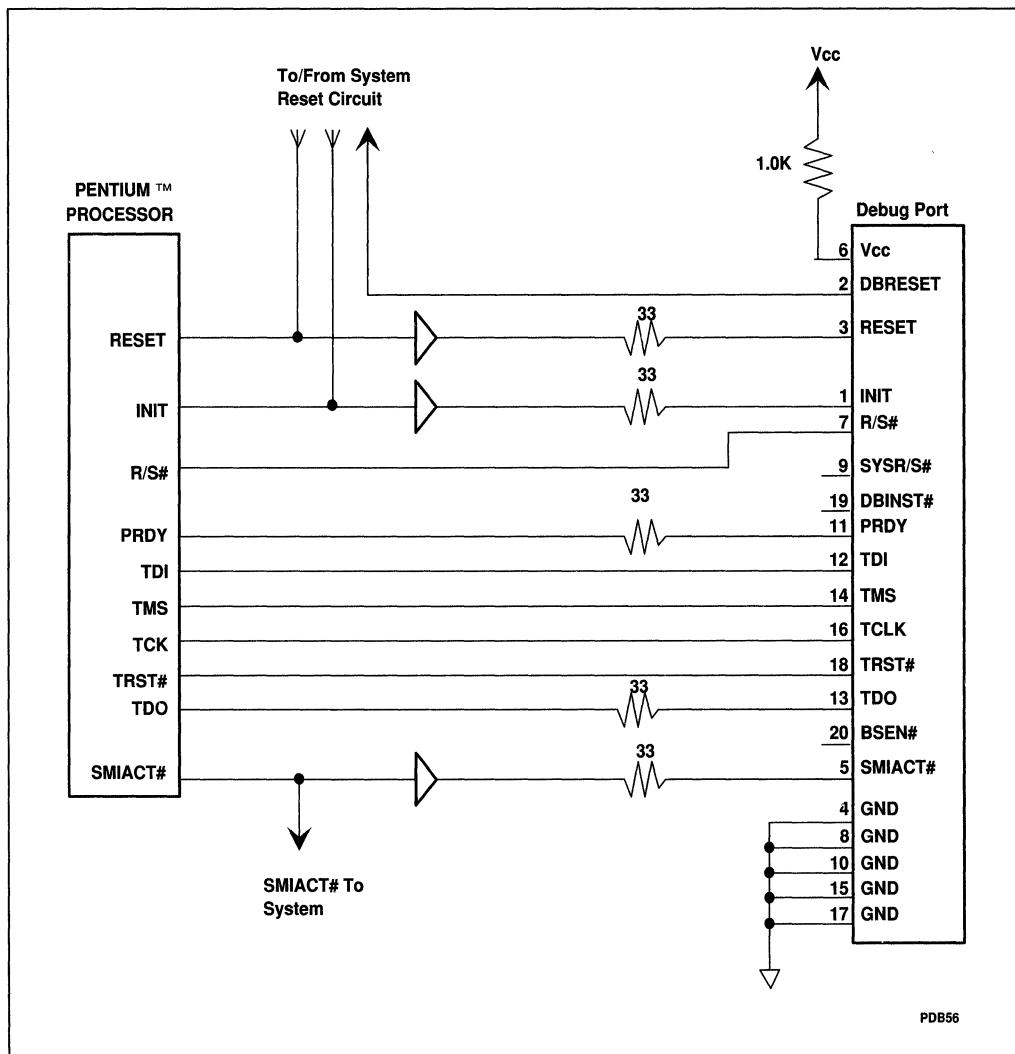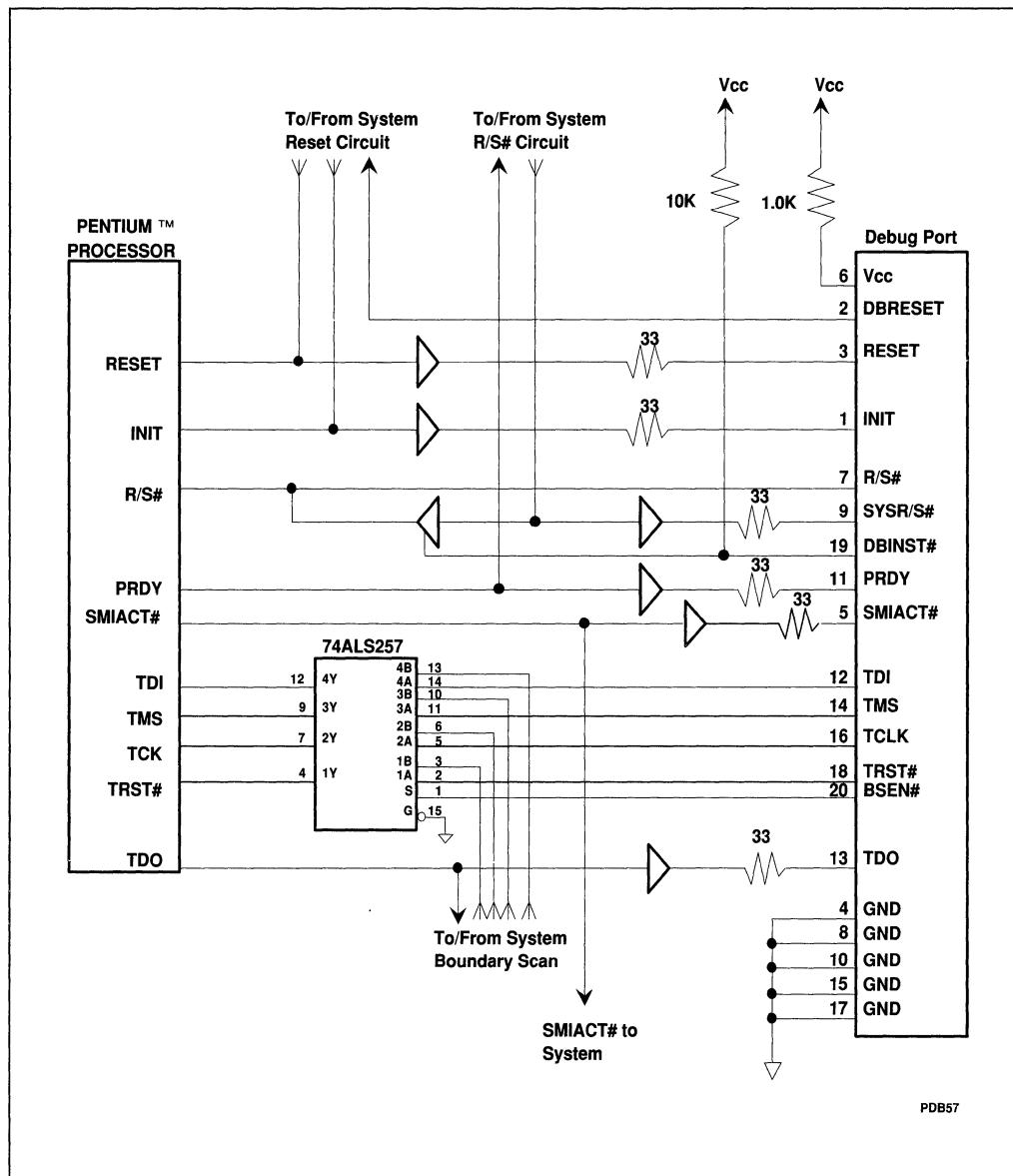
intel®

Figure 15-2. Minimal Debug Port Implementation

intel.

Figure 15-3. Maximal Debug Port Implementation

.

intel®

A

Supplemental Information

intel®

Some non-essential information regarding the Pentium processor are considered Intel confidential and proprietary and have not been documented in this publication. This information is provided in the *Supplement to the Pentium™ Processor User's Manual* once the appropriate non-disclosure agreements are in place. Please contact Intel Corporation for details.

intel®