Pentium® Processor Family Developer's Manual

Volume 1: Pentium® Processors

NOTE: The *Pentium® Processor Family Developer's Manual* consists of three books: *Pentium® Processors*, Order Number 241428; the *82496/82497/82498 Cache Controller and 82491/82492/82493 Cache SRAM*, Order Number 241429; and the *Architecture and Programming Manual*, Order Number 241430. Please refer to all three volumes when evaluating your design needs.

1995

TABLE OF CONTENTS

PART I — Pentium® Processor (510\60, 567\66)

PART II — Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133)

CHAPTER 17
PINOUT

CHAPTER 18
FEATURES OF PENTIUM® PROCESSOR (610\75, 735\90, 815\100, 1000\120, 1110\133)

CHAPTER 21
HARDWARE INTERFACE

Page

Figures

Tables

Part I

Pentium® Processor (510\60, 567\66)

1

Pinout

CHAPTER 1
PINOUT

1.1. PINOUT AND PIN DESCRIPTIONS

1.1.1. Pentium® Processor (510\60, 567\66) Pinout

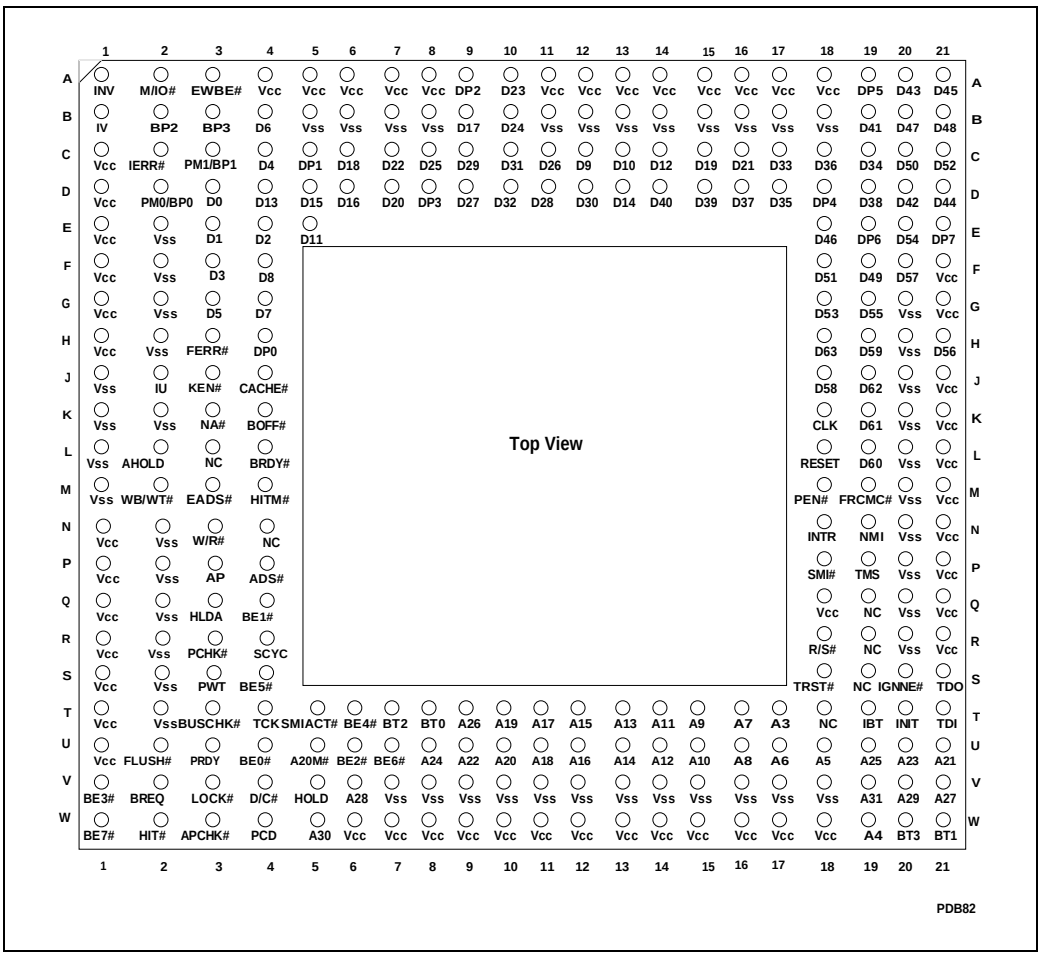


Figure 1-1. Pentium® Processor (510\60, 567\66) Pinout (Top View)

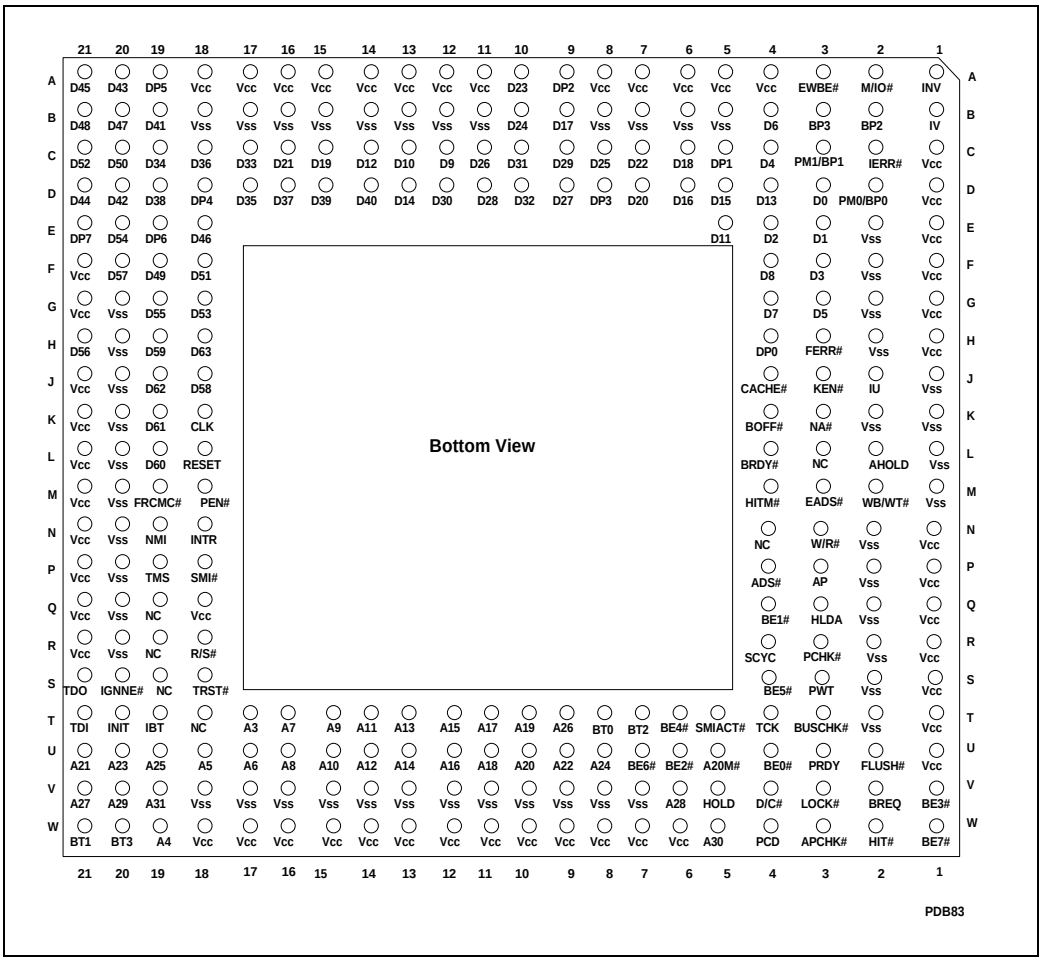


Figure 1-2. Pentium® Processor (510\60, 567\66) Pinout (Bottom View)

Table 1-1. Pentium® Processor (510\60, 567\66) Pin Cross Reference Table by Pin Name

Signal	Location
A3	T17
A4	W19
A5	U18
A6	U17
A7	T16
A8	U16
A9	T15
A10	U15
A11	T14
A12	U14
A13	T13
A14	U13
A15	T12
A16	U12
A17	T11
A18	U11
A19	T10
A20	U10
A21	U21
A22	U09
A23	U20
A24	U08
A25	U19
A26	T09
A27	V21
A28	V06
A29	V20
A30	W05
A31	V19

Signal	Location
A20M#	U05
ADS#	P04
AHOLD	L02
AP	P03
APCHK#	W03
BE0#	U04
BE1#	Q04
BE2#	U06
BE3#	V01
BE4#	T06
BE5#	S04
BE6#	U07
BE7#	W01
BOFF#	K04
BP2	B02
BP3	B03
BRDY#	L04
BREQ	V02
BT0*	T08
BT1*	W21
BT2*	T07
BT3*	W20
BUSCHK#	T03
CACHE#	J04
CLK	K18
D0	D03
D1	E03
D2	E04
D3	F03

Table 1-1. Pentium® Processor (510\60, 567\66) 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 (510\60, 567\66) Pin Cross Reference Table by Pin Name (Contd.)

Signal	Location
D62	J19
D63	H18
D/C#	V04
DP0	H04
DP1	C05
DP2	A9
DP3	D08
DP4	D18
DP5	A19
DP6	E19
DP7	E21
EADS#	M03
EWBE#	A03
FERR#	H03
FLUSH#	U02
FRCMC#	M19
HIT#	W02
HITM#	M04
HLDA	Q03
HOLD	V05
IBT*	T19
IERR#	C02
IGNNE#	S20
INIT	T20
INTR	N18
INV	A01
IU*	J02

Signal	Location
IV*	B01
KEN#	J03
LOCK#	V03
M/IO#	A02
NA#	K03
NMI	N19
PCD	W04
PCHK#	R03
PEN#	M18
PM0/BP0	D02
PM1/BP1	C03
PRDY	U03
PWT	S03
RESET	L18
R/S#	R18
SCYC	R04
SMI#	P18
SMIACT#	T05
TCK	T04
TDI	T21
TDO	S21
TMS	P19
TRST#	S18
WB/WT#	M02
W/R#	N03
NC	L03, N04, Q19, R19, S19, T18

Table 1-1. Pentium® Processor (510\60, 567\66) Pin Cross Reference Table by Pin Name (Contd.)

Signal	Location
V_{CC}	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
V_{SS}	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

NOTE:

*Not supported on the future Pentium® OverDrive® processor.

1.2. DESIGN NOTES

For reliable operation, always connect unused inputs to an appropriate signal level. Unused active low inputs should be connected to V_{CC}. 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,* Order Number 241814, 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.

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 (510\60, 567\66).
AHOLD	I	In response to the assertion of *address hold*, the Pentium processor will stop driving the address lines (A31-A3), BT3-BT0 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.
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). Because of the fractional speed bus implemented in the future Pentium OverDrive® processor, the breakpoint pins, BP[3:0], may indicate that one or more BP matches occurred.

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

Symbol	Type*	Name and Function
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. If the internal request for the bus is removed, the BREQ pin will be deasserted. Note that this means that every assertion of BREQ is NOT guaranteed to have a corresponding assertion of ADS#. For example, assume that the processor has internally requested a code prefetch which is a miss in the processor's code cache. BREQ is asserted to indicate that the processor has a bus request pending internally. If the request can not be serviced immediately (due to bus HOLD or AHOLD, or because the bus is busy), and a branch or serializing instruction is executed, the Pentium processor may recall the request for the code prefetch and deassert BREQ without ever having driven the code prefetch cycle to the 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. These signals are not supported on the future Pentium OverDrive processor.
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. **It is recommended that CLK begin toggling within 150 ms after V_{CC} reaches its proper operating level. This recommendation is only to ensure long-term reliability of the device.**
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. The Pentium processor's data bus (D63-D0) is floated during T1, TD, or Ti bus states. During write cycles, the data bus is driven during the T2, T12, or T2P states. During read cycles, the processor 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.

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 (510\60, 567\66). This signal is not supported on the future Pentium OverDrive 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.

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

Symbol	Type*	Name and Function
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.
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 (510\60, 567\66). This signal is not supported on the future Pentium OverDrive 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 (510\60, 567\66). This signal is not supported on the future Pentium OverDrive 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 two 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.

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

Symbol	Type*	Name and Function
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.
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]BP [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). This pin is provided for use with the Intel debug port described in the "Debugging" chapter.
PWT	O	The *page writethrough* 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 (510\60, 567\66) 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 writeback or writethrough 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.

NOTES:

*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#).
*These signals will be internally tied inactive (low) on the future Pentium® OverDrive® processor.

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 equal tristate test mode (except TDO) and checker mode (except IERR# and TDO).

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

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#.

2

Microprocessor Architecture Overview

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 (510\60, 567\66) contains all of the features of the Intel486 CPU, and provides significant enhancements and additions including the following:

- Superscalar Architecture
- Dynamic Branch Prediction
- Pipe-lined 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 Pipe-lining
- Address Parity
- Internal Parity Checking
- Function Redundancy Checking
- Execution Tracing
- Performance Monitoring
- IEEE 1149.1 Boundary Scan
- System Management Mode
- Virtual Mode Extensions
- Upgradable with a future Pentium OverDrive® processor

The application instruction set of the Pentium processor family 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 family implements several enhancements to increase performance. The two instruction pipe-lines and floating-point unit on the Pentium processor are capable of independent operation. Each pipe-line 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 a 10X speed-up for common operations including ADD, MULTIPLY, and LOAD. Many applications can achieve five times the performance or more with instruction scheduling and overlapped (pipe-lined) execution.

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 pipe-lining 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 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 pipe-lines, 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.

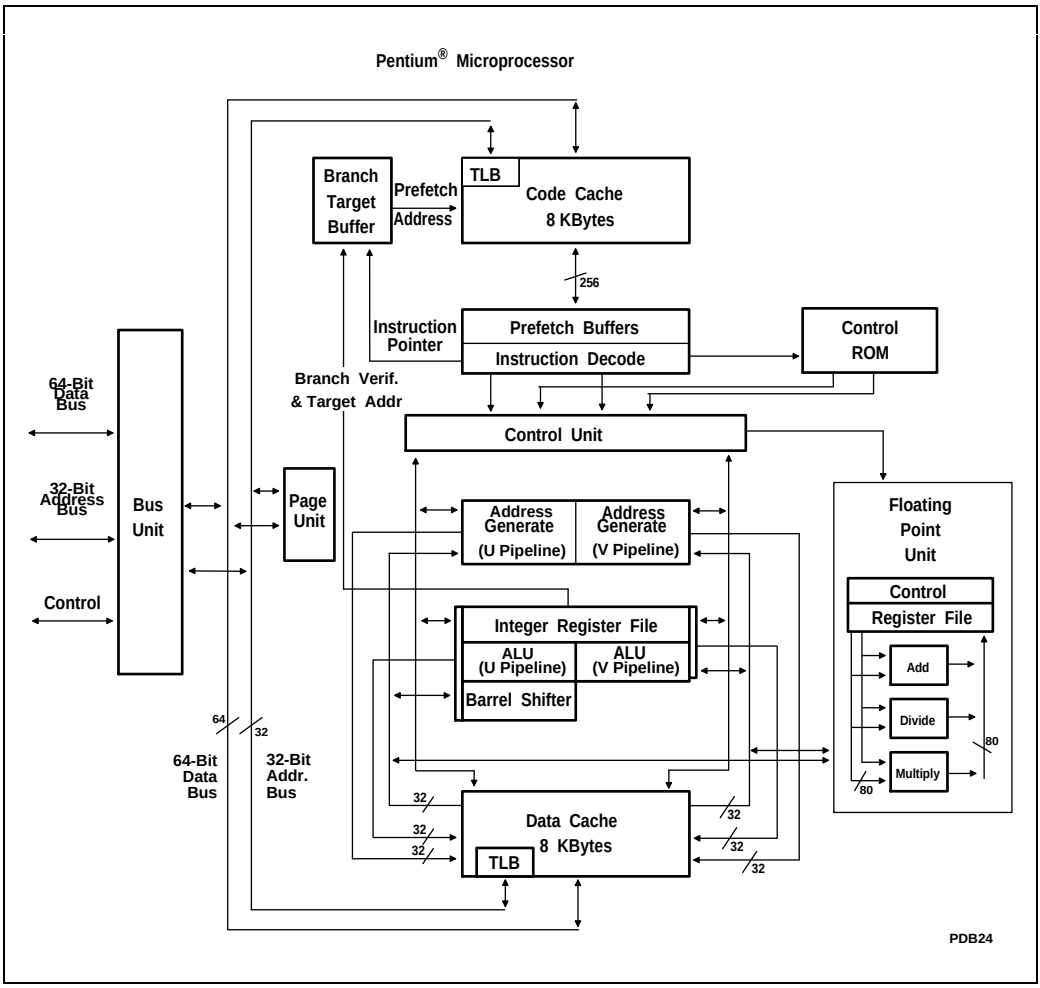


Figure 2-1. Pentium® Processor Block Diagram

The block diagram shows the two instruction pipe-lines, 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 Intel Architecture-based CPUs.

Supporting an upgrade socket (Socket 4) in the system will provide end user upgradability by the addition of a Future Pentium OverDrive processor. Typical applications will realize a 40% to 70% performance increase by the addition of a Future Pentium OverDrive processor. Refer to Chapter 16 for details on the Future Pentium OverDrive processor for Pentium processor (510\60, 567\66)-based systems.

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

The Pentium processor may contain design defects or errors known as errata. Current characterized errata are available upon request.

3

Component Operation

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 pipe-lined floating-point unit combine to sustain the high execution rate. These architectural features and their operation are discussed in this chapter.

3.1. PIPE-LINE AND INSTRUCTION FLOW

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

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

Figure 3-1 shows how instructions move through the Intel486 CPU pipe-line.

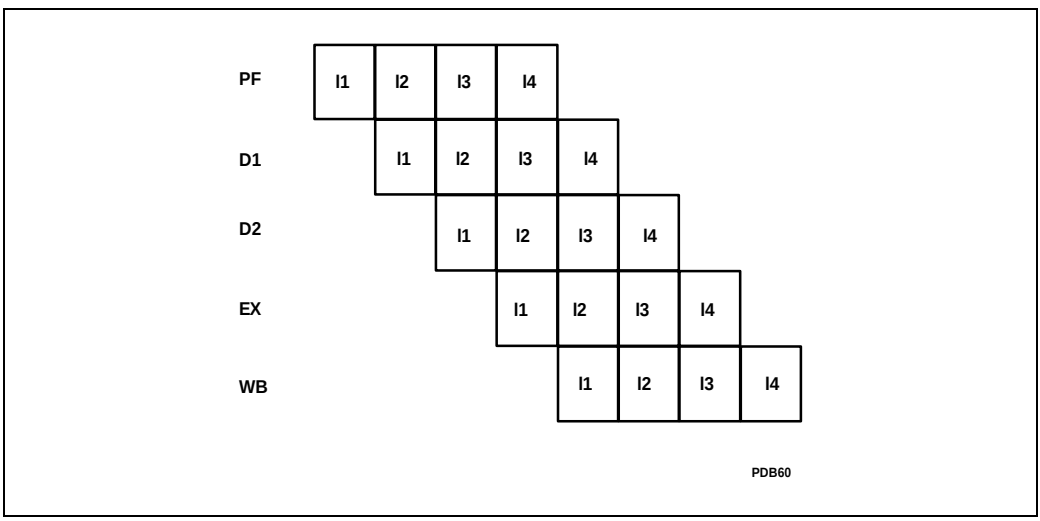


Figure 3-1. Intel486™ CPU Pipe-line Execution

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

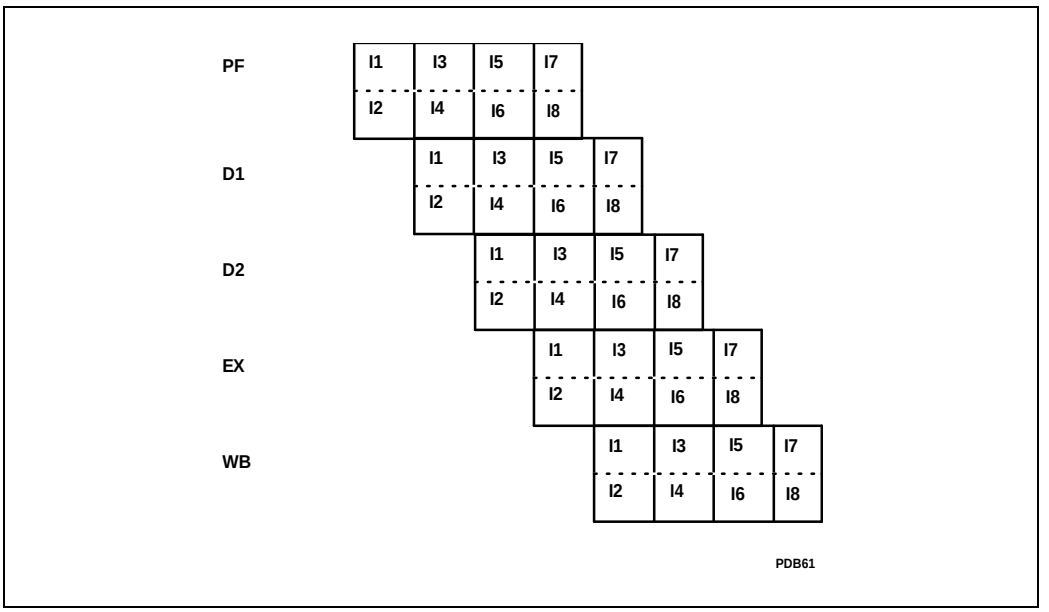


Figure 3-2. Pentium® Processor Pipe-line Execution

The pipe-lines 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 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 Pipe-line Description and Improvements

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

The first stage of the pipe-line 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 pipe-line 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 pipe-line.

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 pipe-line 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 pipe-lines 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 pipe-line 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 pipe-line 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 pipe-line until the instructions in both pipe-lines 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.

For more information on branch prediction, see section 3.2.

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-write 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 three and two 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 be paired with an ADC or an 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 refer to *Optimizing for Intel's 32-Bit CPUs*, Order Number 241799.

3.2. BRANCH PREDICTION

The Pentium processor uses a Branch Target Buffer to predict the outcome of branch instructions which minimizes pipe-line 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 pipe-line 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 pipe-lines 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`)

```
inner_loop:
     mov byte ptr flags[edx],al
     add edx,ecx
     cmp edx, SIZE
     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.

NOTE

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.

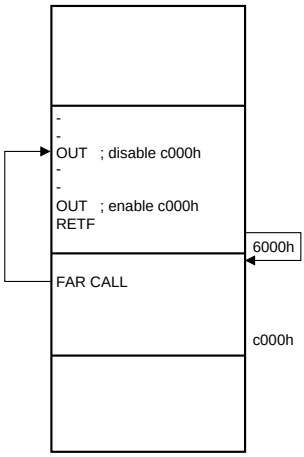
One effect of the branch prediction mechanism is that the Pentium processor may run code fetch bus cycles to retrieve instructions which are never executed. Although the opcodes retrieved are discarded, the system must complete the code fetch bus cycle by returning BRDY#. It is particularly important 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 (approximately 100 bytes past end of last executed instruction). Although the Pentium processor may prefetch beyond the CS limit, it will not attempt to execute beyond the CS limit. Instead, it will raise a GP fault. 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 (i.e., not present pages or pages with incorrect access rights), 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. If paging is not used, branch prediction can be disabled by setting TR12.NBP (bit 0)* and flushing the BTB by loading CR3 before disabling any areas of memory. Branch prediction can be re-enabled after re-enabling memory.

The following is an example of a situation that may occur:

1. Code passes control to segment at address c000h.
2. Code transfers control to code at different address (6000h) by using FAR CALL instruction.
3. This portion of the code does an I/O write to a port that disables memory at address c000h.
4. At the end of this segment, an I/O write is performed to re-enable memory at address c000h.
5. Following the OUT instruction, there is a RETF instruction to c000h segment.



The branch prediction mechanism of the Pentium processor, however, predicts that the RETF instruction is going to transfer control to the segment at address c000h and performs a prefetch from that address prior to the OUT instruction that re-enables that memory address. The result is that no BRDY is returned for that prefetch cycle and the system hangs.

In this case, branch prediction should be disabled (by setting TR12.NBP* and flushing the BTB by loading CR3) prior to disabling memory at address c000h and re-enabled after the RETF instruction by clearing TR12.NBP* as indicated above.

* Please refer to Chapter 33 of this volume.

3.3. WRITE BUFFERS AND MEMORY ORDERING

The Pentium processor has two write buffers, one corresponding to each of the pipe-lines, 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 pipe-lines. 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 inactive.

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: MOV to Debug Register, MOV to Control Register, 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 writeback 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/disable paging (that is, change the PG bit of CR0), this instruction must be followed with a jump. The instruction at the target of the branch is fetched with the new value of PG (i.e., paging enabled/disabled), however, the jump instruction itself is fetched with the previous value of PG. Intel386, Intel486 and Pentium processors have slightly different requirements to enable and disable paging. In all other respects, an MOV to CR0 that changes PG is serializing. Any MOV to CR0 that does not change PG is completely serializing.
5. Whenever an instruction is executed to change the contents of CR3 while paging is enabled, the next instruction is fetched using the translation tables that correspond to the new value of CR3. Therefore the next instruction and the sequentially following instructions should have a mapping based upon the new value 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 pipe-lines, 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 pipe-line. 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 pipe-line 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 (510\60, 567\66) 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 (510\60, 567\66) and 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.

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

For information on Model specific Registers and instructions, refer to Chapter 33 of this document.

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 pipe-lined. 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 refer to *Optimizing for Intel's 32-Bit CPUs*, Order Number 241799.

3.6.1. Floating-Point Pipe-line Stages

The Pentium processor FPU has 8 pipe-line 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 pipe-line 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; bypass 1 (bypass 1 described in the "Bypasses" section).

X2 Floating-Point Execute stage two;

WF Perform rounding and write floating-point result to register file; bypass 2 (bypass 2 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 pipe-line. 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 ah to the lower 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 and Intel486 architectures.

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.

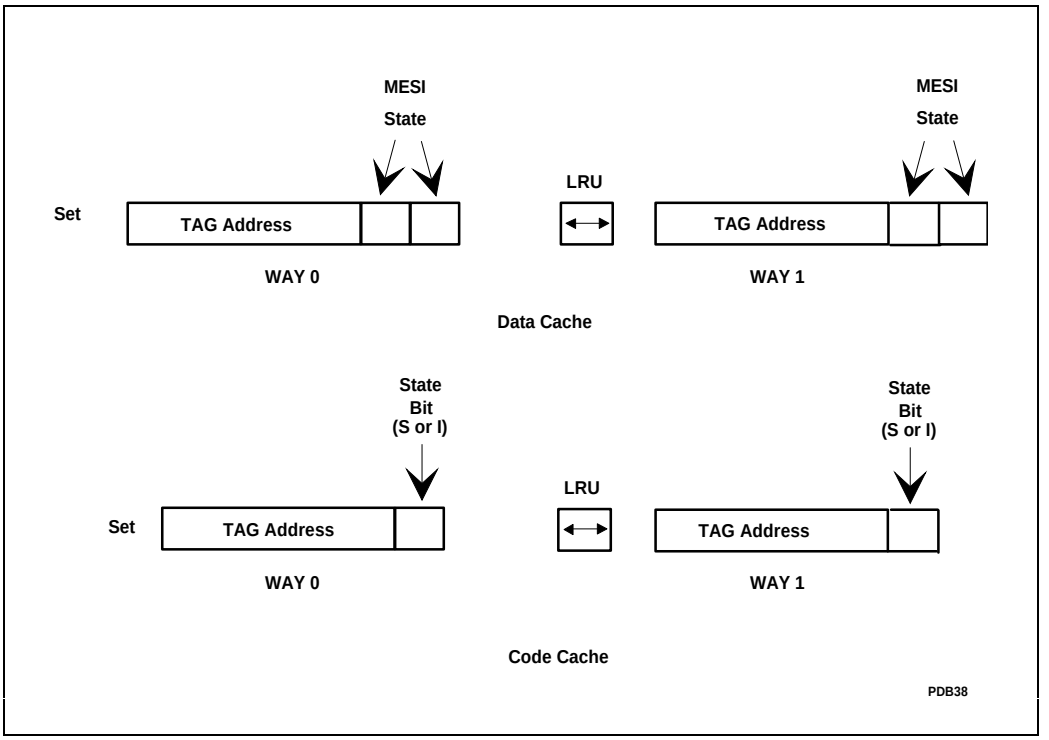


Figure 3-3. Conceptual Organization of Code and Data Caches

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 pipe-lines. 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 writethrough) bits in CR0. See Table 3-1 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.

Table 3-1. 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 affect 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 writethrough 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 PWT 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 3-2 and 3-3.

Table 3-2. 32-Bits/4-Kbyte Pages

PCD/PWT Taken From	During Accesses To
CR3	PDE
PDE	PTE
PTE	All other paged mem references

Table 3-3. 32-Bits/4-Mbyte 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.

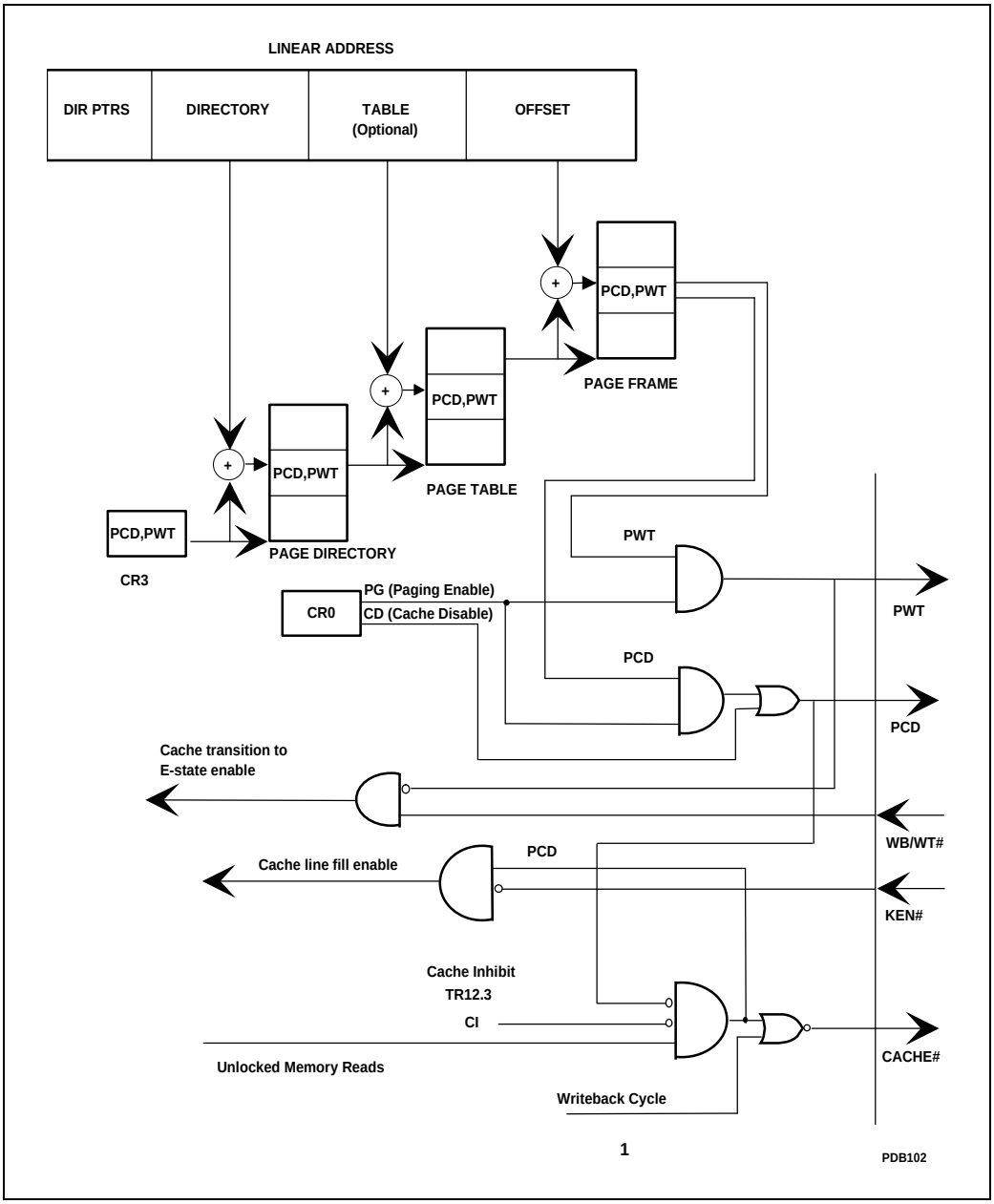


Figure 3-4. PCD and PWT Generation

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 writeback 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 3-4 shows the state transitions for lines in the data cache during unlocked read cycles.

Table 3-4. 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 (510\60, 567\66).

*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 pipe-line 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.

Table 3-5. 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 writethrough page; status of WB/WT# pin is ignored.
I	n/a	I	Write MISS; a writethrough cycle is generated on the bus to update external memory. No allocation done.

NOTE: 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 3-6. 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.

4

Microprocessor Initialization and Configuration

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 V_{CC} is approaching nominal operating voltage to prevent internal bus contention which could negatively affect the reliability of the processor.

It is recommended that CLK begin toggling within 150ms after V_{CC} reaches its proper operating level. This recommendation is only to ensure long term reliability of the device.

In order for RESET to be recognized, the CLK input needs to be toggling. RESET must remain asserted for 1 millisecond after V_{CC} 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 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 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 V_{CC} 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 and TLBs
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
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 and Code Cache	Invalid	Invalid	Unchanged
Code Cache TLB, Data Cache TLB, BTB, SDC	Invalid	Invalid	Invalid

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

4.3.1. Recognition of Interrupts after RESET

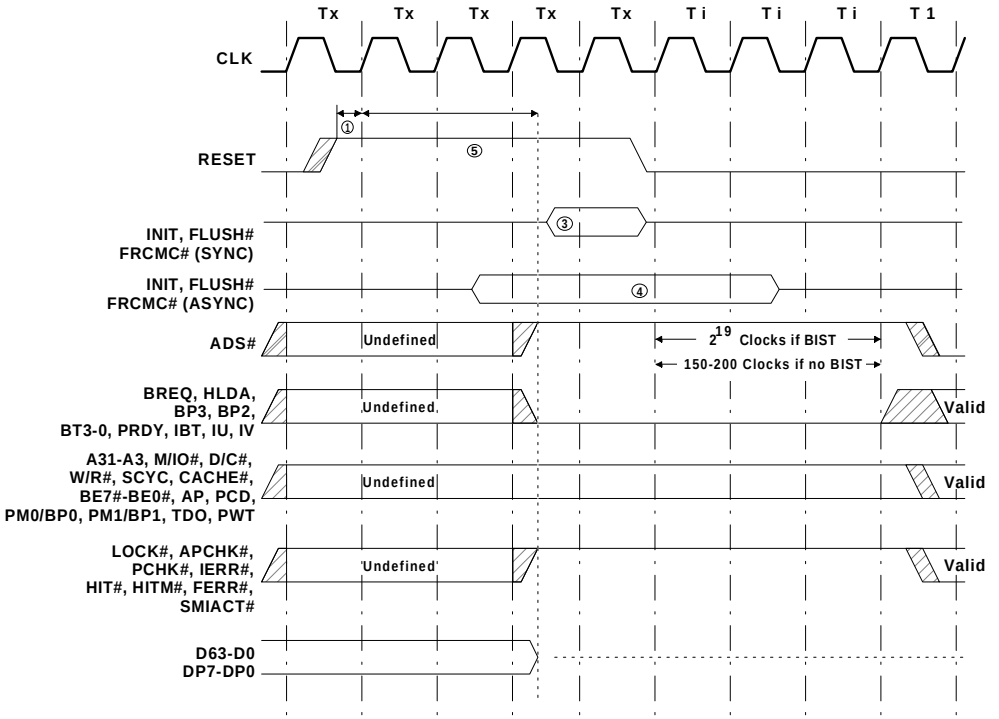
In order to guarantee recognition of the edge sensitive interrupts (FLUSH#, NMI, R/S#, SMI#) after RESET or after RESET with BIST, the interrupt input must not be asserted until four clocks after RESET is deasserted, regardless of whether BIST is run or not.

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, PM0/BP0, PM1/BP1 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, 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

5

Hardware Interface

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 pipe-lining 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 pipe-lines 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.

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 V_{CC} through a pullup resistor 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.

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 writeback 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.

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.

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.

NOTE

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#, then the sequence of bus cycles is as follows. Upon negation of BOFF#, the Pentium processor will drive out a writeback resutling from the internal snoop hit. After completion of the writeback, the processor will then restart the original read cycle. Thus, like external snoop writebacks, internal snoop writebacks may also be reordered in front of cycles that encounter a BOFF#. Also note that, although the original read encountered both an external BOFF# and an internal snoop hit to an M-state line, it is restarted only once.

This circumstance can occur during accesses to the page tables/directories and during prefetch cycles (these accesses cause a bus cycle to be generated before the internal snoop to the data cache is performed).

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.

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).

Because of the fractional-speed bus implemented in the Future Pentium OverDrive processor, the breakpoint pins, BP3-0, may indicate that one or more BP matches occurred.

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.

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#.

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 (510\60, 567\66) 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.

These signals are not supported on the Future Pentium OverDrive processor.

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.

These outputs are always undefined on the Future Pentium OverDrive processor.

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.

NOTE

The Pentium processor can remember only one machine check exception at a time. This exception is recognized on an instruction boundary. If BUSCHK# is sampled active while servicing the machine check exception for a previous BUSCHK#, it will be remembered by the processor until the original machine check exception is completed. It is then that the processor will service the machine check exception for the second BUSCHK#. Note that only one BUSCHK# will be remembered by the processor while the machine exception for the previous one is being serviced.

When the BUSCHK# is sampled active by the processor, the cycle address and cycle type information for the failing bus cycle is latched upon assertion of the last BRDY# of the bus cycle. The information is latched into the Machine Check Address (MCA) and Machine Check Type (MCT) registers respectively. However, if the BUSCHK# input is not deasserted before the first BRDY# of the next bus cycle, and the machine check exception for the first bus cycle has not occurred, then new information will be latched into the MCA and MCT registers, over-writing the previous information at the completion of this new bus cycle. Therefore, in order for

the MCA and MCT registers to report the correct information for the failing bus cycle when the machine check exception for this cycle is taken at the next instruction boundary, the system must deassert the BUSCHK# input immediately after the completion of the failing bus cycle (i.e., before the first BRDY# of the next bus cycle is returned).

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#.

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. It is recommended that CLK begin toggling within 150ms after V_{CC} reaches its proper operating level. This recommendation is only to ensure long term reliability of the device. V_{CC} 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 V_{CC} 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 writethrough 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 writethrough 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. In the event of a pending unmasked floating-point exception the FNINIT, FNCLEX, FNSTENV, FNSAVE, FNSTSW, FNSTCW, FNENI, FNDISI, and FNSETPM instructions assert the FERR# pin. Shortly after the assertion of the pin, an interrupt window is opened during which the processor samples and services interrupts, if any. If no interrupts are sampled within this window, the processor will then execute these instructions with the pending unmasked exception. However, for the FNCLEX, FNINIT, FNSTENV, and FNSAVE instructions, the FERR# pin is deasserted to enable the execution of these instructions. For details please refer to Section 23.3.7 in the Volume 3: *Architecture and Programming Manual* of the *Pentium® Processor Family Developer's Manual.*

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

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 a branch trace message special cycle will be driven subsequent to the assertion of IBT.

This signal is not supported on the future Pentium OverDrive processor.

When Driven

This output is always driven by the Pentium processor (510\60, 567\66). It is driven high for 1 clock each time a branch is taken. It is not floated during bus HOLD or BOFF#.

This signal is always driven inactive (LOW) by the Future Pentium OverDrive processor.

NOTE

The Pentium processor treats **some segment descriptor loads as** causing taken branches. This operation causes the IBT pin to be asserted. If execution tracing is enabled, then this operation will also cause a corresponding Branch Trace Message Special Cycle to be driven.

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.

NOTE

When paging is turned on, an additional parity check occurs to page 0 for all TLB misses. If this access is a valid entry in the cache and this entry also has a parity error, then IERR# will be asserted and shutdown will occur even though the pipe-line is frozen to service the TLB miss.

During a TLB miss, a cache lookup occurs (to the data cache for a data TLB miss, or the code cache for a code TLB miss) to a default page 0 physical address until the correct page translation becomes available. At this time, if a valid cache entry is found at the page 0 address, then parity will be checked on the data read out of the cache. However, the data is not

used until after the correct page address becomes available. If this valid line contains a true parity error, then the error will be reported. This will not cause an unexpected parity error. It can cause a parity error and shutdown at a time when the data is not being used because the pipe-line is frozen to service the TLB miss. However, it still remains that a true parity error must exist within the cache in order for IERR# assertion and shutdown to occur. For more details on TLB, refer to Chapter 11 of the *Pentium® Processor Family Developer's Manual*, Volume 3.

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 FNINIT, FNCLEX, FNSTENV, FNSAVE, FNSTSW, FNSTCW, FNENI, FNDISI, or FNSETPM, the Pentium processor will assert the FERR# pin. Subsequently, the processor opens an interrupt sampling window. The interrupts are checked and serviced during this window. If no interrupts are sampled within this window the processor will then execute these instructions in spite of the pending unmasked exception. For further details please refer to Section 23.3.7 in the Volume 3: *Architecture and Programming Manual* of the *Pentium® Processor Family Developer's Manual.*

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 completed 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.

This signal is not supported on the Future Pentium OverDrive processor.

When Driven

This pin is always driven by the Pentium processor (510\60, 567\66). It is not floated during bus HOLD or BOFF#.

This signal is always driven inactive (LOW) by the Future Pentium OverDrive processor.

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.

This signal is not supported on the Future Pentium OverDrive processor.

When Driven

This pin is always driven by the Pentium processor (510\60, 567\66). It is not floated during bus HOLD or BOFF#.

This signal is always driven inactive (LOW) by the Future Pentium OverDrive processor.

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.

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 pipe-line 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.

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 pipe-lined 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 pipe-lining.

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 pipe-line an additional cycle if LOCK# is asserted, or during a writeback cycle. In addition, ADS# will not be asserted to pipe-line 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 pipe-line 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. If this pin is sampled inactive, it does not prevent PCHK# from being asserted in response to a bus parity error. If systems are using PCHK#, they should be aware of this usage of PEN#.

This pin may be tied to V_{SS}.

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 an 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	Pin Name: Performance Monitoring and Breakpoint
	Function: PM1-0 externally indicate the status of the performance monitor counter.
	Input/Output: Output pins

Signal Description

The performance monitoring pins can be individually configured to externally indicate either that the associated performance monitoring counter has incremented or that it has overflowed. PM1 indicates the status of CTR1, PM0 indicates the status of CTR0.

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 .

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#.

NOTE

The PM1/PM0 pins externally indicate the status of the performance monitoring counters on the Pentium processor. These counters are undefined after RESET, and must be cleared or pre-set (using the WRMSR instruction) before they are assigned to specific events. The Pentium processor databook in section 4.3.2. lists the state of these two output pins (PM0/PM1) as undefined until at least 2 clocks after RESET is asserted. It is however possible for these pins to toggle even during RESET. This may occur ONLY if the RESET pin was asserted while the Pentium processor was in the process of counting a particular performance monitoring event. Since the even counters continue functioning until the CESR (Control and Event Select Register) is cleared by RESET, it is possible for the event counters to increment even during RESET. Externally the state of the event counters would also be reflected on the PM1/PM0 pins. Any assertion of the PM1/PM0 pins during RESET should be ignored until after the start of the first bus cycle.

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.

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 Writethrough
	Externally reflects the writethrough paging attribute bit in CR3, PDE, or PTE.
	Output

Signal Description

PWT is driven to externally reflect the cache writethrough 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 effect 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 where it is optionally capable of executing probe mode instructions. 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 (510\60, 567\66). 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 (510\60, 567\66) then generates a prefetch cycle to re-fill the instruction pipe-line 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.

NOTE

Deasserting R/S# while PRDY is asserted will cause the processor to exit probe mode and return to normal executing. The behavior of the processor is unpredictable if R/S# is deasserted to exit probe mode while PRDY is deasserted.

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 V_{CC} and CLK have reached their AC/DC specifications for the "cold" or "power on" reset. During power up, RESET should be asserted while V_{CC} 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 V_{CC} 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 only driven during the length of the locked cycle that is split. SCYC is asserted with the first ADS# of a misaligned split cycle and remains valid until the clock after the earlier of NA# or the last BRDY# of the last split 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.

NOTE

The current Pentium processor specification allows SMI# to be recognized while in shutdown state. However, if SMM is entered from shutdown state, the following must be considered:

1. if FLUSH# is asserted after the processor has entered SMM from a shutdown state, the processor will service the FLUSH# and then re-enter the shutdown state rather than returning to SMM. The systems should either assert SMI# and FLUSH# simultaneously or prevent FLUSH# from being asserted while SMIACT# is active.
2. Servicing an SMI# request during the shutdown state could potentially further corrupt the system if the shutdown state occurred as a result of an error encountered during the RSM instruction (misaligned SMBASE, reserved bit of CR4 is set to '1', etc.)

Once the system has detected that the processor has entered shutdown state (through the special bus cycle), it should generate an NMI interrupt or

invoke reset initialization to get the processor out of the shutdown state before allowing an SMI# to be asserted to the processor.

Relation to Other Signals

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

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).

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 V_{CC} 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 writeback or writethrough on a line by line basis.
	Synchronous Input

Signal Description

This pin allows a cache line to be defined as writeback or writethrough 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."

6

Bus Functional Description

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).

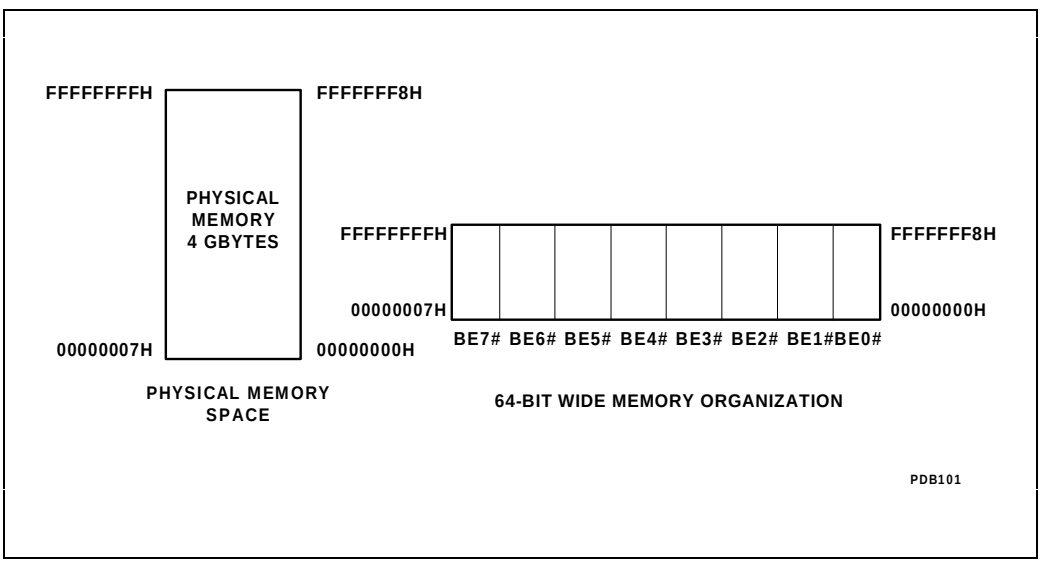


Figure 6-1. Memory Organization

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.

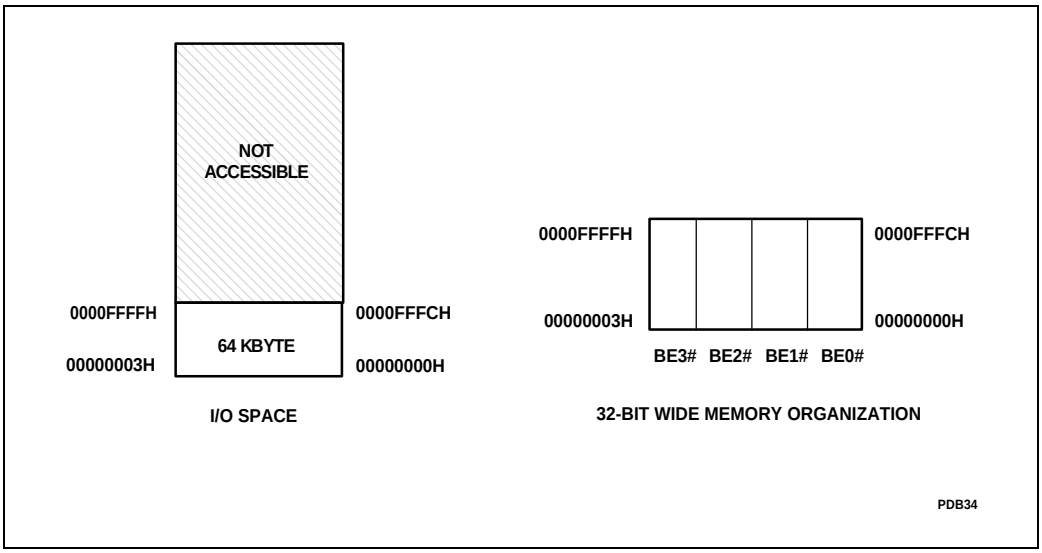


Figure 6-2. I/O Space Organization

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, BHE#, and 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 an 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. For both memory and I/O accesses, the byte enable outputs indicate which of the associated data bus bytes are driven valid for write cycles and on which bytes data is expected back for read cycles. 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 6-8.

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 an 8-byte qword select and BE7#-BE0# select individual bytes within the word.

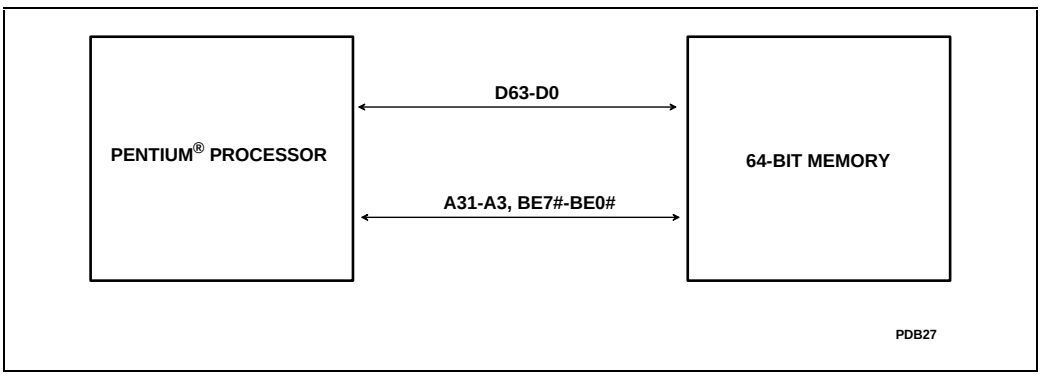


Figure 6-3. Pentium® Processor with 64-Bit Memory

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.

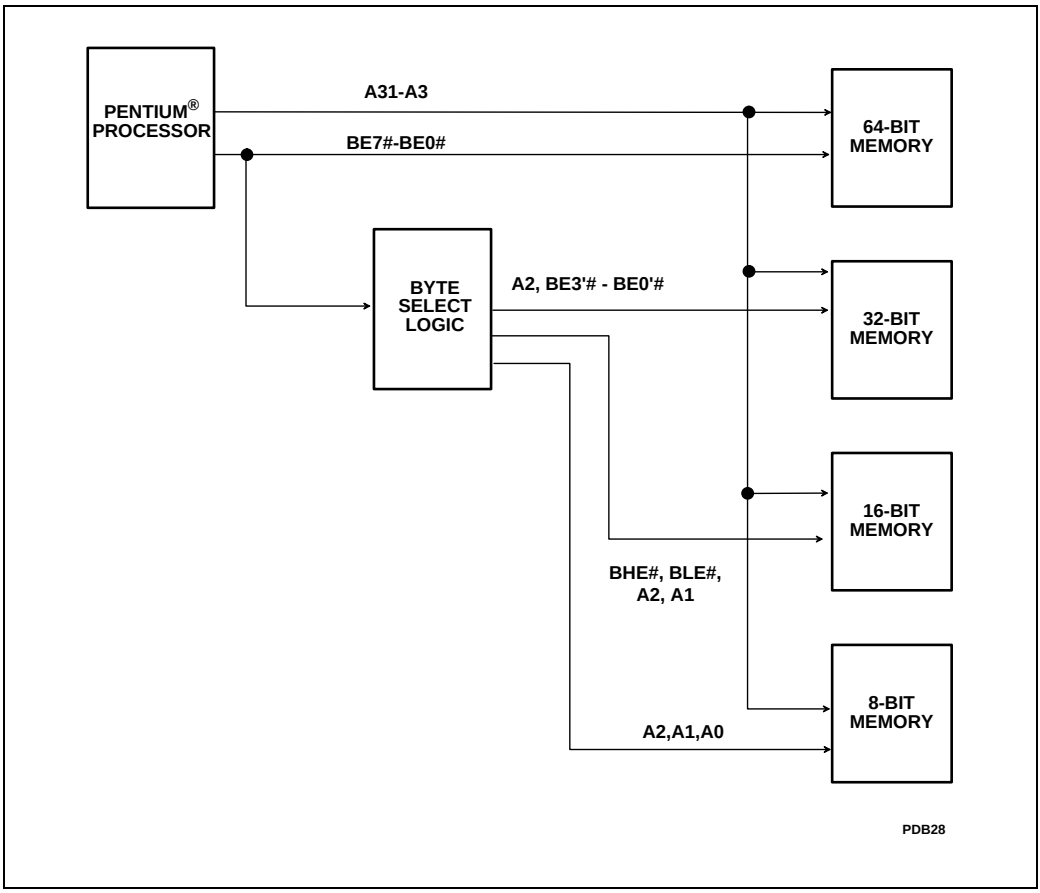


Figure 6-4. Addressing 32-, 16- and 8-Bit Memories

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.

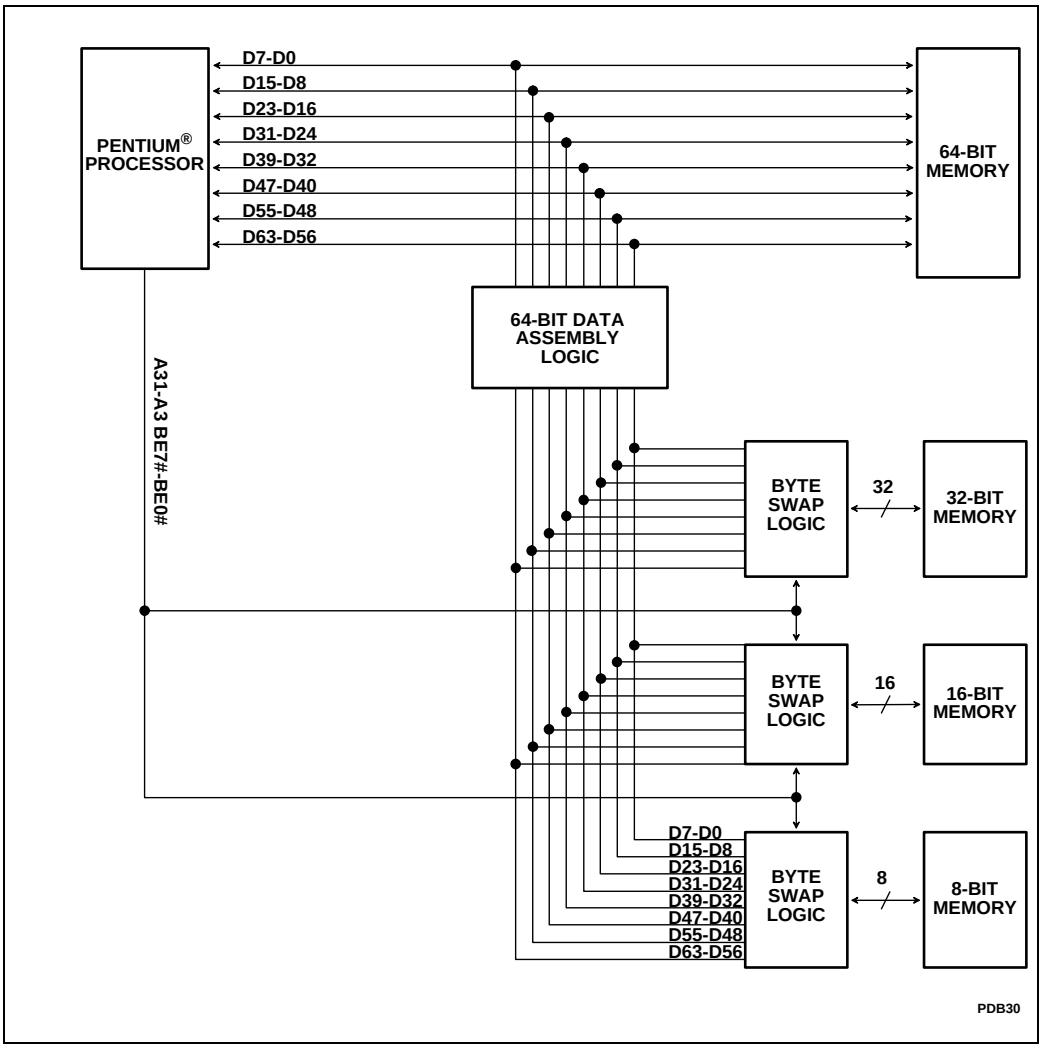


Figure 6-5. Data Bus Interface to 32-, 16- and 8-Bit Memories

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 and 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.

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	Writeback (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 wait states 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.

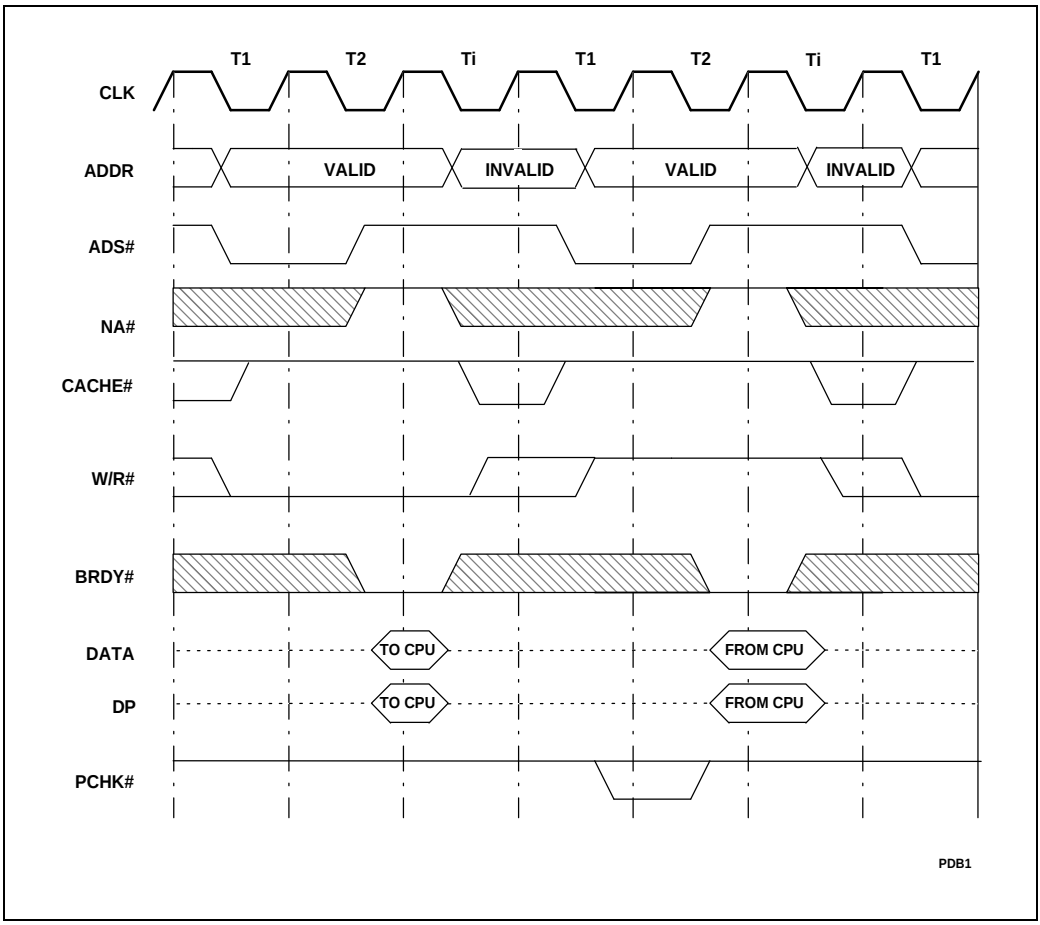


Figure 6-6. Non-Pipelined Read and Write

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.

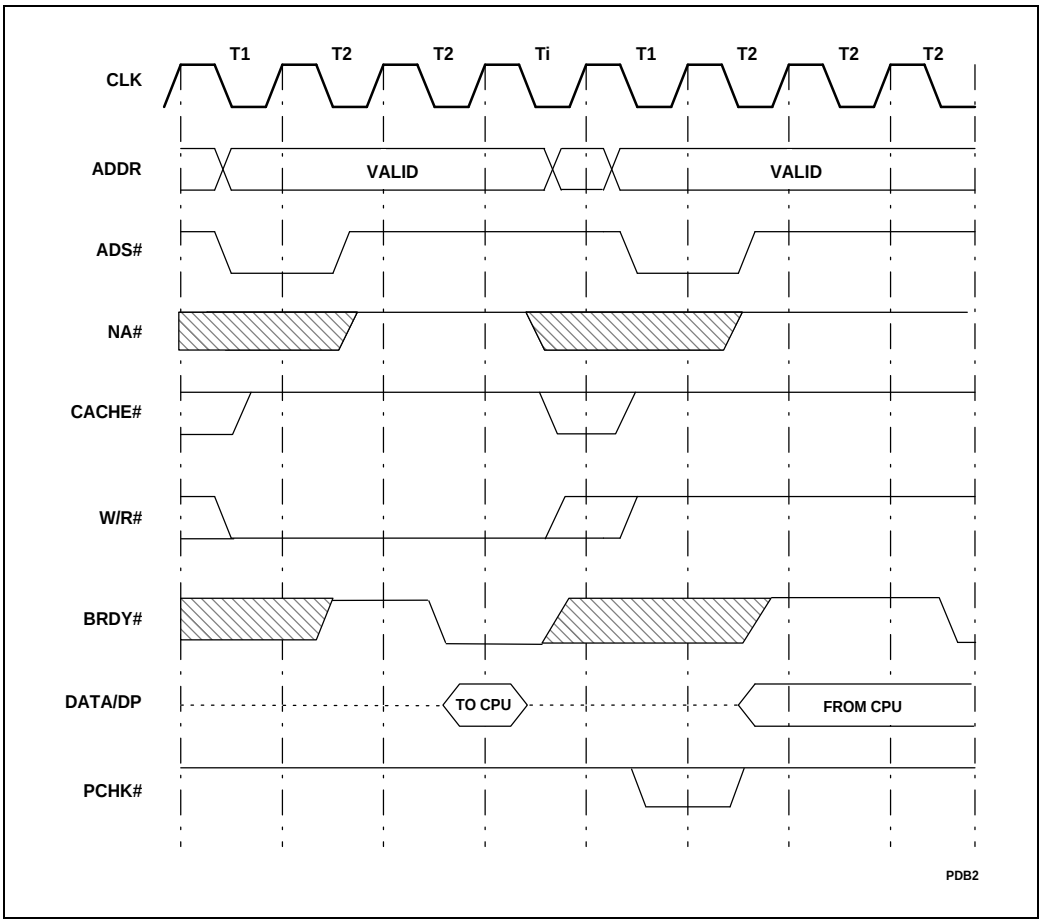


Figure 6-7. Non-Pipelined Read and Write with Wait States

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 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#.

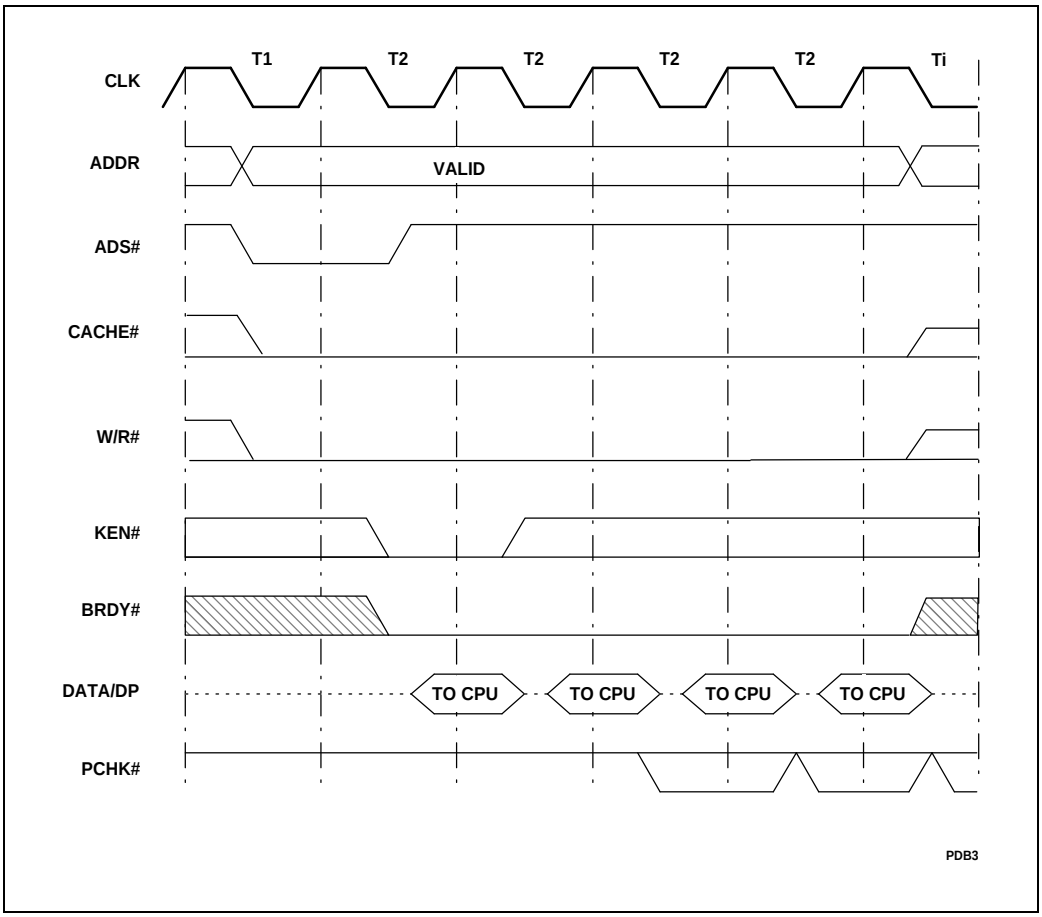


Figure 6-8. Basic Burst Read Cycle

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.

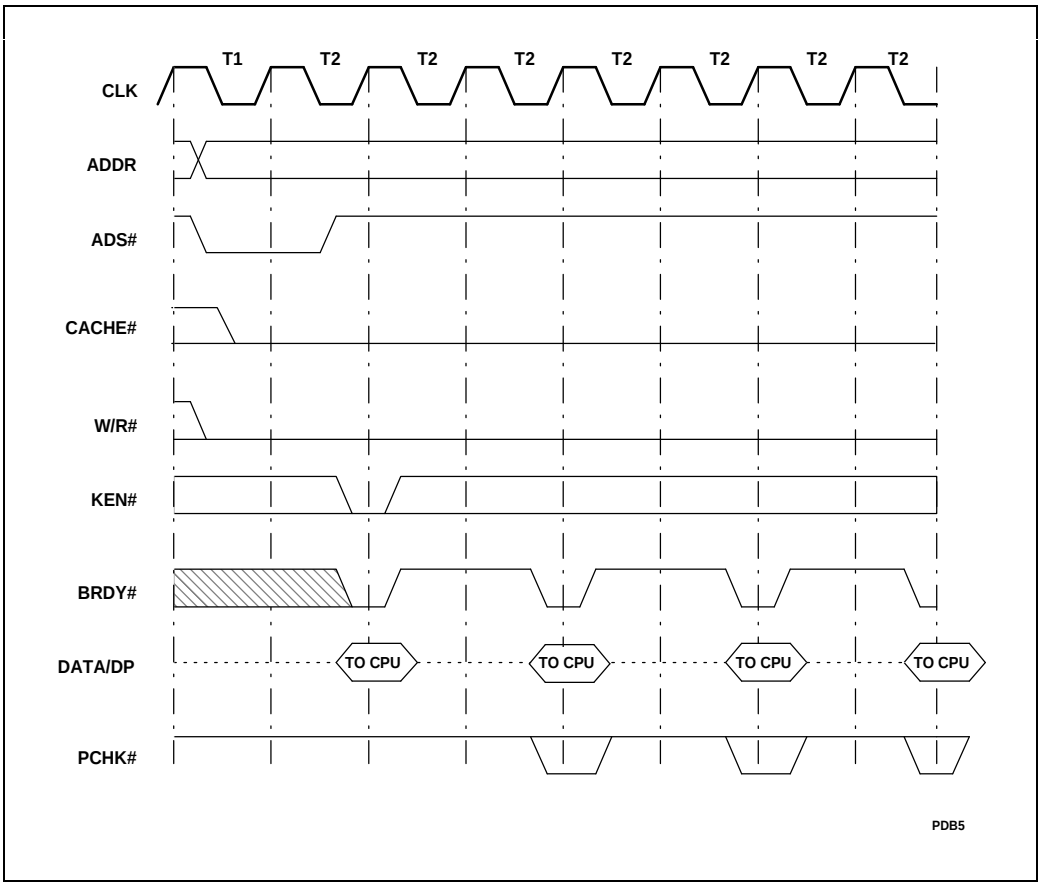


Figure 6-9. Slow Burst Read Cycle

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 bits or less, single transfer bus cycles.

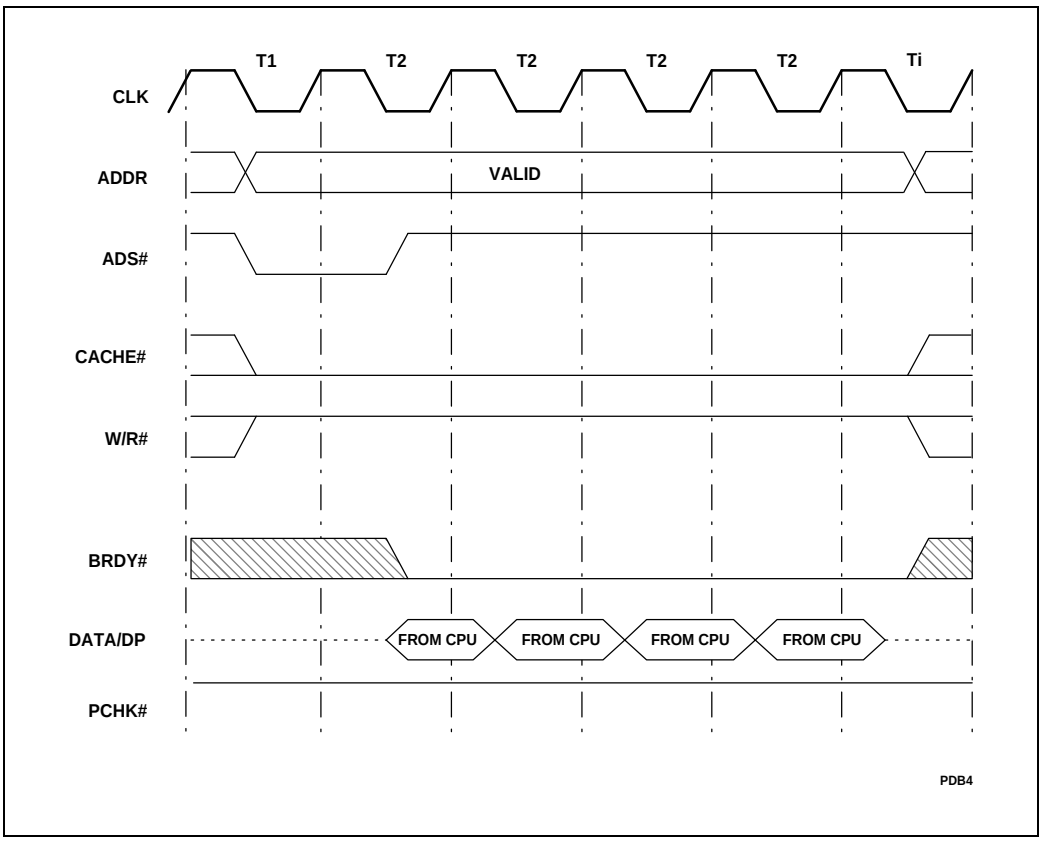


Figure 6-10. Basic Burst Write Cycle

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 an 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 LOCK# 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.

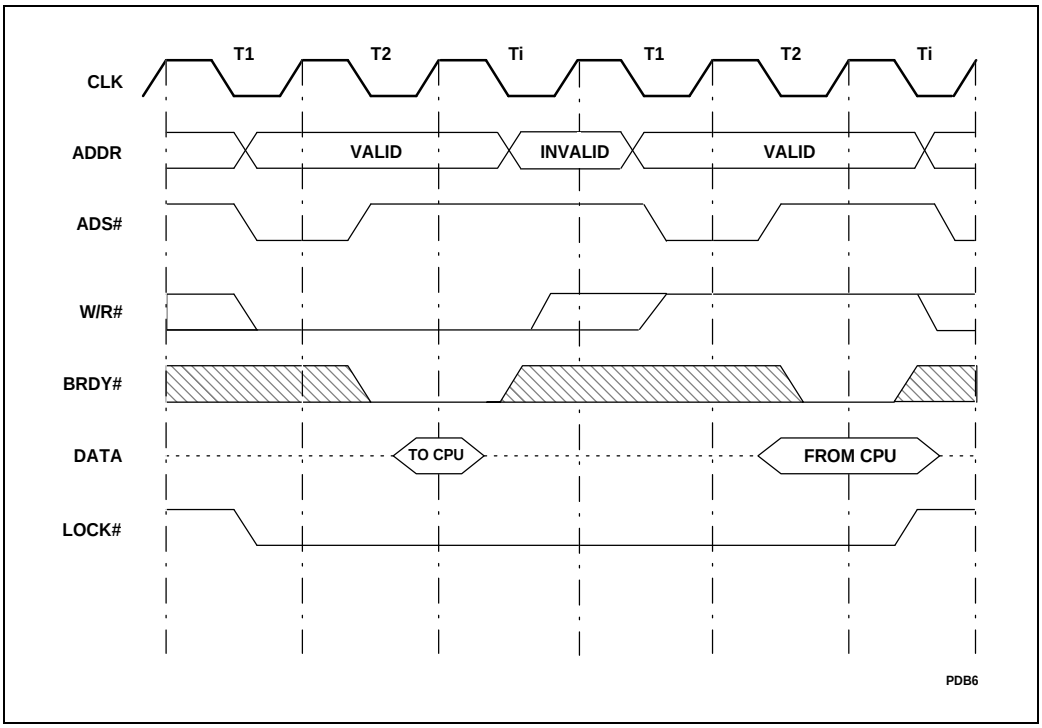


Figure 6-11. LOCK# Timing

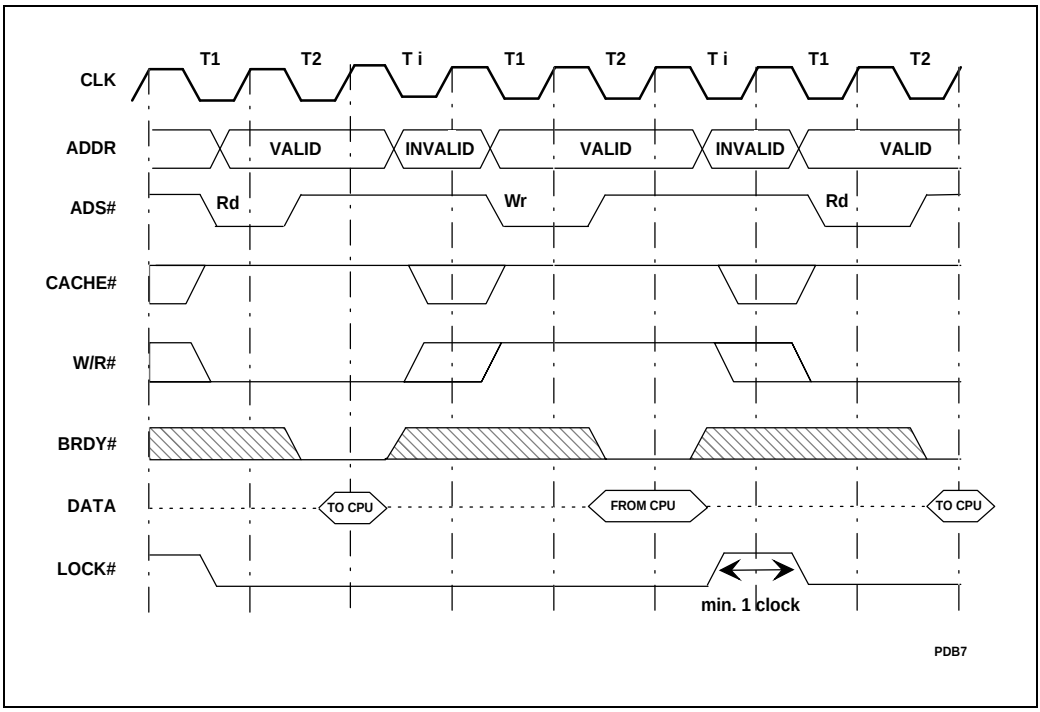


Figure 6-12. Two Consecutive Locked Operations

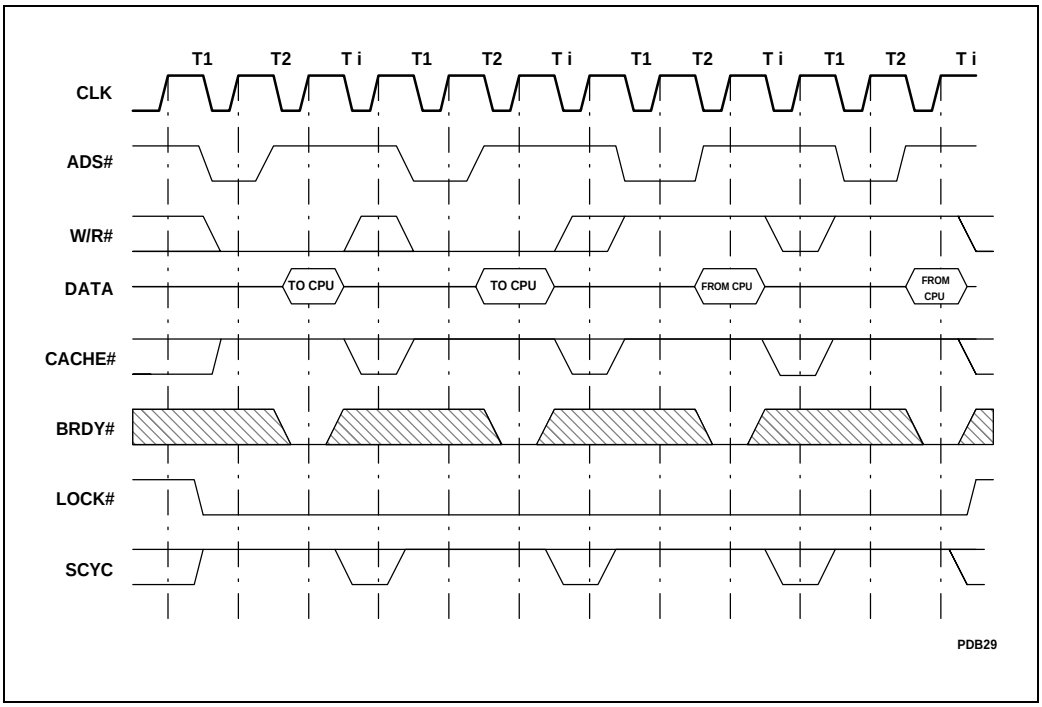


Figure 6-13. Misaligned Locked Cycles

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.

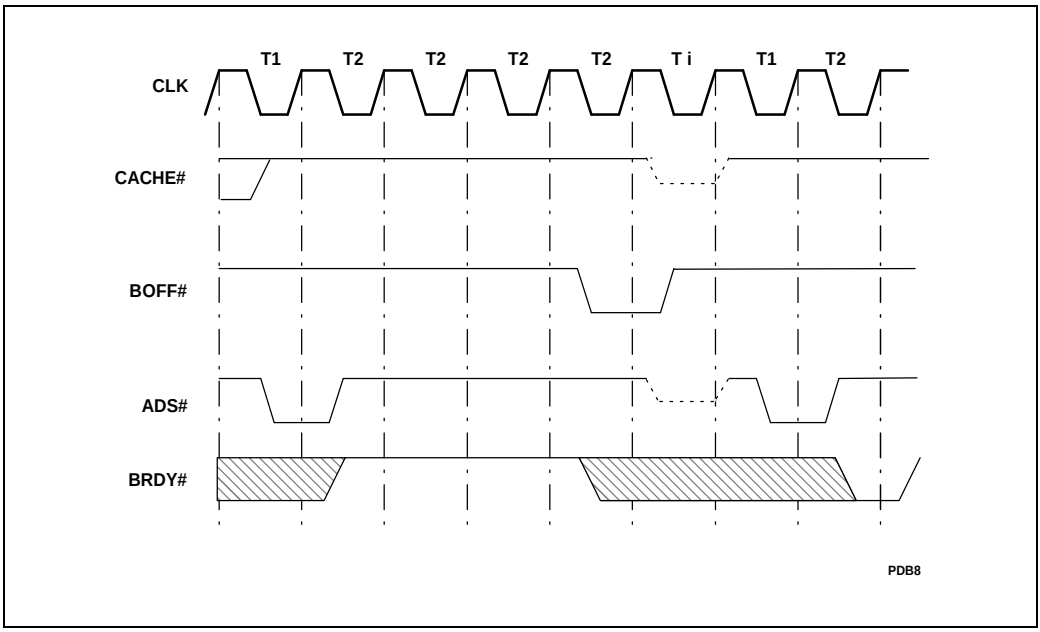


Figure 6-14. Back Off Timing

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.

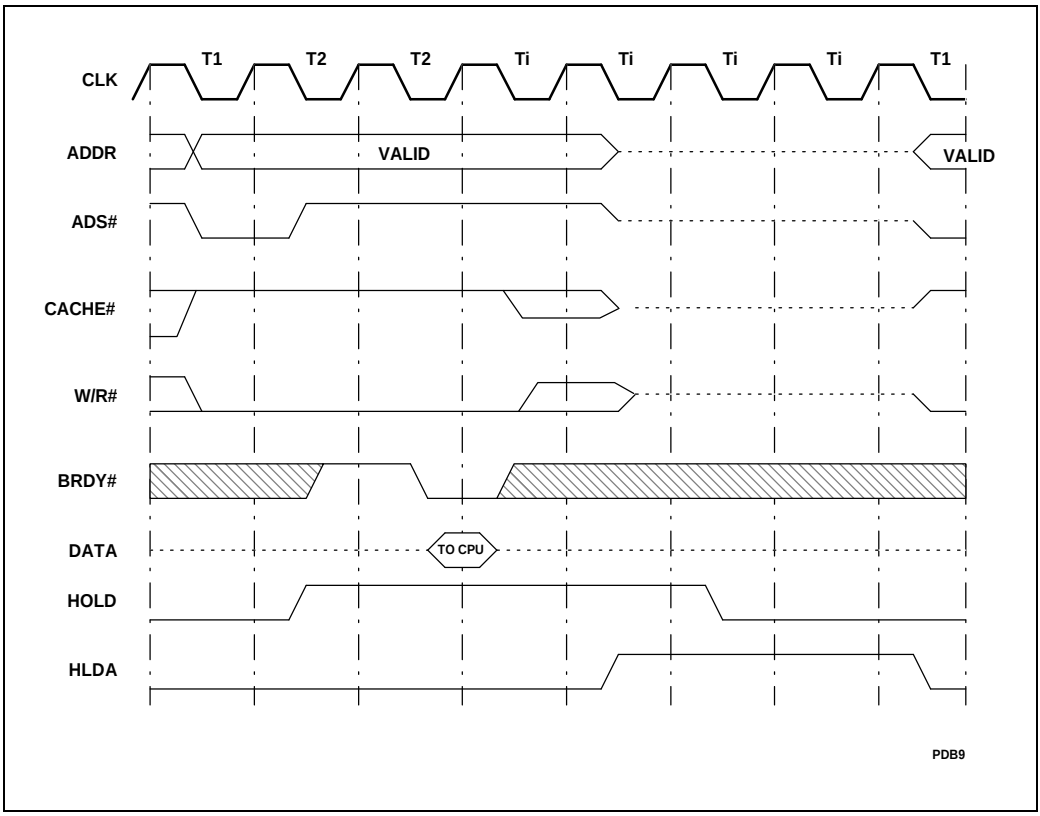


Figure 6-15. HOLD/HLDA Cycles

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.

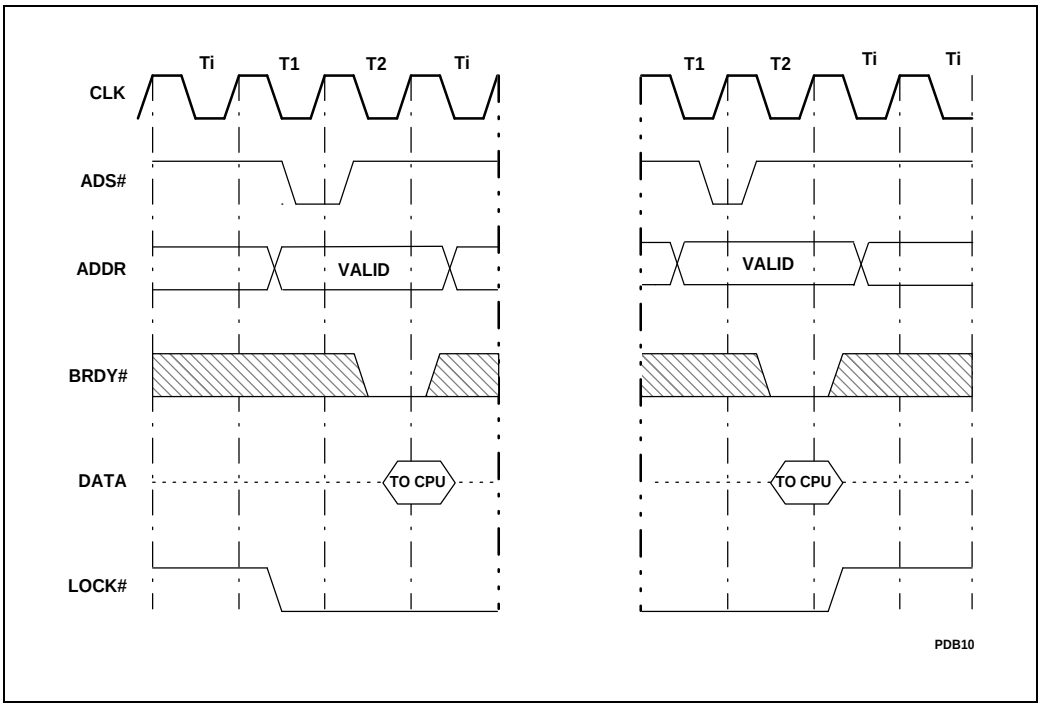


Figure 6-16. Interrupt Acknowledge Cycles

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	Writeback (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.
2. An internal parity error is detected.

Prior to going into shutdown, the Pentium processor will not writeback 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 (writeback 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

INVD should be used with care. This instruction does not write back modified cache lines. See Instruction set Volume 3 for details.

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.
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 Writeback 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 Writeback 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 writeback 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 (510\60, 567\66) 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.

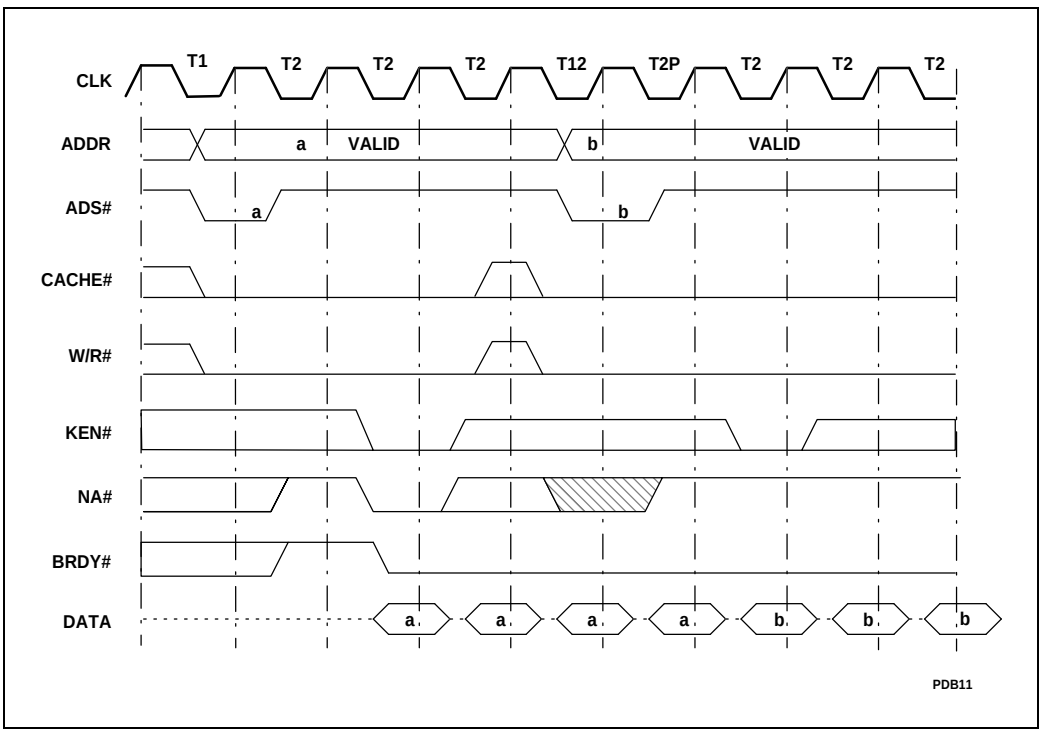


Figure 6-17. Two Pipelined Cache Line Fills

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.

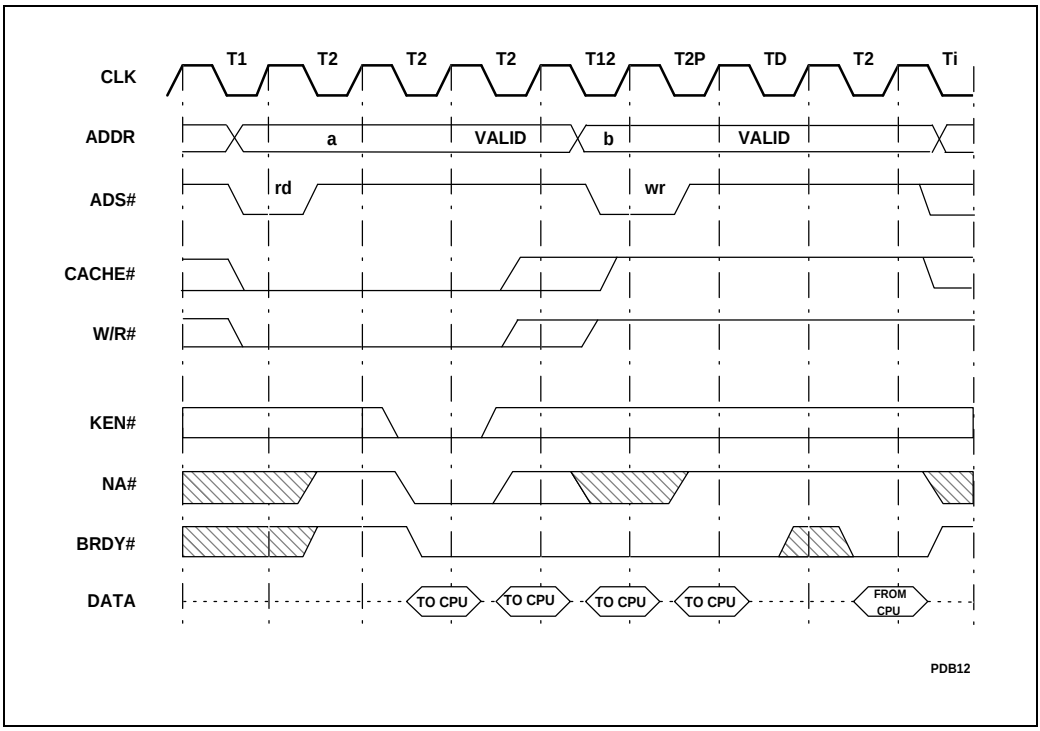


Figure 6-18. Pipelined Back-to-Back Read/Write Cycles

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.

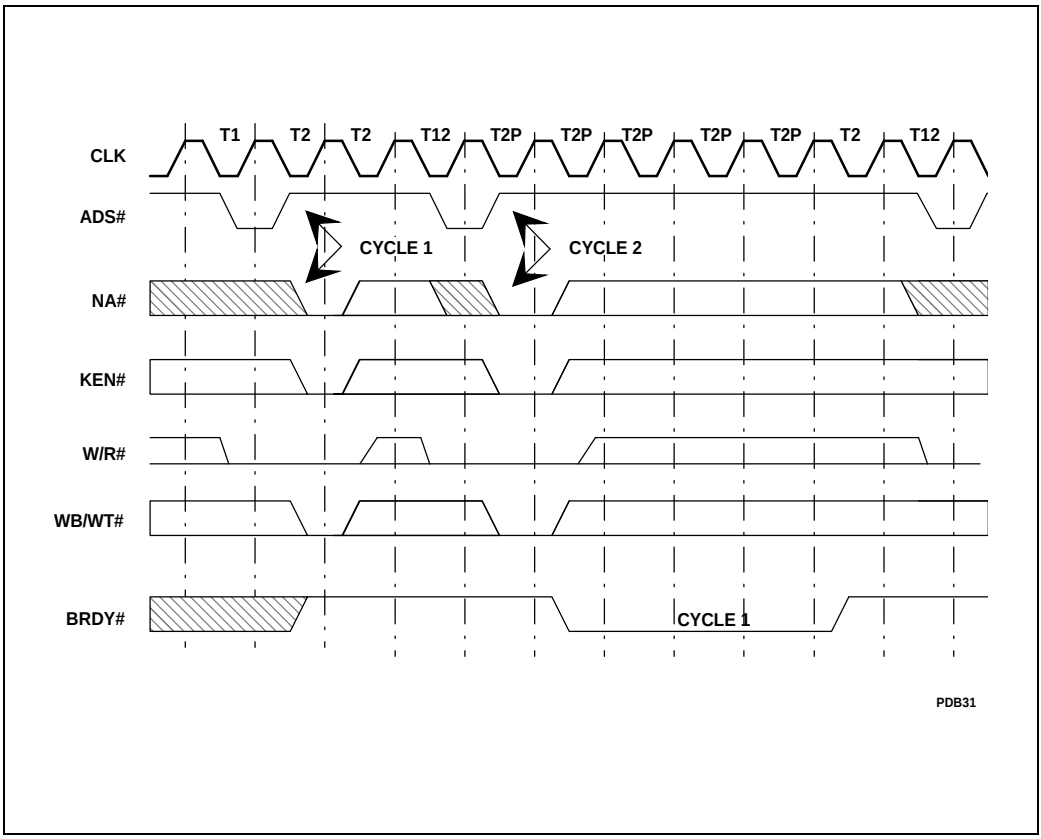


Figure 6-19. KEN# and WB/WT# Sampling with NA#

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.

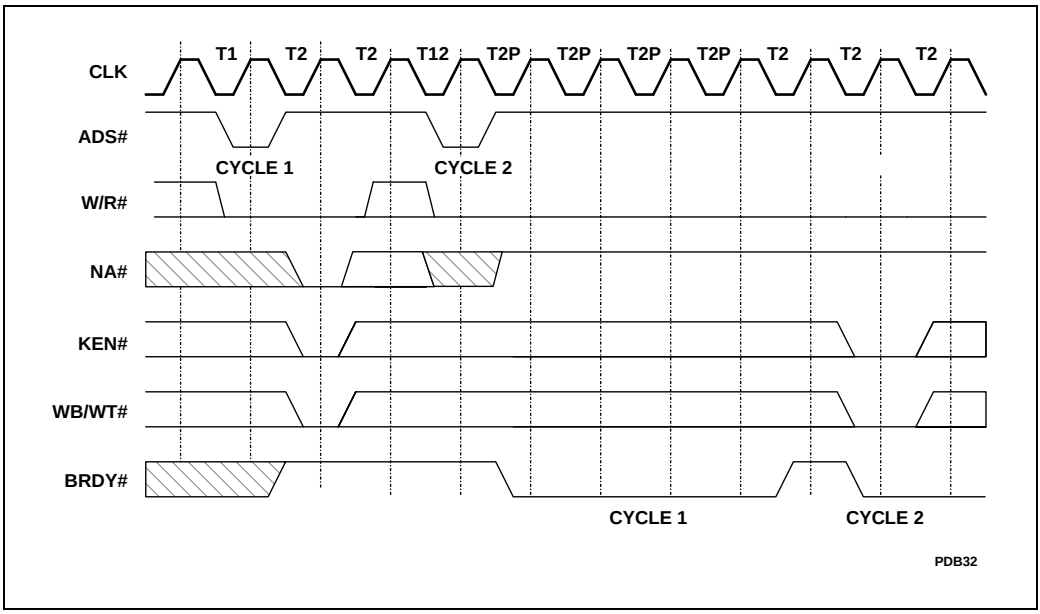


Figure 6-20. KEN# and WB/WT# Sampling with BRDY#

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#.

---

[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.

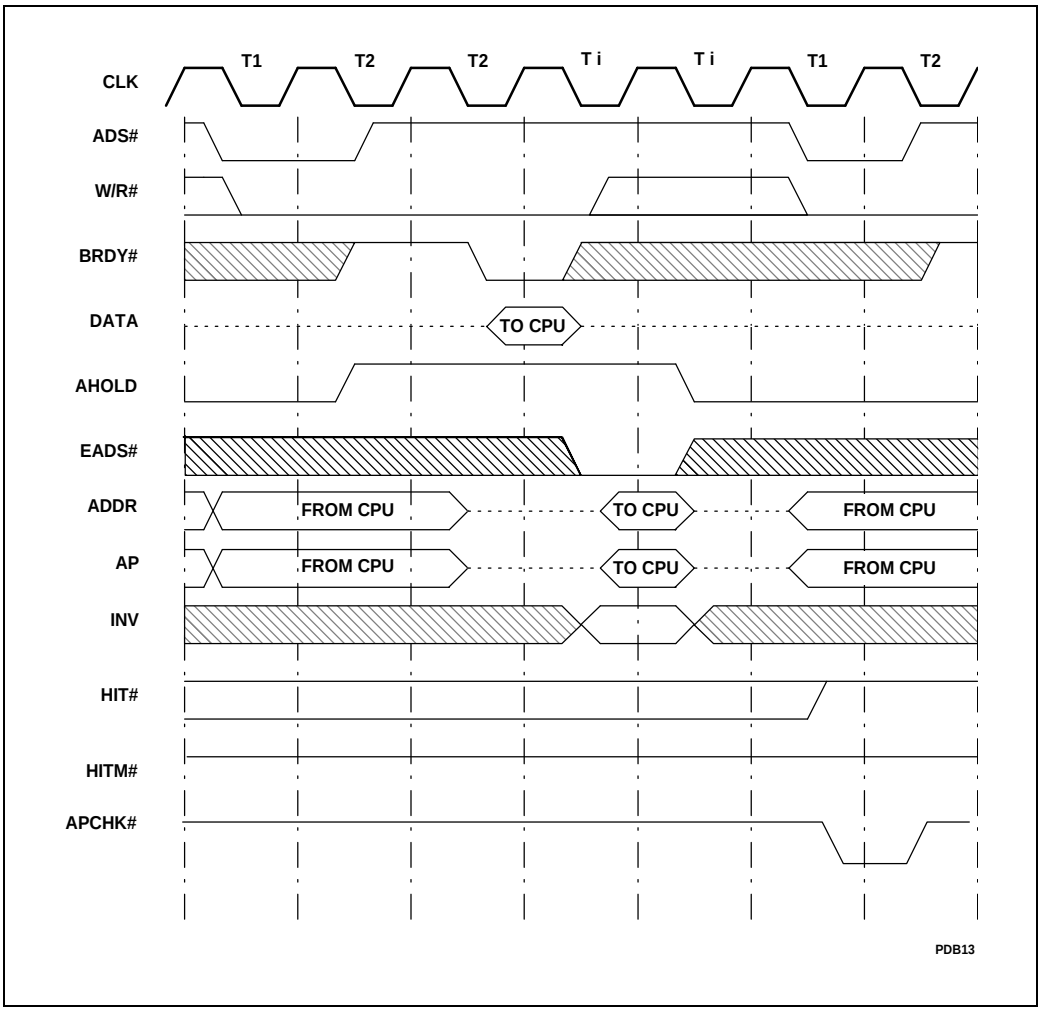


Figure 6-21. Inquire Cycle that Misses Pentium® Processor Cache

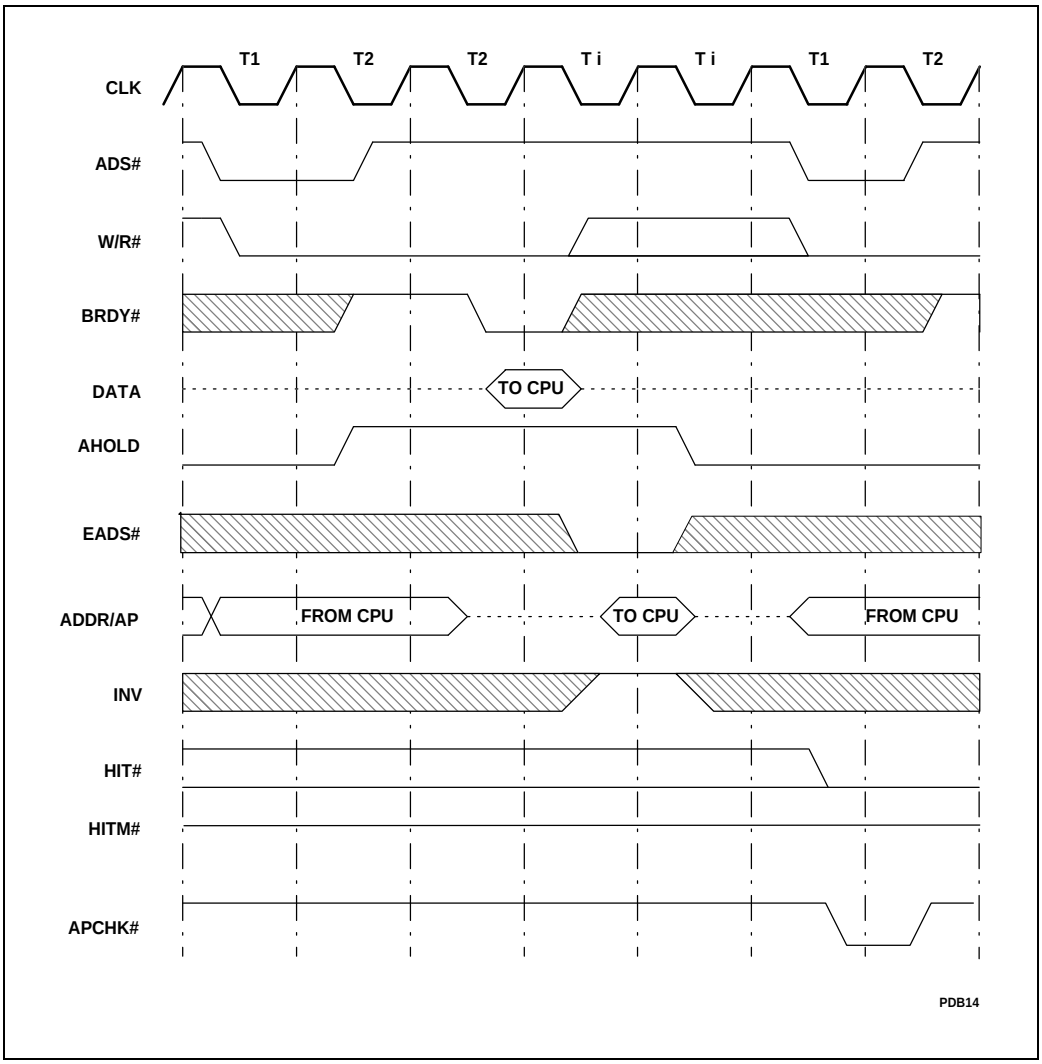


Figure 6-22. Inquire Cycle that Invalidates Non-M-State Line

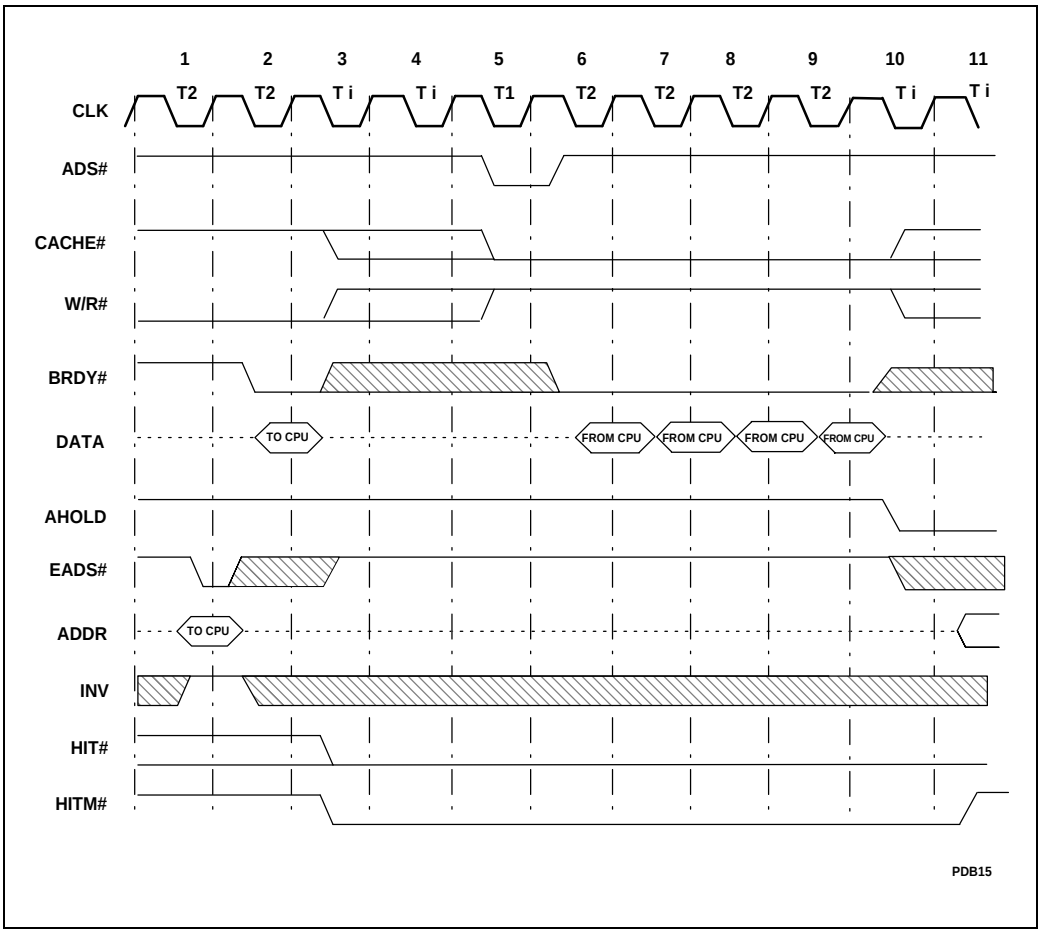


Figure 6-23. Inquire Cycle that Invalidates M-State Line

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 writeback 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 writeback 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.

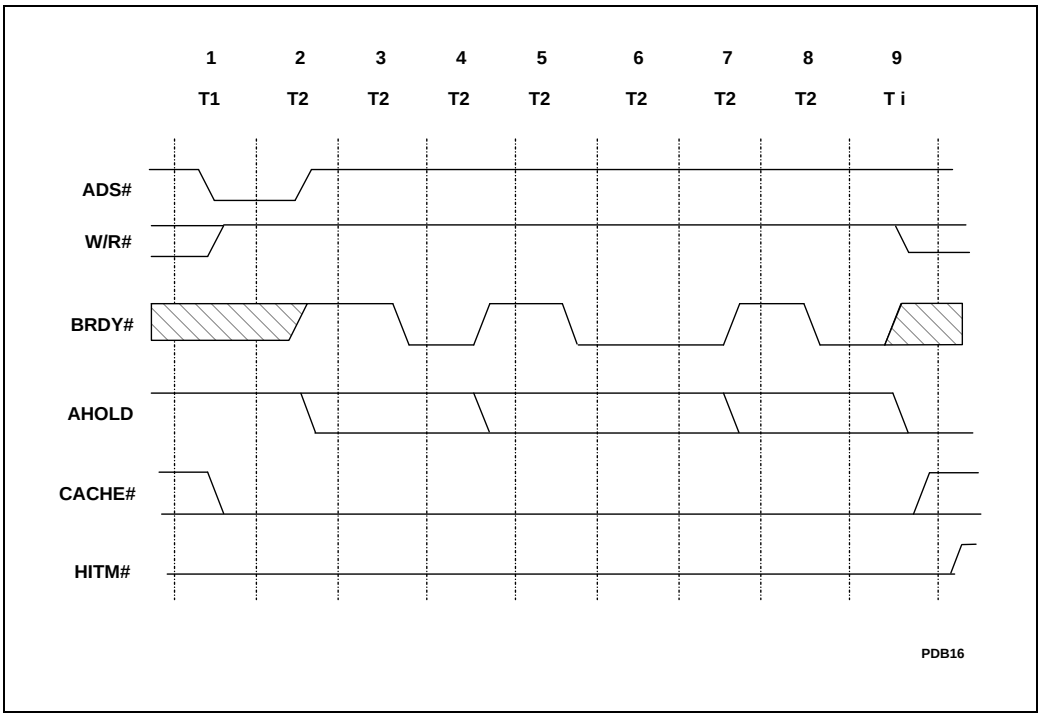


Figure 6-24. AHOLD Restriction during Write Cycles

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.

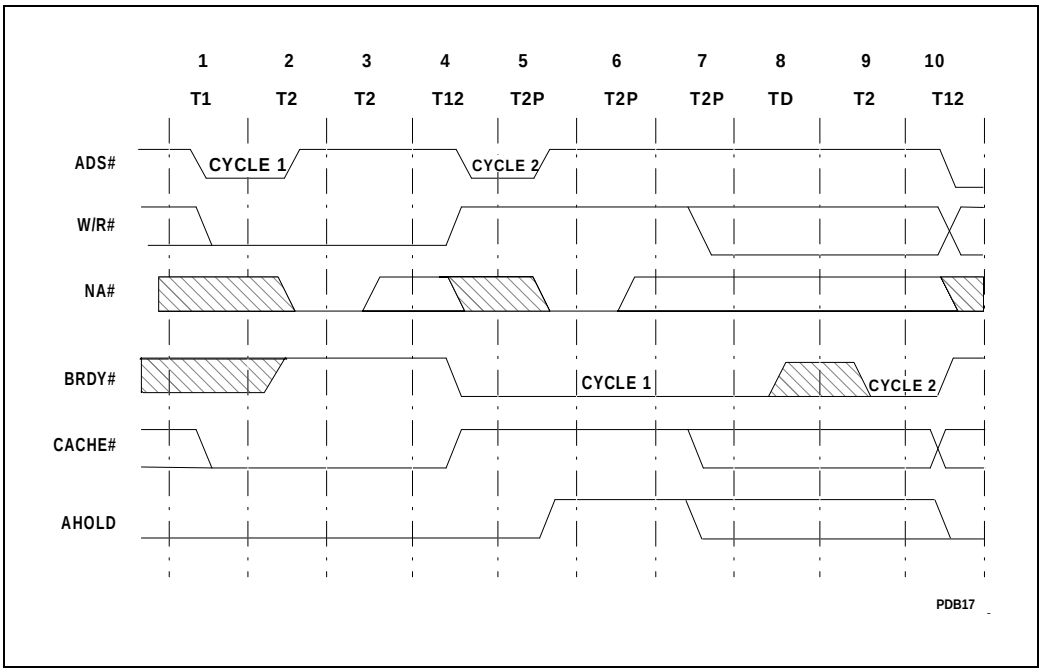


Figure 6-25. AHOLD Restriction during TD

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 there is a write to the accessed and/or dirty bits in the page table/directory entries. 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#.

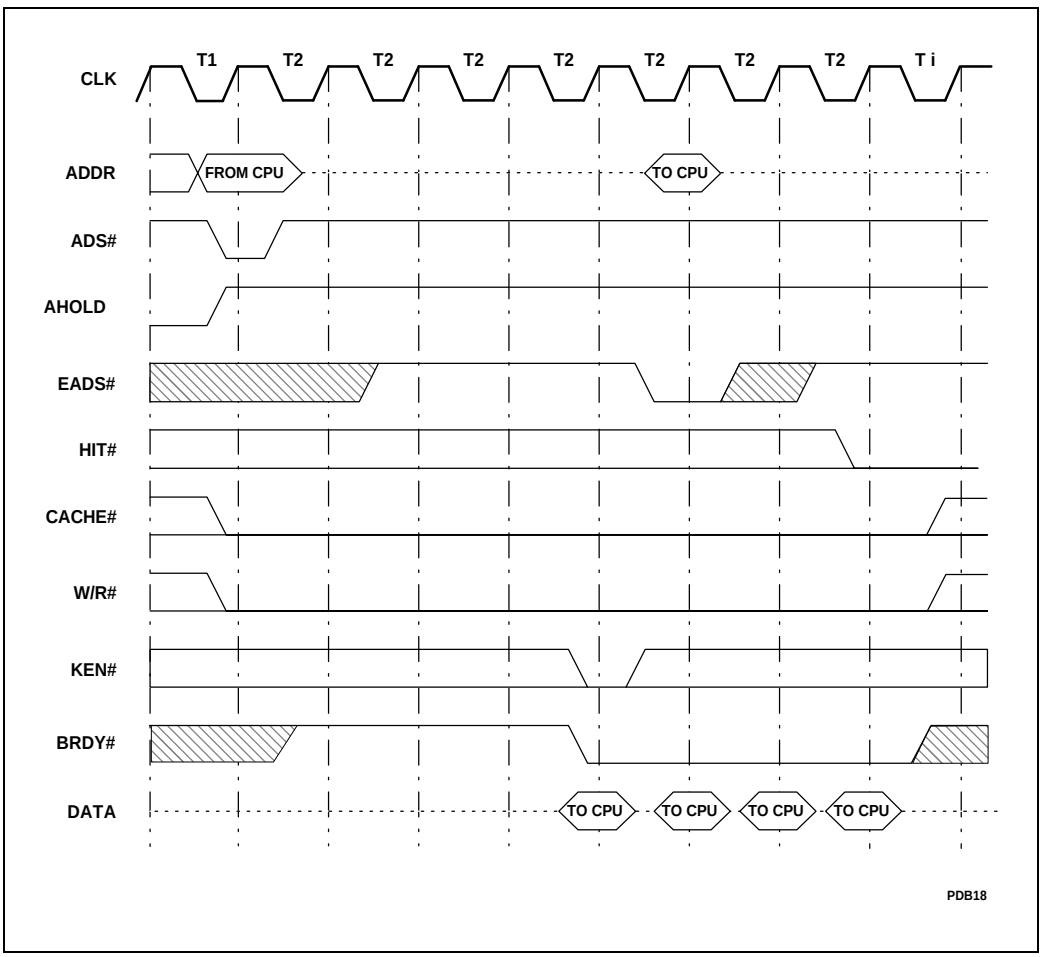


Figure 6-26. Snoop Responsibility Pickup — Non-Pipelined Cycles

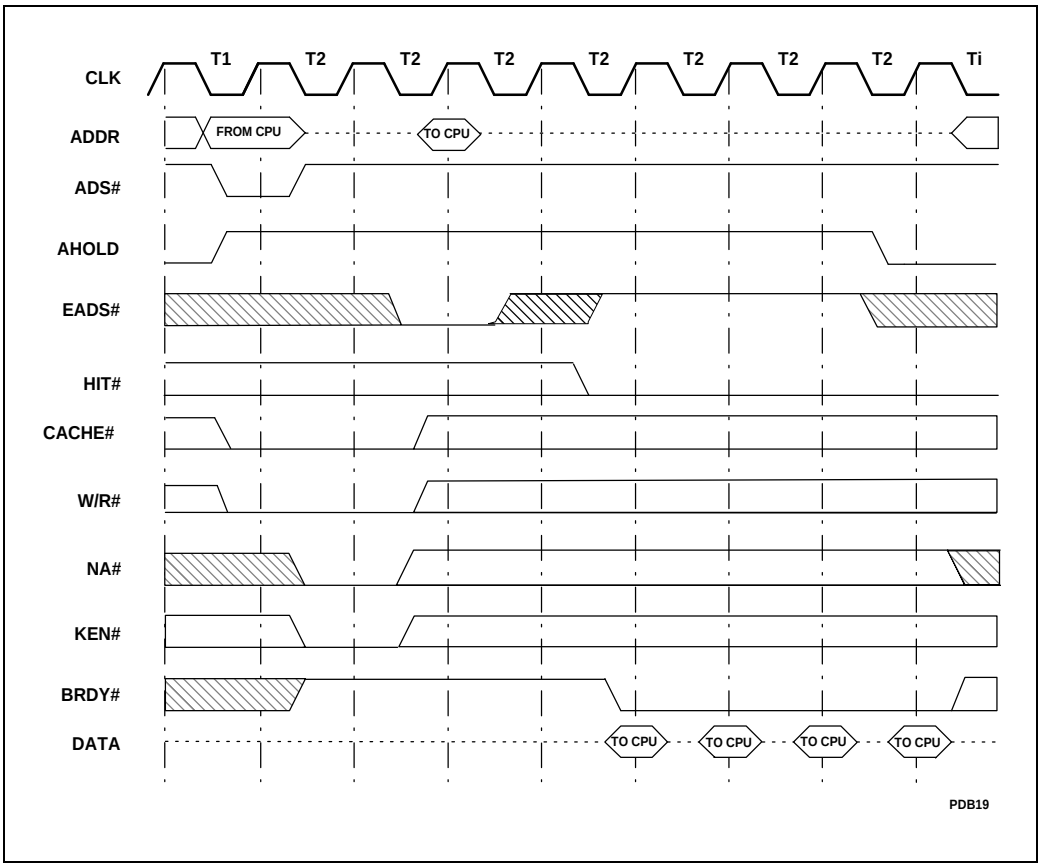


Figure 6-27. Snoop Responsibility Pickup — Pipelined Cycle

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#.

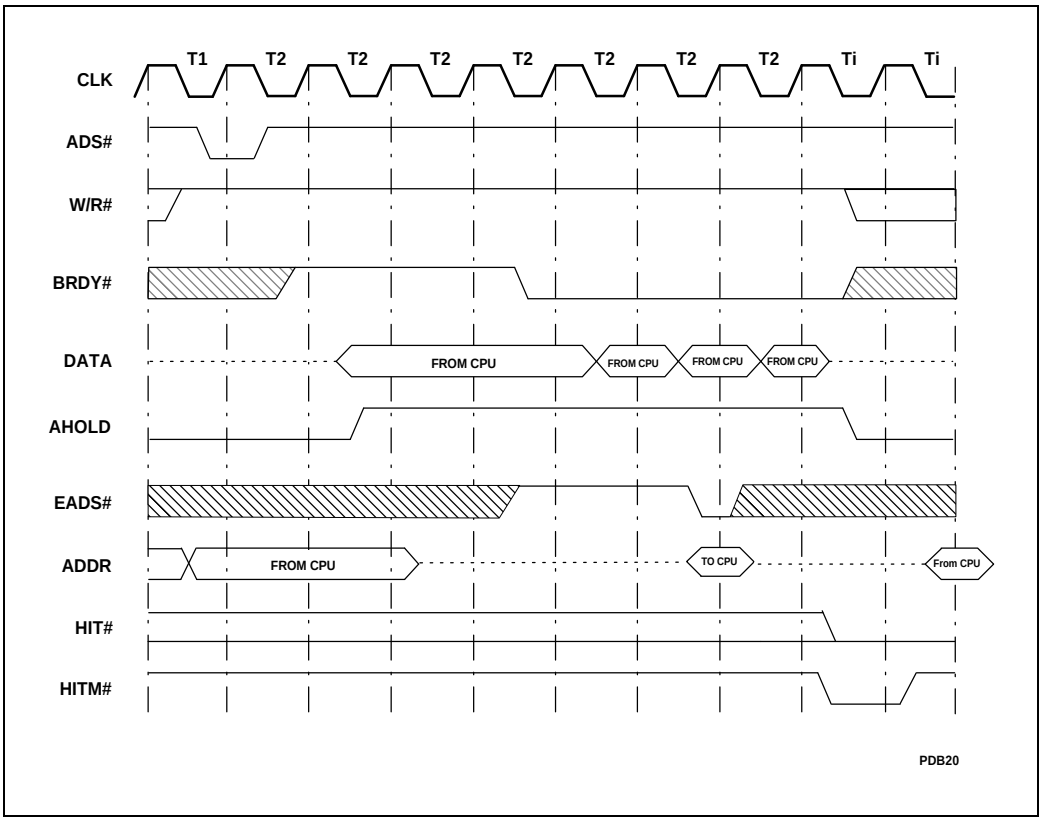


Figure 6-28. Latest Snooping of Writeback Buffer

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 writeback 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 (510\60, 567\66) 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.

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

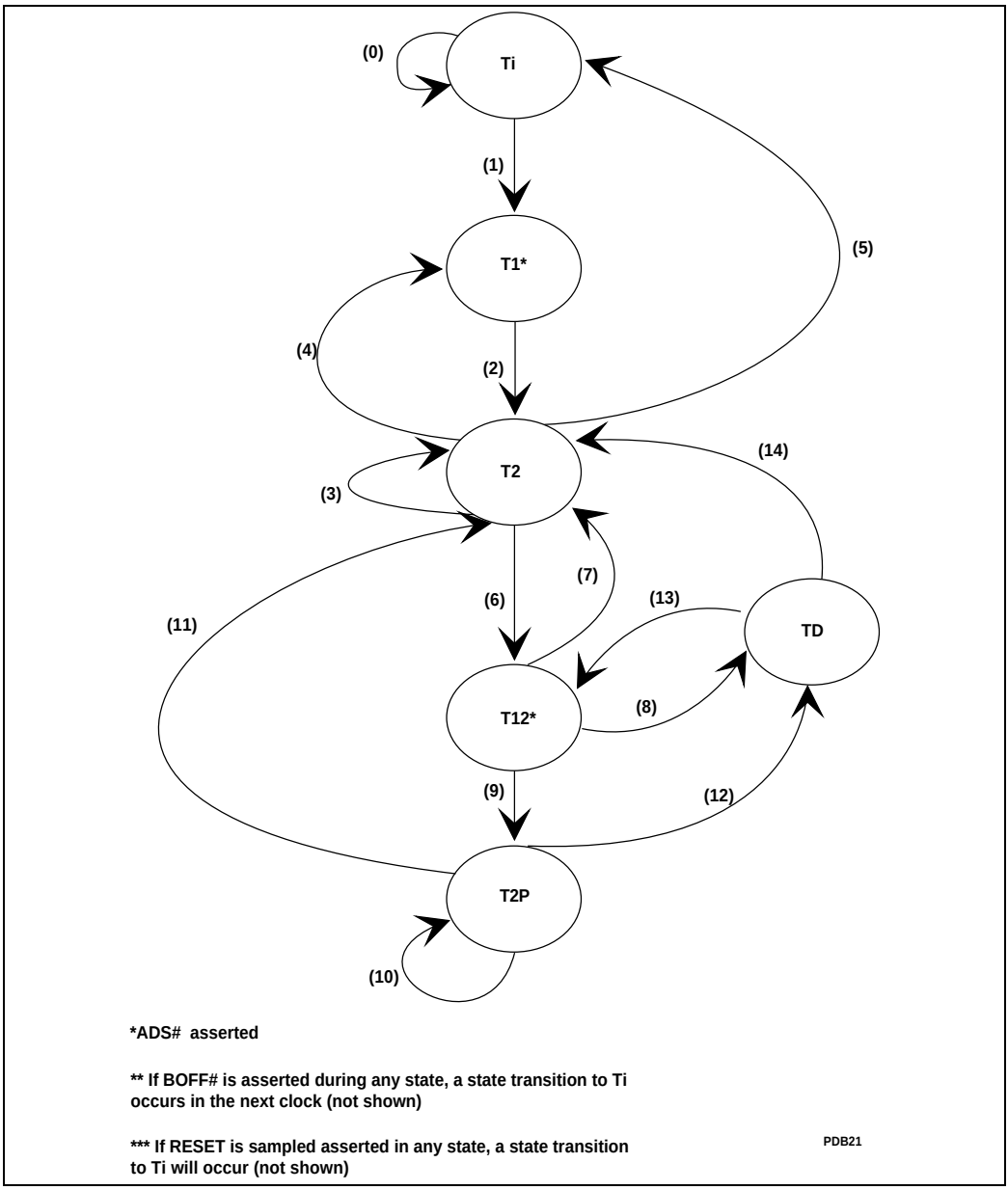


Figure 6-29. Pentium® Processor Bus Control State Machine

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.

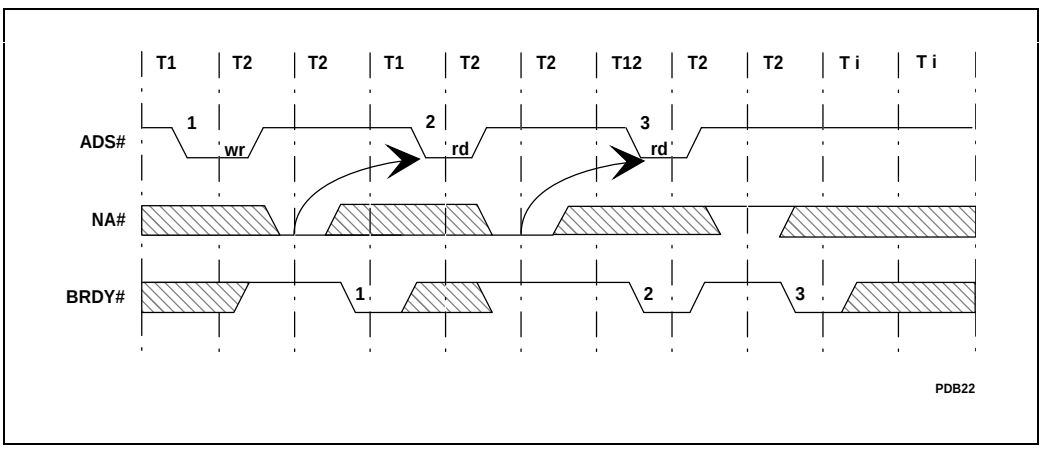


Figure 6-30. Bus Cycles Without Dead Clock

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 of the "same type" as

cycle 2. That is if cycle 2 is a read cycle, cycle 3 must also be a read cycle in order to prevent a dead clock. If cycle 2 is a write cycle, cycle 3 must also be a write cycle to prevent a dead clock.

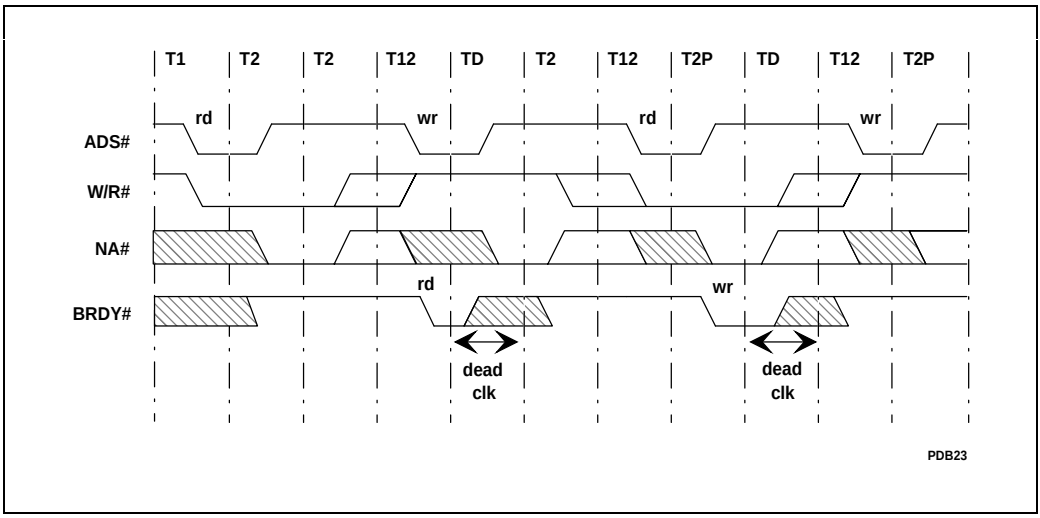


Figure 6-31. Bus Cycles with TD Dead Clock

7

Electrical Specifications

CHAPTER 7
ELECTRICAL SPECIFICATIONS

7.1. POWER AND GROUND

For clean on-chip power distribution, the Pentium Processor (510\60, 567\66) has 50 V_{CC} (power) and 49 V_{SS} (ground) inputs. Power and ground connections must be made to all external V_{CC} and V_{SS} pins of the Pentium Processor (510\60, 567\66). On the circuit board, all V_{CC} pins must be connected to a V_{CC} plane. All V_{SS} pins must be connected to a V_{SS} 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 V_{CC} and V_{SS} 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 V_{CC}. 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 AC and DC 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 V_{CC} to V_{CC} + 0.5 (V)
Supply voltage with respect to V_{SS}	-0.5V to +6.5V

7.5. DC SPECIFICATIONS

Table 7-2 lists the DC specifications associated with the Pentium processor.

Table 7-2. Pentium® Processor (510\60, 567\66) DC Specifications

V_{CC} = See Notes 10, 11; T_{CASE} = See Notes 12, 13					
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		$\pm$15	uA	$0 \le V_{IN} \le V_{CC}$, (4)
I_{LO}	Output Leakage Current		$\pm$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 4 mA load.

(2) Parameter measured at 1 mA 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.

(10) V_{CC} = 5V ± 5% at 60 MHz.

(11) V_{CC} = 4.90V to 5.40V at 66 MHz

(12) T_{CASE} = 0°C to +80°C at 60 MHz

(13) T_{CASE} = 0°C to +70°C at 66 MHz

7.6. AC SPECIFICATIONS

The 66-MHz and 60-MHz AC specifications given in Tables 7-3 and 7-4 consist of output delays, input setup requirements and input hold requirements. All AC 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 567\66 AC Specifications

V_{CC} = 4.90V to 5.40V; T_{CASE} = 0°C to 70°C; C_L = 0 pF						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
	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)

Table 7-3. 66-MHz Pentium® Processor 567\66 AC Specifications (Contd.)

V_{CC} = 4.90V to 5.40V; T_{CASE} = 0°C to 70°C; C_L = 0 pF						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
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	
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	3.8		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)

Table 7-3. 66-MHz Pentium® Processor 567\66 AC Specifications (Contd.)

V_{CC} = 4.90V to 5.40V; T_{CASE} = 0°C to 70°C; C_L = 0 pF						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
t_{37}	RESET Hold Time	1.5		nS	7.5	(11), (13)
t_{38}	RESET Pulse Width, V_{CC} & CLK Stable	15		CLKs	7.5	(11)
t_{39}	RESET Active After V_{CC} & 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)
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. 1 ns 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 ($t_{55\text{-}58}$).
11. FRCMC# should be tied to V_{CC} (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 minimum 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 magnitude 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. If this occurs, I/O timings are degraded by twice the jitter component within this frequency range. For example, if 15% of the jitter energy is within this range degrade I/O timings by 2 × 0.15 × magnitude of jitter.
21. The amount of jitter present must be accounted for as a component of CLK skew between devices.

Table 7-4. 60-MHz Pentium® Processor 510\60 AC Specifications

V_{CC} = 5V ± 5%; T_{CASE} = 0°C to 80°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 510\60 AC Specifications (Contd.)

V_{CC} = 5V ± 5%; T_{CASE} = 0°C to 80°C; C_L = 0 pF						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
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	4.3		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)
t_{38}	RESET Pulse Width, V_{CC} & CLK Stable	15		CLKs	7.5	(11)
t_{39}	RESET Active After V_{CC} & 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)

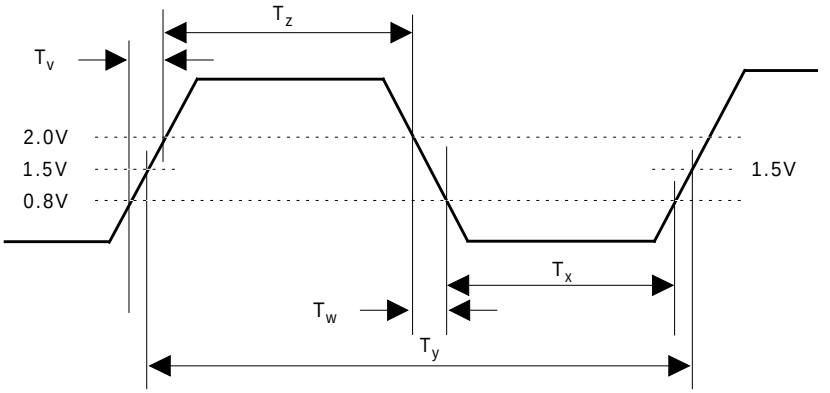
Table 7-4. 60-MHz Pentium® Processor 510\60 AC Specifications (Contd.)

V_{CC} = 5V ± 5%; T_{CASE} = 0°C to 80°C; C_L = 0 pF						
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Figure**	**Notes**
t_{42}	Pentium processor Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time, Async.	2		CLKs	7.5	(16)
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. 1 ns 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 ($t_{55\text{-}58}$).
11. FRCMC# should be tied to V_{CC} (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 minimum 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 magnitude 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. If this occurs, I/O timings are degraded by twice the jitter component within this frequency range. For example, if 15% of the jitter energy is within this range degrade I/O timings by 2 × 0.15 × magnitude of jitter.
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.



T_v = t5, t49

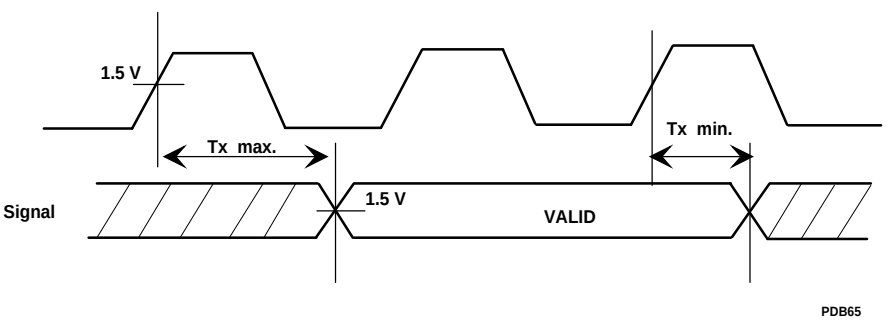
T_w = t4, t48

T_x = t3, t47

T_y = t1, t45

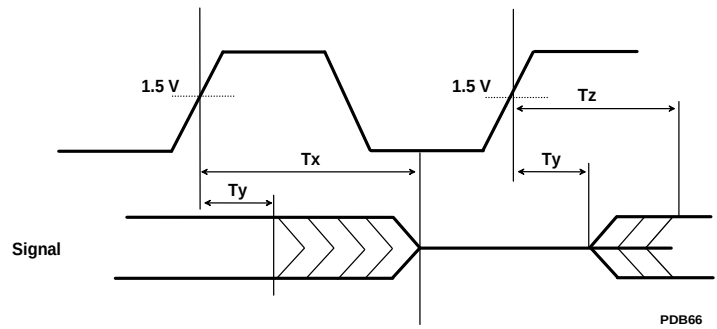
T_z = t2, t46

Figure 7-1. Clock Waveform



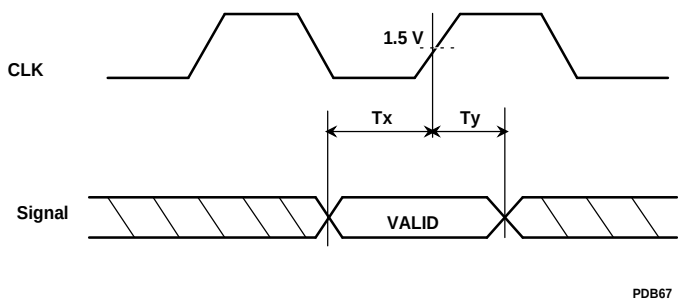
Tx = t6, t6a, t8, t9, t10, t11, t11a, t12

Figure 7-2. Valid Delay Timings



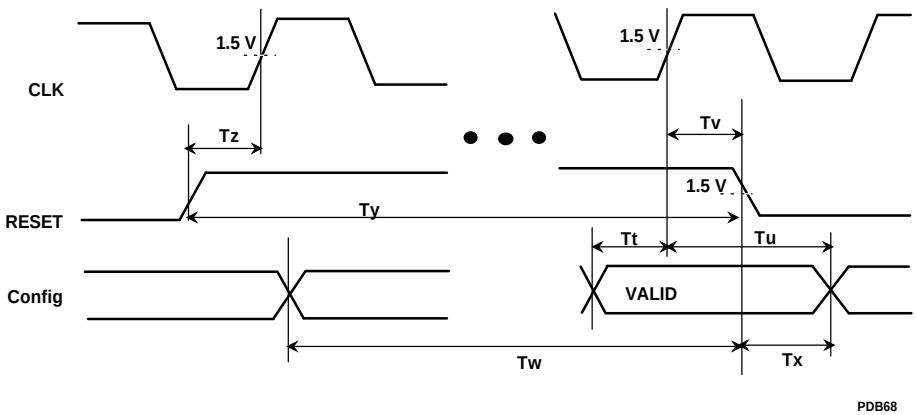
Tx = t7, t13
Ty = t6min, t12min

Figure 7-3. Float Delay Timings



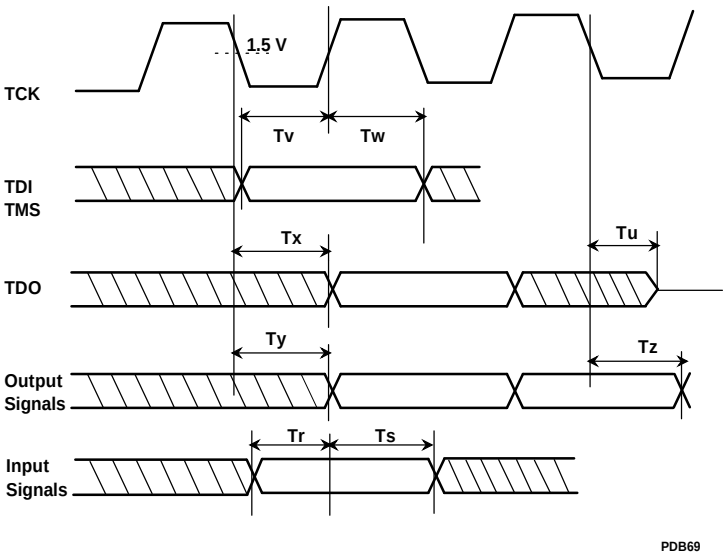
Tx = t14, t16, t18, t18a, t20, t22, t24, t26, t28, t31, t34, t34a
Ty = t15, t17, t19, t21, t23, t25, t27, t29, t32, t35

Figure 7-4. Setup and Hold Timings



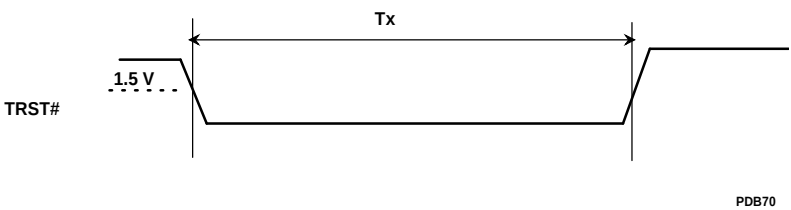
Tt = t40
Tu = t41
Tv = t37
Tw =t42
Tx = t43
Ty = t38, t39
Tz = t36

Figure 7-5. Reset and Configuration Timings



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 V_{CC} or V_{SS} 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 V_{CC} (below V_{SS}). 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)

Ringback is the absolute value of the maximum voltage at the receiving pin below V_{CC} (or above V_{SS}) relative to V_{CC} (or V_{SS}) 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 V_{CC} (below V_{SS}).

Maximum Ringback on Inputs = 0.8 Volts
(with diodes)

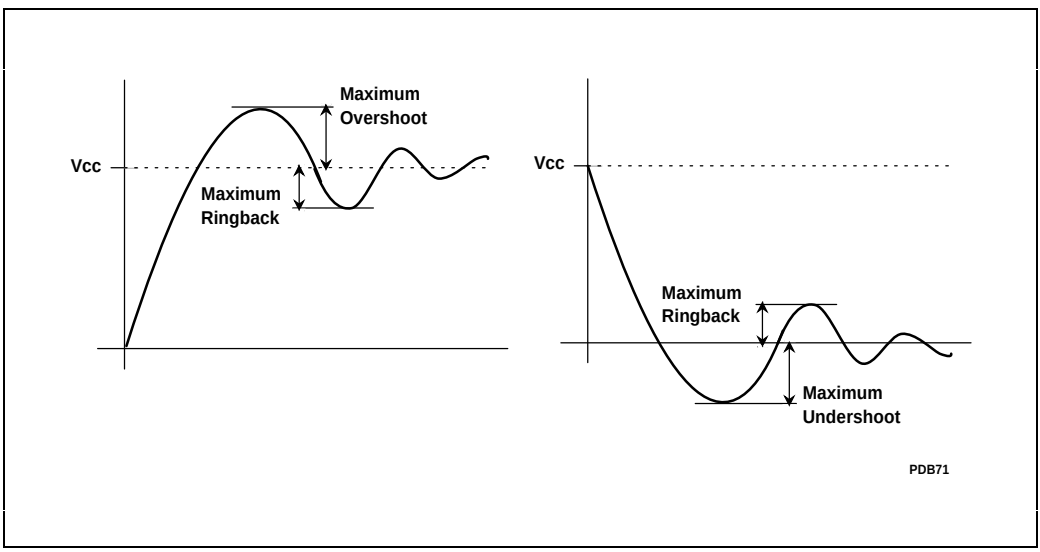


Figure 7-8. Overshoot/Undershoot and Ringback Guidelines

8

I/O Buffer Models

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 (510\60, 567\66). 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.

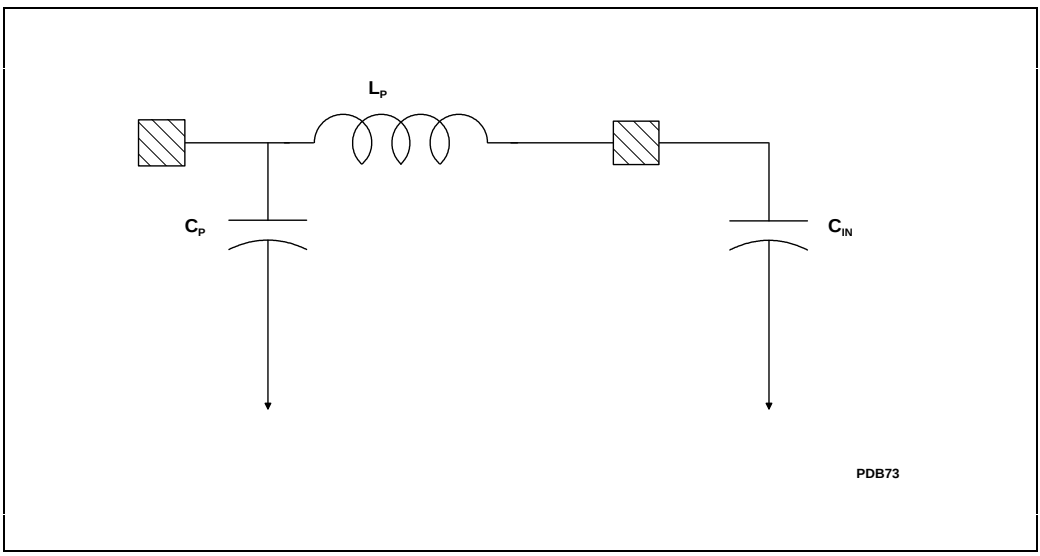


Figure 8-1. First Order Input Buffer

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.

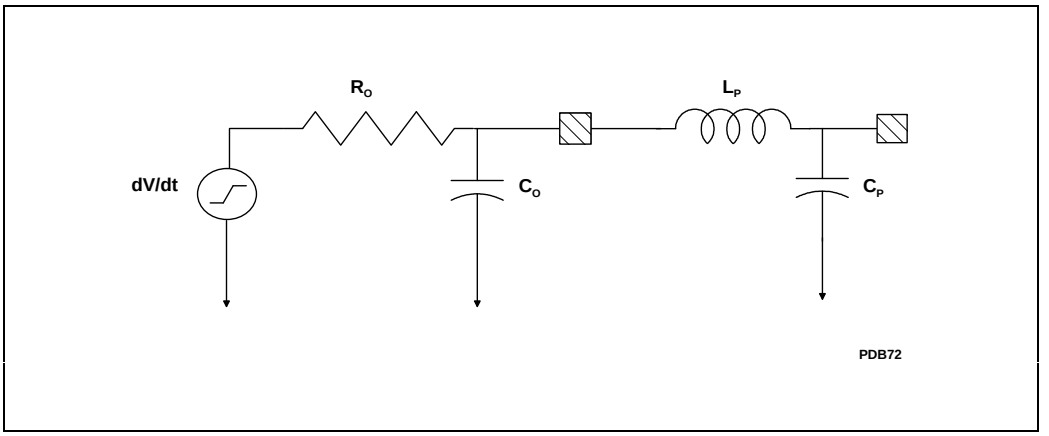


Figure 8-2. First Order Output Buffer

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 8-5 and Table 8-**Error! Bookmark not defined.** list the minimum and maximum parameters for each buffer type within the Pentium Processor (510\60, 567\66). 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.

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

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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	4.5/2.6	5.5/1.1	18	54	4.2	5.6	5.3	15.2	0.5	6.6

Table 8-5. Specification of Output External Interface Buffer Model Parameters (Contd.)

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
ED5	Rising	Pentium Processor (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 (510\60, 567\66)	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 V_{CC}, D2, and one to V_{SS}, D1. Each diode is modeled by the combination of an ideal diode in series with a resistance.

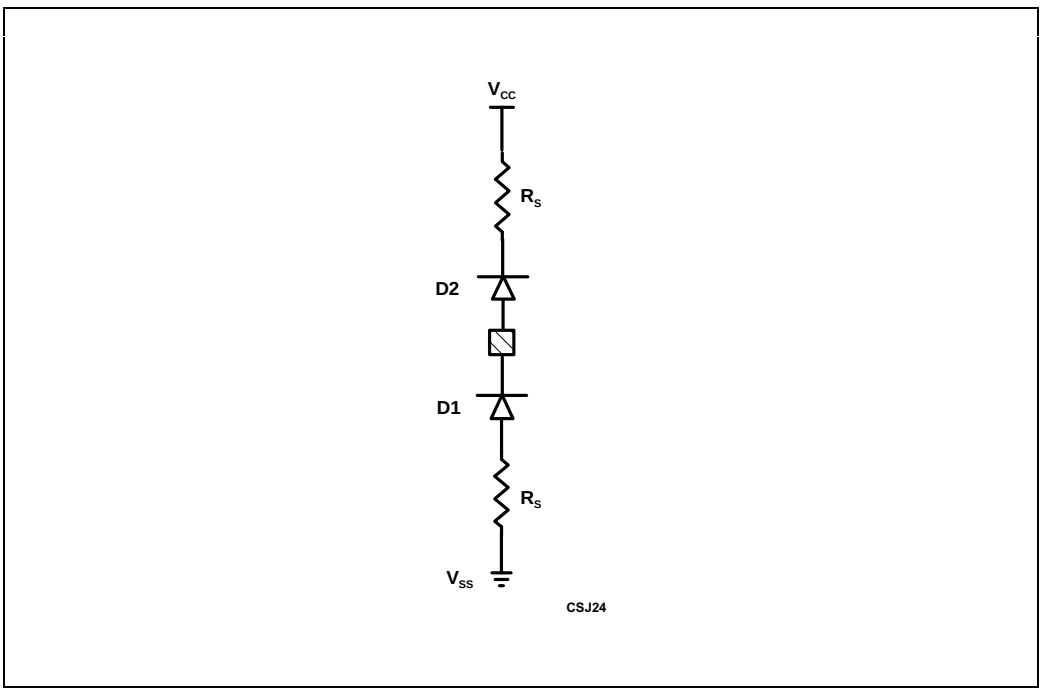


Figure 8-3. Input Diode Model

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.

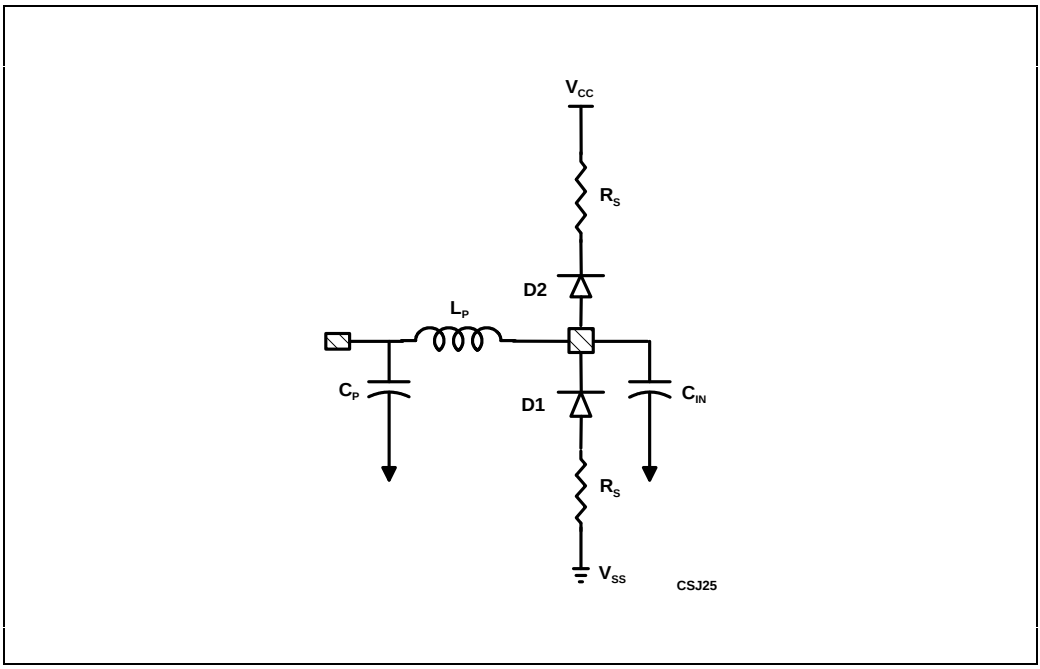


Figure 8-4. Complete Input Model Including Diode

The specific parameters associated with each diode are listed below. Table 8-**Error! Bookmark not defined.** lists the buffer types with their corresponding diode I-V curve and series resistance. Table 8-**Error! Bookmark not defined.** 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			
IV1		IV2	
Vd	Id	Vd	Id
0v	0a	0v	0a
25mv	0.0053pa	25mv	0.0037pa
50mv	0.0062pa	50mv	0.0038pa
75mv	0.0083pa	75mv	0.0039pa
100mv	0.0134pa	100mv	0.0041pa
0.125v	0.0260pa	0.125v	0.0046pa
0.15v	0.0572pa	0.15v	0.0062pa
0.15v	0.0572pa	0.15v	0.0062pa
0.175v	0.13pa	0.175v	0.0107pa
0.2v	0.33pa	0.2v	0.0237pa
0.225v	0.8pa	0.225v	0.0621pa
0.25v	1.98pa	0.25v	0.18pa
0.275v	4.91pa	0.275v	0.51pa
0.3v	12.18pa	0.3v	1.49pa
0.325v	30.22pa	0.325v	4.4pa
0.35v	74.98pa	0.35v	12.96pa
0.375v	0.19na	0.375v	38.22pa
0.4v	0.46na	0.4v	0.11na
0.425v	1.15na	0.425v	0.33na
0.45v	2.84na	0.45v	0.98na
0.475v	7.26na	0.475v	2.89na
0.5v	18.02na	0.5v	8.95na
0.525v	44.72na	0.525v	26.4na
0.55v	0.11ua	0.55v	77.85na
0.575v	0.28ua	0.575v	0.23ua
0.6v	0.68ua	0.6v	0.68ua
0.625v	1.69ua	0.625v	1.99ua
0.65v	4.18ua	0.65v	5.82ua
0.674v	10.26ua	0.674v	16.79ua

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.699v	24.81ua	0.698v	46.58ua
0.722v	58.06ua	0.719v	0.12ma
0.744v	0.13ma	0.737v	0.26ma
0.762v	0.25ma	0.751v	0.47ma
0.778v	0.44ma	0.762v	0.76ma
0.79v	0.69ma	0.772v	0.76ma
0.8v	1ma	0.779v	1.07ma
0.809v	0.99ma	0.784v	1.49ma
0.817v	1.32ma	0.788v	1.91ma
0.822v	1.73ma	0.792v	2.3ma
0.826v	2.13ma	0.795v	2.7ma
0.83v	2.52ma	0.798v	3.13ma
0.834v	2.93ma	0.801v	3.57ma
0.837v	3.35ma	0.803v	4.01ma
0.84v	3.79ma	0.806v	4.46ma
0.843v	4.22ma	0.808v	4.91ma
0.845v	4.67ma	0.809v	5.37ma
0.848v	5.11ma	0.811v	5.83ma
0.85v	5.57ma	0.813v	6.3ma
0.852v	6.02ma	0.814v	6.76ma
0.854v	6.48ma	0.816v	7.23ma
0.856v	6.94ma	0.817v	7.7ma
0.857v	7.4ma	0.819v	8.17ma
0.859v	7.87ma	0.82v	8.64ma
0.86v	8.34ma	0.821v	9.12ma
0.862v	8.81ma	0.822v	9.59ma
0.863v	9.28ma	0.823v	10.07ma
0.864v	9.75ma	0.824v	10.55ma

Table 8-7. Data for Diode I-V Curves (Contd.)

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.866v	10.22ma	0.825v	11.03ma
0.867v	10.7ma	0.826v	11.51ma
0.868v	11.17ma	0.827v	11.99ma
0.869v	11.65ma	0.828v	12.47ma
0.87v	12.13ma	0.829v	12.95ma
0.871v	12.61ma	0.83v	13.43ma
0.872v	13.09ma	0.83v	13.92ma
0.873v	13.56ma	0.831v	14.4ma
0.874v	14.05ma	0.832v	14.89ma
0.875v	14.53ma	0.833v	15.37ma
0.876v	15.01ma	0.833v	15.86ma
0.877v	15.49ma	0.834v	16.34ma
0.878v	15.97ma	0.835v	16.83ma
0.878v	16.46ma	0.835v	17.32ma
0.879v	16.94ma	0.836v	17.8ma
0.88v	17.43ma	0.836v	18.29ma
0.881v	17.91ma	0.837v	18.78ma
0.881v	18.39ma	0.838v	19.27ma
0.882v	18.88ma	0.838v	19.75ma
0.883v	19.37ma	0.839v	20.24ma
0.883v	19.85ma	0.839v	20.73ma
0.884v	20.34ma	0.84v	21.22ma
0.885v	20.83ma	0.84v	21.71ma
0.885v	21.31ma	0.841v	22.2ma
0.886v	21.8ma	0.841v	22.69ma
0.886v	22.29ma	0.842v	23.18ma
0.887v	22.78ma	0.842v	23.67ma

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.888v	23.26ma	0.843v	24.16ma
0.888v	23.75ma	0.843v	24.65ma
0.889v	24.24ma	0.844v	25.14ma
0.889v	24.73ma	0.844v	25.63ma
0.89v	25.22ma	0.845v	26.12ma
0.89v	25.71ma	0.845v	26.61ma
0.891v	26.2ma	0.845v	27.11ma
0.891v	26.69ma	0.846v	27.6ma
0.892v	27.18ma	0.846v	28.09ma
0.892v	27.67ma	0.847v	28.58ma
0.893v	28.16ma	0.847v	29.07ma
0.893v	28.65ma	0.847v	29.57ma
0.894v	29.14ma	0.848v	30.06ma
0.894v	29.63ma	0.848v	30.55ma
0.895v	30.12ma	0.848v	31.04ma
0.895v	30.61ma	0.849v	31.53ma
0.895v	31.1ma	0.849v	32.03ma
0.896v	31.59ma	0.849v	32.52ma
0.896v	32.09ma	0.85v	33.01ma
0.897v	32.58ma	0.85v	33.51ma
0.897v	33.07ma	0.85v	34ma
0.898v	33.56ma	0.851v	34.49ma
0.898v	34.05ma	0.851v	34.99ma
0.898v	34.54ma	0.851v	35.48ma
0.899v	35.04ma	0.852v	35.97ma
0.899v	35.53ma	0.852v	36.47ma
0.899v	36.02ma	0.852v	36.96ma

Table 8-8. Data for Diode I-V Curves (Contd.)

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.9v	36.51ma	0.853v	37.45ma
0.9v	37.01ma	0.853v	37.95ma
0.901v	37.5ma	0.853v	38.44ma
0.901v	37.99ma	0.854v	38.94ma
0.901v	38.49ma	0.854v	39.43ma
0.902v	38.98ma	0.854v	39.92ma
0.902v	39.47ma	0.854v	40.42ma
0.902v	39.96ma	0.855v	40.91ma
0.903v	40.46ma	0.855v	41.41ma
0.903v	40.95ma	0.855v	41.9ma
0.903v	41.44ma	0.856v	42.4ma
0.904v	41.94ma	0.856v	42.89ma
0.904v	42.43ma	0.856v	43.39ma
0.904v	42.92ma	0.856v	43.88ma
0.904v	43.42ma	0.857v	44.38ma
0.905v	43.91ma	0.857v	44.87ma
0.905v	44.41ma	0.857v	45.37ma
0.905v	44.9ma	0.857v	45.86ma
0.906v	45.39ma	0.858v	46.36ma
0.906v	45.89ma	0.858v	46.85ma
0.906v	46.38ma	0.858v	47.35ma
0.907v	46.88ma	0.858v	47.84ma
0.907v	47.37ma	0.859v	48.34ma
0.907v	47.86ma	0.859v	48.83ma
0.907v	48.36ma	0.859v	49.33ma
0.908v	48.85ma	0.859v	49.82ma
0.908v	49.35ma	0.859v	50.32ma

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.908v	49.84ma	0.86v	50.81ma
0.909v	50.34ma	0.86v	51.31ma
0.909v	50.83ma	0.86v	51.8ma
0.909v	51.33ma	0.86v	52.3ma
0.909v	51.82ma	0.861v	52.79ma
0.91v	52.32ma	0.861v	53.29ma
0.91v	52.81ma	0.861v	53.79ma
0.91v	53.3ma	0.861v	54.28ma
0.91v	53.8ma	0.861v	54.78ma
0.911v	54.29ma	0.862v	55.27ma
0.911v	54.79ma	0.862v	55.77ma
0.911v	55.28ma	0.862v	56.27ma
0.911v	55.78ma	0.862v	56.76ma
0.912v	56.27ma	0.862v	57.26ma
0.912v	56.77ma	0.863v	57.75ma
0.912v	57.27ma	0.863v	58.25ma
0.912v	57.76ma	0.863v	58.74ma
0.913v	58.26ma	0.863v	59.24ma
0.913v	58.75ma	0.863v	59.74ma
0.913v	59.25ma	0.864v	60.23ma
0.913v	59.74ma	0.864v	60.73ma
0.913v	60.24ma	0.864v	61.23ma
0.914v	60.73ma	0.864v	61.72ma
0.914v	61.23ma	0.864v	62.22ma
0.914v	61.72ma	0.864v	62.71ma
0.914v	62.22ma	0.865v	63.21ma
0.915v	62.71ma	0.865v	63.71ma

Table 8-9. Data for Diode I-V Curves (Contd.)

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.915v	63.21ma	0.865v	64.2ma
0.915v	63.71ma	0.865v	64.7ma
0.915v	64.2ma	0.865v	65.2ma
0.915v	64.7ma	0.866v	65.69ma
0.916v	65.19ma	0.866v	66.19ma
0.916v	65.69ma	0.866v	66.69ma
0.916v	66.18ma	0.866v	67.18ma
0.916v	66.68ma	0.866v	67.68ma
0.916v	67.18ma	0.866v	68.18ma
0.917v	67.67ma	0.867v	68.67ma
0.917v	68.17ma	0.867v	69.17ma
0.917v	68.66ma	0.867v	69.67ma
0.917v	69.16ma	0.867v	70.16ma
0.917v	69.66ma	0.867v	70.66ma
0.918v	70.15ma	0.867v	71.16ma
0.918v	70.65ma	0.868v	71.65ma
0.918v	71.14ma	0.868v	72.15ma
0.918v	71.64ma	0.868v	72.65ma
0.918v	72.14ma	0.868v	73.14ma

Curve Type			
IV1		IV2	
Vd	Id	Vd	Id
0.919v	72.63ma	0.868v	73.64ma
0.919v	73.13ma	0.868v	74.14ma
0.919v	73.63ma	0.868v	74.63ma
0.919v	74.12ma	0.869v	75.13ma
0.919v	74.62ma	0.869v	75.63ma
0.919v	75.11ma	0.869v	76.12ma
0.92v	75.61ma	0.869v	76.62ma
0.92v	76.11ma	0.869v	77.12ma
0.92v	76.6ma	0.869v	77.62ma
0.92v	77.1ma	0.87v	78.11ma
0.92v	77.6ma	0.87v	78.61ma
0.921v	78.09ma	0.87v	79.11ma
0.921v	78.59ma	0.87v	79.6ma
0.921v	79.09ma	0.87v	80.1ma
0.921v	79.58ma	0.87v	80.6ma
0.921v	80.08ma	0.87v	81.09ma
0.921v	80.58ma	0.871v	81.59ma
0.922v	81.07ma	0.871v	82.09ma

Table 8-10. Data for Diode I-V Curves (Contd.)

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0v	0a	0v	0a
25mv	0.021pa	25mv	0.013pa
50mv	0.024pa	50mv	0.013pa
75mv	0.032pa	75mv	0.014pa
100mv	0.051pa	100mv	0.014pa
0.125v	0.098pa	0.125v	0.016pa
0.15v	0.21pa	0.15v	0.021pa
0.175v	0.5pa	0.175v	0.037pa
0.2v	1.21pa	0.2v	0.083pa
0.225v	2.98pa	0.225v	0.22pa
0.25v	7.35pa	0.25v	0.62pa
0.275v	18.22pa	0.275v	1.79pa
0.3v	45.17pa	0.3v	5.26pa
0.325v	0.11na	0.325v	15.47pa
0.35v	0.28na	0.35v	45.6pa
0.375v	0.69na	0.375v	0.13na
0.4v	1.71na	0.4v	0.4na
0.425v	4.25na	0.425v	1.17na
0.45v	10.85na	0.45v	3.45na
0.475v	26.93na	0.475v	10.68na
0.5v	66.83na	0.5v	31.49na
0.525v	0.17ua	0.525v	92.86na
0.55v	0.41ua	0.55v	0.27ua
0.575v	1.02ua	0.575v	0.81ua
0.6v	2.52ua	0.6v	2.37ua
0.625v	6.22ua	0.625v	6.93ua
0.649v	15.2ua	0.649v	19.9ua
0.673v	36.33ua	0.672v	54.64ua

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.696v	83ua	0.693v	0.14ma
0.716v	0.17ma	0.711v	0.29ma
0.734v	0.33ma	0.724v	0.52ma
0.748v	0.55ma	0.735v	0.81ma
0.759v	0.82ma	0.744v	0.81ma
0.768v	1.14ma	0.751v	1.13ma
0.776v	1.14ma	0.756v	1.55ma
0.783v	1.48ma	0.76v	1.97ma
0.788v	1.9ma	0.763v	2.36ma
0.792v	2.31ma	0.767v	2.77ma
0.796v	2.7ma	0.77v	3.2ma
0.799v	3.12ma	0.772v	3.64ma
0.803v	3.54ma	0.775v	4.09ma
0.805v	3.98ma	0.777v	4.54ma
0.808v	4.42ma	0.779v	4.99ma
0.81v	4.86ma	0.781v	5.45ma
0.813v	5.31ma	0.782v	5.91ma
0.815v	5.77ma	0.784v	6.37ma
0.817v	6.22ma	0.786v	6.84ma
0.819v	6.68ma	0.787v	7.31ma
0.82v	7.15ma	0.788v	7.78ma
0.822v	7.61ma	0.79v	8.25ma
0.823v	8.08ma	0.791v	8.72ma
0.825v	8.55ma	0.792v	9.2ma
0.826v	9.01ma	0.793v	9.67ma
0.828v	9.49ma	0.794v	10.15ma
0.829v	9.96ma	0.795v	10.63ma
0.83v	10.43ma	0.796v	11.11ma

Table 8-11. Data for Diode I-V Curves (Contd.)

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.831v	10.91ma	0.797v	11.59ma
0.832v	11.38ma	0.798v	12.07ma
0.834v	11.86ma	0.799v	12.55ma
0.835v	12.34ma	0.8v	13.03ma
0.836v	12.82ma	0.801v	13.51ma
0.837v	13.3ma	0.801v	14ma
0.838v	13.78ma	0.802v	14.48ma
0.838v	14.26ma	0.803v	14.97ma
0.839v	14.74ma	0.804v	15.45ma
0.84v	15.22ma	0.804v	15.94ma
0.841v	15.7ma	0.805v	16.42ma
0.842v	16.19ma	0.806v	16.91ma
0.843v	16.67ma	0.806v	17.39ma
0.843v	17.15ma	0.807v	17.88ma
0.844v	17.64ma	0.807v	18.37ma
0.845v	18.12ma	0.808v	18.86ma
0.846v	18.61ma	0.809v	19.34ma
0.846v	19.1ma	0.809v	19.83ma
0.847v	19.58ma	0.81v	20.32ma
0.848v	20.07ma	0.81v	20.81ma
0.848v	20.55ma	0.811v	21.3ma
0.849v	21.04ma	0.811v	21.79ma
0.849v	21.53ma	0.812v	22.28ma
0.85v	22.02ma	0.812v	22.77ma
0.851v	22.5ma	0.813v	23.26ma
0.851v	22.99ma	0.813v	23.75ma
0.852v	23.48ma	0.814v	24.24ma
0.852v	23.97ma	0.814v	24.73ma

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.853v	24.46ma	0.815v	25.22ma
0.853v	24.95ma	0.815v	25.71ma
0.854v	25.44ma	0.815v	26.2ma
0.854v	25.93ma	0.816v	26.69ma
0.855v	26.41ma	0.816v	27.19ma
0.855v	26.9ma	0.817v	27.68ma
0.856v	27.39ma	0.817v	28.17ma
0.856v	27.88ma	0.818v	28.66ma
0.857v	28.38ma	0.818v	29.15ma
0.857v	28.87ma	0.818v	29.65ma
0.858v	29.36ma	0.819v	30.14ma
0.858v	29.85ma	0.819v	30.63ma
0.859v	30.34ma	0.819v	31.12ma
0.859v	30.83ma	0.82v	31.62ma
0.86v	31.32ma	0.82v	32.11ma
0.86v	31.81ma	0.82v	32.6ma
0.86v	32.3ma	0.821v	33.09ma
0.861v	32.79ma	0.821v	33.59ma
0.861v	33.29ma	0.821v	34.08ma
0.862v	33.78ma	0.822v	34.57ma
0.862v	34.27ma	0.822v	35.07ma
0.862v	34.76ma	0.822v	35.56ma
0.863v	35.25ma	0.823v	36.05ma
0.863v	35.75ma	0.823v	36.55ma
0.864v	36.24ma	0.823v	37.04ma
0.864v	36.73ma	0.824v	37.54ma
0.864v	37.22ma	0.824v	38.03ma
0.865v	37.72ma	0.824v	38.52ma

Table 8-12. Data for Diode I-V Curves (Contd.)

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.865v	38.21ma	0.825v	39.02ma
0.865v	38.7ma	0.825v	39.51ma
0.866v	39.2ma	0.825v	40.01ma
0.866v	39.69ma	0.825v	40.5ma
0.866v	40.18ma	0.826v	40.99ma
0.867v	40.68ma	0.826v	41.49ma
0.867v	41.17ma	0.826v	41.98ma
0.867v	41.66ma	0.826v	42.48ma
0.868v	42.16ma	0.827v	42.97ma
0.868v	42.65ma	0.827v	43.47ma
0.868v	43.14ma	0.827v	43.96ma
0.869v	43.64ma	0.828v	44.46ma
0.869v	44.13ma	0.828v	44.95ma
0.869v	44.62ma	0.828v	45.45ma
0.869v	45.12ma	0.828v	45.94ma
0.87v	45.61ma	0.829v	46.44ma
0.87v	46.11ma	0.829v	46.93ma
0.87v	46.6ma	0.829v	47.43ma
0.871v	47.09ma	0.829v	47.92ma
0.871v	47.59ma	0.829v	48.42ma
0.871v	48.08ma	0.83v	48.91ma
0.872v	48.58ma	0.83v	49.41ma
0.872v	49.07ma	0.83v	49.9ma
0.872v	49.57ma	0.83v	50.4ma
0.872v	50.06ma	0.831v	50.89ma
0.873v	50.55ma	0.831v	51.39ma
0.873v	51.05ma	0.831v	51.88ma

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.873v	51.54ma	0.831v	52.38ma
0.873v	52.04ma	0.832v	52.88ma
0.874v	52.53ma	0.832v	53.37ma
0.874v	53.03ma	0.832v	53.87ma
0.874v	53.52ma	0.832v	54.36ma
0.874v	54.02ma	0.832v	54.86ma
0.875v	54.51ma	0.833v	55.35ma
0.875v	55.01ma	0.833v	55.85ma
0.875v	55.5ma	0.833v	56.35ma
0.875v	56ma	0.833v	56.84ma
0.876v	56.49ma	0.833v	57.34ma
0.876v	56.99ma	0.834v	57.83ma
0.876v	57.48ma	0.834v	58.33ma
0.876v	57.98ma	0.834v	58.83ma
0.877v	58.47ma	0.834v	59.32ma
0.877v	58.97ma	0.834v	59.82ma
0.877v	59.47ma	0.835v	60.31ma
0.877v	59.96ma	0.835v	60.81ma
0.877v	60.46ma	0.835v	61.31ma
0.878v	60.95ma	0.835v	61.8ma
0.878v	61.45ma	0.835v	62.3ma
0.878v	61.94ma	0.835v	62.8ma
0.878v	62.44ma	0.836v	63.29ma
0.879v	62.93ma	0.836v	63.79ma
0.879v	63.43ma	0.836v	64.28ma
0.879v	63.93ma	0.836v	64.78ma
0.879v	64.42ma	0.836v	65.28ma
0.879v	64.92ma	0.837v	65.77ma

Table 8-13. Data for Diode I-V Curves (Contd.)

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.88v	65.41ma	0.837v	66.27ma
0.88v	65.91ma	0.837v	66.77ma
0.88v	66.4ma	0.837v	67.26ma
0.88v	66.9ma	0.837v	67.76ma
0.88v	67.4ma	0.837v	68.26ma
0.881v	67.89ma	0.838v	68.75ma
0.881v	68.39ma	0.838v	69.25ma
0.881v	68.88ma	0.838v	69.75ma
0.881v	69.38ma	0.838v	70.24ma
0.881v	69.88ma	0.838v	70.74ma
0.882v	70.37ma	0.838v	71.24ma
0.882v	70.87ma	0.839v	71.73ma
0.882v	71.36ma	0.839v	72.23ma
0.882v	71.86ma	0.839v	72.73ma
0.882v	72.36ma	0.839v	73.22ma
0.883v	72.85ma	0.839v	73.72ma
0.883v	73.35ma	0.839v	74.22ma

Curve Type			
IV3		IV4	
Vd	Id	Vd	Id
0.883v	73.85ma	0.839v	74.71ma
0.883v	74.34ma	0.84v	75.21ma
0.883v	74.84ma	0.84v	75.71ma
0.883v	75.33ma	0.84v	76.21ma
0.884v	75.83ma	0.84v	76.7ma
0.884v	76.33ma	0.84v	77.2ma
0.884v	76.82ma	0.84v	77.7ma
0.884v	77.32ma	0.84v	78.19ma
0.884v	77.82ma	0.841v	78.69ma
0.885v	78.31ma	0.841v	79.19ma
0.885v	78.81ma	0.841v	79.68ma
0.885v	79.31ma	0.841v	80.18ma
0.885v	79.8ma	0.841v	80.68ma
0.885v	80.3ma	0.841v	81.18ma
0.885v	80.8ma	0.841v	81.67ma
0.886v	81.29ma	0.842v	82.17ma
0.886v	81.79ma	0.842v	82.67ma

Table 8-14. Data for Diode I-V Curves (Contd.)

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0v	0a	0v	0a
25mv	0.0009pa	25mv	0.0008pa
50mv	0.0012pa	50mv	0.0008pa
75mv	0.0022pa	75mv	0.0008pa
100mv	0.0044pa	100mv	0.0009pa
0.125v	0.0100pa	0.125v	0.001pa
0.15v	0.0239pa	0.15v	0.0013pa
0.175v	0.0584pa	0.175v	0.0022pa
0.2v	0.14pa	0.2v	0.0047pa
0.225v	0.36pa	0.225v	0.0124pa
0.25v	0.88pa	0.25v	0.0348pa
0.275v	2.19pa	0.275v	0.1pa
0.3v	5.44pa	0.3v	0.3pa
0.325v	13.49pa	0.325v	0.87pa
0.35v	33.48pa	0.35v	2.57pa
0.375v	83.09pa	0.375v	7.59pa
0.4v	0.21na	0.4v	22.37pa
0.425v	0.51na	0.425v	65.99pa
0.45v	1.27na	0.45v	0.19na
0.475v	3.15na	0.475v	0.57na
0.5v	8.05na	0.5v	1.69na
0.525v	19.97na	0.525v	5.24na
0.55v	49.56na	0.55v	15.46na
0.575v	0.12ua	0.575v	45.58na
0.6v	0.31ua	0.6v	0.13ua
0.625v	0.76ua	0.625v	0.4ua
0.65v	1.87ua	0.65v	1.17ua

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.675v	4.63ua	0.675v	3.43ua
0.699v	11.35ua	0.7v	9.97ua
0.724v	27.37ua	0.724v	28.32ua
0.747v	63.69ua	0.746v	75.71ua
0.768v	0.14ma	0.766v	0.18ma
0.787v	0.27ma	0.782v	0.36ma
0.802v	0.47ma	0.795v	0.61ma
0.814v	0.73ma	0.804v	0.92ma
0.823v	1.03ma	0.812v	0.92ma
0.832v	1.03ma	0.819v	1.25ma
0.839v	1.36ma	0.824v	1.68ma
0.845v	1.77ma	0.828v	2.1ma
0.849v	2.18ma	0.831v	2.5ma
0.853v	2.57ma	0.834v	2.92ma
0.857v	2.98ma	0.837v	3.35ma
0.86v	3.4ma	0.84v	3.79ma
0.863v	3.84ma	0.842v	4.24ma
0.865v	4.27ma	0.844v	4.69ma
0.868v	4.72ma	0.846v	5.14ma
0.87v	5.17ma	0.848v	5.6ma
0.872v	5.62ma	0.849v	6.07ma
0.874v	6.07ma	0.851v	6.53ma
0.876v	6.53ma	0.853v	7ma
0.878v	6.99ma	0.854v	7.47ma
0.88v	7.46ma	0.855v	7.94ma
0.881v	7.92ma	0.857v	8.41ma
0.883v	8.39ma	0.858v	8.88ma

Table 8-15. Data for Diode I-V Curves (Contd.)

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.884v	8.86ma	0.859v	9.36ma
0.885v	9.33ma	0.86v	9.83ma
0.887v	9.8ma	0.861v	10.31ma
0.888v	10.28ma	0.862v	10.79ma
0.889v	10.75ma	0.863v	11.27ma
0.89v	11.23ma	0.864v	11.75ma
0.891v	11.7ma	0.865v	12.23ma
0.892v	12.18ma	0.866v	12.71ma
0.894v	12.66ma	0.867v	13.2ma
0.894v	13.14ma	0.867v	13.68ma
0.895v	13.62ma	0.868v	14.16ma
0.896v	14.1ma	0.869v	14.65ma
0.897v	14.58ma	0.87v	15.13ma
0.898v	15.06ma	0.87v	15.62ma
0.899v	15.55ma	0.871v	16.1ma
0.9v	16.03ma	0.872v	16.59ma
0.901v	16.51ma	0.872v	17.07ma
0.901v	17ma	0.873v	17.56ma
0.902v	17.48ma	0.874v	18.05ma
0.903v	17.96ma	0.874v	18.54ma
0.904v	18.45ma	0.875v	19.02ma
0.904v	18.94ma	0.875v	19.51ma
0.905v	19.42ma	0.876v	20ma
0.906v	19.91ma	0.876v	20.49ma
0.906v	20.39ma	0.877v	20.98ma
0.907v	20.88ma	0.877v	21.47ma
0.907v	21.37ma	0.878v	21.96ma

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.908v	21.86ma	0.878v	22.45ma
0.909v	22.34ma	0.879v	22.94ma
0.909v	22.83ma	0.879v	23.43ma
0.91v	23.32ma	0.88v	23.92ma
0.91v	23.81ma	0.88v	24.41ma
0.911v	24.3ma	0.881v	24.9ma
0.911v	24.79ma	0.881v	25.39ma
0.912v	25.27ma	0.882v	25.88ma
0.913v	25.76ma	0.882v	26.37ma
0.913v	26.25ma	0.883v	26.86ma
0.914v	26.74ma	0.883v	27.35ma
0.914v	27.23ma	0.883v	27.85ma
0.915v	27.72ma	0.884v	28.34ma
0.915v	28.21ma	0.884v	28.83ma
0.915v	28.7ma	0.885v	29.32ma
0.916v	29.19ma	0.885v	29.81ma
0.916v	29.69ma	0.885v	30.31ma
0.917v	30.18ma	0.886v	30.8ma
0.917v	30.67ma	0.886v	31.29ma
0.918v	31.16ma	0.886v	31.78ma
0.918v	31.65ma	0.887v	32.28ma
0.919v	32.14ma	0.887v	32.77ma
0.919v	32.63ma	0.887v	33.26ma
0.919v	33.12ma	0.888v	33.76ma
0.92v	33.62ma	0.888v	34.25ma
0.92v	34.11ma	0.888v	34.74ma
0.921v	34.6ma	0.889v	35.24ma

Table 8-16. Data for Diode I-V Curves (Contd.)

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.921v	35.09ma	0.889v	35.73ma
0.921v	35.58ma	0.889v	36.22ma
0.922v	36.08ma	0.89v	36.72ma
0.922v	36.57ma	0.89v	37.21ma
0.922v	37.06ma	0.89v	37.7ma
0.923v	37.56ma	0.891v	38.2ma
0.923v	38.05ma	0.891v	38.69ma
0.923v	38.54ma	0.891v	39.19ma
0.924v	39.03ma	0.891v	39.68ma
0.924v	39.53ma	0.892v	40.17ma
0.924v	40.02ma	0.892v	40.67ma
0.925v	40.51ma	0.892v	41.16ma
0.925v	41.01ma	0.893v	41.66ma
0.925v	41.5ma	0.893v	42.15ma
0.926v	41.99ma	0.893v	42.65ma
0.926v	42.49ma	0.893v	43.14ma
0.926v	42.98ma	0.894v	43.64ma
0.927v	43.47ma	0.894v	44.13ma
0.927v	43.97ma	0.894v	44.63ma
0.927v	44.46ma	0.894v	45.12ma
0.928v	44.96ma	0.895v	45.62ma
0.928v	45.45ma	0.895v	46.11ma
0.928v	45.94ma	0.895v	46.61ma
0.929v	46.44ma	0.895v	47.1ma
0.929v	46.93ma	0.896v	47.6ma
0.929v	47.43ma	0.896v	48.09ma
0.929v	47.92ma	0.896v	48.59ma

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.93v	48.41ma	0.896v	49.08ma
0.93v	48.91ma	0.896v	49.58ma
0.93v	49.4ma	0.897v	50.07ma
0.93v	49.9ma	0.897v	50.57ma
0.931v	50.39ma	0.897v	51.06ma
0.931v	50.89ma	0.897v	51.56ma
0.931v	51.38ma	0.898v	52.05ma
0.932v	51.88ma	0.898v	52.55ma
0.932v	52.37ma	0.898v	53.04ma
0.932v	52.87ma	0.898v	53.54ma
0.932v	53.36ma	0.898v	54.04ma
0.933v	53.86ma	0.899v	54.53ma
0.933v	54.35ma	0.899v	55.03ma
0.933v	54.85ma	0.899v	55.52ma
0.933v	55.34ma	0.899v	56.02ma
0.934v	55.84ma	0.899v	56.52ma
0.934v	56.33ma	0.9v	57.01ma
0.934v	56.83ma	0.9v	57.51ma
0.934v	57.32ma	0.9v	58ma
0.934v	57.82ma	0.9v	58.5ma
0.935v	58.31ma	0.9v	59ma
0.935v	58.81ma	0.901v	59.49ma
0.935v	59.3ma	0.901v	59.99ma
0.935v	59.8ma	0.901v	60.48ma
0.936v	60.29ma	0.901v	60.98ma
0.936v	60.79ma	0.901v	61.48ma
0.936v	61.28ma	0.902v	61.97ma

Table 8-17. Data for Diode I-V Curves (Contd.)

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.936v	61.78ma	0.902v	62.47ma
0.937v	62.28ma	0.902v	62.97ma
0.937v	62.77ma	0.902v	63.46ma
0.937v	63.27ma	0.902v	63.96ma
0.937v	63.76ma	0.902v	64.45ma
0.937v	64.26ma	0.903v	64.95ma
0.938v	64.75ma	0.903v	65.45ma
0.938v	65.25ma	0.903v	65.94ma
0.938v	65.75ma	0.903v	66.44ma
0.938v	66.24ma	0.903v	66.94ma
0.938v	66.74ma	0.904v	67.43ma
0.939v	67.23ma	0.904v	67.93ma
0.939v	67.73ma	0.904v	68.43ma
0.939v	68.22ma	0.904v	68.92ma
0.939v	68.72ma	0.904v	69.42ma
0.939v	69.22ma	0.904v	69.92ma
0.94v	69.71ma	0.905v	70.41ma
0.94v	70.21ma	0.905v	70.91ma
0.94v	70.7ma	0.905v	71.41ma
0.94v	71.2ma	0.905v	71.9ma

Curve Type			
IV5		IV6	
Vd	Id	Vd	Id
0.94v	71.7ma	0.905v	72.4ma
0.941v	72.19ma	0.905v	72.9ma
0.941v	72.69ma	0.905v	73.39ma
0.941v	73.19ma	0.906v	73.89ma
0.941v	73.68ma	0.906v	74.39ma
0.941v	74.18ma	0.906v	74.88ma
0.941v	74.67ma	0.906v	75.38ma
0.942v	75.17ma	0.906v	75.88ma
0.942v	75.67ma	0.906v	76.38ma
0.942v	76.16ma	0.907v	76.87ma
0.942v	76.66ma	0.907v	77.37ma
0.942v	77.16ma	0.907v	77.87ma
0.943v	77.65ma	0.907v	78.36ma
0.943v	78.15ma	0.907v	78.86ma
0.943v	78.65ma	0.907v	79.36ma
0.943v	79.14ma	0.907v	79.85ma
0.943v	79.64ma	0.908v	80.35ma
0.943v	80.14ma	0.908v	80.85ma
0.944v	80.63ma	0.908v	81.35ma

For information on IBIS models, please contact Intel Corporation.

9

Mechanical Specifications

CHAPTER 9
MECHANICAL SPECIFICATIONS

The Pentium Processor (510\60, 567\66) 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" 5.49 cm X 5.49 cm	16

NOTE: See DC Specifications for more detailed power specifications.

Figure 9-1 and Figure 9-2 shows the package dimensions for the Pentium Processor (510\60, 567\66). The mechanical specifications are provided in Table 9-2.

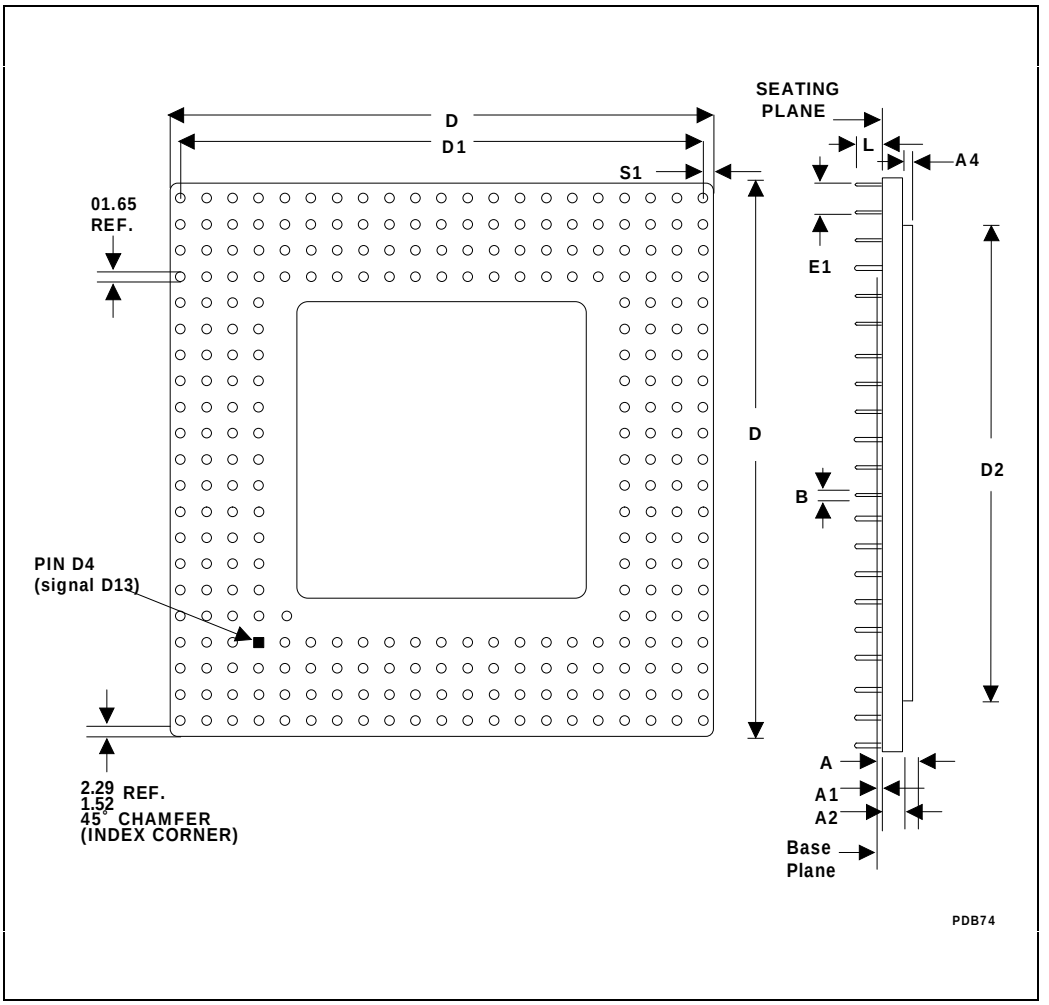


Figure 9-1. Pentium® Processor (510\60, 567\66) Package Dimensions

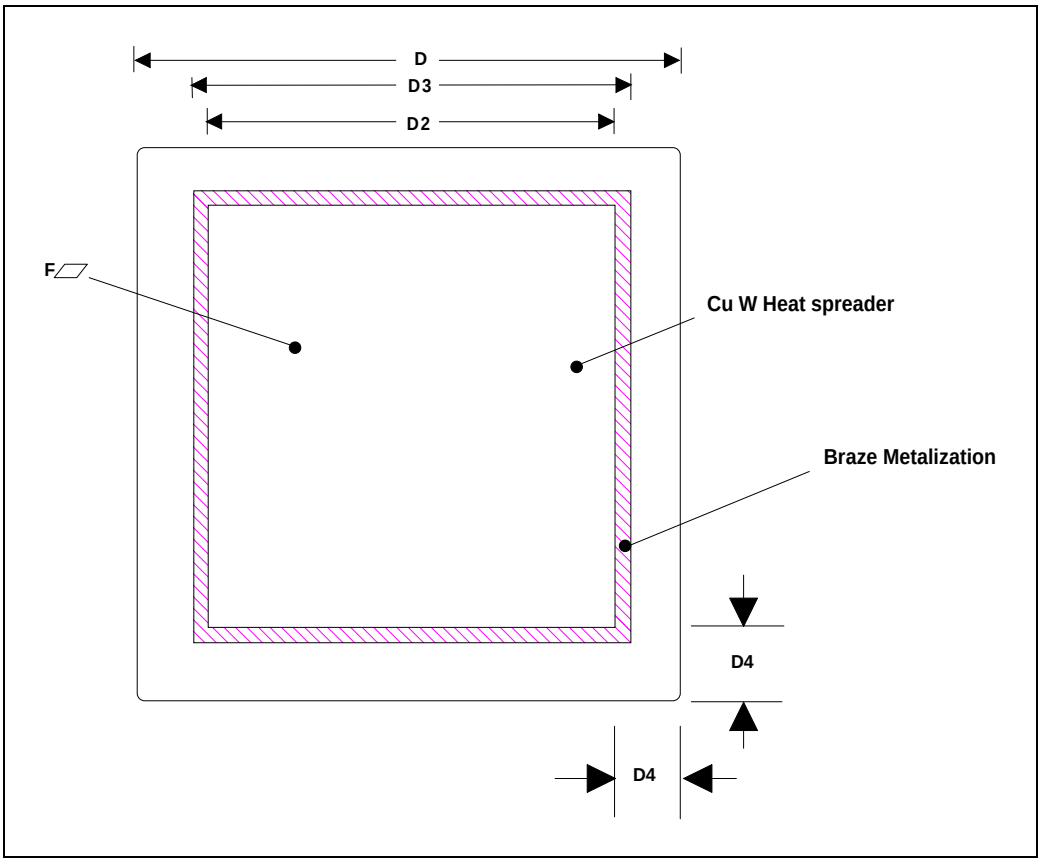


Figure 9-2. Pentium® Processor (510\60, 567\66) Package Dimensions (Top View)

Table 9-2. Pentium® Processor (510\60, 567\66) Mechanical Specifications

Family: Ceramic Pin Grid Array Package						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	3.91	4.70	Solid Lid	0.154	0.185	Solid Lid
A1	0.38	0.43	Solid Lid	0.015	0.017	Solid Lid
A2	2.62	2.97		0.103	0.117	
A4	0.97	1.22		0.038	0.048	
B	0.43	0.51		0.017	0.020	
D	54.66	55.07		2.152	2.168	
D1	50.67	50.93		1.995	2.005	
D2	37.85	38.35	Spreader Size	1.490	1.510	Spreader Size
D3	40.335	40.945	Braze	1.588	1.612	Braze
D4	8.382			0.330		
E1	2.29	2.79		0.090	0.110	
F	0.127		Flatness of spreader measured diagonally		0.005	Flatness of spreader measured diagonally
L	3.05	3.30		0.120	0.130	
N	273		Total Pins	273		Total Pins
S1	1.651	2.16		0.065	0.085	

The weight of the heat spreader package increases to approximately twice the weight of the standard PGA package (70.7 grams vs. 33.2 grams).

10

Thermal Specifications

CHAPTER 10
THERMAL SPECIFICATIONS

10.1. THERMAL MEASUREMENTS

The Pentium Processor (510\60, 567\66) is specified for proper operation when T_C (case temperature) is within the specified range (T_C = 0°C to + 80°C at 60 MHz; T_C = 0°C to + 70°C at 66 MHz). 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° angle as shown in Figure 10-1. When a heat sink is attached, a hole (no larger than 0.15 × 0.15) should be drilled through the heat sink to allow probing the center of the package as shown in Figure 10-1.
- If the case temperature is measured with a heat sink attached to the package, drill a hole through the heat sink to route the thermocouple wire out.

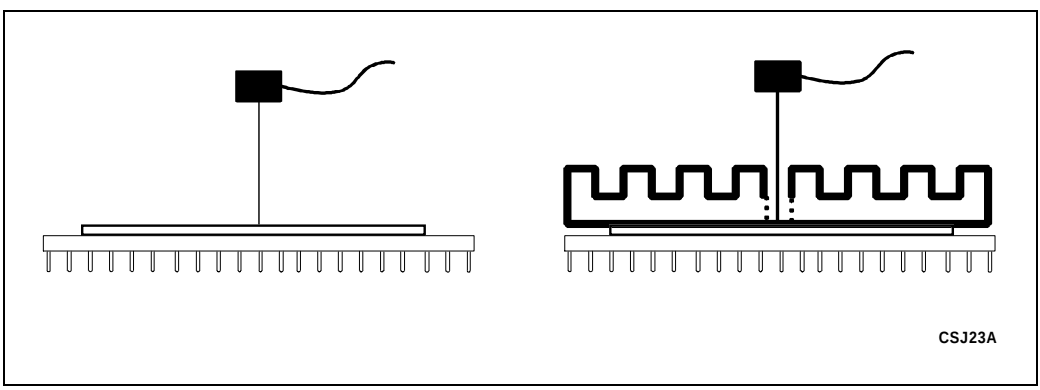


Figure 10-1. Technique for Measuring Tcase*

* Though the figure shows the package with a heat spreader, the same technique applies to measuring T_C of the package without a heat spreader.

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 (510\60, 567\66).

Table 10-1. Junction-to-Case and Case-to-Ambient Thermal Resistances for the Pentium® Processor (510\60, 567\66) (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	5.4	3.5	2.6	2.1	1.8
With 0.35" Heat Sink	0.6	7.4	4.5	3.0	2.2	1.8	1.6
With 0.65" Heat Sink	0.6	5.9	3.0	1.9	1.5	1.2	1.1
Without Heat Sink	1.2	10.5	7.9	5.5	3.8	2.8	2.4

NOTES:

1. Heat Sink: 2.1 sq. in. base, 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. Heat sinks are attached to the package with a 2 to 4 mil thick layer of typical thermal grease. The thermal conductivity of this grease is about 1.2 w/m °C.
2. Θca values shown in Table 10-1 are typical values. The actual Θca values depend on the air flow in the system (which is typically unsteady, non uniform and turbulent) and thermal interactions between Pentium® processor and surrounding components through PCB and the ambient.

10.2. PACKAGE THERMAL PERFORMANCE WITH THE HEAT SPREADER

The following is a description of how the heat spreader improves the package thermal performance:

Since the Pentium processor requires an external heat sink in order to maintain the junction and case temperatures below the acceptable levels, the main contributors to the total junction

to ambient thermal resistance are junction to case (Θ_{JC}), case to heat sink (Θ_{CS}), and heat sink to ambient (Θ_{SA}) thermal resistances.

Θ_{JC} is mainly a function of internal construction of the package and packaging material, thermal properties such as the die size and die attach, and ceramic thermal conductivity. Θ_{CS} is a function of the thickness and thermal properties of the interface material between the package and heat sink, package and heat sink flatness, and surface finish and effective heat transer are between the package and the heat sink. Θ_{SA} is a function of both the heat sink design and the airflow type and rate.

Using a heat spreader in the package lowers the overall thermal resistance in two ways:

1. It increases the effective heat transfer area between the package and the heat sink and as a result lowers Θ_{CS}. The actual reduction in Θ_{CS} depends on the magnitude of Θ_{CS} without a heat spreader. The larger the value of Θ_{CS} without using a heat spreader, the larger will be the reduction in the value of Θ_{JA} if a heat spreader is used.
2. A heat spreader may also improve the heat sink thermal performance by increasing the effective heat transfer area in the heat sink and making the fins away from the die more effective.

Using a heat spreader with a thermal grease interface will result in about .4 c/w lower Θ_{CA} than that for the package without the heat spreader. Thermal grease is considered one of the more thermally efficient materials for use as an interface between heat sink and package. Thermally conductive adhesives and conductive tapes or films typically have poorer thermal performance when compared to a thin layer of thermal grease, because grease facilitates a larger reduction in thermal resistance.

10.2.1. Case Temperature Specifications

Following are the case temperature specifications for the Pentium processor with and without the heat spreader on the package:

10.2.1.1. PENTIUM® PROCESSOR PACKAGE WITHOUT HEAT SPREADER

The case temperature specifications for the Pentium processor package without heat spreader at 60 and 66 MHz are as follows (Note: This applies to B1" and previous steppings):

1. T_c (case temperature) 0°C to 85°C @ 60 MHz.
2. T_c (case temperature) 0°C to 75°C @ 66 MHz.

10.2.1.2. PENTIUM® PROCESSOR WITH HEAT SPREADER PACKAGE

The case temperature specifications for the Pentium processor heat spreader package at 60 and 66 MHz are as follows (Note: This applies to C1 and later steppings):

1. T_c (case temperature) 0°C to 80°C @ 60 MHz.
2. T_c (case temperature) 0°C to 70°C @ 66 MHz.

In the case of the Pentium processor with heat spreader package, the case temperature is measured at the center of the package top surface on the heat spreader. The procedure to measure the case temperature, is outlined in Chapter 10 of the *Pentium® Processor Family Developer's Manual*, Volume 1 (Order Number 241428).

The thermal specification of the heat spreader package calls for 5 degrees Celsius lower case temperature than the non-spreader package for both 60- and 66-MHz versions. The lower case temperature requirement of the heat spreader package is due to its lower junction to ambient thermal resistance compared to a non-heat spreader package with the same heat sink. For example, at 66-MHz, the heat spreader package will have $0.4 \times 16 = 6.4$ degree Celsius lower case temperature than a non-heat spreader package with the same heat sink design and grease interface. This implies that in a system designed for a non-spreader package, if the non-spreader package is replaced with a heat spreader package, the measured case temperature will be lower by 6.4 degrees Celsius for the 66-MHz and 5.8 degrees Celsius for the 60-MHz versions. The actual reduction in the case temperature will be slightly higher or lower depending on the efficiency of the thermal interface. Therefore, a more conservative value of 5 degrees Celsius is used as the difference between the case temperature specifications of the two package types for both frequency versions. The expectation is that the ambient temperature in the system will be maintained while gaining the benefits of lower junction and case temperatures when the heat spreader package is added to an existing system with the same airflow and unmodified heat sink.

11

Testability

CHAPTER 11
TESTABILITY

This chapter describes the features which are included in the Pentium Processor (510\60, 567\66) or the purpose of enhancing the testability of the Pentium Processor (510\60, 567\66). The capability of the Intel486 CPU test hooks are included in the Pentium Processor (510\60, 567\66), 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 (510\60, 567\66) 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.

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.

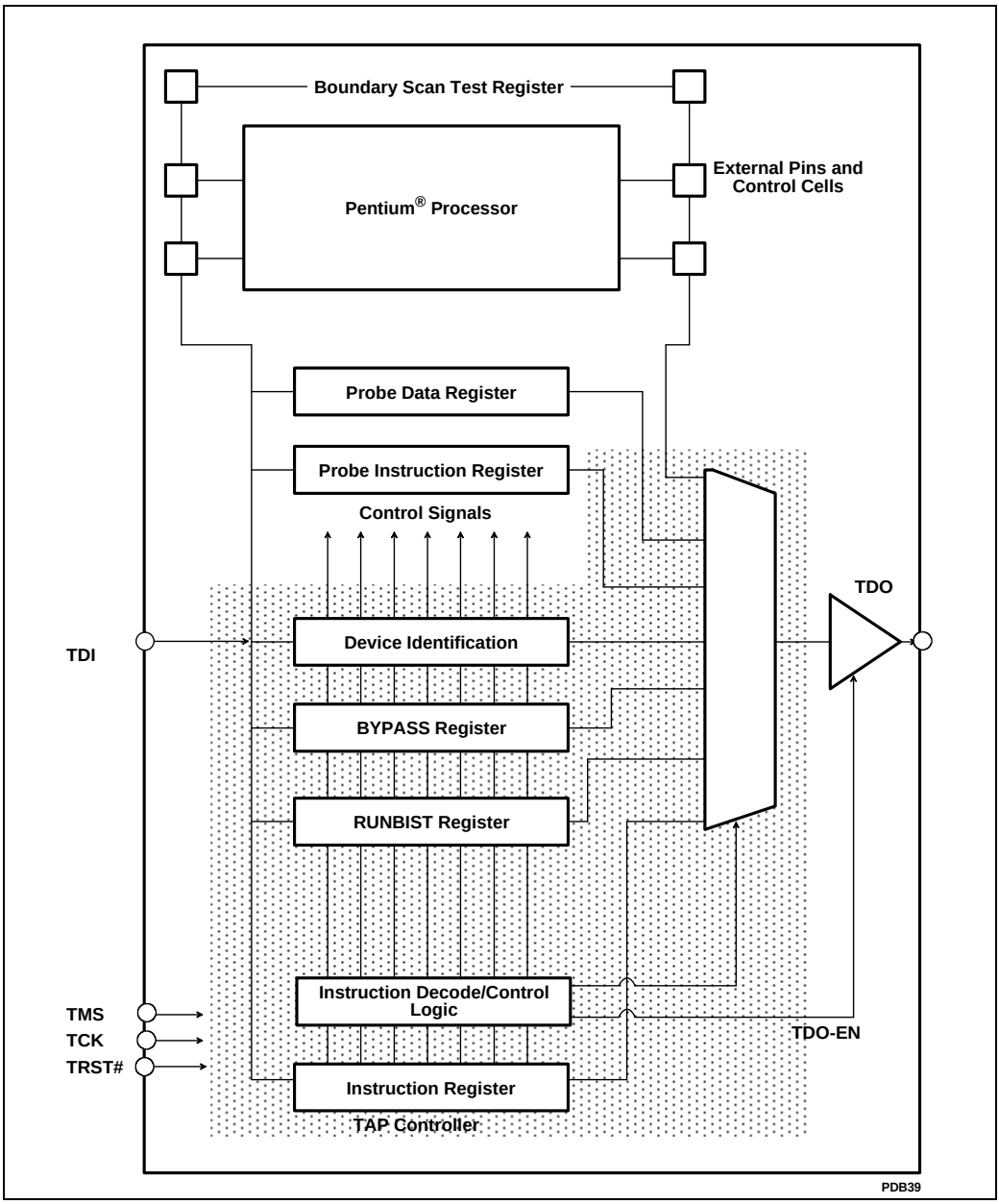


Figure 11-1. Test Access Port Block Diagram

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.

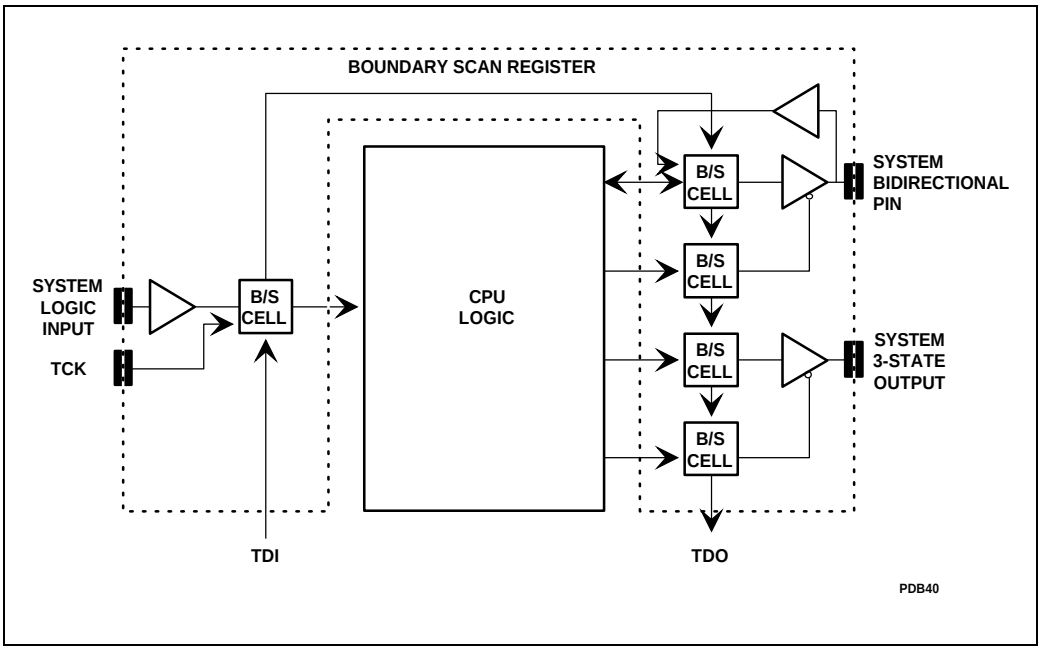


Figure 11-2. Boundary Scan Register

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.

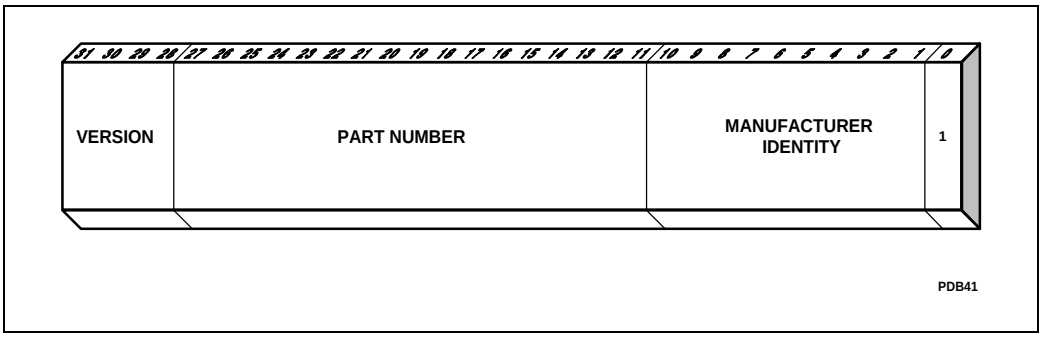


Figure 11-3. Format of the Device ID Register

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 (510\60, 567\66) 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.

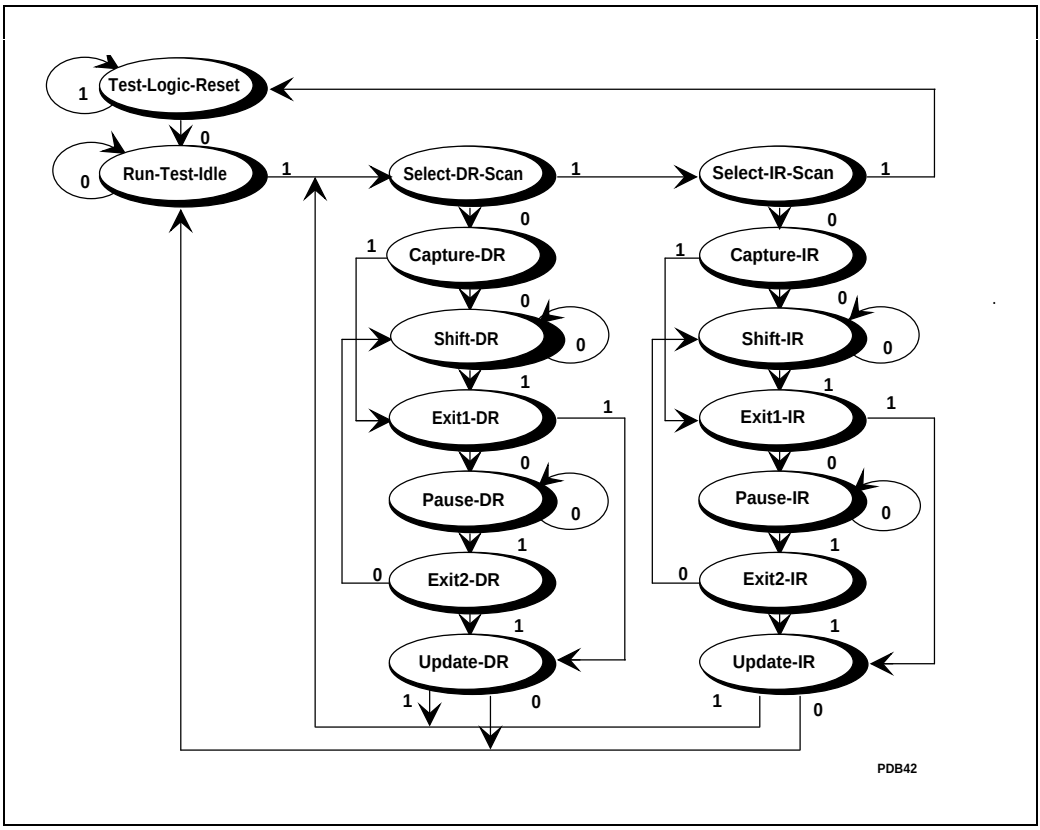


Figure 11-4. TAP Controller State Diagram

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 (510\60, 567\66) 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 (510\60, 567\66) contains a cell for each pin. The following is the bit order of the Pentium Processor (510\60, 567\66) 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*, Order Number 241814, 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.

12

Error Detection

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 writeback 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.

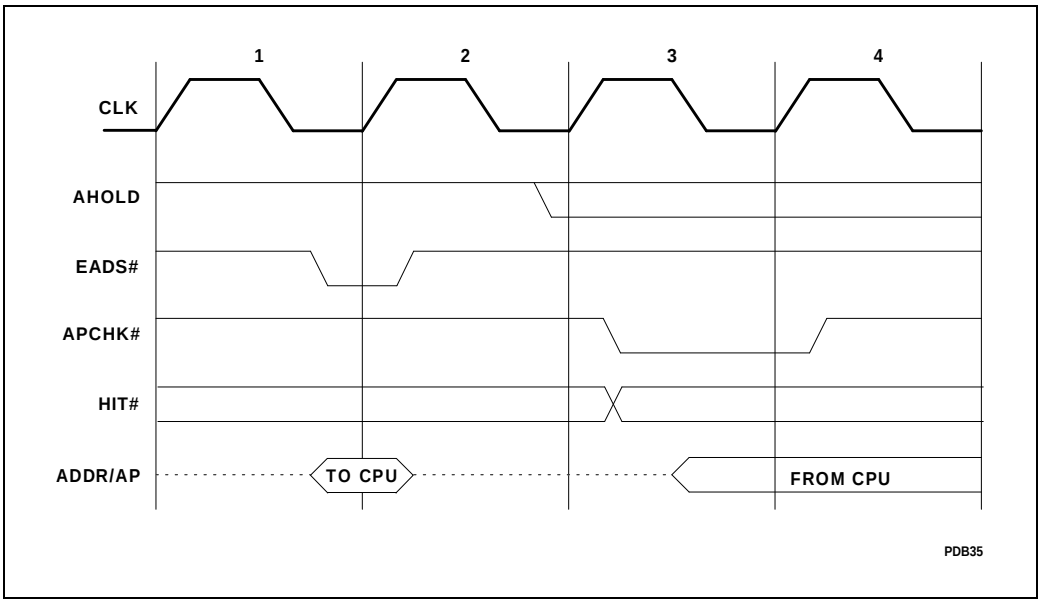


Figure 12-1. Inquire Cycle Address Parity Checking

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.

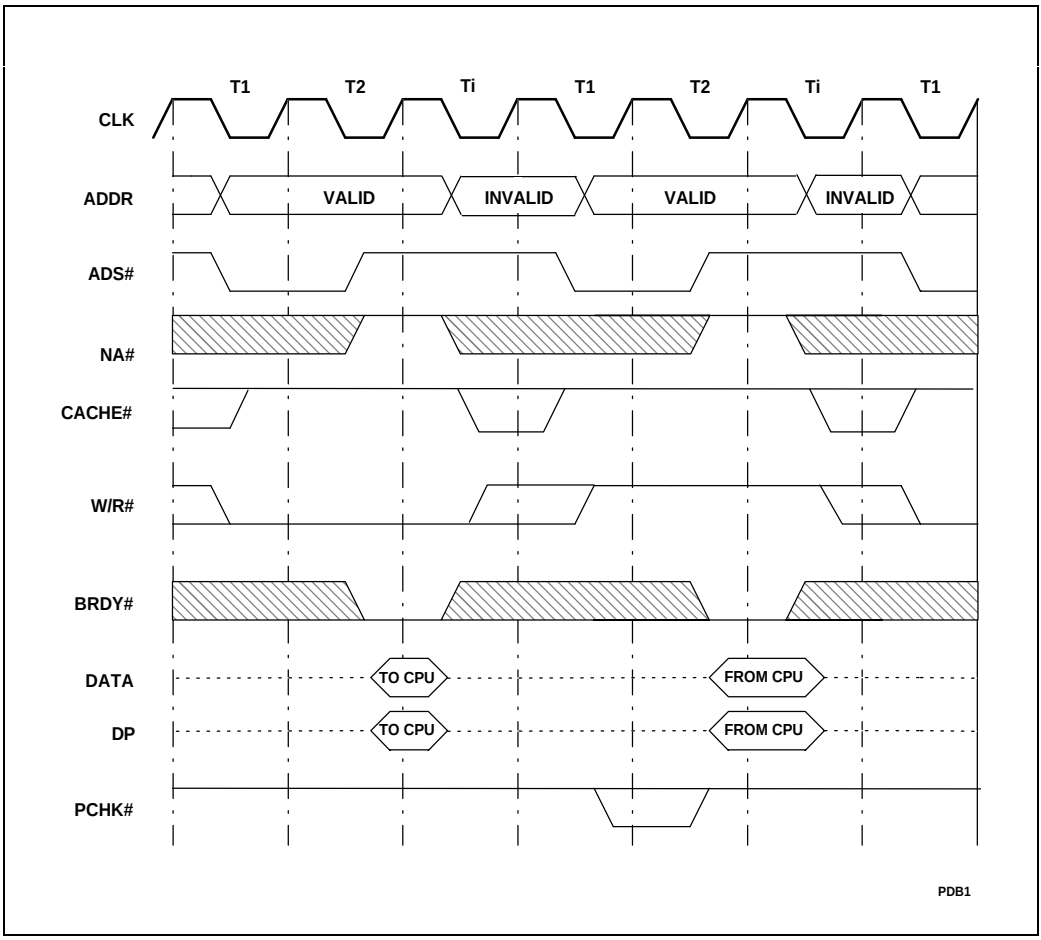


Figure 12-2. Data Parity during a Read and Write Cycle

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. Machine Check Exception

As mentioned in the earlier section, 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.

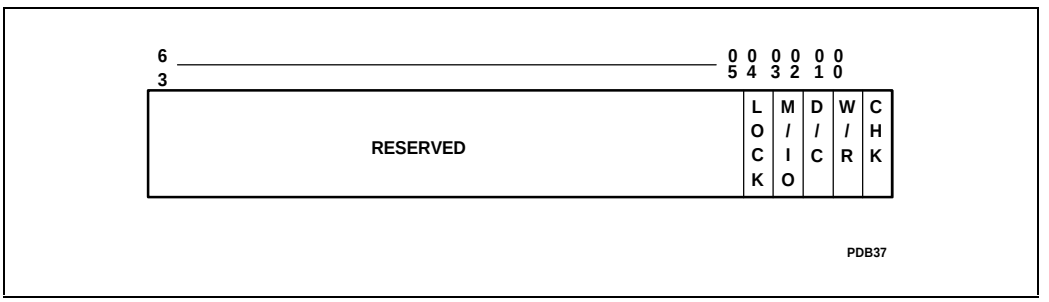


Figure 12-3. Machine Check Type Register

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

12.2.4. 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.

The Pentium processor can remember only one machine check exception at a time. This exception is recognized on an instruction boundary. If BUSCHK# is sampled active while servicing the machine check exception for a previous BUSCHK#, it will be remembered by the processor until the original machine check exception is completed. It is then that the processor will service the machine check exception for the second BUSCHK#. Note that only one BUSCHK# will be remembered by the processor while the machine exception for the previous one is being serviced.

For use of BUSCHK# with STPCLK#, please see Table 17-2.

When the BUSCHK# is sampled active by the processor, the cycle address and cycle type information for the failing bus cycle is latched upon assertion of the last BRDY# of the bus cycle. The information is latched into the Machine Check Address and Machine Check Type registers respectively. However, if the BUSCHK# input is not deasserted before the first BRDY# of the next bus cycle, and the machine check exception for the first bus cycle has not occurred, then new information will be latched into the MCA and MCT registers, over-writing the previous information at the completion of this new bus cycle. Therefore, in order for the MCA and MCT registers to report the correct information for the failing bus cycle when the machine check exception for this cycle is taken at the next instruction boundary, the system must deassert the BUSCHK# input immediately after the completion of the failing bus cycle and before the first BRDY# of the next bus cycle is returned.

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.

Figure 12-4 illustrates the configuration of output pins with respect to FRC. 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.

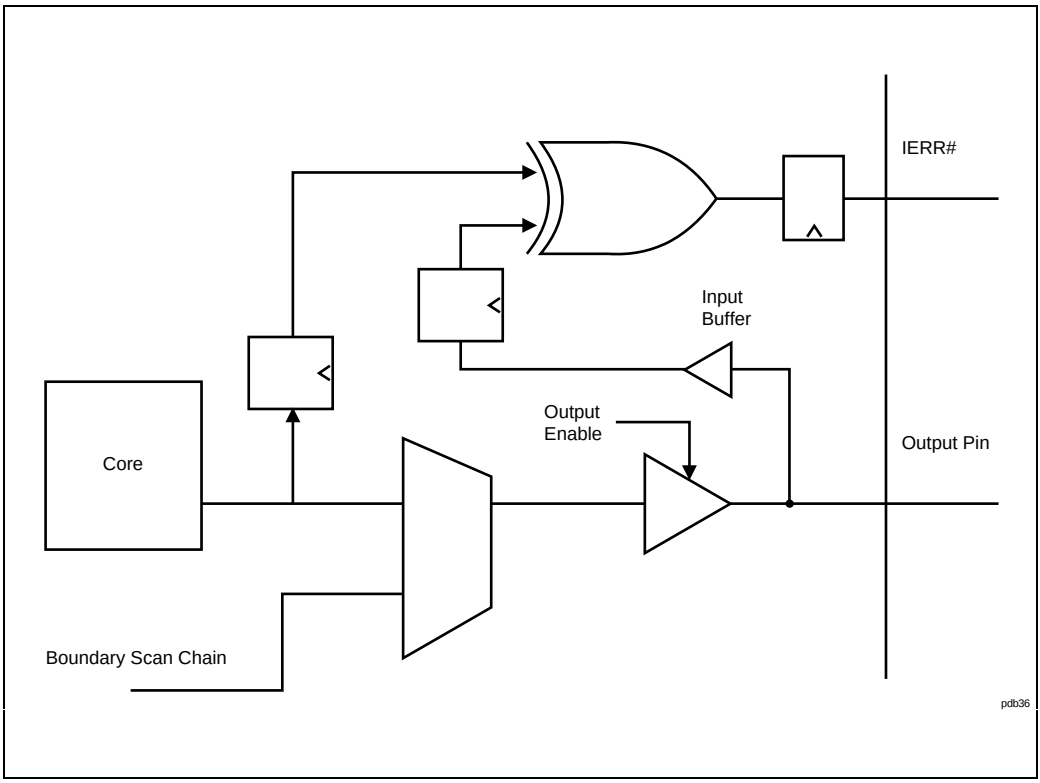


Figure 12-4. Conceptual IERR# Implementation for FRC

12.2.5.1. FRC TIMINGS

Please refer to Pentium processor specifications update for the AC specifications for FRC. Note that the Pentium processor configured as a master operates with the output delays and input setup time requirements as specified in Chapter 7 of the *Pentium® Processor Developer's Manual*, Volume 1.

13

Execution Tracing

CHAPTER 13
EXECUTION TRACING

The Pentium processor (510\60, 567\66) 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 (510\60, 567\66) 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.

NOTE

The Future Pentium® OverDrive® processor does not support execution tracing.

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 (510\60, 567\66) 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 (510\60, 567\66) 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.

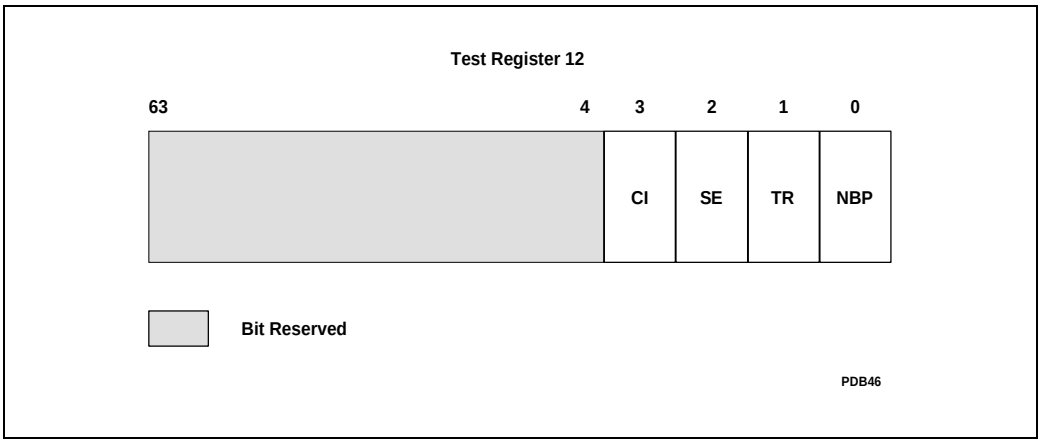


Figure 13-1. Test Register TR12

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 (510\60, 567\66).

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.

14

Pentium® Processor (510\60, 567\66) Power Management

CHAPTER 14
PENTIUM® PROCESSOR (510\60, 567\66) POWER MANAGEMENT

The Pentium® processor (510\60, 567\66) implements Intel's System Management Mode (SMM) architecture. This chapter describes the hardware interface to SMM. For a detailed architectural description, refer to the Power Management chapter in the *Pentium® Processor Family Developer's Manual*, Volume 3.

14.1. SYSTEM MANAGEMENT INTERRUPT PROCESSING

The system interrupts the normal program execution and invokes SMM by generating a System Management Interrupt (SMI#) to the CPU. The CPU will service the SMI# by executing the following sequence.

1. Wait for all pending bus cycles to complete and EWBE# to go active.
2. The CPU asserts the SMIACT# signal while in SMM indicating to the system that it should enable the SMRAM.
3. The CPU saves its state (context) to SMRAM, starting at address location SMBASE + 0FFFFH, proceeding downward in a stack-like fashion.
4. The CPU switches to the System Management Mode processor environment (a pseudo-real mode).
5. The CPU will then jump to the absolute address of SMBASE + 8000H in SMRAM to execute the SMI handler. This SMI handler performs the system management activities.
6. The SMI handler will then execute the RSM instruction which restores the CPU's context from SMRAM, de-asserts the SMIACT# signal, and then returns control to the previously interrupted program execution.

NOTE

The default SMBASE value following RESET is 30000H.

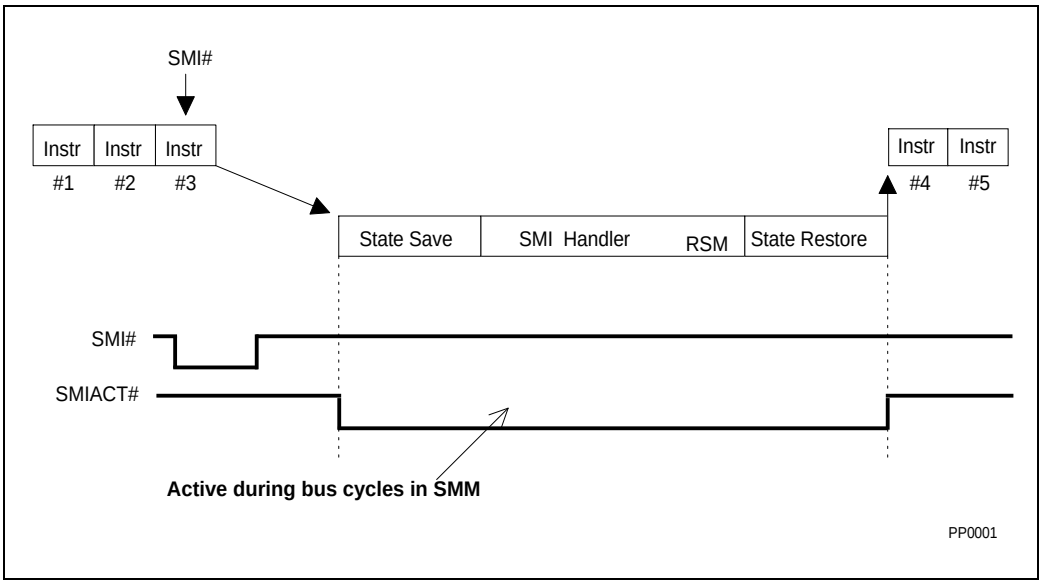


Figure 14-1. Basic SMI# Interrupt Service

The System Management Interrupt hardware interface consists of the SMI# interrupt request input and the SMIACT# output used by the system to decode the SMRAM.

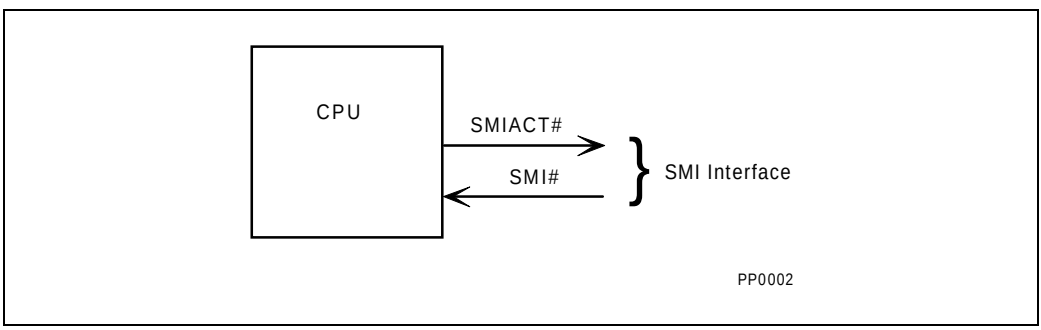


Figure 14-2. Basic SMI# Hardware Interface

14.1.1. System Management Interrupt (SMI#)

SMI# is a falling-edge triggered, non-maskable interrupt request signal. SMI# is an asynchronous signal, but setup and hold times, t28 and t29, must be met in order to guarantee recognition on a specific clock. The SMI# input need not remain active until the interrupt is actually serviced. The SMI# input only needs to remain active for a single clock if the required setup and hold times are met. SMI# will also work correctly if it is held active for an arbitrary number of clocks.

The SMI# input must be held inactive for at least four clocks after it is asserted to reset the edge triggered logic. A subsequent SMI# might not be recognized if the SMI# input is not held inactive for at least four clocks after being asserted.

SMI#, like NMI, is not affected by the IF bit in the EFLAGS register and is recognized on an instruction boundary. An SMI# will not break locked bus cycles. The SMI# has a higher priority than NMI and is not masked during an NMI. In the Pentium® processor (510\60, 567\66), the interrupt priorities are as follow: FLUSH#, SMI#, INIT, NMI, INTR.

After the SMI# interrupt is recognized, the SMI# signal will be masked internally until the RSM instruction is executed and the interrupt service routine is complete. Masking the SMI# prevents recursive SMI# calls. If another SMI# occurs while the SMI# is masked, the pending SMI# will be recognized and executed on the next instruction boundary after the current SMI# completes. This instruction boundary occurs before execution of the next instruction in the interrupted application code, resulting in back to back SMM handlers. Only one SMI# can be pending while SMI# is masked.

The SMI# signal is synchronized internally and must be asserted at least three (3) CLK periods prior to asserting the BRDY# signal in order to guarantee recognition on a specific instruction boundary.

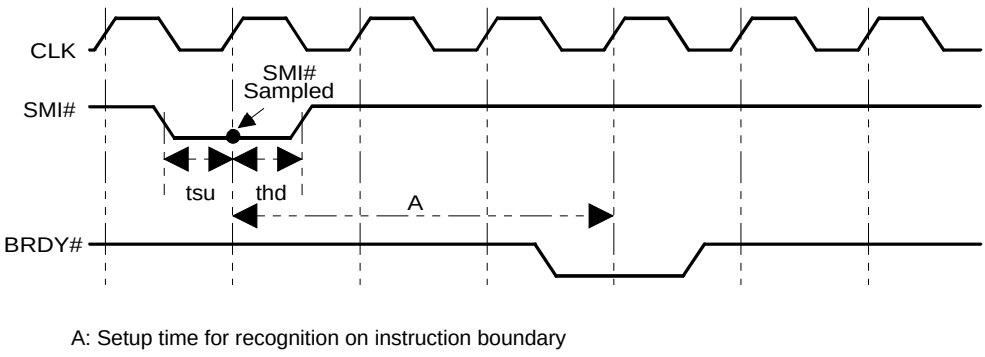
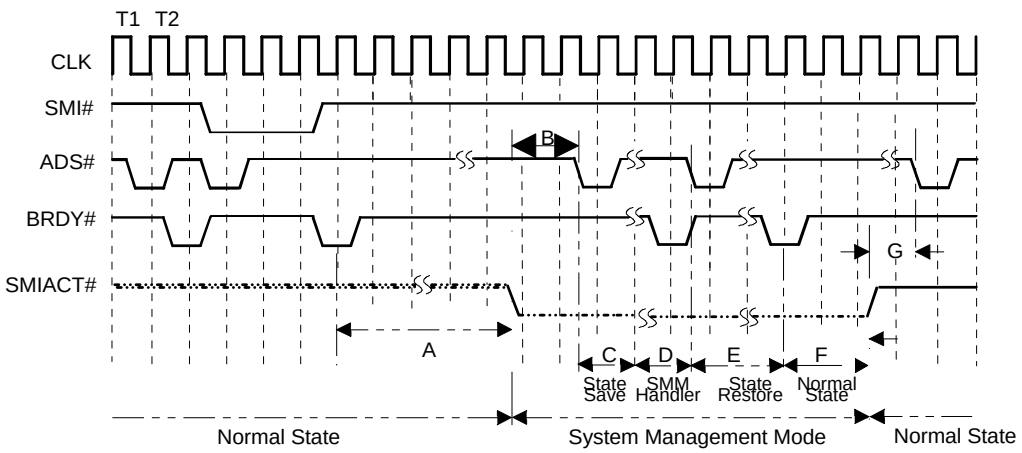


Figure 14-3. SMI# Timing

14.1.2. SMI Active (SMIACT#)

SMIACT# indicates that the CPU is operating in System Management Mode. The CPU asserts SMIACT# in response to an SMI interrupt request on the SMI# pin or through the APIC message. SMIACT# is driven active for accesses only after the CPU has completed all pending write cycles (including emptying the write buffers - EWBE# returned active by the system). SMIACT# will be asserted for all accesses in SMM beginning with the first access to SMRAM when the CPU saves (writes) its state (or context) to SMRAM. SMIACT# is driven active for every access until the last access to SMRAM when the CPU restores (reads) its state from SMRAM. The SMIACT# signal is used by the system logic to decode SMRAM.



	Pentium® Processor
A: Last BRDY# from non-SMM transfer to SMIACT# assertion	2 CLKs minimum
B: SMIACT# assertion to first ADS# for SMM state save	2 CLKs minimum
C: SMM state save (dependent on memory performance)	Approximately 180 CLKs
D: SMM handler	User determined
E: SMM state restore (dependent on memory performance)	Approximately 200 CLKs
F: Last RDY# from SMM transfer to de-assertion of SMIACT#	2 CLKs minimum
G: SMIACT# de-assertion to first non-SMM ADS#	5 CLKs minimum

PP0004

Figure 14-4. SMIACT# Timing

NOTE

The number of CLKs required to complete the SMM state save and restore is very dependent on system memory performance and the CPU bus frequency.

As shown in Figure 14-41.4, the approximate time required to enter an SMI handler routine for the Pentium® processor (510\60, 567\66) (from the completion of the interrupted instruction) is given by:

Latency to beginning of SMI handler = A + B + C = ~184 CLKs

The approximate time required to return to the interrupted application (following the final SMM instruction before RSM) is given by:

Latency to continue interrupted application = E + F + G = ~207 CLKs

14.2. SMM — SYSTEM DESIGN CONSIDERATIONS

14.2.1. SMRAM Interface

The hardware designed to control the SMRAM space must follow these guidelines:

1. A provision should be made to allow for initialization of SMRAM space during system boot up. This initialization of SMRAM space must happen before the first occurrence of an SMI# interrupt. Initializing the SMRAM space must include installation of an SMM handler, and may include installation of related data structures necessary for particular SMM applications. The memory controller providing the interface to the SMRAM should provide a means for the initialization code to manually open the SMRAM space.
2. A minimum initial SMRAM address space of SMBASE + 8000H to SMBASE + 0FFFFH should be decoded by the memory controller.
3. Alternate bus masters (such as DMA controllers) should not be allowed to access SMRAM space. Only the CPU, either through SMI or during initialization, should be allowed access to SMRAM.
4. In order to implement a zero-volt suspend function, the system must have access to all of normal system memory from within an SMM handler routine. If the SMRAM is going to overlay normal system memory, there must be a method of accessing any system memory that is located underneath SMRAM.

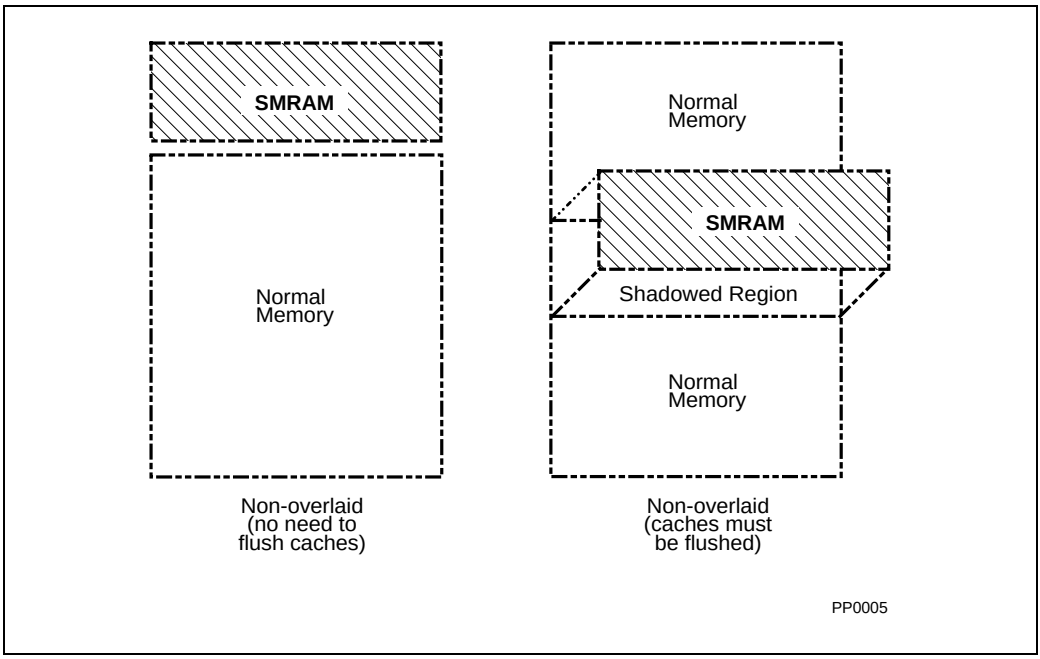


Figure 14-5. SMRAM Location

14.2.2. Cache Flushes

The Pentium® processor (510\60, 567\66) does not unconditionally write back and invalidate its cache before entering SMM (this option is left to the system designer). If the SMRAM is in an area that is cacheable and overlaid on top of normal memory that is visible to the application or operating system (default), then it is necessary for the system to flush both the CPU cache and any second level cache upon entering SMM. This may be accomplished by asserting flush the same time as the request to enter SMM (i.e. Cache flushing during SMM entry is accomplished by asserting the FLUSH# pin at the same time as the request to enter SMM through SMI#). The priorities of FLUSH# and SMI# are such that the FLUSH# will be serviced first. To guarantee this behavior, the constraints on setup and hold timings on the interaction of FLUSH# and SMI# as specified for a processor should be obeyed. When the default SMRAM location is used, SMRAM is overlaid on top of system main memory (at SMBASE + 8000H to SMBASE + 0FFFFH).

In a system where FLUSH# and SMI# pins are synchronous and setup/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 the cache contents before entering SMM, the Pentium® processor (510\60, 567\66) 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 driving of SMIACT# for SMRAM accesses. It is the obligation of the system to ensure that these lines are not cached by returning KEN# inactive.

If SMRAM is located in its own distinct memory space, which can be completely decoded with only the CPU address signals, it is said to be non-overlaid. In this case, there is one new requirement for maintaining cache coherency, refer to Table 14-1 below.

Table 14-1. Scenarios for Cache Flushes with WB Caches

Is SMRAM overlapped with normal memory?	Is Normal Memory cacheable ?	Is SMRAM cacheable?	Flush required during SMM entry?	Flush required during SMM exit?	Comments
No	No	No	No	No	
	No	WT	No	No	
	WT	No	No	No	
	WB	No	No*	No	*Snoop WB's must always go to normal memory space
	WT	WT	No	No	
	WB	WT	No*	No	*Snoop and Replacement WB's must go to normal memory space.
Yes	No	No	No	No	
	No	WT	No	Yes	
	WT	No	Yes	No	
	WB	No	Yes	No	
	WT	WT	Yes	Yes	
	WB	WT	Yes	Yes	

NOTE:

Write-back cacheable SMRAM is not recommended. When flushing upon SMM exit, SMIACT# will be deasserted and may cause regular memory to be overwritten.

The Pentium® processor (510\60, 567\66) implements write back caches. Hence the performance hit due to flushing the cache for SMM execution can be more significant. Due to the write back nature of the cache, flushing the cache has the following penalties:

1. Before entry into SMM (when SMRAM is cacheable), the cache has to be flushed. Hence, all dirty lines need to be written back. This may cause a large number of bus cycles and increase SMM entry latency.
2. If the cache had to be flushed upon SMM exit, execution starts with cache miss 100%. The cache fill cycles reduce performance.

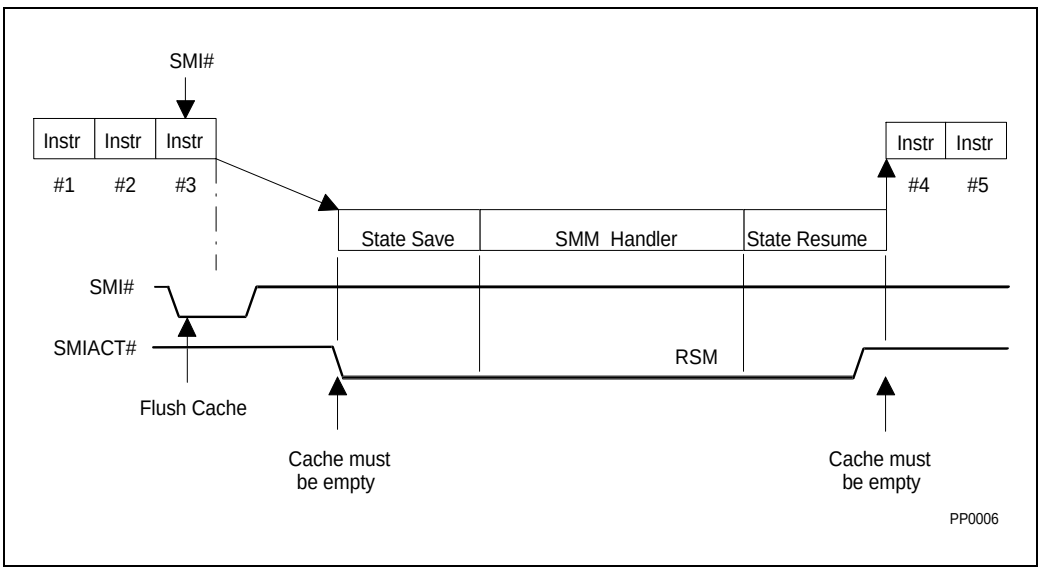


Figure 14-6. FLUSH# Mechanism during SMM with Overlay

The method suggested is shown in Figure 14-7.

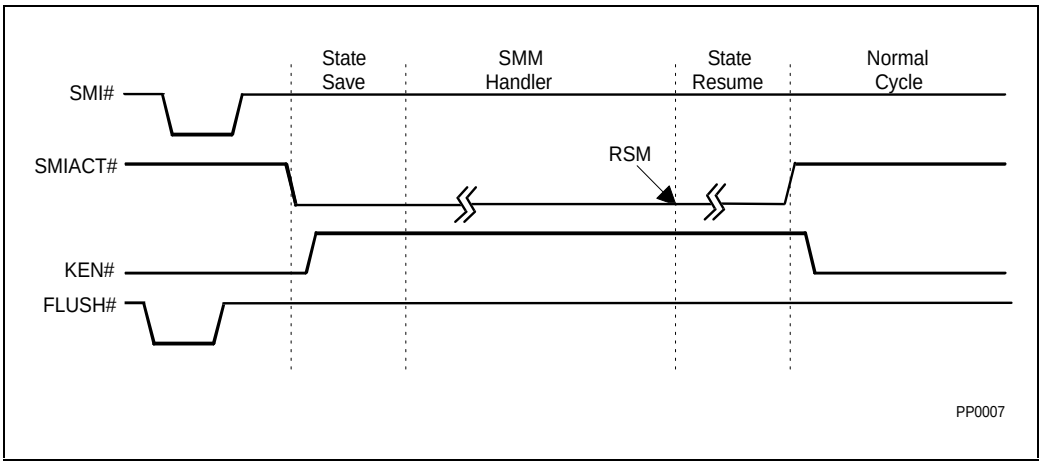


Figure 14-7. Flush with Non-Cached SMM with Overlay

14.2.3. A20M# Pin

Systems based on the MS-DOS operating system contain a feature that enables the CPU address bit A20 to be forced to 0. This limits physical memory to a maximum of 1 Mbyte, and is provided to ensure compatibility with those programs that relied on the physical address wrap around functionality of the original IBM PC. The A20M# pin on Pentium® processor (510\60, 567\66) provides this function. When A20M# is active, all external bus cycles will drive A20 low, and all internal cache accesses will be performed with A20 low.

The A20M# pin is recognized while the CPU is in SMM. The functionality of the A20M# input must be recognized in two instances:

1. If the SMM handler needs to access system memory space above 1 Mbyte (for example, when saving memory to disk for a zero-volt suspend), the A20M# pin must be de-asserted before the memory above 1 Mbyte is addressed.
2. If SMRAM has been relocated to address space above 1 Mbyte, and A20M# is active upon entering SMM, the CPU will attempt to access SMRAM at the relocated address, but with A20 low. This could cause the system to crash, since there would be no valid SMM interrupt handler at the accessed location.

In order to account for the above two situations, the system designer must ensure that A20M# is de-asserted on entry to SMM. A20M# must be driven inactive before the first cycle of the SMM state save, and must be returned to its original level after the last cycle of the SMM

state restore. This can be done by blocking the assertion of A20M# whenever SMIACT# is active.

14.2.4. SMM and Second Level Write Buffers

Before the Pentium® processor (510\60, 567\66) enters SMM, it empties its internal write buffers. This is necessary so that the data in the write buffers is written to normal memory space, not SMM space. Once the CPU is ready to begin writing an SMM state save to SMRAM, it asserts the SMIACT# signal for SMRAM references. SMIACT# may be driven active by the CPU before the system memory controller has had an opportunity to empty the second level write buffers.

To prevent the data from these second level write buffers from being written to the wrong location, the system memory controller needs to direct the memory write cycles to either SMM space or normal memory space. This can be accomplished by saving the status of SMIACT# along with the address for each word in the write buffers.

EWBE# can also be used to prevent the CPU from asserting SMIACT# before write buffers are empty. The processor will wait for an active EWBE# before asserting SMIACT#.

15

Debugging

CHAPTER 15
DEBUGGING

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:

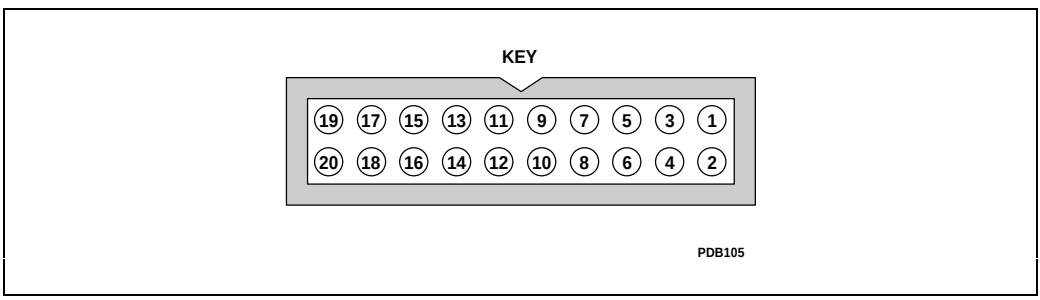


Figure 15-1. Debug Port Connector

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.

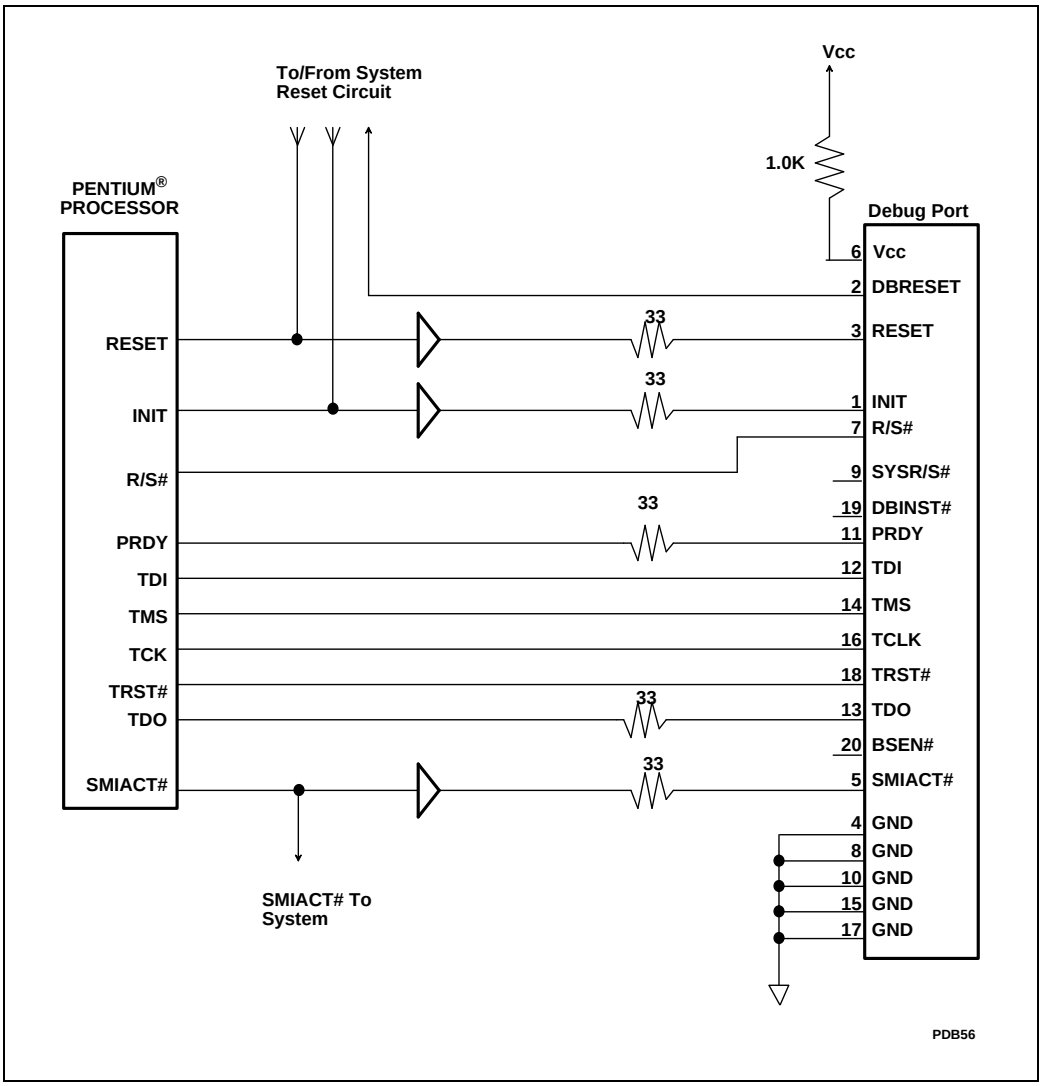


Figure 15-2. Minimal Debug Port Implementation

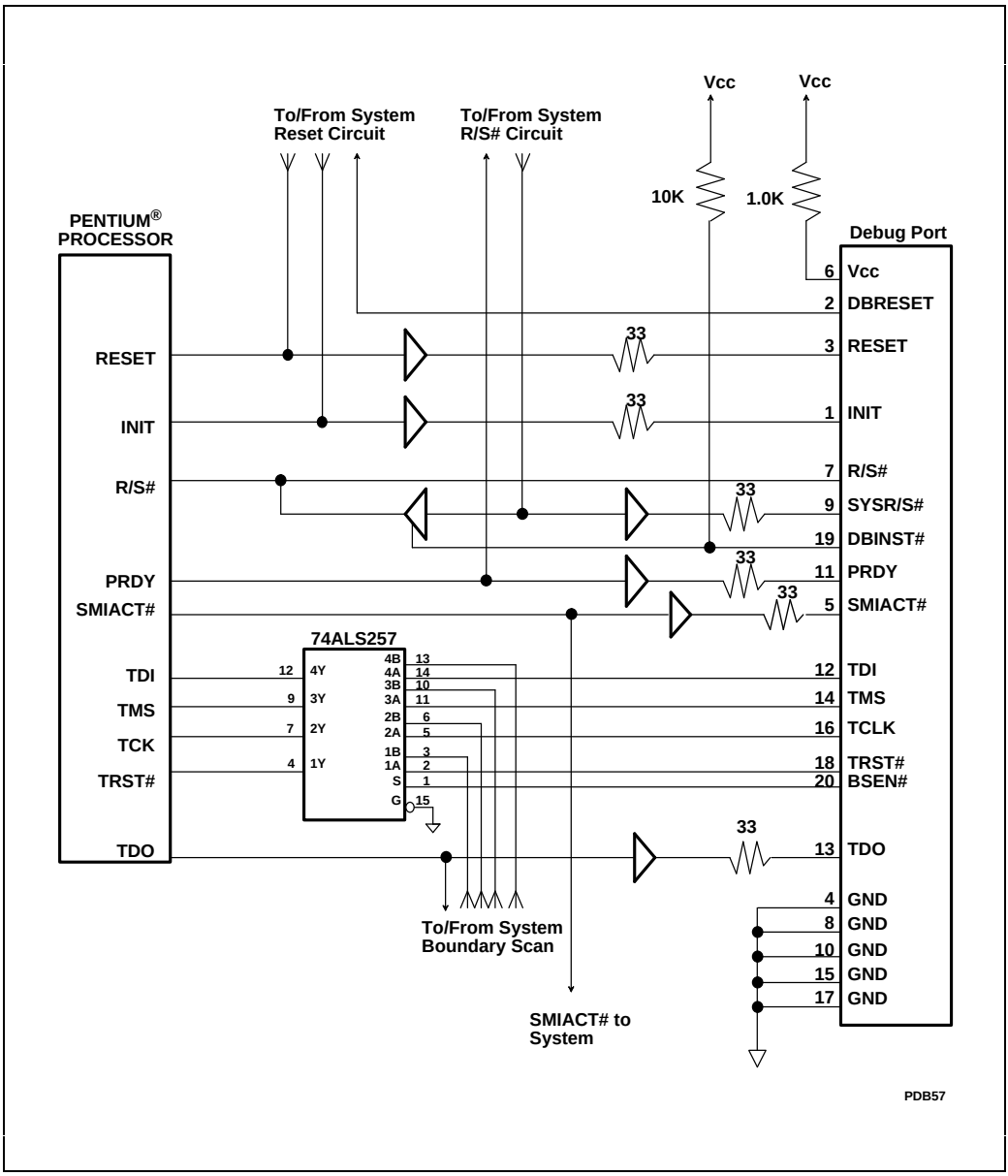


Figure 15-3. Maximal Debug Port Implementation

16

Future Pentium® OverDrive® Processor for Pentium Processor (510\60, 567\66)-Based Systems Socket Specification

CHAPTER 16
FUTURE PENTIUM® OverDrive® PROCESSOR FOR PENTIUM PROCESSOR (510\60, 567\66)-BASED SYSTEMS SOCKET SPECIFICATION

16.1. INTRODUCTION

The Future Pentium OverDrive processor is an end user single-chip CPU upgrade product for Pentium processor (510\60, 567\66)-based systems. The Future Pentium OverDrive processor will speed up most software applications by 40% to 70%. It is binary compatible with the Pentium processor (510\60, 567\66).

An upgrade socket (Socket 4) has been defined along with the Pentium processor (510\60, 567\66) as part of the processor architecture. The Future Pentium OverDrive processor will be socket compatible with the Pentium processor (510\60, 567\66). The Future Pentium OverDrive processor is packaged in a 273-pin ceramic pin grid array package with an attached fan/heatsink present on the turbo upgrade processor component.

Execution tracing is not supported in the Future Pentium OverDrive processor, and performance monitoring is implemented differently than in the Pentium processor (510\60, 567\66). Refer to section 16.1.3 .

16.1.1. Upgrade Objectives

Systems using the Pentium processor (510\60, 567\66) must use Socket 4 to also accept the Future Pentium OverDrive processor. Inclusion of upgrade Socket 4 in Pentium processor (510\60, 567\66) systems provides the end user with an easy and cost effective way to increase system performance. The process of simply installing an upgrade component into an easy to use Zero Insertion Force (ZIF) socket to achieve enhanced system performance is familiar to the millions of end users and dealers who have purchased Intel Math CoProcessor upgrades to boost system floating-point performance.

Inclusion of Socket 4 in Pentium processor (510\60, 567\66) systems provides the end-user with an easy and cost-effective way to increase system performance. The paradigm of simply installing an additional component into an easy to use Zero Insertion Force (ZIF) Socket to achieve enhanced system performance is familiar to the millions of end-users and dealers who have purchased Intel math coprocessor upgrades to boost system floating point performance.

The majority of upgrade installations which take advantage of Socket 4 will be performed by end users and resellers. Therefore, it is important that the design be "end user easy," and that the amount of training and technical expertise required to install the upgrade processors be minimized. Upgrade installation instructions should be clearly described in the system user's

manual. In addition, by making installation simple and foolproof, PC manufacturers can reduce the risk of system damage, warranty claims and service calls.

Feedback from Intel's Math CoProcessor upgrade customers highlights three main characteristics of end user easy designs:

- accessible socket location
- clear indication of upgrade component orientation
- minimization of insertion force

The Future Pentium OverDrive processor will support the Intel 82430 PCIset. Unlike the Pentium processor (510\60, 567\66), the Future Pentium OverDrive processor will not support the 82496 Cache Controller and 82491 Cache SRAM chip set.

16.1.2. Intel Verification Program

The Intel Platform Support Labs ensures that a Pentium processor (510\60, 567\66)-based personal computer meets a minimum set of design criteria for reliable and straightforward CPU upgradability with the Future Pentium OverDrive processor. Evaluation performed at the Intel Platforms Support Labs confirms that future Pentium OverDrive processor specifications for mechanical, thermal, electrical, functional, and end-user installation attributes have been met. While system designs may exceed these minimum design criteria, the intent is to provide end-users with confidence that computer systems based on verified designs can be upgraded with Future Pentium OverDrive processors.

The OEM submits production-ready designs to one of Intel's worldwide Platform Support Labs for evaluation. The OEM benefits from advance testing of the design prior to availability of the Future Pentium OverDrive processor. Section 16.8.2 briefly describes the areas which will be evaluated at Intel Platform Support Labs.

Contact your local Intel representative for more information on the Intel Platform Support Labs for Pentium processor (510\60, 567\66)-based systems.

16.1.3. Functional Differences to the Pentium® Processor (510\60, 567\66)

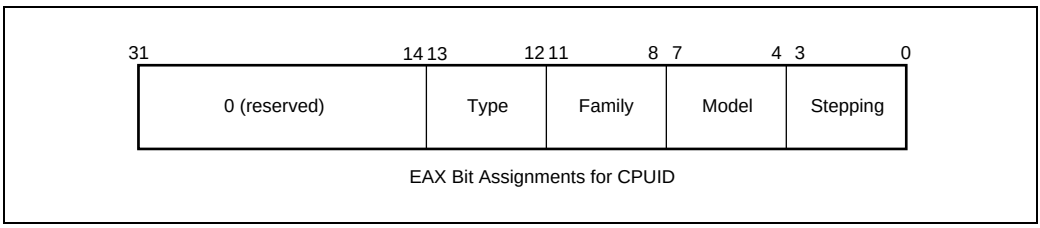
16.1.3.1. DEBUG FEATURE DIFFERENCES

The following Pentium processor (510\60, 567\66) features are either not supported or functionality has been modified in the Future Pentium OverDrive processor:

- The Pentium processor (510\60, 567\66) IU, IV and IBT functions have been removed.
- The branch trace (BT[3:0]) functions have been removed.
- The breakpoint pins (BP[3:0]) have been redefined such that each assertion of one of these pins indicates that 1 to N BP matches occurred, where N = core/bus frequency ratio.

16.1.3.2. FEATURE ENHANCEMENTS

- The CPUID instruction is used to deliver processor-specific information. The Future Pentium OverDrive processor CPUID status has been extended to supply the pocessor type information which includes "Turbo-Upgrade" classification ("type" field: bits 13-12 = 0-1).



EAX Bit Assignments for CPUID

- STI/CLI latencies are each 2 clocks shorter.
- BIOS should not depend on the internal Future Pentium OverDrive processor caches being identical to those of the Pentium processor (510\60, 567\66) (except for the line size).

16.2. Future Pentium® OverDrive® Processor Socket

The following drawings in Figure 16-1 show the preliminary worst case socket footprints from two qualified Socket 4 vendors, AMP and Yamaichi. OEMs should work directly with socket vendors for the most current socket information.

To order Socket 4 from AMP and Yamaichi, the phone numbers and part numbers are:

AMP:	1-800-522-6752	Part #: 916510-1
Yamaichi:	1-800-769-0797	Part #: NP11J-273K13221

Figure 16-2 shows the Future Pentium OverDrive processor chip's orientation in the Socket 4.

For a complete list of qualified sockets and vendor order numbers, call the Intel Faxback number for your geographical area and have document number 7209 automatically faxed to you.

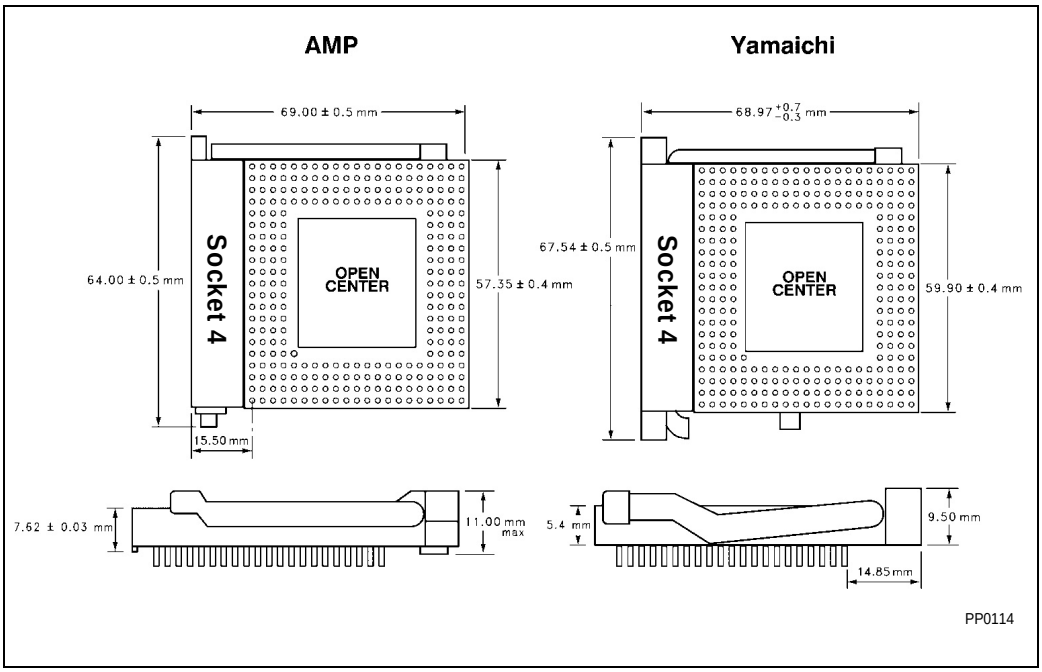


Figure 16-1. Socket 4 Footprint Dimensions
(See socket manufacturer for the most current information.)

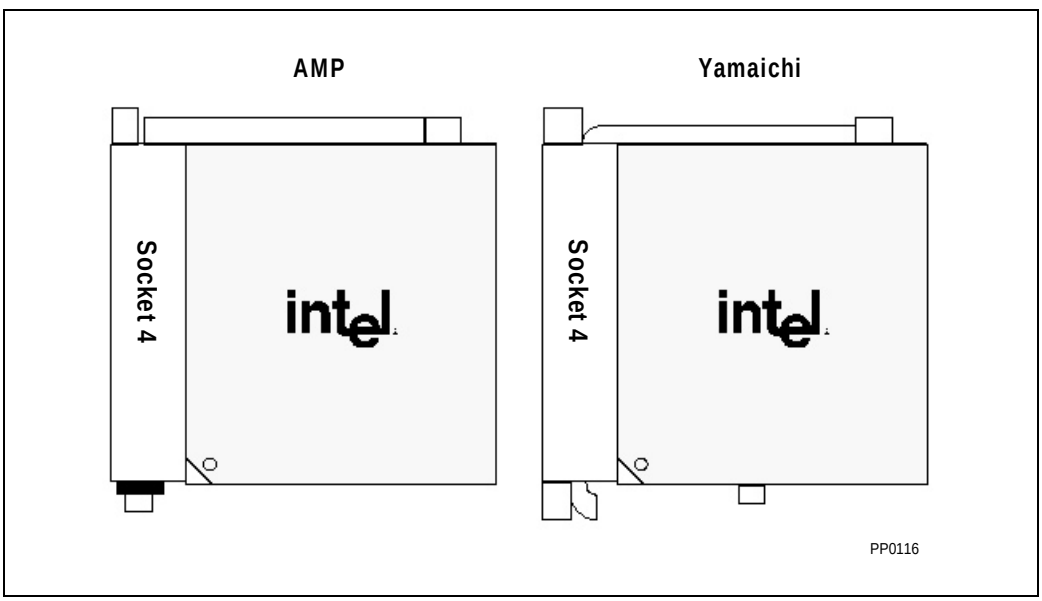


Figure 16-2. Socket 4 Chip Orientation

16.3. SOCKET 4 PINOUT

The Future Pentium OverDrive processor pinout is identical to that of the Pentium processor (510\60, 567\66). Note that all input pins must meet their A.C./D.C. specifications to guarantee proper functional behavior. Figure 16-3 and Figure 16-4 show the Socket 4 pinout.

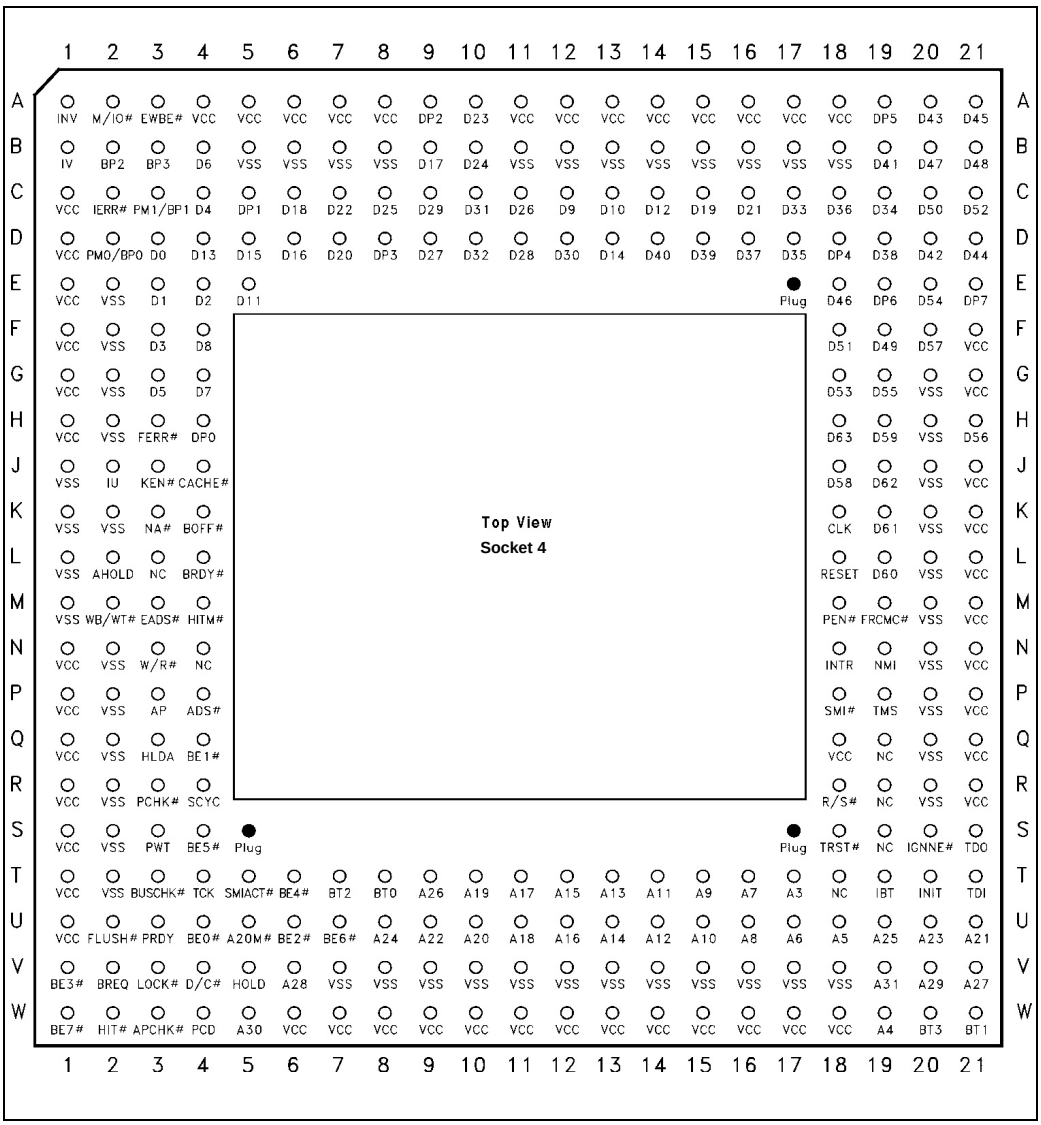


Figure 16-3. Socket 4 Pinout (Top View)

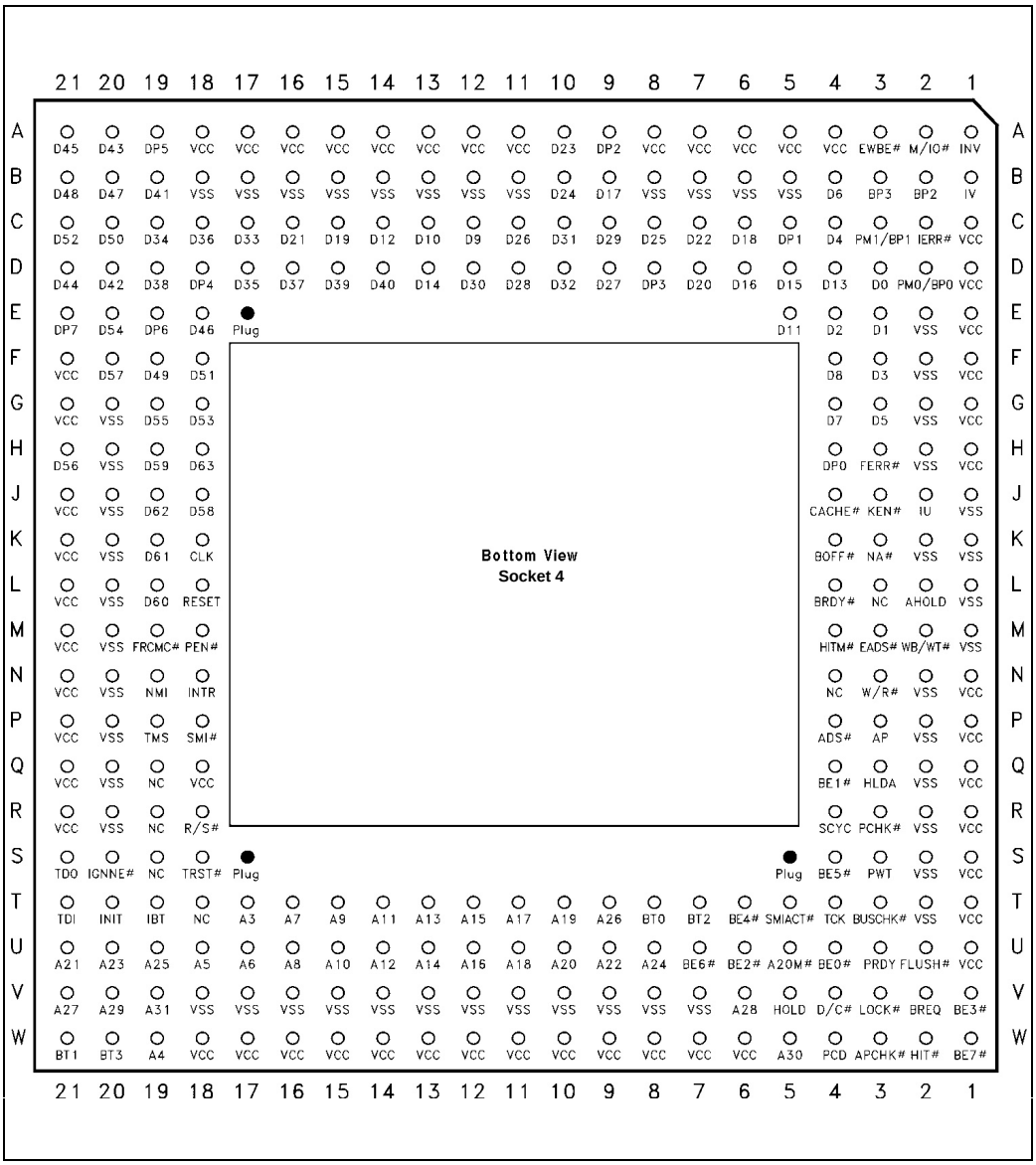


Figure 16-4. OverDrive® Processor Pinout (Bottom View)

Locations E17, S17 and S5 should be plugged on the Socket 4 in order to ensure that the Pentium processor (510\60, 567\66) or OverDrive processor chip is installed in the socket with the correct orientation.

16.4. ELECTRICAL SPECIFICATIONS

The Future Pentium OverDrive processor will have the same power and ground specifications, decoupling recommendations, connection specifications and maximum ratings as the Pentium processor (510\60, 567\66).

16.4.1. Absolute Maximum Ratings for Upgrade

The on-chip voltage regulation and fan/heatsink devices included on the Future Pentium OverDrive processor require different stress ratings than the Pentium processor (510\60, 567\66). The voltage regulator is surface mounted on the Future Pentium OverDrive processor and is, therefore, an integral part of the assembly. The Future Pentium OverDrive processor storage temperature ratings area tightened as a result. The fan is a detachable unit, and the storage temperature is stated separately in the table below. Functional operation of the Future Pentium OverDrive processor remains 0°C to 70°C.

Table 16-1. Absolute Maximum Ratings

Future Pentium® OverDrive® Processor and Voltage Regulator Assembly:					
	Parameter	**Min**	**Max**	**Unit**	**Notes**
	Storage Temperature	-30	100	°C	
	Case Temperature Under Bias	-30	100	°C	
Fan:					
	Parameter	**Min**	**Max**	**Unit**	**Notes**
	Storage Temperature	-30	75	°C	
	Case Temperature Under Bias	-30	75	°C	

WARNING

Stressing the devices beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

16.4.2. DC Specifications

The Future Pentium OverDrive processor will have the same DC specifications as the Pentium processor (510\60, 567\66), including that I_{CC} (Power Supply Current) as shown in Table 16-2

Table 16-2. OverDrive® Processor I_{CC} Specifications (2)

Symbol	Parameter	Min	Max	Unit	Notes
I_{CC}	Power Supply Current		3200 2910	mA mA	66 MHz(1) 60 MHz(1)

NOTE:

1. Worst case average I_{CC} for a mix of test patterns. (The mix of test patterns will be determined after silicon is examined.)
2. See Chapter 7 for Pentium processor (510\60, 567\66) V_{CC} specification.

See Chapter 7 for a listing of the remaining DC specifications.

16.4.3. AC Specifications

The Future Pentium OverDrive processor will have the same AC specifications as the Pentium processor (510\60, 567\66). See Chapter 7 for a listing of the AC specifications. The functional parameters for the Future Pentium OverDrive processor's AC specifications are the same as those for Pentium processor (510\60, 567\66) except T_{SINK} as shown below:

$$T_{SINK} = 0°C \text{ to } + 70°C$$

16.5. MECHANICAL SPECIFICATIONS

The Future Pentium OverDrive processor for Pentium processor (510\60, 567\66)-based systems is packaged in a 273-pin ceramic pin grid array (PGA) with attached fan/heatsink. The pins are arranged in a 21 ´ 21 matrix and the package dimensions will be 2.16² ´ 2.16² (5.49 cm ´ 5.49 cm). See Table 16-3.

Table 16-3. OverDrive® Processor Package Information Summary

Package Type	Total Pins	Pin Array	Package Size
PGA	273	21 x 21	2.16^2 x 2.16^2 (5.49 cm x 5.49 cm)

NOTE: See DC Specifications for more detailed power specifications.

Table 16-4. OverDrive® Processor Mechanical Specifications

Family: Ceramic Pin Grid Array Package						
Symbol	**Millimeters**			**Inches**		
	Min	**Max**	**Notes**	**Min**	**Max**	**Notes**
A		33.98	Solid Lid		1.338	Solid Lid
A1	2.84	3.50	Solid Lid	0.112	0.138	Solid Lid
A2	0.33	0.43	Solid Lid	0.013	0.017	Solid Lid
A3	2.51	3.07		0.099	0.121	
A4		20.32			0.800	
A5	10.16			0.400		
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	

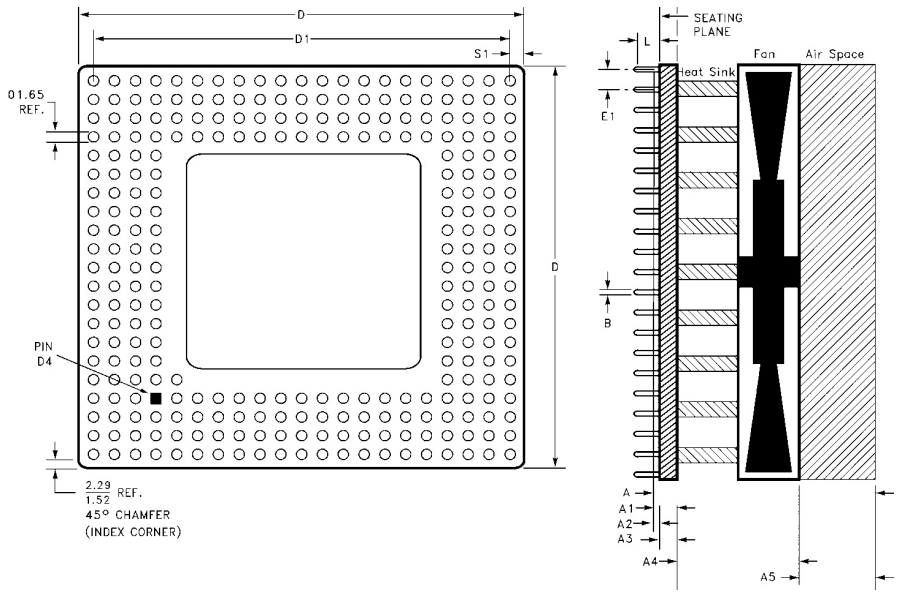


Figure 16-5. Processor Package Dimensions

As can be seen in the mechanical dimensions in Table 16-4 and Figure 16-5, the actual height required by the heatsink and fan is less than the total space allotted. Since the Future Pentium OverDrive processor for Pentium processor (510\60, 567\66)-based systems employs a fan/heatsink, a certain amount of space is required above the fan/heatsink unit to ensure that the airflow is not blocked. Figure 16-6 shows unacceptable blocking of the airflow for the Future Pentium OverDrive processor fan/heatsink. Figure 16-7 details the minimum space needed around the PGA package to ensure proper heatsink airflow.

As shown in Figure 16-7, it is acceptable to allow any device (i.e., add-in cards, surface mount device, chassis, etc.) to enter within the free space distance of 0.2² from the PGA package if it is not taller than the level of the heatsink base. In other words, if a component is taller than height "B," it cannot be closer to the PGA package than distance "A." This applies to three of the four sides of the PGA package, although the back and handle sides of a ZIF socket will generally automatically meet this specification since they have widths larger than distance "A."

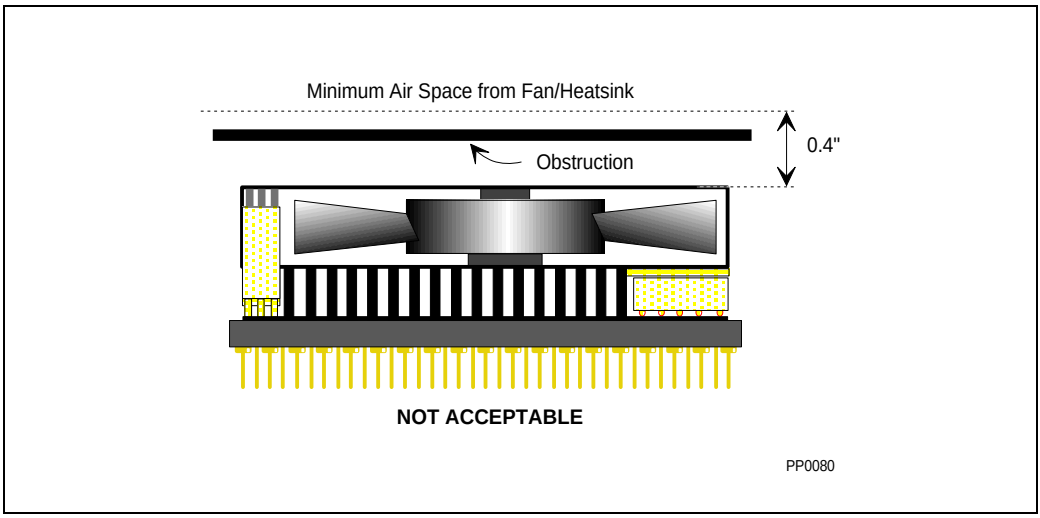


Figure 16-6. Fan/Heatsink Top Space Requirements

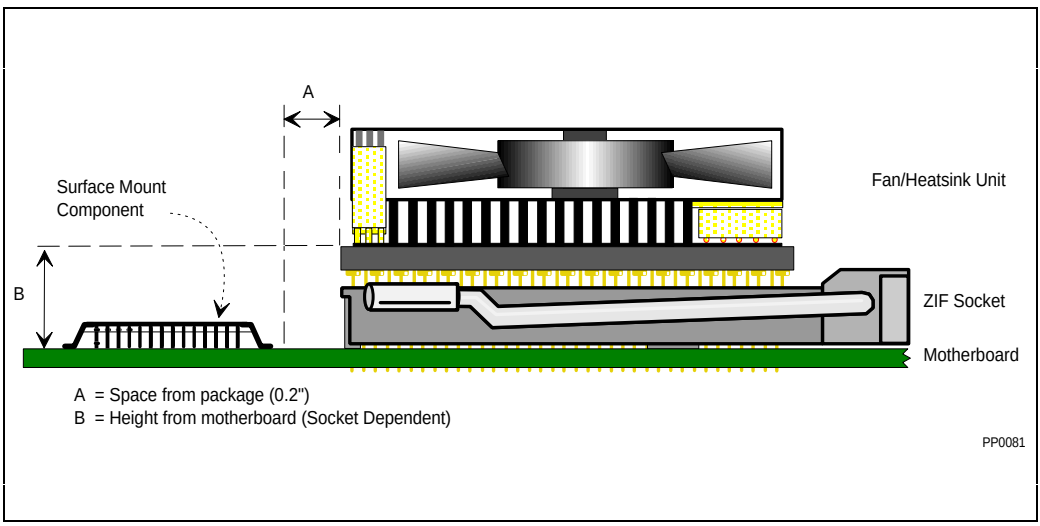


Figure 16-7. Required Free Space from Sides of PGA Package

16.6. THERMAL SPECIFICATIONS

The fan/heatsink cooling solution will properly cool the Future Pentium OverDrive processor as long as the maximum air temperature entering the fan/heatsink cooling solution (T_A(In)) does not exceed 45°C. It is left up to the OEM to ensure that T_A(In) meets this specification by providing sufficient airflow around the Future Pentium OverDrive processor heatsink unit.

Intel's fan/heatsink will dissipate approximately 1W and is powered by the chip such that no external wires or connections are required. The extra power needed for the fan/heatsink is taken into account in the I_{CC} numbers of the processor. Additionally, Intel is evaluating the feasibility of having the Future Pentium OverDrive processor monitor its temperature. No BIOS or hardware changes will be needed for this thermal protection mechanism. The shutdown temperature will be greater than the maximum temperature specification of the processor. The fan/heatsink unit will be designed to be removable so that if fan failure should occur, the unit may be easily replaced. Figure 16-8 gives a functional representation of the processor and fan/heatsink unit. The actual fan/heatsink unit may be different from the one shown in the figure.

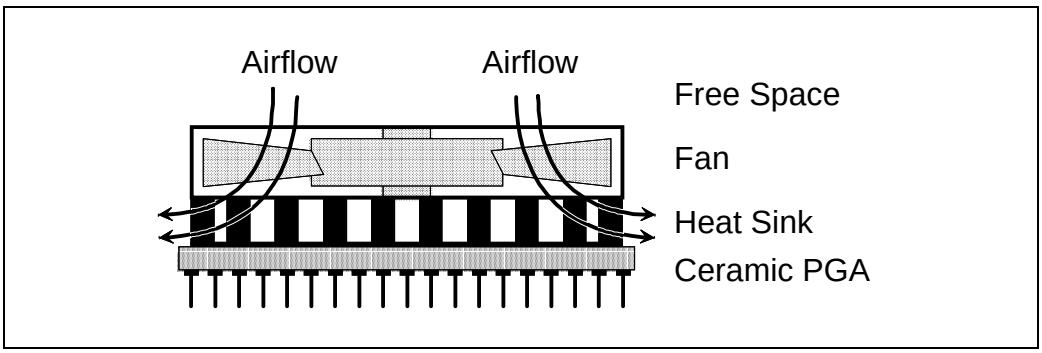


Figure 16-8. Fan/Heatsink Example

Since the Future Pentium OverDrive processor for Pentium processor (510\60, 567\66)-based systems employs a fan/heatsink, it is not as important that the processor heatsink receive direct airflow, rather that the system has sufficient capability to remove the warm air that the Future Pentium OverDrive processor will generate. This implies that enough airflow exists at the Socket 4 to keep localized heating from occurring. This can be accomplished by a standard power supply fan with a clear path to the processor. Figure 16-9 shows how system design can cause localized heating to occur by limiting the airflow in the area of the processor. The airflow supplied in the system should also be enough to ensure that the OEM processor shipped with the system will meet the OEM processor thermal specifications before the system is upgraded with the Future Pentium OverDrive processor.

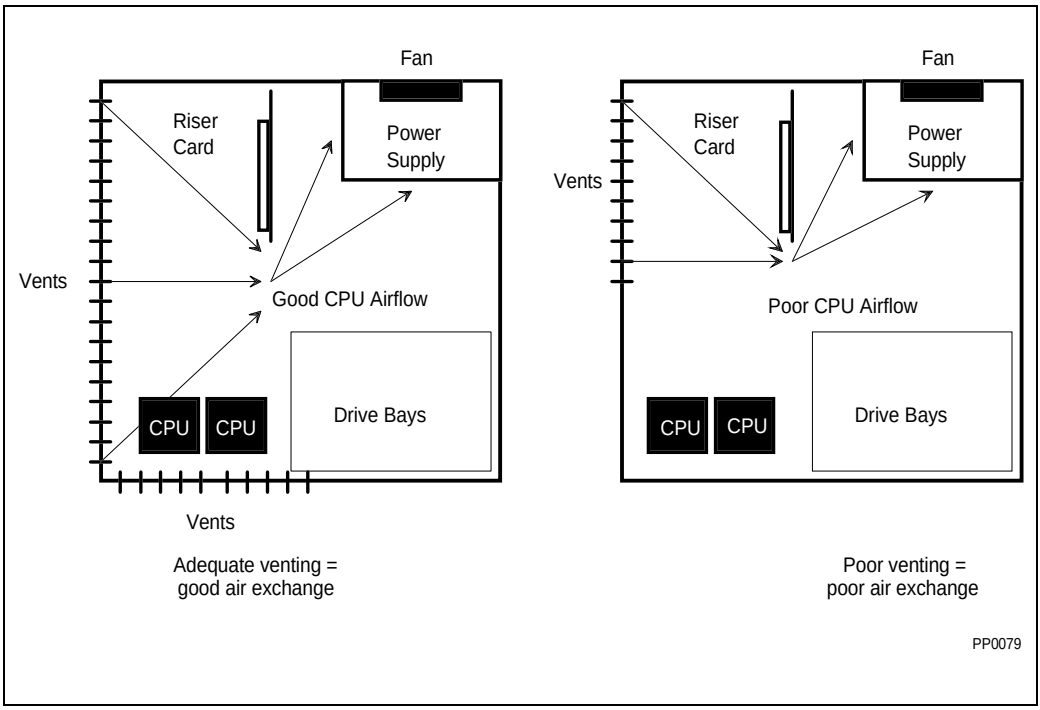


Figure 16-9. Airflow Design Examples

16.7. TESTABILITY

16.7.1. Boundary Scan

The Future Pentium OverDrive processor supports the IEEE Standard 1149.1 boundary scan using the Test Access Port (TAP) and TAP Controller as described in Chapter 11. The boundary scan register for the Future Pentium OverDrive processor contains a cell for each pin. The turbo upgrade component will have a different bit order than the Pentium processor (510\60, 567\66). If the TAP port on your system will be used by an end user following installation of the Future Pentium OverDrive processor, please contact Intel for the bit order of the OverDrive processor boundary scan register.

16.8. INTEL PLATFORM SUPPORT LABS CRITERIA

16.8.1. Introduction

The purpose of this section is to provide the OEM engineers with information regarding the Intel Platform Support Labs and the OverDrive processor upgradability that can be incorporated during the system design and manufacturing phase. The evaluatation criteria of the Intel Platform Support Labs for the Future Pentium OverDrive processor for the Pentium Processor (510\60, 567\66)-based systems are specified in this section. A thorough review of the Criteria section is recommended for all engineering groups.

Important Note

The Future Pentium® OverDrive® processor for Pentium processor (510\60, 567\66)-based systems does not support subsystems which require 5V CMOS drive levels, such as the 82496 cache controller.

16.8.1.1. RELATED DOCUMENTS

Several other documents are available that will help the engineer understand the Intel Platform Support Labs and design systems that meet the Intel Platform Support Labs evaluation criteria. These references will provide detailed information required for the system design to support the Future Pentium OverDrive processor. To obtain information regarding these documents, contact your local Intel field sales representative.

You can also call the Intel Faxback number for your geographical area to obtain any of these below documents that have a Faxback document number next to them. You will be asked to enter the Faxback number for the desired document on a digital touch tone telephone key pad. The Intel Faxback telephone numbers are as follows: 1-916-356-3105 (worldwide), 1-800-628-2283 (U.S. and Canada), or 44(0)793-496646 (Europe).

To obtain documents from Intel Corporation Literature Sales call 1-800-628-8686 and order the desired document using the associated Intel literature order number.

Processor Specifications:

- *Pentium® Processor Family Developer's Manual, Volume 3: Architecture and Programming Manual* (Intel Literature Order # 241430)

Intel Platform Support Labs Documentation:

- Intel Platform Support Labs Guide (Faxback document # 7300)
- System Design Checklist (Faxback document # 7307)
- Motherboard Design checklist (Faxback document # 7312)
- List of Qualified Sockets and Vendor Order Numbers (Faxback document # 7209)

16.8.2. Intel Platform Support Labs Verification Criteria

The Intel Platform Support Labs helps assure OEMs and PC buyers that their systems meet minimum design criteria for reliable and straightforward CPU upgradability with the Future Pentium OverDrive processor for Pentium processor (510\60, 567\66)-based systems. This section specifies the criteria used to evaluate whether systems meet the criteria for the Future Pentium OverDrive processor socket specification.

The Intel Platform Support Labs establishes the minimum system design criteria for Future Pentium OverDrive processor upgradability. The details of the test criteria are described in the following sections and are divided into five key areas:

End User Installation

Mechanical Specifications

Thermal Specifications

Electrical Specifications

Functional Operation

16.8.2.1. END-USER TEST CRITERIA

One of the key aspects of that upgradability is that it be straightforward and well documented for the end user. The end user should be able to open the system, readily locate the CPU\OverDrive processor socket, install the OverDrive processor, close and boot the system, all within a matter of minutes. The criteria that follows provide guidelines to evaluate whether this has been accomplished.

16.8.2.1.1. Qualified Future Pentium® OverDrive® Processor Socket

A system design verified as part of the Intel Platform Support Labs must employ an Intel-qualified OverDrive processor socket:

- Socket 4, 273-hole ZIF (5V only).

16.8.2.1.2. Primary CPU

A system design verified as part of the Intel Platform Support Labs must be shipped with a genuine Intel Pentium processor (510\60, 567\66) as the original CPU.

16.8.2.1.3. Socket Visibility

If Socket 4 is not visible upon opening the system case, simple diagrams or other indicators should be visible upon removal of the system case or clear instructions in the users manual should guide the user to the CPU/OverDrive processor socket.

16.8.2.1.4. Socket Accessibility

It is required that the area around the ZIF socket lever be clear for unobstructed operation, to ensure proper component removal and insertion, and to reduce possible damage to nearby components during the process. Heatsink clips, when used, must be accessible and must be restrained to prevent the clip from ejecting out of the system if and when tension is released. Substantial force (greater than 10 lbs, 4.5 kg) must not be required to remove (or reinstall) a heatsink clip.

It is required that the upgrade installation not necessitate the end user to remove any cables (disk drive, power, etc.) that are shipped with the system. Removal of I/O cards from expansion slots is acceptable.

It is required that no special tools be needed to access the OverDrive processor socket or remove heatsink clips, if used and their removal is necessary for the upgrade installation process. Standard screwdrivers (Phillips, slotted) are not considered special tools. If a screwdriver is required, its use must not jeopardize other components in the system should the tool slip during the process. Torx and Allen wrenches or tools from the OEM only available after purchasing the system are considered special tools. If a special tool is necessary, it must be shipped with the system and contained inside the chassis.

16.8.2.1.5. Jumper Configuration

End-user configured jumpers are not recommended. Installation of Future Pentium OverDrive processor should not require jumper configuration. If jumper changes are required to install the Future Pentium OverDrive processor, they must be documented in the system End-User manual.

16.8.2.1.6. BIOS Changes

The system should not require a BIOS change or additional software (other than a standard system configuration utility shipped with the system) are needed to complete the upgrade. Any reconfiguration of the system must be able to be performed with the Future Pentium OverDrive processor installed.

16.8.2.1.7. Documentation

The Intel Platform Support Labs requires that system documentation contains clear and concise end-user installation instructions for the Future Pentium OverDrive processor. The system must contain the following minimum documentation:

- Installation instructions including illustrations of the system and socket location, which contain a clear description of the procedures for safe removal (and re-installation) of the OEM CPU, and operation of heatsink clips, if used. If heatsink clips are used their removal is necessary for the upgrade process (a diagram is strongly recommended).
- Pin 1 alignment information. If a heatsink clip is used and its orientation is important for proper operation of the clip, or for meeting verification requirements, this orientation must be indicated in the event that re-installation of the OEM CPU is required.

- Jumper locations and settings.
- Warnings which address:
  - Required clearance (sections 16.8.2.2.1 and 16.8.2.2.2).
  - ESD precautions.
  - Powering down the system to allow the chip to cool prior to starting the installation process.
- Loss of bus slot information due to processor upgrade (section 16.8.2.2.4) if slots are lost.
- Standard system configuration utility usage if applicable (e.g. EISA systems or systems with separate configuration utility necessary to configure system).

16.8.2.1.8. Warranty

The Future Pentium OverDrive processor installation must not affect the OEM system warranty and should be so stated in the system documentation. Labels on the CPU stating that the warranty is void if removed are not permitted.

16.8.2.1.9. Upgrade Removal

The upgrade installation process must be reversible in case of device failure or hardware and software incompatibilities discovered after verification. The end user must be able to install the original processor with no more than the complexity of the original upgrade process. Reinstallation of the original processor must return the system to its original functionality. The original CPU cooling solution must also return to its original effectiveness.

16.8.2.1.10. Full Configuration

Verification testing requires submission of all motherboard designs configured to a level that is representative of its ultimate use. As a result, all system designs must be submitted in a fully configured state, where fully configured means that all optional proprietary slots, adapters, sockets, or footprints on the motherboard are populated with the options that are intended to be sold with the system or made available after purchase of the system.

16.8.2.2. MECHANICAL SPECIFICATIONS

Mechanical specifications ensure that the Future Pentium OverDrive processor will fit properly in a system and be afforded the proper clearance for component removal/insertion and proper thermal management.

16.8.2.2.1. Vertical Clearance

The active fan/heatsink on the Future Pentium OverDrive processor requires 0.4" (1.02cm) of space above the device for proper airflow. For physical measurements, the clearance area extends from the socket surface to 1.4" (3.56cm) perpendicular to the socket surface. There can be no obstructions in this area. The system must be able to meet this specification in a

fully configured state, with the exception of documented I/O bus slot losses. (See section 16.8.2.2.4.)

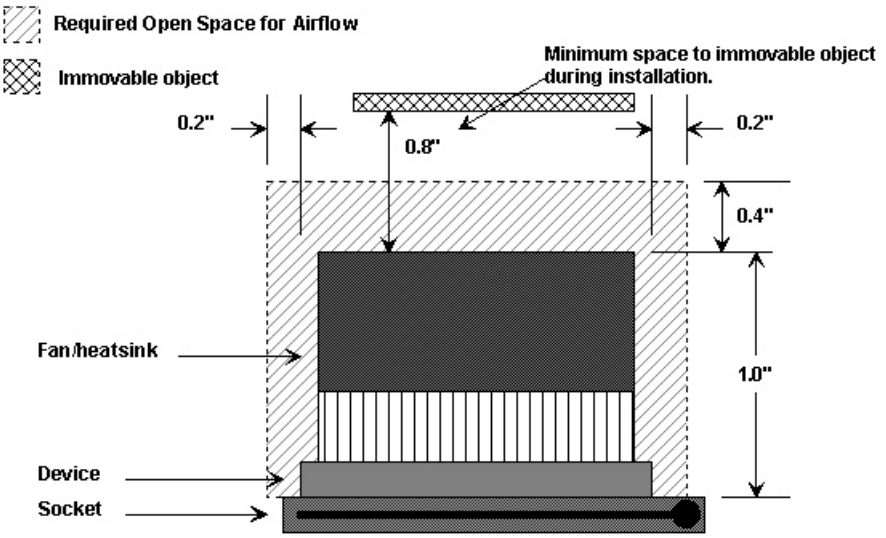


Figure 16-10. IVP Criteria — Space Requirements

In order to enable the easy installation and removal of the Future Pentium OverDrive processor, an additional 0.4" is required above the 1.4" vertical clearance requirement to provide clearance for the device pins and maneuverability. This 1.8" (4.57cm) total vertical clearance requirement applies only to immovable obstructions present during the installation process.

If there is a possibility for movable obstructions (power supply cable, floppy disk cables), the documentation must include a warning to the user to keep such obstructions clear of the component. The OEM must ensure that all system cables and free-moving parts will not become obstructions.

It is recommended that the system design should not permit an end user to install an obstruction such as an I/O card within the 0.4" airflow area. The OEM is required to document the loss of an I/O slot where an installed card would intrude into the specified clearance space .

16.8.2.2.2. Horizontal Clearance

Absolute minimum space requirement around the Future Pentium OverDrive processor is specified as 0.2" (0.51 cm) on at least three of four sides when the system is fully configured with the exception of documented I/O bus slot losses. (See section 16.8.2.2.4.) This is necessary to provide proper airflow through the active fan/heatsink.

If there is an obstruction within the 0.2" airflow area on one side of the Future Pentium OverDrive processor, the system design must not permit installation of an obstruction, such as an I/O card, within the 0.2" airflow area on any other side of the Future Pentium OverDrive processor.

If there is a possibility for movable obstructions (power supply cables, floppy disk cables, etc.), the documentation must include a warning to the user to keep such obstructions clear of the component. The OEM must ensure that all system cables and free-moving parts will not become obstructions.

16.8.2.2.3. ZIF Socket Lever Clearance

The ZIF socket lever must operate from a closed position to the fully open position without interference. If the lever action is obstructed or prevented from extending to the full open position, damage may occur to the OEM CPU or OverDrive processor during removal/installation.

16.8.2.2.4. Loss of Local Bus Slot and Expansion Slot

The Future Pentium OverDrive processor is designed to increase the performance of a system; however, installation of the OverDrive processor before installation of other devices (such as local bus graphics cards) may prevent the installation of these performance enhancing devices, thereby decreasing the overall performance of the system. To prevent this from occurring, the Intel Platform Support Labs has established the following slot loss criteria for system upgrade designs:

All Slots:

- Microchannel, EISA, and ISA slots are defined as I/O slots. VESA Local (VL) and PCI slots are defined as local bus slots.
- Installation of the Future Pentium OverDrive processor must not result in the loss of more than 50% of the available local bus or expansion slots; see Table 16-5.
- Any slots that are lost as a result of the Future Pentium OverDrive processor installation must be clearly specified in the OEM documentation. These are counted as lost slots only if the slots were usable before the upgrade.

I/O Slots:

- After installation of the Future Pentium OverDrive processor in a system with 1 or more usable full length expansion slots, a minimum of 1 usable full length I/O slots must remain.

Local Bus Slots:

- After installation of the Future Pentium OverDrive processor in a system with 1 or more usable full length local bus slots, a minimum of 1 usable full length local bus slot must remain.

Table 16-5. Minimum Number of Slots Available after OverDrive® Processor Installation

Total I/O Slots	I/O Slots Available After Upgrade	Total Local Bus Slots	Local Bus Slots Available After Upgrade
1	1	1	1
2	1	2	1
3	2	3	2
4	2	4	2
5	3	5	3
6	3	6	3
7	4	7	4
8	4	8	4

16.8.2.3. THERMAL TESTING

Thermal testing is performed to verify that the system has proper thermal management to dissipate the heat from the Future Pentium OverDrive processor. Thermal measurements should be performed with worst case thermal loading and the maximum system ambient temperature as specified by the OEM. Worst case thermal loading requires every expansion slot to be populated by the longest add-in card that will fit in each slot without violating the required clearance for airflow around the OverDrive processor specifies (see sections 16.8.2.2.1 and 16.8.2.2.2). Furthermore, the Intel Platform Support Labs requires that each add-in card should dissipate the average power for its form factor. For PCI and Microchannel add-in cards the average power dissipation is 10 Watts. For Full length and 3/4 length ISA cards the average power dissipation is also 10 Watts. For 1/2 length ISA cards the average power dissipation is 5 Watts, and the average power dissipation for 1/4 length ISA cards is 3.3 Watts. The OEM specified maximum system ambient temperature must be no less than 32°C.

The requirement for the Future Pentium OverDrive processor is that the temperature of the air entering the fan/heatsink not exceed 45°C.

16.8.2.4. ELECTRICAL TESTING

The tests performed determine whether the system design meets the specifications and characteristics of the OverDrive processor. For active measurements the system should be tested after the processor has completed the reset sequence. The system design must meet the specification under the worst case load and thermal conditions.

16.8.2.4.1. V_{CC} Continuity

All V_{CC} pins must be connected to the proper supply. Failure to connect all V_{CC} pins may cause functionality and reliability problems. All pins are checked for a resistance of less than 0.5 Ohm to the supply plane.

16.8.2.4.2. V_{SS} Continuity

All V_{SS} pins must be connected to ground. Failure to connect all V_{SS} pins to ground may cause functionality and reliability problems. All pins are checked for a resistance of less than 0.5 Ohm to the ground plane.

16.8.2.4.3. V_{CC} Specification

V_{CC} must meet the Future Pentium OverDrive processor V_{CC} specifications under any operating condition.

Table 16-6. Minimum and Maximum V_{CC} Specifications Across Loading Conditions

Loading Conditions	Minimum V_{CC}	Maximum V_{CC}
Steady State Conditions for 60mhz System Bus Maximum (2.91A) and Minimum (0A) Icc Load	4.75V	5.25V
Steady State Conditions for 66mhz System Bus Maximum (3.2A) and Minimum (0A) Icc Load	4.9V	5.4V
Worst Case Load Switch for 60mhz System Bus 200mA to 2.91A within 200nS 2.91A to 200mA within 200ns	4.75V	5.67V (V_{CC} max + 5%)
Worst Case Load Switch for 66mhz System Bus 200mA to 3.2A within 200nS 3.2A to 200mA within 200ns	4.75V	5.67V (V_{CC} max + 5%)

16.8.2.4.4. No Connect (NC) and Reserve Pins

There must be no connections to the NC and Reserve pins. Failure to comply may cause problems due to reserved test features of the Future Pentium OverDrive Processor. All NC and Reserve pins are checked for conformity to specification.

16.8.2.4.5. Signal Usage

There are differences in pin usage between the 5V Pentium processor (510\60, 567\66) and the Future Pentium OverDrive Processor, specifically, the IU, IV and IBT pins. System designs must not depend on identical behavior (on these pins) to the 5V Pentium processor (510\60, 567\66).

16.8.2.4.6. Cold RESET Pulse Width

During cold system boot, the RESET signal must be active for a minimum of 1mS to properly synchronize the internal phase-locked loop. RESET is measured after V_{CC} and CLK reach valid levels. Failure to meet specification may cause the Future Pentium OverDrive processor to malfunction.

16.8.2.4.7. Warm RESET Pulse Width

If RESET is pulsed for warm reset (instead of INIT), the RESET signal must be active for a minimum of 15 clocks. Failure to meet specification may cause the Future Pentium OverDrive processor to malfunction. If INIT is used, it must meet setup and hold times to the CLK signal.

16.8.2.4.8. CLK Waveform

The input clock waveforms must meet the specifications for the Future Pentium OverDrive processor to work properly. Refer to Chapter 7 for the CLK specifications.

16.8.2.4.9. Input Signal Quality

Input signals to the processor must meet the electrical specifications of the Pentium processor (510\60, 567\66). Furthermore, input signals to the processor on input or bi-directional pins must meet the ringback specifications for the Pentium processor (510\60, 567\66). The ringback of the input signal as defined in Chapter 7 must not exceed 0.8V.

16.8.2.4.10. CPUID Instruction

The CPUID instruction allows software like the PC BIOS, to determine the type and features of the microprocessor on which it is executing.

When executing CPUID, the Future Pentium OverDrive Processor behaves like the Pentium Processor:

If the value in EAX is '0' then the 12-byte ASCII string "GenuineIntel" (little endian) is returned in EBX, EDX, and ECX. Also, a '1' is returned to EAX.

If the value in EAX is '1' then the processor version is returned in EAX and the processor capabilities are returned in EDX. The values of EAX and EDX for the Future Pentium OverDrive Processor are given below.

If the value in EAX is neither '0' nor '1', the Future Pentium OverDrive Processor writes '0' to all registers.

The following EAX and EDX values are defined for the CPUID instruction executed with EAX = '1'.

The stepping fields has the same format as the Pentium processor's and will be the same for the Future Pentium OverDrive Processor. The Pentium Processor (510\60, 567\66) is CPUID = 051xH and the Future Pentium OverDrive Processor is CPUID = 152xH. The type field is defined as follows:

Table 16-7. EAX Bit Values Definition for CPUID

CPU	31.....14	13.12	11...8	7...4	3...0
Field Definition	(reserved)	type	family	model	stepping
Pentium® Processor (567\66, 510\60)	(reserved)	table	5H	1H	(varies)
Future Pentium OverDrive® Processor	(reserved)	table	5H	2H	(varies)

Table 16-8. EAX Bit Values Definition for Processor Type

Bit 13	Bit 12	Processor Type
0	0	Primary Pentium® Processor
0	1	Future Pentium OverDrive® Processor
1	0	Dual Pentium Processor *
1	1	Reserved

Note: * The Future Pentium® OverDrive® Processor does not support Dual mode.

After masking the reserve bits, the Future Pentium OverDrive Processor will have a value of 0x000001BF in the EDX (when APIC is disabled, using APICEN boot pin).

16.8.2.5. FUNCTIONAL TESTING

Functional testing verifies whether the hardware and firmware are likely to operate properly with the Future Pentium OverDrive processor. The purpose of the functional testing is to uncover hardware or BIOS problems that may exist when the Future Pentium OverDrive processor is installed. Areas of testing include checking for BIOS timing loops that may not behave properly when executed at a significantly higher speed, CPU ID recognition, test register usage, etc.

16.8.2.5.1. AC Timing Specifications

The system must meet all AC timing specifications as documented in Chapter 7.

16.8.2.5.2. Error-Free Boot

The system must boot properly without error messages when the Future Pentium OverDrive processor is installed. There should be no new error messages, beeps or other indications of anomalous behavior. An exception to this is a message declaring the presence of an "OverDrive processor." If the system displays the CPU type, then it must display it correctly; otherwise, the CPU type should not be displayed at all.

16.8.2.5.3. Software Compatibility

The system hardware and configuration must not cause any software that operates properly with the host CPU to not function properly with the OverDrive processor.

16.8.2.5.4. No 5V CMOS Drive Levels

Since the OverDrive processor, unlike the Pentium processor (510\60, 567\66), does not support 5V CMOS drive levels, subsystems which require 5V CMOS drive level inputs (such as the 82496 cache controller) can *not* be connected to the processor.

16.8.3. Submitting a System to Intel Platform Support Labs

To submit a system design to Intel Platform Support Labs, first obtain a SystemDesign Checklist from your local Intel field sales representative or Intel Faxback (see Section 16.8.1.1 for Faxback document information). Next, throughly complete the System Design Checklist. Send the completed System Design Checklist along with the fully configured system design to Intel Platform Support Lab nearest to your geographical location.

Part II

Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133)

17

Pinout

CHAPTER 17
PINOUT

The physical pinouts of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), and the Pentium processor (510\60, 567\66) are different. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses a 296-pin Staggered PGA package, while the Pentium processor (510\60, 567\66) uses a 273-pin PGA.

17.1. PINOUT AND CROSS REFERENCE TABLES

The text orientation on the top side view drawings in this section represents the orientation of the ink mark on the actual packages (Note that the text shown in this section is not the actual text which will be marked on the packages).

17.1.1. Pinout

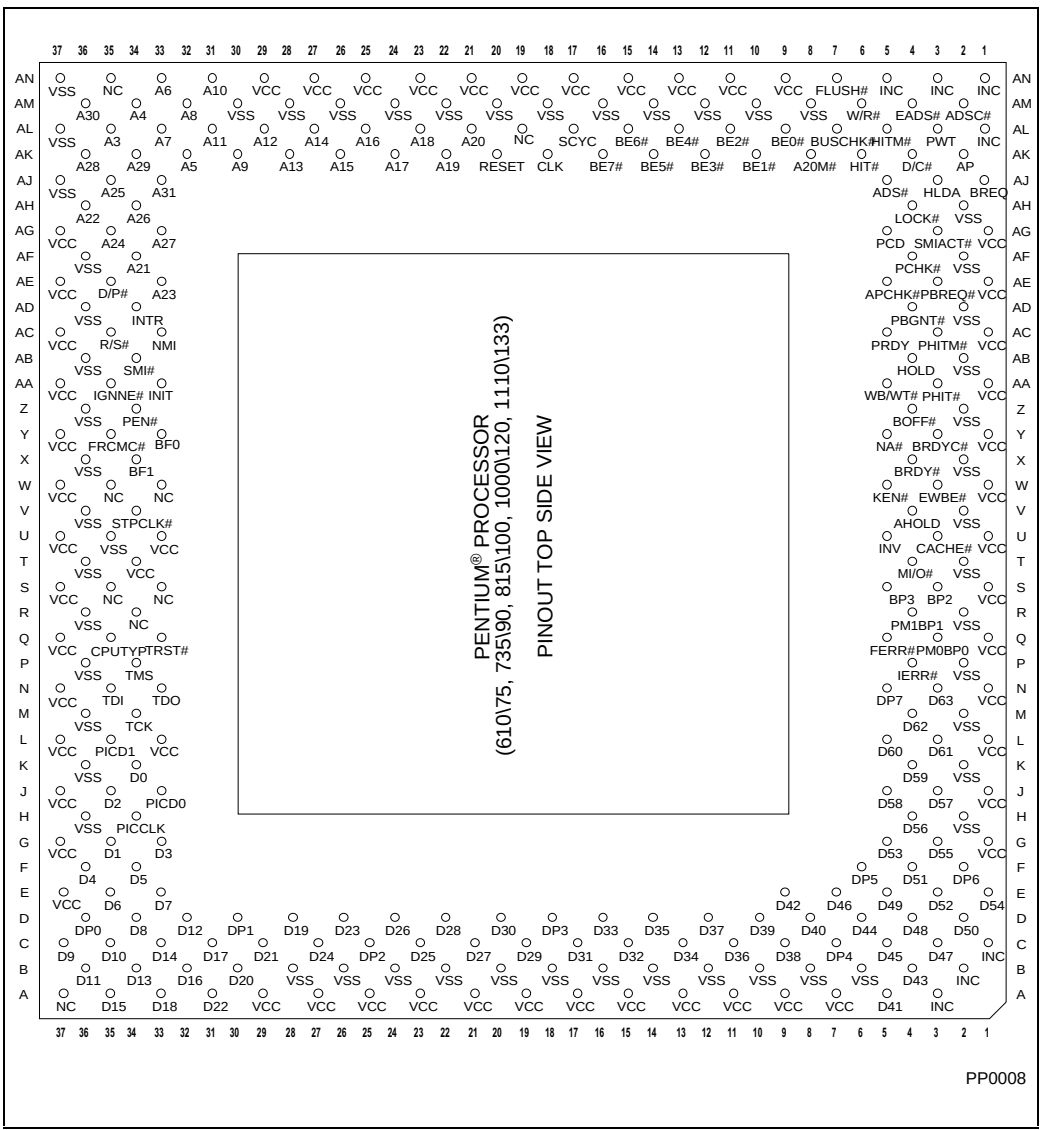


Figure 17-1. Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Pinout — Top Side View

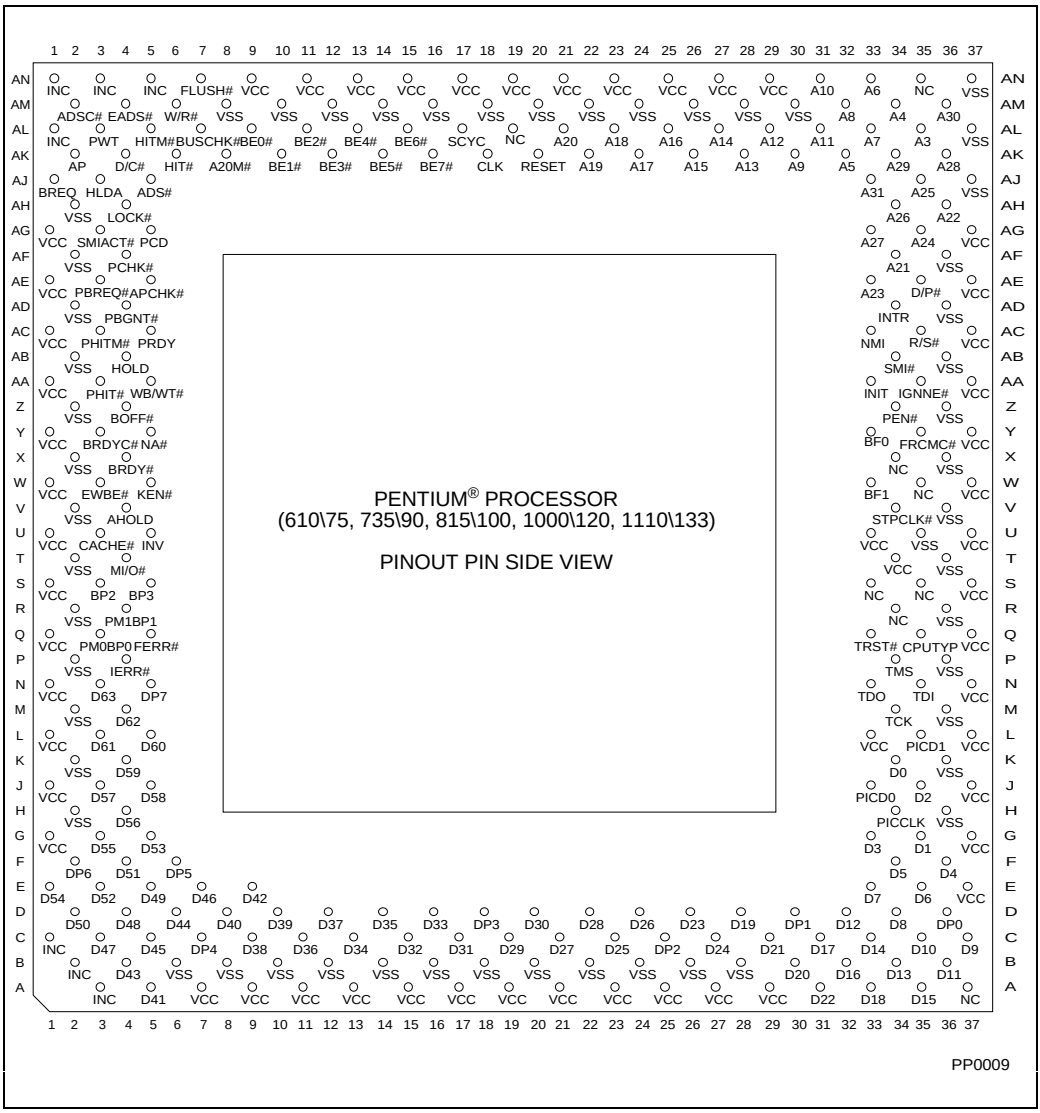


Figure 17-2. Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Pinout — Pin Side View

17.1.2. Pin Cross Reference Table

Table 17-1. Pin Cross Reference by Pin Name

Address									
A3	AL35	A9	AK30	A15	AK26	A21	AF34	A27	AG33
A4	AM34	A10	AN31	A16	AL25	A22	AH36	A28	AK36
A5	AK32	A11	AL31	A17	AK24	A23	AE33	A29	AK34
A6	AN33	A12	AL29	A18	AL23	A24	AG35	A30	AM36
A7	AL33	A13	AK28	A19	AK22	A25	AJ35	A31	AJ33
A8	AM32	A14	AL27	A20	AL21	A26	AH34		
Data									
D0	K34	D13	B34	D26	D24	D39	D10	D52	E03
D1	G35	D14	C33	D27	C21	D40	D08	D53	G05
D2	J35	D15	A35	D28	D22	D41	A05	D54	E01
D3	G33	D16	B32	D29	C19	D42	E09	D55	G03
D4	F36	D17	C31	D30	D20	D43	B04	D56	H04
D5	F34	D18	A33	D31	C17	D44	D06	D57	J03
D6	E35	D19	D28	D32	C15	D45	C05	D58	J05
D7	E33	D20	B30	D33	D16	D46	E07	D59	K04
D8	D34	D21	C29	D34	C13	D47	C03	D60	L05
D9	C37	D22	A31	D35	D14	D48	D04	D61	L03
D10	C35	D23	D26	D36	C11	D49	E05	D62	M04
D11	B36	D24	C27	D37	D12	D50	D02	D63	N03
D12	D32	D25	C23	D38	C09	D51	F04		

Table 17-1. Pin Cross Reference by Pin Name (Contd.)

Address							
A20M#	AK08	BRDYC#	Y03	FLUSH#	AN07	PEN#	Z34
ADS#	AJ05	BREQ	AJ01	FRCMC#	Y35	PM0/BP0	Q03
ADSC#	AM02	BUSCHK#	AL07	HIT#	AK06	PM1/BP1	R04
AHOLD	V04	CACHE#	U03	HITM#	AL05	PRDY	AC05
AP	AK02	CPUTYP	Q35	HLDA	AJ03	PWT	AL03
APCHK#	AE05	D/C#	AK04	HOLD	AB04	R/S#	AC35
BE0#	AL09	D/P#	AE35	IERR#	P04	RESET	AK20
BE1#	AK10	DP0	D36	IGNNE#	AA35	SCYC	AL17
BE2#	AL11	DP1	D30	INIT	AA33	SMI#	AB34
BE3#	AK12	DP2	C25	INTR/LINT0	AD34	SMIACT#	AG03
BE4#	AL13	DP3	D18	INV	U05	TCK	M34
BE5#	AK14	DP4	C07	KEN#	W05	TDI	N35
BE6#	AL15	DP5	F06	LOCK#	AH04	TDO	N33
BE7#	AK16	DP6	F02	M/IO#	T04	TMS	P34
BOFF#	Z04	DP7	N05	NA#	Y05	TRST#	Q33
BP2	S03	EADS#	AM04	NMI/LINT1	AC33	W/R#	AM06
BP3	S05	EWBE#	W03	PCD	AG05	WB/WT#	AA05
BRDY#	X04	FERR#	Q05	PCHK#	AF04		
APIC		**Clock Control**		**Dual Processor Private Interface**			
PICCLK	H34	CLK	AK1 8	PBGNT#	AD04		
PICD0	J33	BF0	Y33	PBREQ#	AE03		
[DPEN#]		BF1	Y34				
		STPCLK#	V34	PHIT#	AA03		
PICD1	L35			PHITM#	AC03		
[APICEN]							

Table 17-1. Pin Cross Reference by Pin Name (Contd.)

V_{CC}							
A07	A21	G37	N37	U33	AA37	AN09	AN23
A09	A23	J01	Q01	U37	AC01	AN11	AN25
A11	A25	J37	Q37	W01	AC37	AN13	AN27
A13	A27	L01	S01	W37	AE01	AN15	AN29
A15	A29	L33	S37	Y01	AE37	AN17	
A17	E37	L37	T34	Y37	AG01	AN19	
A19	G01	N01	U01	AA01	AG37	AN21	
V_{SS}							
B06	B20	K02	R36	X36	AF02	AM12	AM26
B08	B22	K36	T02	Z02	AF36	AM14	AM28
B10	B24	M02	T36	Z36	AH02	AM16	AM30
B12	B26	M36	U35	AB02	AJ37	AM18	AN37
B14	B28	P02	V02	AB36	AL37	AM20	
B16	H02	P36	V36	AD02	AM08	AM22	
B18	H36	R02	X02	AD36	AM10	AM24	
NC/INC							
A03	AL01	AN01	AN05	B02	R34	S35	W35
A37	AL19	AN03	AN35	C01	S33	W33	

17.2. DESIGN NOTES

For reliable operation, always connect unused inputs to an appropriate signal level. Unused active low inputs should be connected to V_{CC}. Unused active HIGH inputs should be connected to GND (V_{SS}).

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

17.3. QUICK PIN REFERENCE

This section gives a brief functional description of each of the pins. **Note that all input pins must meet their AC/DC specifications to guarantee proper functional behavior.**

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. Square brackets around a signal name indicate that the signal is defined only at RESET. See Chapter 23 for the timing requirements of these signals.

The following pins exist on the Pentium processor (510\60, 567\66) but have been removed from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133):

- IBT, IU, IV, BT0-3

The following pins become I/O pins when two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating in a dual processing environment:

- ADS#, CACHE#, HIT#, HITM#, HLDA#, LOCK#, M/IO, D/C#, W/R#, SCYC

Please refer to Chapter 32 for information how to connect the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) pins if an upgrade socket is designed in the system.

Table 17-2. Quick Pin Reference

Symbol	Type*	Name and Function
A20M#	I	When the **address bit 20 mask** pin is asserted, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) emulates the address wraparound at 1 Mbyte which occurs on the 8086. When A20M# is asserted, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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. A20M# is internally masked by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) when configured as a Dual processor.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) .
ADSC#	O	ADSC# is functionally identical to ADS#.
AHOLD	I	In response to the assertion of **address hold**, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) with even parity information on all Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) generated cycles in the same clock that the address is driven. Even parity must be driven back to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) during inquire cycles on this pin in the same clock as EADS# to ensure that correct parity check status is indicated by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).
APCHK#	O	The **address parity check** status pin is asserted two clocks after EADS# is sampled active if the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (including during dual processing private snooping).
[APICEN] PICD1	I	**Advanced Programmable Interrupt Controller enable** is a new pin that enables or disables the on-chip APIC interrupt controller. If sampled high at the falling edge of RESET, the APIC is enabled. APICEN shares a pin with the **Programmable Interrupt Controller Data 1** signal.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
BE7#-BE5# BE4#-BE0#	O I/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). Unlike the Pentium processor (510\60, 567\66), the lower four byte enables (BE3#-BE0#) are used on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) as APIC ID inputs and are sampled at RESET. After RESET, these behave exactly like the Pentium processor (510\60, 567\66) byte enables. In dual processing mode, BE4# is used as an input during flush cycles.
BF[1:0]	I	**Bus Frequency** determines the bus-to-core frequency ratio. BF[1:0] is sampled at RESET, and cannot be changed until another non-warm (1 ms) assertion of RESET. Additionally, BF[1:0] must not change values while RESET is active. For proper operation of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) these pins should be strapped high or low. When BF0 is strapped to V_{CC}, the processor will operate at a 2/3 bus/core frequency ratio. When BF0 is strapped to V_{SS}, the processor will operate at a 1/2 bus/core frequency ratio. If BF0 is left floating, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) defaults to a 2/3 bus ratio. Note the Pentium processor (610\75, 735\90) will not operate at a 1/2 bus/core frequency ratio. For descriptions on BF1 and more detailed information, please refer to section 21.2.7 of this document.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) restarts the aborted bus cycle(s) in their entirety.
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.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) data in response to a write request. This signal is sampled in the T2, T12 and T2P bus states.
BRDYC#	I	This signal has the same functionality as BRDY#.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
BREQ	O	The **bus request** output indicates to the external system that the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has internally generated a bus request. This signal is always driven whether or not the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is driving its bus.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will vector to the machine check exception. **NOTE:** To assure that the BUSCHK# will always be recognized, STPCLK# must be deasserted any time BUSCHK# is asserted by the system, before the system allows another external bus cycle. If BUSCHK# is asserted by the system for a snoop cycle while STPCLK# remains asserted, usually (if MCE = 1) the processor will vector to the exception after STPCLK# is deasserted. But if another snoop to the same line occurs during STPCLK# assertion, the processor can lose the BUSCHK# request.
CACHE#	O	For Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133)-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, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133). Its frequency is the operating frequency of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) external bus and requires TTL levels. All external timing parameters except TDI, TDO, TMS, TRST#, and PICD0-1 are specified with respect to the rising edge of CLK. **Note: It is recommended that CLK begin toggling within 150ms after V_{CC} reaches its proper operating level. This recommendation is only to ensure long-term reliability of the device.**
CPUTYP	I	**CPU type** distinguishes the Primary processor from the Dual processor. In a single processor environment, or when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is acting as the Primary processor in a dual processing system, CPUTYP should be strapped to V_{SS}. The Dual processor should have CPUTYP strapped to V_{CC}. For the future Pentium OverDrive processor, CPUTYP will be used to determine whether the bootup handshake protocol will be used (in a dual socket system) or not (in a single socket system).
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.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
D/P#	O	The **dual / primary** processor indication. The Primary processor drives this pin low when it is driving the bus, otherwise it drives this pin high. D/P# is always driven. D/P# can be sampled for the current cycle with ADS# (like a status pin). This pin is defined only on the Primary processor. **Dual processing is supported in a system only if both processors are operating at identical core and bus frequencies. Within these restrictions, two processors of different steppings may operate together in a system.**
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.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) with even parity information on writes in the same clock as write data. Even parity information must be driven back to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) on these pins in the same clock as the data to ensure that the correct parity check status is indicated by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). DP7 applies to D63-56, DP0 applies to D7-0.
[DPEN#] PICD0	I/O	**Dual processing enable** is an output of the Dual processor and an input of the Primary processor. The Dual processor drives DPEN# low to the Primary processor at RESET to indicate that the Primary processor should enable dual processor mode. DPEN# may be sampled by the system at the falling edge of RESET to determine if Socket 5 is occupied. DPEN# shares a pin with PICD0.
EADS#	I	This signal indicates that a valid **external address** has been driven onto the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) generates a write, and EWBE# is sampled inactive, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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. FERR# is never driven active by the Dual processor.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
FLUSH#	I	When asserted, the **cache flush** input forces the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) indicating completion of the writeback and invalidation. If FLUSH# is sampled low when RESET transitions from high to low, tristate test mode is entered. If two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating in dual processing mode in a system and FLUSH# is asserted, the Dual processor will perform a flush first (without a flush acknowledge cycle), then the Primary processor will perform a flush followed by a flush acknowledge cycle.
FRCMC#	I	The **functional redundancy checking master/checker** mode input is used to determine whether the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is configured in master mode or checker mode. When configured as a master, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) drives its output pins as required by the bus protocol. When configured as a checker, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) data or instruction cache, this pin is asserted two clocks after EADS# is sampled asserted. If the inquire cycle misses the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will resume driving the bus. If the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has a bus cycle pending, it will be driven in the same clock that HLDA is de-asserted.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
HOLD	I	In response to the **bus hold request**, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will float most of its output and input/output pins and assert HLDA after completing all outstanding bus cycles. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will maintain its bus in this state until HOLD is de-asserted. HOLD is not recognized during LOCK cycles. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will recognize HOLD during reset.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will assert the IERR# pin for one clock and then shutdown. If the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will stop execution and wait for an external interrupt. IGNNE# is internally masked when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is configured as a Dual processor.
INIT	I	The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) **initialization** input pin forces the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will perform built-in self test prior to the start of program execution.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
INTR / LINT0	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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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. If the local APIC is enabled, this pin becomes **local interrupt 0**.
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.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) generates a cycle that can be cached (CACHE# asserted) and KEN# is active, the cycle will be transformed into a burst line fill cycle.
LINT0/INTR	I	If the APIC is enabled, this pin is **local interrupt 0**. If the APIC is disabled, this pin is **interrupt**.
LINT1/NMI	I	If the APIC is enabled, this pin is **local interrupt 1**. If the APIC is disabled, this pin is **non-maskable interrupt**.
LOCK#	O	The **bus lock** pin indicates that the current bus cycle is locked. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will issue ADS# for a pending cycle two clocks after NA# is asserted. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports up to 2 outstanding bus cycles.
NMI/LINT1	I	The **non-maskable interrupt** request signal indicates that an external non-maskable interrupt has been generated. If the local APIC is enabled, this pin becomes **local interrupt 1**.
PBGNT#	I/O	**Private bus grant** is the grant line that is used when two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are configured in dual processing mode, in order to perform private bus arbitration. PBGNT# should be left unconnected if only one Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) exists in a system.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
PBREQ#	I/O	**Private bus request** is the request line that is used when two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are configured in dual processing mode, in order to perform private bus arbitration. PBREQ# should be left unconnected if only one Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) exists in a system.
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. When two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating in dual processing mode, PCHK# may be driven two or three clocks after BRDY# is returned.
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 (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) will vector to the machine check exception before the beginning of the next instruction.
PHIT#	I/O	**Private hit** is a hit indication used when two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are configured in dual processing mode, in order to maintain local cache coherency. PHIT# should be left unconnected if only one Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) exists in a system.
PHITM#	I/O	**Private modified hit** is a hit indication used when two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are configured in dual processing mode, in order to maintain local cache coherency. PHITM# should be left unconnected if only one Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) exists in a system.
PICCLK	I	The APIC interrupt controller serial data bus clock is driven into the **programmable interrupt controller clock** input of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).
PICD0-1 [DPEN#] [APICEN]	I/O	**Programmable interrupt controller data lines 0-1** of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) comprise the data portion of the APIC 3-wire bus. They are open-drain outputs that require external pull-up resistors. These signals share pins with DPEN# and APICEN.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
PM/BP[1:0]	O	These pins function as part of the performance monitoring feature. The breakpoint 1-0 pins are multiplexed with the **performance monitoring 1-0** pins. 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.
PRDY	O	The **probe ready** output pin indicates that the processor has stopped normal execution in response to the R/S# pin going active, or Probe Mode being entered.
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 **run / stop** 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.
RESET	I	**RESET** forces the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to begin execution at a known state. All the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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).
STPCLK#	I	Assertion of the **stop clock** input signifies a request to stop the internal clock of the Pentium processor (610\75, 735\90,815\100, 1000\120) thereby causing the core to consume less power. When the CPU recognizes STPCLK#, the processor will stop execution on the next instruction boundary, unless superseded by a higher priority interrupt, and generate a stop grant acknowledge cycle. When STPCLK# is asserted, the Pentium processor (610\75, 735\90,815\100, 1000\120) will still respond to interprocessor and external snoop requests.
TCK	I	The **testability clock** input provides the clocking function for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) during boundary scan.

Table 17-2. Quick Pin Reference (Contd.)

Symbol	Type*	Name and Function
TDI	I	The **test data input** is a serial input for the test logic. TAP instructions and data are shifted into the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 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 (610\75, 735\90, 815\100, 1000\120, 1110\133) on the TDO pin on TCK's falling edge 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.
V_{CC}	I	The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has 53 3.3V **power** inputs.
V_{SS}	I	The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has 53 **ground** inputs.
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. Refer to Chapter 21 for further information.

17.4. PIN REFERENCE TABLES

Table 17-3. Output Pins

Name	Active Level	When Floated
ADS#*	Low	Bus Hold, BOFF#
ADSC#	Low	Bus Hold, BOFF#
APCHK#	Low	
BE7#-BE5#	Low	Bus Hold, BOFF#
BREQ	High	
CACHE#*	Low	Bus Hold, BOFF#
D/P#**	n/a	
FERR#**	Low	
HIT#*	Low	
HITM#*	Low	
HLDA*	High	
IERR#	Low	
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

NOTES:

All output pins are floated during tristate test mode (except TDO) and checker mode (except IERR# and TDO). There are two pins that have pullups attached during dual processor mode, HIT# and HITM#. These pins are pulled high during tri-state test mode. The pull-up on HITM# has a value of about 30K ohms, HIT# is about 2K ohms.

* These signals are Input/Output when two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in Dual Processing Mode.

** These signals are undefined when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is configured as a Dual processor.

Table 17-4. Input Pins

Name	Active Level	Synchronous/ Asynchronous	Internal Resistor	Qualified
A20M#*	Low	Asynchronous		
AHOLD	High	Synchronous		
BF1-0	High	Synchronous/RESET	Pullup	
BOFF#	Low	Synchronous		
BRDY#	Low	Synchronous	Pullup	Bus State T2,T12,T2P
BRDYC#	Low	Synchronous	Pullup	Bus State T2,T12,T2P
BUSCHK#	Low	Synchronous	Pullup	BRDY#
CLK	n/a			
CPUTYP	High	Synchronous/RESET		
EADS#	Low	Synchronous		
EWBE#	Low	Synchronous		BRDY#
FLUSH#	Low	Asynchronous		
FRCMC#	Low	Asynchronous	Pullup	
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		
PICCLK	High	Asynchronous	Pullup	
PEN#	Low	Synchronous		BRDY#
R/S#	n/a	Asynchronous	Pullup	
RESET	High	Asynchronous		
SMI#	Low	Asynchronous	Pullup	
STPCLK#	Low	Asynchronous	Pullup	

Table 17-4. Input Pins (Contd.)

Name	Active Level	Synchronous/ Asynchronous	Internal Resistor	Qualified
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#

NOTE:

*Undefined when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is configured as a Dual processor.

Table 17-5. Input/Output Pins

Name	Active Level	When Floated	Qualified (When an Input)	Internal Resistor
A31-A3	n/a	Address hold, Bus Hold, BOFF#	EADS#	
AP	n/a	Address hold, Bus Hold, BOFF#	EADS#	
BE4#-BE0#	Low	Bus Hold, BOFF#	RESET	Pulldown*
D63-D0	n/a	Bus Hold, BOFF#	BRDY#	
DP7-DP0	n/a	Bus Hold, BOFF#	BRDY#	
PICD0[DPEN#]				Pullup
PICD1[APICEN]				Pulldown

NOTES:

* BE4#-BE0# have Pulldowns during RESET only.

All output and input/output pins are floated during tristate test mode (except TDO) and checker mode (except IERR# and TDO). Two pins, PICD0 and PICD1 have a pullup and a pulldown respectively in dual processor mode. These pins are pulled high and low respectively during tri-state test mode.

Table 17-6. Interprocessor I/O Pins

Name	Active Level	Internal Resistor
PHIT#	Low	Pullup
PHITM#	Low	Pullup
PBGNT#	Low	Pullup
PBREQ#	Low	Pullup

NOTES: For proper inter-processor operation, the system cannot load these signals

PHIT#, PHITM#, PBGNT# and PBREQ# have pullups in dual processor mode. These pins are pulled high during tri-state test mode. The pullups on these pins have a value of about 30K ohms.

Table 17-7 organizes the pins with respect to their function.

Table 17-7. Pin Functional Grouping

Function	Pins
Clock	CLK
Initialization	RESET, INIT, BF1-0
Address Bus	A31-A3, BE7# - BE0#
Address Mask	A20M#
Data Bus	D63-D0
Address Parity	AP, APCHK#
APIC Support	PICCLK, PICD0-1
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#, ADSC#, BRDY#, BRDYC#, 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
Dual Processing Private Bus Control	PBGNT#, PBREQ#, PHIT#, PHITM#
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
Clock Control	STPCLK#
Miscellaneous Dual Processing	CPUTYP, D/P#
Probe Mode	R/S#, PRDY

18

Features of Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133)

CHAPTER 18
Features of Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133)

The Pentium processor family consists of the new Pentium processor at iCOMP® index 610\75 MHz, iCOMP index 735\90 MHz, iCOMP index 815\100 MHz and iCOMP index 1000\120 MHz, and iCOMP index 1110\133 MHz (product order code 80502), described in this document, and the original Pentium processor (510\60, 567\66) (order code 80501). The name Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will be used in this document to refer to the Pentium processor at iCOMP index 610\75 MHz, iCOMP index 735\90 MHz, iCOMP index 815\100 MHz, iCOMP index 1000\120, and iCOMP index 1110\133 MHz. Also, the name "Pentium processor (510\60, 567\66)" will be used to refer to the original 60- and 66-MHz version product.

In addition to the architecture described in Chapter 2 for the Pentium processor family, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) offers the following enhancements over the Pentium processor (510\60, 567\66):

- iCOMP performance rating of 1110 at 133 MHz in single processor configuration
- iCOMP performance rating of 1000 at 120 MHz in single processor configuration
- iCOMP performance rating of 815 at 100 MHz in single processor configuration
- iCOMP performance rating of 735 at 90 MHz in single processor configuration
- iCOMP performance rating of 610 at 75 MHz in single processor configuration
- Dual processing support
- SL power management features
- Fractional bus operation
- On-chip local APIC device

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) offers higher performance and higher operating frequencies than the Pentium processor (510\60, 567\66).

Pentium® Processor Core Frequency	External Bus Interface	iCOMP® Index
133 MHz	66 MHz	1110
120 MHz	60 MHz	1000
100 MHz	66/50 MHz	815
90 MHz	60 MHz	735
75 MHz	50 MHz	610

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) includes new features to support multi-processor systems, namely an on-chip Advanced Programmable Interrupt Controller (APIC). This APIC implementation supports multiprocessor interrupt management (with symmetric interrupt distribution across all processors), multiple I/O subsystem support, 8259A compatibility, and inter-processor interrupt support.

The dual processor configuration allows two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to share a single L2 cache for a low-cost symmetric multi-processor system. The two processors appear to the system as a single Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). Multiprocessor operating systems properly schedule computing tasks between the two processors. This scheduling of tasks is transparent to software applications and the end-user. Logic built into the processors support a "glueless" interface for easy system design. Through a private bus, the two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) arbitrate for the external bus and maintain cache coherency. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) can also be used in a conventional multi-processor system in which one L2 cache is dedicated to each processor.

In this document, in order to distinguish between two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) in dual processing mode, one CPU will be designated as the Primary processor with the other being the Dual processor. Note that this is a different concept than that of "master" and "checker" processors described in the discussion on functional redundancy.

Dual processing is supported in a system only if both processors are operating at identical core and bus frequencies. Within these restrictions, two processors of different steppings may operate together in a system.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is produced on a 3.3V BiCMOS process. 3.3V operation results in much lower power dissipation in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) as compared to the Pentium processor (510\60, 567\66). The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) also includes SL enhanced power management features. When the clock to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is stopped, power dissipation is virtually eliminated. The combination of 3.3V operation and SL enhanced power management features makes the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) a good choice for energy-efficient desktop designs.

Supporting an upgrade socket (Socket 5/Socket 7) in the system will provide end-user upgradability by the addition of a future Pentium OverDrive processor. Typical applications will realize a 40% to 70% performance increase by addition of a Future Pentium OverDrive processor.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports fractional bus operation. This allows the internal processor core to operate at high frequencies, while communicating with the external bus at lower frequencies. The Pentium processor 1110\133 and 1000\120 support a 1/2 bus/core frequency ratio (a 66-MHz bus frequency for a 133-MHz core, and a 60-MHz bus frequency for a 120-MHz core frequency), the Pentium processor 735\90 supports a 2/3 bus/core frequency ratio (a 60-MHz bus frequency for a 90-MHz core frequency), the Pentium processor 610\75 supports a 2/3 bus/core frequency ratio

(a 50-MHz bus frequency for a 75-MHz core frequency) while the Pentium processor 815\100 supports a selectable bus/core frequency ratio of 2/3 (66-MHz bus, 100-MHz core) or 1/2 (50-MHz bus, 100-MHz core).

18.1. PROCESSOR OBJECTIVES

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is the highest performance member of the Pentium Processor family. Intel has designed the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to allow new systems to be designed from Pentium processor (510\60, 567\66) systems with few changes needed to the memory bus. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses the Pentium processor (510\60, 567\66) as a core, so both parts behave similarly internally and on the bus.

A new package allows sufficient pins to support new features, and to allow further proliferations in the future.

Intel has designed the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) on a 3.3V process for higher performance and lower power consumption. The power pins and I/O of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are 3.3V. Pentium processor (510\60, 567\66) systems must accommodate this change to use a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

The Primary and Dual processors are designed to operate together "gluelessly" and behave like a single processor. Although both processors are operating simultaneously and are sharing the same bus, they behave like a single Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

18.2. PENTIUM® PROCESSOR (610\75, 735\90, 815\100, 1000\120, 1110\133) DIFFERENCES FROM PENTIUM PROCESSOR (510\60, 567\66)

This section describes the major hardware and software differences between the Pentium processor (510\60, 567\66) and Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to help modify a Pentium processor (510\60, 567\66) system to accommodate a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). This is not a list of features. Each difference has a *Description* and *Implications* section. The differences are summarized below:

- Pinout and Package
- CPUTYP Pin
- Up to 133 MHz internal core frequency
- Selectable frequency CPU bus (Pentium processor 815\100)
- 3.3V power supply and I/O

- Lower power dissipation
- Dual-processor and Socket 5 support
- APIC interrupt controller
- Power management features
- Breakpoint pins
- Hi-Z TAP instruction
- Bus cycles
- HOLD latency
- Interrupt priorities

18.2.1. Pinout and Package

The signals on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and Pentium processor (510\60, 567\66) are almost identical. The V_{CC} pins of the Pentium processor (510\60, 567\66) are now 3.3V supply inputs. A small subset of other pins on the Pentium processor (510\60, 567\66) have been defined differently for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to enable new features. The physical pinouts of the package are different, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses a 296-pin SPGA package, while the Pentium processor (510\60, 567\66) uses a 273-lead PGA. See Chapter 17 for details.

Implications

The different pinout results in a different layout from a Pentium processor (510\60, 567\66)-based system. There are several added features that would require the same redesign effort to make those features accessible.

18.2.2. CPUTYP Pin

The CPUTYP pin is a new configuration signal which, when sampled by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) at the falling edge of RESET, indicates the type of OEM processor which will be placed in each socket site.

For the future Pentium OverDrive processor, CPUTYP will be used to determine whether the bootup handshake protocol will be used (in a dual socket system) or not (in a single socket system).

Implications

CPUTYP must be strapped to either V_{CC} or V_{SS}, depending upon one or two sockets and which socket site.

18.2.3. Up To 133-MHz Internal Core Frequency

The multiplied Internal Core frequency allows the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to execute instructions faster than the Pentium processor (510\60, 567\66). The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) multiplies the input clock by a factor of 1.5 or 2 depending on the component, and uses this as the internal clock. This internal clock can operate up to 133 MHz.

Implications

This increased frequency requires no hardware or software modifications, but will cause memory bus utilization to be increased.

18.2.4. Fractional Speed Bus

The Pentium processor 815\100 uses the BF pin to select the ratio of the CPU bus/core speeds. The CLK pin inputs either a 50-MHz or a 66-MHz CPU bus speed. The BF pin is used to select a bus to core ratio of either 1/2 or 2/3. This is explained further in Chapter 19.

Implications

No hardware modifications are necessary. If the BF pin is left floating, like in a system designed around the Pentium processor (510\60, 567\66) pinout, the Pentium processor (610\75, 735\90, 815\100) assumes a 2/3 bus/core ratio. If system software expects a certain CPU frequency or relationship between internal and external frequencies, that software may not function properly. The Pentium processor (1000\120, 1110\133) only support a 1/2 bus/core ratio. Hence the BF pin should be pulled low in this case.

18.2.5. 3.3V and 5V Power Supply

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) requires a voltage range of 3.135 to 3.6 on all of its V_{CC} inputs. This causes the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to have input and output levels of 3.3V. However, two clock inputs are 5V safe, CLK and PICCLK. See Chapter 23 for more information.

Implications

The power plane or traces that supply power to the CPU must supply 3.3V. In addition, all devices that drive signals into the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) must not drive more than 3.3V. Outputs of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), although only driving to 3.3V levels, meet 5V TTL high and low specifications. Since the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) CLK is 5V safe, the CPU may be clocked with a 5V or 3.3V clock driver.

18.2.6. Lower Power Dissipation

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) consumes less power than the Pentium processor (510\60, 567\66). Please see Chapter 23 for further information.

Implications

Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and Pentium processor (510\60, 567\66) thermal solutions are different (refer to Chapter 26 for details).

18.2.7. Dual Processor Support

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) contains logic to interface to the Dual processor. In a dual-processor system, both the Primary and Dual processors connect to the same system signals. There are dedicated pins on each device that are used for dual-processing communication. See Chapters 18 and 20 for further information.

Implications

The Dual processor socket (socket 5/socket 7) connects to most signals of the Primary processor. Therefore, timing analysis must be done with processors in both sockets, each capable of being a bus master. The dual-processing interface is performed gluelessly; however, the system must use the on-chip APIC in the Primary and Dual processors at bootup time.

18.2.8. Local APIC Interrupt Controller

The APIC interrupt controller has been included in both the Primary and Dual processor. It is capable of supporting a multiprocessing interrupt scheme with an external APIC-compatible controller. Refer to Chapter 19.

Implications

If used, the APIC requires that the system supply an I/O APIC-compatible interrupt controller (see the APIC section). Also, system software must be able to support the APIC. If not enabled, the APIC is invisible and interrupts behave like in the Pentium processor (510\60, 567\66).

18.2.9. Power Management Features

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports SMM and other features similar to the *SL Enhanced Intel486™ Microprocessor Family*. Refer to Chapter 30.

Implications

These features are inactive or invisible to the system unless they are specifically used.

18.2.10. Breakpoint Pins

Because of the fractional-speed bus, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) breakpoint pins BP0-3 are defined differently from the Pentium processor (510\60, 567\66). Each assertion of a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) BP pin indicates that one or more BP matches occurred.

Implications

External breakpoint logic handling of the breakpoint function will have to take this into account. Each assertion of BP may have to be treated differently than in the Pentium processor (510\60, 567\66).

18.2.11. Hi-Z TAP Instruction

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) have all of the TAP instructions of the Pentium processor (510\60, 567\66) plus one additional instruction: Hi-Z. Refer to Chapter 19.

Implications

None.

18.2.12. Bus Cycles

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) may initiate bus cycles differently than the Pentium processor (510\60, 567\66). This is due mainly to the fractional-speed bus and the dual-processor. Refer to Chapter 19.

Implications

None.

18.2.13. HOLD Latency

When the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) on-chip APIC is being accessed, the CPU will not respond to a HOLD request. Thus HLDA may be delayed by up to 6 clocks.

Implications

HOLD latency is increased (just as if APIC was an external device) which may affect other bus masters.

18.2.14. Interrupt Priorities

Four interrupts have different priorities in order to support the I/O Instruction Restart feature of SMM. Refer to section 19.1.6 for details.

Implications

System designers must be aware of the changes to interrupt priorities.

18.3. SUMMARY OF DUAL PROCESSING DIFFERENCES FROM UNI-PROCESSING

The following is a list of bus cycles or bus cycle sequences which would not occur in Pentium processor (510\60, 567\66) or Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uni-processor systems, but may be seen in Dual processor systems. Refer to Section 18.2 for a list of Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uni-processor differences to Pentium processor (510\60, 567\66) bus cycles. These two sections form a complete list of dual processing bus differences versus the Pentium processor (510\60, 567\66).

- Locked cycle sequences
- Cycle pipe-lining
- Cycle ordering due to BOFF#
- Cache line state
- Back-to-back cycles
- Address parity checking
- Flush cycles
- PCHK# assertion
- Synchronous FLUSH# and RESET
- Floating point error handling

18.3.1. Locked Cycle Sequences

1. Locked read to address X
2. Locked write back to address X
3. Locked read to address X
4. Locked write to address X

May occur due to the inter-processor cache consistency mechanism. Refer to Chapter 20.

Implications

Processor bus hardware needs to handle this locked sequence. The only other time the system will see a locked write back is when an external snoop hits a modified line while a locked cycle is in progress (this will occur in a uni-processor or a dual-processor system).

18.3.2. Cycle Pipe-lining

Inter-processor (Primary/Dual processor) back-to-back write cycles will not be pipe-lined even if NA# has been asserted. The purpose of this rule is to prevent data bus contention during bus arbitration from one processor to the other. In dual processor mode, the Primary processor may pipe-line I/O cycles into I/O cycles from the Dual processor (and vice versa) for any I/O instruction combination (i.e., except I/O writes into writes).

Implications

System hardware designers should be aware of these bus changes.

18.3.3. Cycle Ordering Due to BOFF#

Cycle ordering following an assertion of BOFF# may be different between uni-processor and dual processor modes. This occurs when there are pipe-lined cycles from both processors, a BOFF# stalls both cycles, and an external snoop hits a modified line in the LRMs cache.

Implications

System hardware designers should be aware of these bus changes.

18.3.4. Cache Line State

In Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uni-processor and Pentium processor (510\60, 567\66) systems, if a line is put into the E state by the system hardware using the WB/WT# signal during the line fill, then all subsequent writes to that line will be handled internally via the on-chip cache. In dual-processor systems, under certain

circumstances, even if the system puts a line into the E state using WB/WT#, the dual-processor protocol may force the line to be stored in the S state. Private snooping in dual processor systems can also cause a line to be placed into the S or I state.

Implications

There are no system implications. The system may be required to handle writes to a line which would not otherwise have been seen.

NOTE

In a dual processing system where NW=1 and CD=1 are set, (i.e., SRAM mode), an inquire cycle will invalidate a cache line with INV on a HIT#.

18.3.5. Back-to-Back Cycles

Due to the dual-processor cache consistency protocol, the Primary and Dual processors may follow a write to address X with a write back to a 32-byte area which contains X. This will not occur in uni-processor systems. Also a read to address X may be followed by a write back to a 32-byte area which contains X.

Implications

There are no system implications.

18.3.6. Address Parity Checking

Address parity is checked during every private snoop between the Primary and Dual processors. Therefore, APCHK# may be asserted due to an address parity error during this private snoop. If an error is detected, APCHK# will be asserted 2 clocks after ADS# for one processor clock period. The system can choose to acknowledge this parity error indication at this time or do nothing.

Implications

There are no system implications. The system designers get extra address parity checking with dual processors due to the automatic private snooping.

18.3.7. Synchronous FLUSH# and RESET

When the Dual processor is present in Socket 5, the FLUSH# and RESET signals must be recognized by both processors at the same time.

Implications

FLUSH# and RESET must be asserted on the same clock to both the Primary and Dual processors.

18.3.8. PCHK# Assertion

In a dual-processor configuration, there is the possibility that the PCHK# signal can be asserted either 2 OR 3 CLKs following incorrect parity being detected on the data bus. This is due to the 2/3 bus to internal clock ratio.

Implications

Chip sets must account for this difference from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in their logic or state machines.

18.3.9. Flush Cycles

The Primary and Dual processors incorporate a mechanism to present a unified view of the cache flush operation to the system when in dual processing mode. The Dual processor performs the cache flush operation first, then grants the bus to the Primary processor. The Primary processor flushes its internal caches, and then runs the cache flush special cycle.

Implications

The system hardware **must** not assert a subsequent FLUSH# to the processors until the flush acknowledge special cycle has completed on the processor bus. The assertion of FLUSH# to the processors prior to this point would result in a corruption of the dual processing bus arbitration state machines.

18.3.10. Floating Point Error Handling

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as a Dual processor, ignores the IGNNE# input. The FERR# output is also undefined in the Dual processor.

Implications

None.

18.4. INTRODUCTION TO DUAL PROCESSOR MODE

Symmetric dual processing in a system is supported with two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) sharing a single second-level cache. The two processors appear to the system as a single Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). Multiprocessor operating systems properly schedule computing tasks between the two processors. This scheduling of tasks is transparent to software applications and the end-user. Logic built into the processors support a "glueless" interface for easy system design. Through a private bus, the two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) arbitrate for the external bus and maintain cache coherency.

In this document, in order to distinguish between two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) in dual processing mode, one CPU will be designated as the Primary processor with the other being the Dual processor. Note that this is a different concept than that of "master" and "checker" processors.

The Dual processor is a configuration option of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). The Dual processor is targeted to operate with up to a 133 MHz internal core clock and supports two core/bus ratios: 1/2 and 2/3. The bus frequencies supported are: 50 MHz, 60 MHz, and 66 MHz.

The Dual processor enables a two-socket system to operate without requiring logic changes to a Pentium processor (510\60, 567\66) memory bus controller. Bus cycles issued by the dual processor pair are compatible with the Pentium processor (510\60, 567\66). Differences between Pentium processor (510\60, 567\66) and dual processor cycles are listed in Sections 18.2 and section 18.3.

The Primary and Dual processors include logic to maintain cache consistency between the processors and to arbitrate for the common bus. The cache consistency and bus arbitration activity will cause the dual processor pair to issue extra bus cycles that will not appear in a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uniprocessor system. These extra bus cycles are compatible with the equivalent Pentium processor (510\60, 567\66) cycles.

Chapter 20 describes in detail how the DP bootup, cache consistency, and bus arbitration mechanisms operate. The dual processor pair should require little or no logic changes to a Pentium processor (510\60, 567\66) memory bus controller. In order to operate properly in dual processing mode, the Primary and Dual processors require private APIC, cache consistency, and bus arbitration interfaces, as well as a multiprocessing-ready operating system.

The dual processor interface allows the Dual processor to be inserted into Socket 5 for a substantial increase in system performance. The interface allows the Primary and Dual processor to operate in a coherent manner that is transparent to the system.

The memory subsystem transparency was the primary goal of the cache coherency and bus arbitration mechanisms. Another goal of the Primary and Dual processors, whether running individually or in conjunction, is to behave as similarly to a Pentium processor (510\60, 567\66) as possible.

18.4.1. Dual Processing Terminology

This section defines some terms used in the following discussions. They are here to ensure your understanding of the explanations and examples in remainder of this document.

Symmetric Multi-Processing:	Two or more processors operating with equal priorities in a system. No individual processor is a master, and none is a slave.
DP or Dual Processing:	The Primary and Dual processor operating symmetrically in a system sharing a second-level cache.
MRM or Most Recent Master:	The processor (either the Primary or Dual) which currently owns the processor address bus. When interprocessor pipe-lining, this is the processor which last issued an ADS#.
LRM or Least Recent Master:	The processor (either the Primary or Dual) which does not own the address bus. The LRM automatically snoops every ADS# from the MRM processor in order to maintain level one cache coherency.
Primary Processor:	The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) when CPUTYP = V_{SS}.
Upgrade Processor:	The Future Pentium OverDrive processor.
Dual Processor:	The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) when CPUTYP = V_{CC}.
OverDrive Processor:	The Future Pentium OverDrive processor.

18.4.2. New Pins / Pin Modifications

The following table outlines pin changes that have been made to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) from the Pentium processor (510\60, 567\66) in normal operation:

Table 18-1. Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Pin Directions vs. Pentium Processor (510\60, 567\66)

Pin Name	Pentium® Processor (510\60, 567\66) Pin Direction	Pentium Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Pin Direction
ADS#	Output	Input/Output
BE[4:0]#	Output	Input/Output *
BF	n/a	Input
BT[3:0]	Output	n/a
CACHE#	Output	Input/Output
CPUTYP	n/a	Input
D/C#	Output	Input/Output
D/P#	n/a	Output **
HIT#	Output	Input/Output
HITM#	Output	Input/Output
HLDA	Output	Input/Output
IBT	Output	n/a
IU	Output	n/a
IV	Output	n/a
LOCK#	Output	Input/Output
M/IO#	Output	Input/Output
PBGNT#	n/a	Input/Output
PBREQ#	n/a	Input/Output
PHIT#	n/a	Input/Output
PHITM#	n/a	Input/Output
PICCLK	n/a	Input
PICD[1:0]	n/a	Input/Output
SCYC	Output	Input/Output
STPCLK#	n/a	Input
W/R#	Output	Input/Output

NOTE:* BE4#-BE0# are I/O during RESET only.
** D/P# is only an Output from the Primary processor

18.4.3. Dual Processing Overview

The Primary and Dual processor both have logic built-in to support "glueless" dual-processing behind a shared L2 cache. Through a set of private handshake signals, the Primary and Dual processors arbitrate for the external bus and maintain cache coherency between themselves. The bus arbitration and cache coherency mechanisms allow the Primary and Dual processors to look like a single Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to the external bus.

The Primary and Dual processors implement a fair arbitration scheme. If the Least Recent Master (LRM) requests the bus from the Most Recent Master (MRM), the bus is granted. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) arbitration scheme provides no penalty to switch from one master to the next. If pipe-lining is used, the two processors will pipe-line into and out of each other's cycles according to the Pentium processor (510\60, 567\66) specification.

Cache coherency is maintained between the two processors by snooping on every bus access. The LRM must snoop with every ADS# assertion of the MRM. Internal cache states are maintained accordingly. If an access hits a modified line, a write back is scheduled as the next cycle in accordance with the Pentium processor (510\60, 567\66) specification.

Using the Dual processor may require special design considerations. Please refer to Chapter 20 for more details.

18.4.3.1. CONCEPTUAL OVERVIEW

Dual processing can be viewed in Figure 18-1:

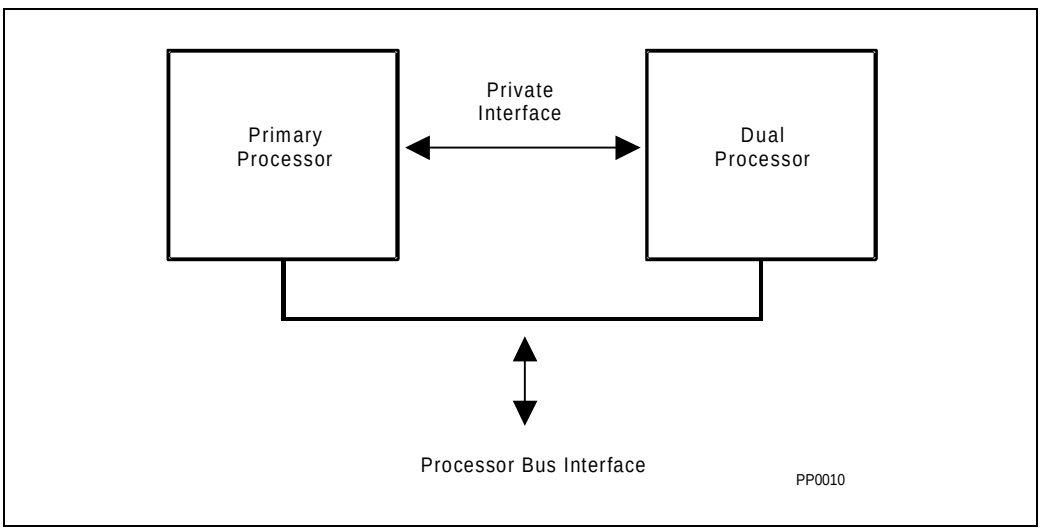


Figure 18-1. Dual Processors

The dual processor pair will appear to the system bus as a single, unified processor. The operation will be identical to a uni-processor Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), except as noted in Section 18.3. The interface shields the system designer from the cache consistency and arbitration mechanisms that are necessary for dual processor operation.

Both the Primary and Dual processors contain local APIC modules. The system designer is recommended to supply an I/O APIC or other multiprocessing interrupt controller in the chip set that interfaces to the local APIC blocks over a three wire bus. The APIC allows directed interrupts as well as inter-processor interrupts.

The Primary and Dual processors, when operating in dual processing mode, require the local APIC modules to be hardware enabled in order to complete the bootup handshake protocol. This method is used to "wake up" the Dual processor at an address other than the normal Intel Architecture high memory execution address. On bootup, if the Primary processor detects that a Dual processor is present in Socket 5, the dual processor cache consistency and arbitration mechanisms are automatically enabled. The bootup handshake process is supported in a protocol that is included in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). See Chapter 19 for more details on the APIC.

18.4.3.2. ARBITRATION OVERVIEW

In the dual processor configuration, there is a single-system bus which provides the processors access to the external system. This bus is a single, shared resource.

The dual processor pair will need to arbitrate for use of the system bus as requests are generated. The processors implement a fair arbitration mechanism.

If the LRM processor needs to run a cycle on the bus it will submit a request for bus ownership to the MRM. The MRM processor will grant the LRM processor bus ownership as soon as all outstanding bus requests have finished on the processor bus. The LRM processor will assume the MRM state, and the processor which was just the MRM, will become the LRM. Figure 18-2 further illustrates this point:

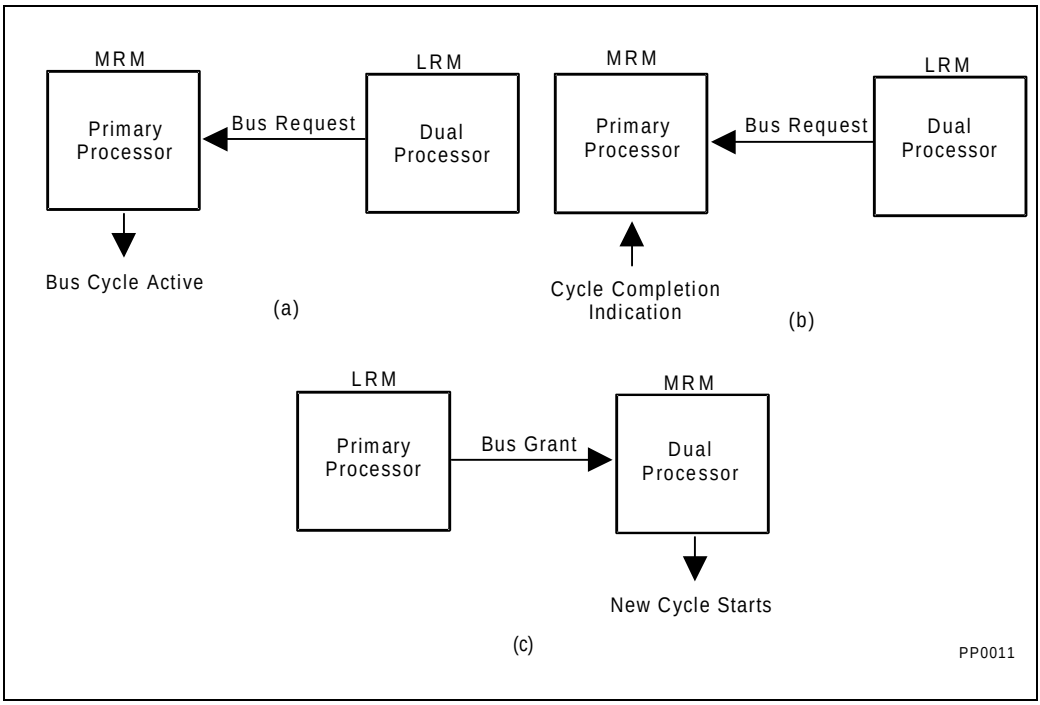


Figure 18-2. Dual Processor Arbitration Mechanism

Diagram (a) of Figure 18-2 shows a configuration where the Primary processor is in the MRM state and the Dual processor is in the LRM state. The Primary processor is running a cycle on the system bus when it receives a bus request from the Dual processor. In diagram (b) of Figure 18-2 the MRM (still the Primary processor) has received an indication that the bus request has finished. The bus ownership has transferred in diagram (c) of Figure 18-2, where the Dual processor is now the MRM. At this point, the Dual processor will start a bus transaction and continue to own the bus until the LRM requests the bus.

18.4.3.3. CACHE COHERENCY OVERVIEW

The Primary and Dual processors both contain an 8 Kbyte instruction cache and an 8 Kbyte data cache. The data cache uses the MESI protocol to enforce cache consistency. A line in the data cache can be in the Modified, Exclusive, Shared or Invalid state whereas a line in the instruction cache can be either in the valid or invalid state.

A situation can arise where the Primary and Dual processors are operating in dual processor mode with shared code or data. The first level caches will attempt to cache this code and data whenever possible (as indicated by the page cacheability bits and the cacheability pins). The private cache coherency mechanism guarantees data consistency across the processors. If any data is cached in one of the processors, and the other processor attempts to access the data, the processor containing the data will notify the requesting processor that it has cached

the data. The state of the cache line in the processor containing the data will change depending on the current state and the type of request that the other processor has made.

In some cases the data returned by the system will be ignored. This constraint is placed on the dual processor cache consistency mechanism so that the dual processor pair will look like a single processor to the system bus. However, in general, bus accesses are minimized to efficiently use the available bus bandwidth.

The basic coherency mechanism requires the processor that is in the LRM state to snoop all MRM bus activity. The MRM processor running a bus cycle will watch the LRM processor for an indication that the data is contained in the LRM cache. The following diagrams illustrate the basic coherency mechanism.

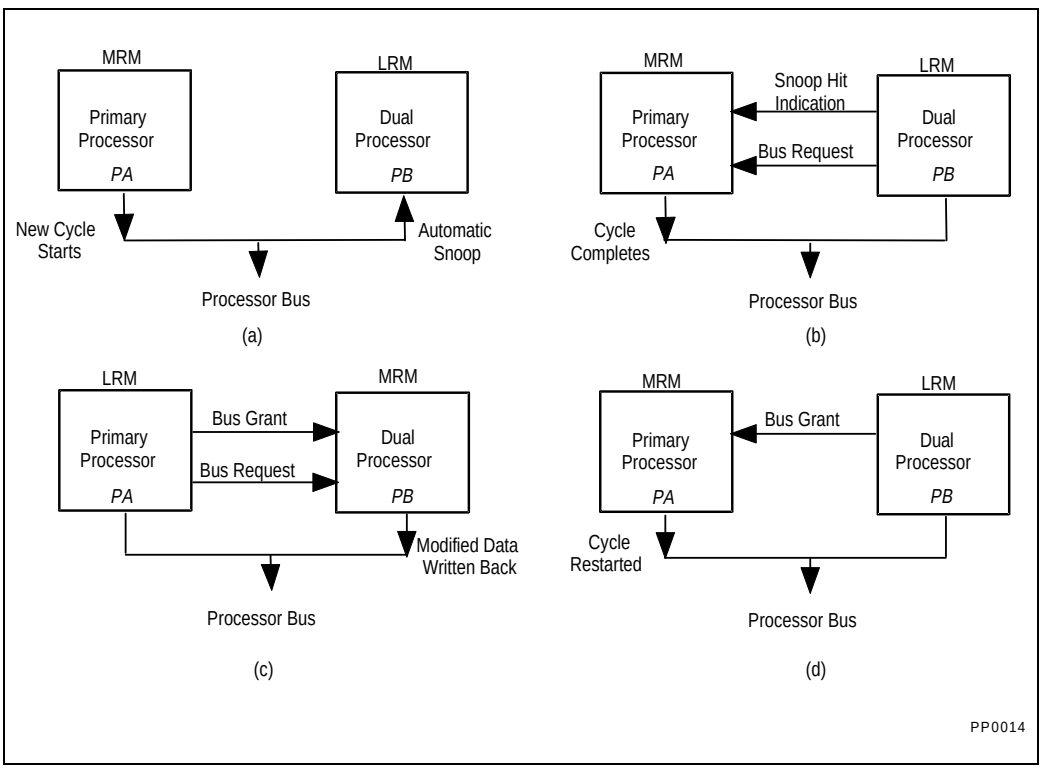


Figure 18-3. Dual Processor L1 Cache Consistency

The series of figures above show an example where the *Primary processor* (the MRM) is performing a cache line fill of data. In this example, the data requested by the *Primary processor* is cached by the *Dual processor* (the LRM), and is in the modified state.

In diagram (a) of Figure 18-3, the *Primary processor* has already negotiated with the *Dual processor* for use of the system bus and started a cycle. As the *Primary processor* starts running the cycle on the system bus, the *Dual processor* processor snoops the transaction.

The key for the start of the snoop sequence for the LRM processor is an assertion of **ADS#** by the MRM processor.

Diagram (b) of Figure 18-3 shows the *Dual processor* indicating to the *Primary processor* that the requested data is cached and modified in the *Dual processor* cache. The snoop notification mechanism uses a dedicated, two-signal interface that is private to the dual processor pair. At the same time that the *Dual processor* indicates that the transaction is contained as Modified in the its cache, the *Dual processor* will request the bus from the *Primary processor* (still the MRM). The MRM processor continues with the transaction that is outstanding on the bus, but will ignore the data returned by the system bus.

After the *Dual processor* notifies the *Primary processor* that the requested data is modified in the *Dual processor* cache, the *Dual processor* will wait for the bus transaction to complete. At this point, the LRM/MRM state will toggle, with the *Primary processor* becoming the LRM processor and the *Dual processor* becoming the MRM processor. This sequence of events is shown in diagram (c) of Figure 18-3.

Diagram (c) of Figure 18-3 also shows the *Dual processor* writing the data back on the system bus. The write back cycle will look like a normal cache line replacement to the system bus. The final state of the line in the *Dual processor* is determined by the value of the **W/R#** pin as sampled during the **ADS#** assertion by the *Primary processor*.

Finally, diagram (d) of Figure 18-3 shows the *Primary processor* re-running the bus transaction that started the entire sequence. The requested data will be returned by the system as a normal line fill request without intervention from the LRM processor.

19

Component Operation

CHAPTER 19
COMPONENT OPERATION

19.1. FEATURES AND DIFFERENCES

In addition to the component operation described in Chapter 3 for the Pentium processor family, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) offers extensions to the Pentium processor (510\60, 567\66) architecture and bus:

- Fractional speed bus
- Selectable buffer sizes
- Power management: I/O instruction restart
- Power management: stop clock and autohalt powerdown
- APIC interrupt controller
- Interrupt priorities
- CPUID instruction operation
- Bus cycle and latency differences
- Breakpoint signals
- New TAP instruction: Hi-Z

19.1.1. Fractional Speed Bus

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will be offered in bus/core speed ratios of 66\133 MHz, 60\120 MHz, 66/100 MHz, 50/100 MHz, 60/90 MHz, and 50\75 MHz. The BF configuration pin is provided to support a 2/3 bus/core ratio and a 1/2 bus/core ratio. The allowable frequencies of the external bus and the CLK pin are 50 MHz, 60 MHz, and 66 MHz. The component will multiply the input CLK to achieve the higher internal core frequencies.

The external bus frequency is set on power-up RESET through the CLK pin. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will sample the BF pin on the falling edge of RESET to determine which bus/core ratio to use. If the BF pin is left unconnected, the Pentium processor (610\75, 735\90, 815\100) assumes the 2/3 ratio. The Pentium processor (1110\133, 1000\120) only supports a 1/2 bus/core ratio. Therefore, the BF pin should be pulled low in this case. **BF must not change its value while RESET is active**. Once a frequency is selected, it may not be changed with a warm-reset (15 clocks). Changing this speed or ratio requires a "power-on" RESET pulse initialization.

The BF pin must meet a 1 ms setup time to the falling edge of RESET. Table 19.1 summarizes the operation of BF.

Table 19-1. Bus Frequency Selections

BF	Clock and Bus Frequency	Bus/Core Ratio	Core Frequency
0	66 MHz	1/2	133 MHz
0	60 MHz	1/2	120 MHz
0	50 MHz	1/2	100 MHz
1	60 MHz	2/3	90 MHz
1	66 MHz	2/3	100 MHz

Despite a fractional-speed CPU bus, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will execute bus cycles according to the Pentium processor (510\60, 567\66) bus definition. However, due to the higher core frequency of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), some cycles in the CPU may be initiated by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in an order which differs from a Pentium processor (510\60, 567\66) or from another Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) with a different bus speed selection. These cycles are prefetch cycles and write back cycles due to a replacement. These are never snoop write back cycles or cycles that could cause read or write reordering. Although the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) may initiate some cycles in a different order, the order of execution of instructions remains the same. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) synchronizes the internal buses and CPU buses without synchronization waitstate penalties. This means that if a cycle is pending internally, it is driven on the next possible CPU clock (CLK) edge out to the bus. Likewise, if data is available at the pins on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), it is read on the next possible internal clock edge. This is exactly how the Pentium processor (510\60, 567\66), with a synchronous core and bus, operates.

The following examples illustrate the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) synchronization mechanism. The Pentium processor (510\60, 567\66) case is given to indicate exactly how a 1/1 bus operates.

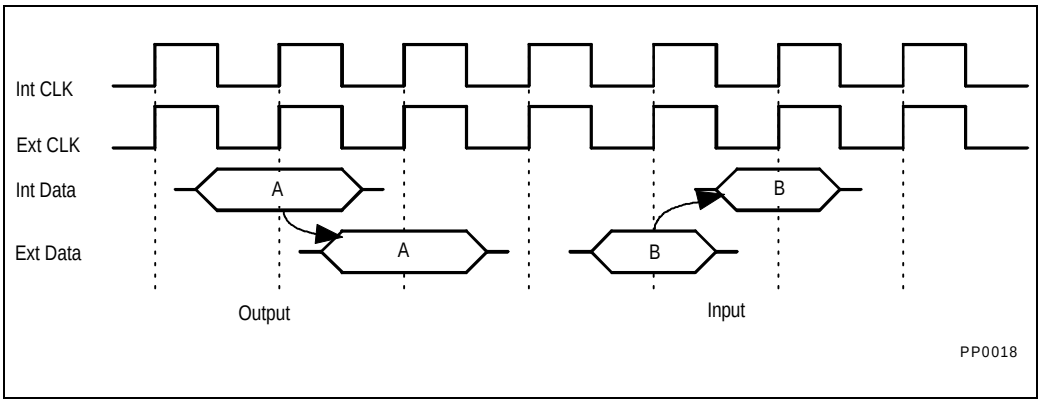


Figure 19-1. Pentium® Processor (510\60, 567\66) Synchronous Internal/External Data Movement

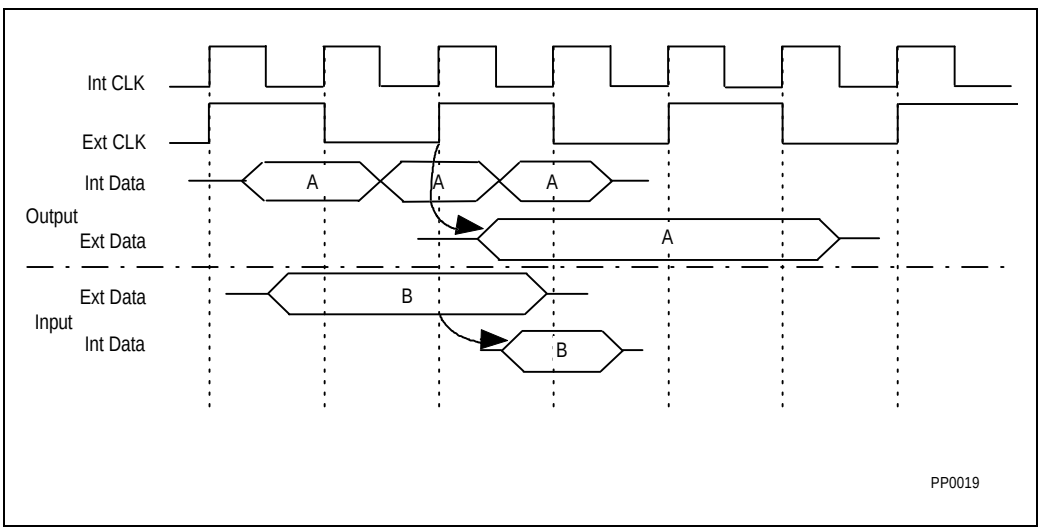


Figure 19-2. Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 1/2 Bus Internal/External Data Movement

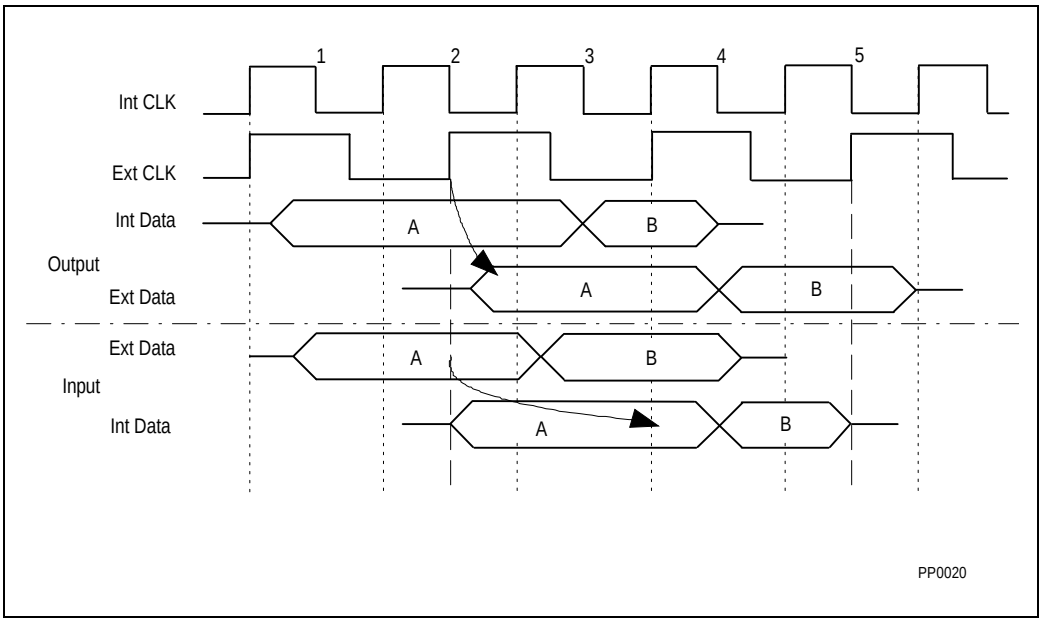


Figure 19-3. Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) 2/3 Bus Internal/External Data Movement

In Figure 19-3, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) prevents data from changing in clock 2, where the 2/3 external clock rising edge occurs in the middle of the internal clock phase, so it can be properly synchronized and driven.

19.1.2. Selectable Buffer Sizes

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has selectable buffer sizes to allow for faster switching of the buffer in heavily loaded environments. The buffer selection is done through the setting of configuration pins at power on RESET. Once selected, these cannot be changed without a power on RESET. The BUSCHK# pin is used to select the different buffer size. All configurable pins get set to the selected buffer size. There is no selection for specific signal groups to get specific buffers. Keep in mind that the largest buffer size is not always the best selection especially in a lightly loaded environment. AC timing and signal quality simulations should be done to ensure that the buffers used meet required timing and signal quality specifications for the components that will be used in the specific board design. Refer to Chapter 24.

19.1.3. Power Management: I/O Instruction Restart

I/O Instruction restart is a power management feature of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) that allows the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to re-execute an I/O instruction. In this way, an I/O instruction can alert a sleeping device in a system and SMI# can be recognized before the I/O instruction is re-executed. SMI# assertion will cause a wake-up routine to be executed, so the restarted I/O instruction can be executed by the system.

Please reference Chapter 20 of the *Pentium® Processor Family Developer's Manual*, Volume 3.

19.1.4. Power Management: Stop Clock and AutoHalt Powerdown

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses stop clock and AutoHalt Powerdown to immediately reduce the power of each device. These features cause the clock to be stopped to most of the CPU's internal units and thus significantly reduce power consumption by the CPU as a whole.

Stop clock is enabled by asserting the STPCLK# pin of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). While asserted, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will stop execution and not service interrupts, but will allow external and interprocessor (Primary and Dual processor) snooping.

AutoHalt Powerdown is entered once the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) executes a HLT instruction. In this state the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) behaves like it executed a HLT instruction, but most internal units are powered-down. In AutoHalt Powerdown, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will recognize all interrupts and snoops.

Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) pin functions (D/P#, etc.) are not affected by STPCLK# or AutoHalt.

For an extensive explanation of this feature, please reference the Power Management Section, Chapter 30, of this document.

19.1.5. APIC Interrupt Controller

The Advanced Programmable Interrupt Controller (APIC) is an on-chip interrupt controller that supports multiprocessing. In a uniprocessor system, APIC may be used as the sole system interrupt controller, or may be disabled and bypassed completely.

In a multiprocessor system, the APIC operates with an additional and external I/O APIC system interrupt controller. The dual processor configuration requires that the APIC be hardware enabled. The APIC of the Primary and Dual processors are used in the bootup procedure to communicate startup information. Note: the APIC is not hardware compatible with the 82489DX.

On the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), the APIC uses 3 pins: PICCLK, PICD0, and PICD1. PICCLK is the APIC bus clock while PICD0-1 form the two-wire communication bus.

To use the 8259A interrupt controller, or to completely bypass, the APIC may be disabled using the APICEN pin. You must use the local APICs when using the Dual processor component.

19.1.6. Interrupt Priorities

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) interrupt priority scheme is different than the Pentium processor (510\60, 567\66) scheme to support new features. The handling of interrupts on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) can be different than the handling on the Pentium processor (510\60, 567\66) because of the priority change. The priority is as follows:

Table 19-2. Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Interrupt Priority Scheme

	ITR = 0 (default)	ITR = 1
1.	Breakpoint (INT 3)	Breakpoint (INT 3)
2.	BUSCHK#	BUSCHK#
3.	Debug Traps (INT 1)	FLUSH#
4.	R/S#	SMI#
5.	FLUSH#	Debug Traps (INT 1)
6.	SMI#	R/S#
7.	INIT	INIT
8.	NMI	NMI
9.	INTR	INTR
10.	Floating-Point Error	Floating-Point Error
11.	STPCLK#	STPCLK#
12.	Faults on Next Instruction	Faults on Next Instruction

Note: ITR is bit 9 of the TR12 register

19.1.7. CPUID Instruction

The CPUID instruction allows software to determine the type and features of the microprocessor on which it is executing.

When executing CPUID, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) behaves like the Pentium processor (510\60, 567\66) :

- If the value in EAX is "0" then the 12-byte ASCII string "GenuineIntel" (little endian) is returned in EBX, EDX, and ECX. Also, EAX contains a value of "1" to indicate the largest value of EAX which should be used when executing CPUID.
- If the value in EAX is "1" then the processor version is returned in EAX and the processor capabilities are returned in EDX. The value of EAX for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is given below.
- If the value in EAX is neither "0" nor "1", the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) writes "0" to EAX, EBX, ECX, and EDX.

For more information on values in the EDX register, see Appendix A.

The following EAX value is defined for the CPUID instruction executed with EAX = 1.

The processor version EAX bit assignments are given in figure Figure 19-4.

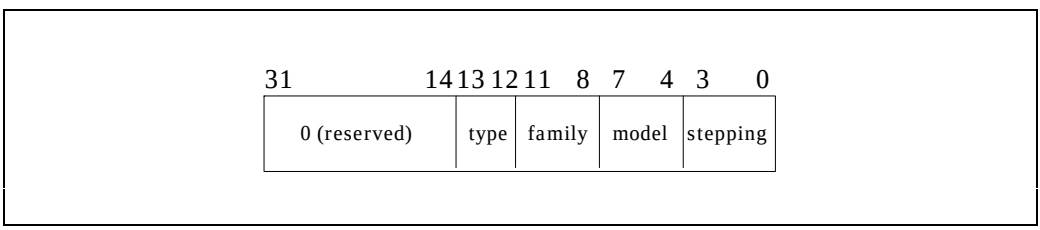


Figure 19-4. EAX Bit Assignments for CPUID

The family field is the same as the Pentium processor (510\60, 567\66) (e.g., family = 5H). The model value is 2H. The stepping field has the same format as the Pentium processor (510\60, 567\66) (see Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Specification Update). The Future Pentium OverDrive processor will not have the same values in EAX as the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). The type field is defined as follows:

Table 19-3. EAX Type Field Values

Bit 13	Bit 12	Processor Type
0	0	Primary Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133)
0	1	Future Pentium OverDrive® processor
1	0	Dual Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133)
1	1	Reserved

19.1.8. Bus Cycle and Latency Differences

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) may initiate cycles to the bus in a different order than the Pentium processor (510\60, 567\66), or produce different types of cycles. This is due to two architectural differences: 1), the fractional-speed bus, and 2), a dual processor configuration. All of the dual processor configuration differences are listed in Chapter 20.

Because of the fractional-speed bus, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) bus scheduler can schedule cycles in an order and number which differ from the Pentium processor (510\60, 567\66) operation.

While the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is accessing the local APIC, the processor will delay a HOLD request by up to 6 CLKs. To external agents which are not aware of the APIC bus, this looks like the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is not responding to HOLD even though ADS# has not been driven and the processor seems idle.

While accessing the APIC, the address and control pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) may toggle without ADS#.

Because of the difference in interrupt priority schemes between the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and the Pentium processor (510\60, 567\66), the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) may respond to multiple interrupts in a different order than the Pentium processor (510\60, 567\66) .

19.1.9. Breakpoint Signals

Because of the fractional-speed bus, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) breakpoint pins BP0-3 are defined differently from the Pentium processor (510\60, 567\66). Each assertion of a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) BP pin indicates that one or more BP matches occurred. The maximum number of matches per assertion is 2 when using the 1/2 bus/core fraction. This is different from the Pentium processor (510\60, 567\66) which will only get a single match per clock.

19.1.10. New Tap Instruction: Hi-Z

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has the same TAP instructions as the Pentium processor (510\60, 567\66) plus one additional instruction: Hi-Z.

Instruction	TAP Encoding
Hi-Z	XXXXXXXX1011

The TAP Hi-Z instruction causes all outputs and I/Os of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to go to a high-impedance state (float) immediately.

The Hi-Z state is terminated by either resetting the TAP with the TRST# pin, by issuing another TAP instruction, or by entering the Test_Logic_Reset state.

The Hi-Z state is enabled or disabled on the first TCK clock after the TAP instruction has entered the UPDATE-IR state of the TAP control state machine.

This instruction overrides all other bus cycles. Resetting the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will not disable this instruction since CPU RESET does not reset the TAP.

19.2. APIC INTERRUPT CONTROLLER

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) contains implementations of the Advanced Programmable Interrupt Controller architecture. These implementations are capable of supporting a multiprocessing interrupt scheme with an external APIC-compatible controller. The main features of the APIC architecture include:

- Multiprocessor interrupt management (static and dynamic symmetric interrupt distribution across all processors)
- Dynamic interrupt distribution that includes routing interrupts to the lowest-priority processor
- Inter-processor interrupt support
- Edge or level triggered interrupt programmability
- Various naming/addressing schemes
- System-wide processor control functions related to NMI, INIT, and SMI (see Chapter 30 for APIC handling of SMI)
- 8259A compatibility by becoming virtually transparent with regard to an externally connected 8259A style controller, making the 8259A visible to software
- A 32-bit wide counter used as a timer to generate time slice interrupts local to that processor.

The AC timings of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) APIC are described in Chapter 23 of this document. Note that while there are minor software differences from the 82489DX, programming to the integrated APIC model ensures compatibility with the external 82489DX. For additional APIC programming information, please refer to the *MultiProcessor Specification*, Order Number 242016.

In a dual-processor configuration, the local APIC may be used with an additional device similar to the I/O APIC. The I/O APIC is a device which captures all system interrupts and directs them to the appropriate processors via various programmable distribution schemes. An external device provides the APIC system clock. Interrupts which are local to each CPU go through the APIC on each chip. A system example is shown in Figure 19-5.

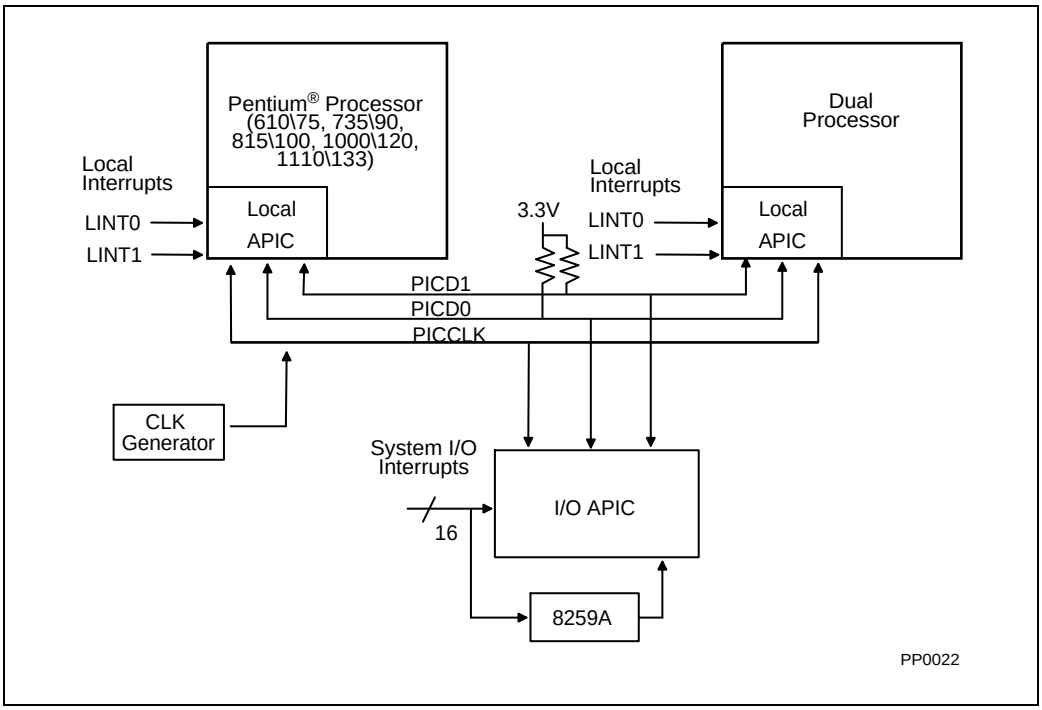


Figure 19-5. APIC System Configuration

The APIC devices in the Primary and Dual processors may receive interrupts from the I/O APIC via the three-wire APIC bus, locally via the local interrupt pins (LINT0, LINT1), or from the other processor via the APIC bus. The local interrupt pins, LINT0 and LINT1, are shared with the INTR and NMI pins, respectively. When the APIC is bypassed (hardware disabled) or programmed in "through local" mode, the 8259A interrupt (INTR) and NMI are connected to the INTR/LINT0 and NMI/LINT1 pins of the processor. Figure 19-6 shows the APIC implementation in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). Note that the PICCLK has a maximum frequency of 16.67 MHz.

When the local APIC is hardware enabled, *data* memory accesses to its 4 KByte address space are executed internally and do not generate an ADS# on the processor bus. However, a *code* memory access in the 4 KByte APIC address space will not be recognized by the APIC and will generate a cycle on the processor bus.

NOTE

Internally executed data memory accesses may cause the address bus to toggle even though no ADS# is issued on the processor bus.

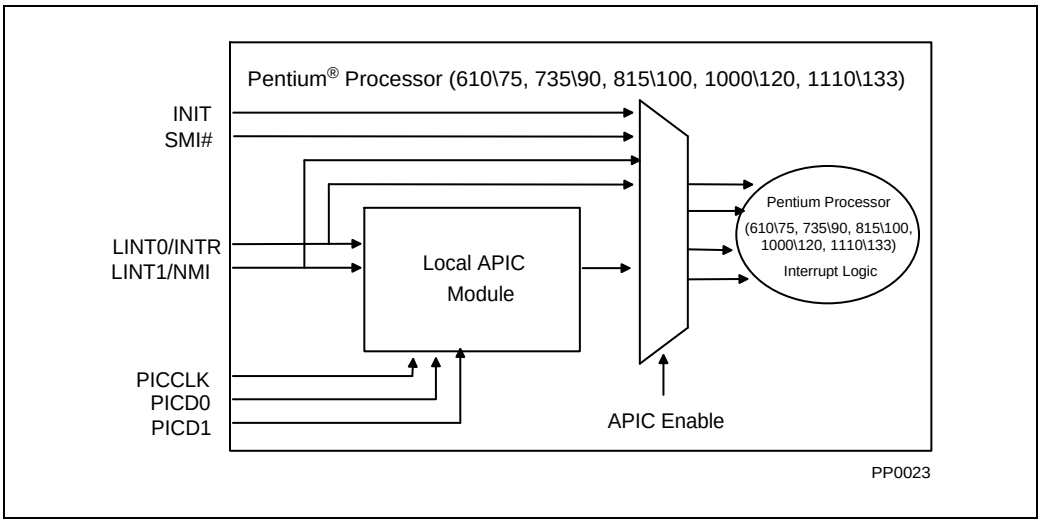


Figure 19-6. Local APIC Interface

19.2.1. APIC Configuration Modes

There are four possible APIC Modes:

1. Normal mode
2. Bypass mode (hardware disable).
3. Through local mode.
4. Masked mode (software disable).

19.2.1.1. NORMAL MODE

This is the normal operating mode of the local APIC. When in this mode, the local APIC is both hardware and software enabled.

19.2.1.2. BYPASS MODE

Bypass mode effectively removes (bypasses) the APIC from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) causing it to operate as if there were no APIC present (i.e., like a Pentium processor (510\60, 567\66)). Any accesses to the APIC address space will go to memory. APICEN is sampled at the falling edge of RESET, and later becomes the PICD1 (part of the APIC 3-wire bus) signal. Bypass mode is entered by driving APICEN low at the falling edge of RESET. Since the APIC must be used to enable the Dual processor after RESET, PICD1 must be driven high at reset to ensure APIC is hardware enabled if a second processor is present.

For hardware disabling operations, the following implications must be considered:

1. The INTR and NMI pins become functionally equivalent to the corresponding interrupt pins in the Pentium processor, and the APIC is bypassed.
2. The APIC PICCLK must be tied high.
3. The system will not operate with the Dual Processor if the local APIC is hardware disabled.

19.2.1.3. THROUGH LOCAL MODE

Configuring in through local mode allows the APICs to be used for the dual-processor bootup handshake protocol and then pass interrupts through the local APIC to the core to support an external interrupt controller.

To use the through local mode of the local APIC, the APIC must be enabled in both hardware and software. This is done by programming two local vector table entries, LVT1 and LVT2 at addresses 0FEE00350H and 0FEE00360H, as external interrupts (ExtInt) and NMI, respectively. The 8259A responds to the INTA cycles and returns the interrupt vector to the processor.

The local APIC should not be sent any interrupts prior to it's being programmed. Once the APIC is programmed it can receive interrupts.

Note that although external interrupts and NMI are passed through the local APIC to the core, the APIC can still receive messages on the APIC bus.

19.2.1.4. MASKED MODE

The local APIC is initialized to masked mode once hardware enabled via the APICEN pin. In order to be programmed in normal or through local modes, the APIC must be "software enabled." Once operating in normal mode or through local mode, the APIC may be disabled by software through clearing bit 8 of the APIC's spurious vector interrupt register (Note: this register is normally cleared at RESET and INIT). This register is at address 0FEE000F0H. Disabling APIC in software will return it to Masked mode. With the exception of NMI, SMI, INIT, remote reads and the startup IPI, all interrupts are masked on the APIC bus. The local APIC does not accept any interrupts on LINT0 or LINT1. See the following section for software disabling implications.

19.2.1.4.1. Software Disabling Implications

For the software disabling operations, the following implications must be considered:

1. The 4 KByte address space for the APIC is always blocked for data accesses (i.e., external memory in this region must not be accessed).
2. The interrupt control register (ICR) can be read and written (e.g. interprocessor interrupts are sent by writing to this register).
3. The APIC can continue to receive SMI, NMI, INIT, "startup," and remote read messages.
4. Local interrupts are masked.
5. Software can enable/disable the APIC at any time. After software disabling the local APICs, pending interrupts must be handled or masked by software.
6. The APIC PICCLK must be driven at all times.

19.2.1.5. DUAL PROCESSING WITH THE LOCAL APIC

The Dual processor bootup protocol may be used in the normal, through local, or masked modes.

19.2.2. Loading the APIC ID

The local APIC module on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) loads its 4-bit APIC ID value from four pins at the falling edge of RESET. The following table shows the four pins that comprise the APIC ID:

Table 19-4. APIC ID

APIC ID Register Bit	Pin Latched at RESET
bit 24	BE0#
bit 25	BE1#
bit 26	BE2#
bit 27	BE3#

Loading the APIC ID may be done with external logic that would drive the proper address at reset. If the BE[3:0]# signals are not driven and do not have external resistors to V_{CC} or V_{SS}, the APIC ID value will default to 0000 for the Primary processor and 0001 for the Dual processor.

WARNING

An APIC ID of all 1s is an APIC special case (i.e., a broadcast) and must not be used. Since the Dual processor inverts the lowest order bit of the APIC ID placed on the BE pins, the value "1110" should also be avoided when operating in Dual Processing mode.

In a dual processor configuration, the OEM and Socket 5 should have the four BE pairs tied together. The OEM processor will load the value seen on these four pins at RESET. The dual processor will load the value seen on these pins and automatically invert bit 24 of the APIC ID Register. Thus the two processors will have unique APIC ID values.

In a general multi-processing system consisting of multiple Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), these pins must not be tied together so each local APIC can have unique ID values.

These four pins must be valid and stable two clocks before and after the falling edge of RESET.

19.2.3. Response to HOLD

While the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is accessing the APIC, the processor will respond to a HOLD request with a maximum delay of six clocks. To external agents which are not aware of the APIC bus, this looks like the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is not responding to HOLD even though ADS# has not been driven and the processor bus seems idle.

20

Microprocessor Initialization and Configuration

CHAPTER 20
MICROPROCESSOR INITIALIZATION AND CONFIGURATION

In addition to the processor initialization and configurations described in Chapter 4 for the Pentium processor family, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports a symmetric dual processing configuration. This section describes this new mode in detail.

20.1. MANAGING AND DESIGNING WITH THE SYMMETRICAL DUAL PROCESSING CONFIGURATION

20.1.1. Dual Processor Bootup Protocol

20.1.1.1. BOOTUP OVERVIEW

Systems using the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) may be equipped with a second processor socket. For correct system operation, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) must be able to identify the presence and type of the second processor (a Dual processor or a Future Pentium OverDrive processor). Furthermore, since upgrade processors will typically be installed in the field by end users, system configuration may change between any two consecutive power-down/up sequences. The system must therefore have a mechanism to ascertain the system configuration during boot time. The boot up handshake protocol provides this mechanism.

20.1.1.2. BIOS / OPERATING SYSTEM REQUIREMENTS

The BIOS or HAL (hardware abstraction layer) of the operating system software should be generic, independent of the kind of OEM or upgrade processor present in the system. BIOS/HAL are specific to the system hardware, and should not need any change when an upgrade processor is installed. For dual processors, if the BIOS is not DP-ready, it will be up to the operating system to initialize and configure the dual processor appropriately.

The CPUID instruction is used to deliver processor-specific information. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) CPUID status has been extended to supply the processor type information which includes "turbo-upgrade" classification ("type" field: bits 13-12 = 0-1). For upgradability with a Future Pentium OverDrive processor, system software must allow the type field of the EAX register following the CPUID instruction to contain the values for both the Pentium processor (610\75, 735\90, 815\100,

1000\120, 1110\133) and the fFuture Pentium OverDrive processor. Refer to Section 19.1.7 for details. Note also that the model field of the CPUID will change for the Future Pentium OverDrive processor, and the new value should not break the system software or BIOS.

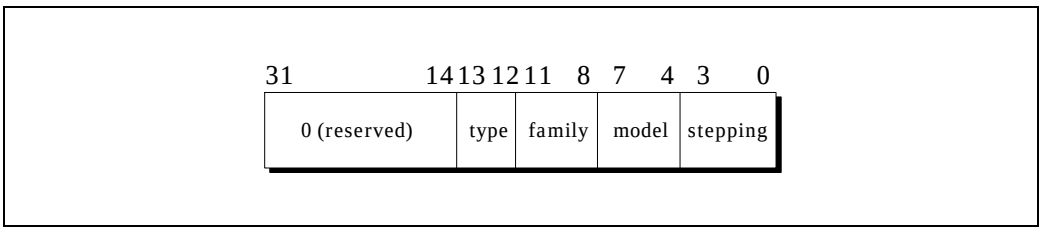


Figure 20-1. EAX Bit Assignments for CPUID

20.1.1.3. SYSTEM REQUIREMENTS

The number of Dual processors per Primary processor is limited to 1.

This bootup handshake protocol requires enabling the local APIC module using the APICEN pin. The startup IPI must be sent via the local APICs. Once the Dual processor has been initialized, software can later disable the local APIC module using several methods. These methods and their considerations are discussed in Chapter 19.

The protocol does not preclude more generic multiprocessing systems where multiple pairs of Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Primary and Dual processors may exist on the system bus.

20.1.1.4. STARTUP BEHAVIOR

On RESET and INIT (message or pin), the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) begins execution at the reset vector (0FFFFFFF0H). The Dual processor waits for a startup IPI from the BIOS or operating system via the local APIC of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). The INIT IPI can be used to put the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) or Dual processor to sleep (since, once the INIT IPI is received, the CPU must wait for the startup IPI).

The startup IPI is specifically provided to start the Dual processor's execution from a location other than the reset vector, although it can be used for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) as well. The startup IPI is sent by the system software via the local APIC by using a delivery mode of 110B. The startup IPI must include an 8-bit vector which is used to define the starting address. The starting address = *000 VV 000 h*, where *VV* indicates the vector field (in hex) passed through the IPI.

The 8-bit vector defines the address of a 4 K-Byte page in the Intel Architecture Real Mode Space (1 Meg space). For example, a vector of 0cdH specifies a startup memory address of 000cd000H. This value is used by the processor to initialize the segment descriptor for the upgrade's CS register as follows:

- The CS selector is set to the startup memory address/16 (real mode addressing)
- The CS base is set to the startup memory address
- The CS limit is set to 64 K-bytes
- The current privilege level (CPL) and instruction pointer (IP) are set to 0

NOTE

Vectors of 0A0H to 0BFH are Reserved by Intel.

The benefit of the startup IPI is that it does not require the APIC to be software enabled (the APIC must be hardware enabled via the APICEN pin) and does not require the interrupt table to be programmed. Startup IPIs are non-maskable and can be issued at any time to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) or Dual processor. If the startup IPI message is not preceded by a RESET or INIT (message or pin), it will be ignored.

It is the responsibility of the system software to resend the startup IPI message if there is an error in the IPI message delivery. Although the APIC need not be enabled in order to send the startup IPI, the advantage to enabling the APIC prior to sending the startup IPI is to allow APIC error handling to occur via the APIC error handling entry of the local vector table (ERROR INT or LVT3 at APIC address 0FEE00370). Otherwise, the system software would have to poll the delivery status bit of the interrupt command register to determine if the IPI is pending (Bit 12 of the ICR=1) and resend the startup IPI if the IPI remains pending after an appropriate amount of time.

20.1.1.5. DUAL PROCESSOR OR UPGRADE PRESENCE INDICATION

The bootup handshake protocol becomes aware that an additional processor is present through the DPEN# pin. The second processor is guaranteed to drive this signal low during RESETs falling edge. If the system needs to remember the presence of a second processor for future use, it must latch the state of the DPEN# pin during the falling edge of RESET.

20.1.2. Dual Processor Arbitration

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) incorporates a private arbitration mechanism that allows the Primary and Dual processors to arbitrate for the shared processor bus without assistance from a bus controller. The arbitration scheme is architected in such a way that the dual processor pair will appear as a single processor to the system.

The arbitration logic uses a fair arbitration scheme. The arbitration state machine was designed to efficiently use the processor bus bandwidth. In this spirit, the dual processor pair supports inter-CPU pipe-lining of most bus transactions. Furthermore, the arbitration mechanism does not introduce any dead clocks on bus transactions.

20.1.2.1. BASIC DP ARBITRATION MECHANISM

The basic set of arbitration premises requires that the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) check the second socket (Socket 5/Socket 7) for a processor every time the processor enters reset. To perform the checking of the Socket 5 and to perform the actual boot sequence, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in the 296-pin socket will always come out of reset as the MRM. This will require the part in the Socket 5 to always come out of reset as the LRM.

The LRM processor will request ownership of the processor bus by asserting the private arbitration request pin, PBREQ#. The processor that is currently the MRM and owns the bus, will grant the bus to the LRM as soon as any pending bus transactions have completed. The MRM will grant the bus to the LRM immediately if that CPU has a pipe-lined cycle to issue. The MRM will notify that the LRM can assume ownership by asserting the private arbitration grant pin, PBGNT#. The PBREQ# pin is always the output of the LRM and the PBGNT# is always an input to the LRM.

A processor can park on the processor bus if there are no requests from the LRM. A parked processor can be running cycles or just sitting idle on the bus. If a processor just ran a cycle on the bus and has another cycle pending without an LRM request, the processor will run the second cycle on the bus.

Locked cycles present an exception to the simple arbitration rules. All locked cycles will be performed as atomic operations without interrupt from the LRM. The case where a locked access causes an assertion of PHITM# by the LRM provides an exception to this rule. In this case, the MRM will grant the bus to the LRM and allow the write back to complete.

The normal system arbitration pins (HOLD, HLDA, BOFF#) will function the same as in uni-processor mode. Thus, the dual processor pair will always factor the state of the processor bus as well as the state of the local arbitration before actually running a cycle on the processor bus.

20.1.2.2. DP ARBITRATION INTERFACE

The following diagram details the hardware arbitration interface.

NOTE

For proper operation, PBREQ# and PBGNT# must not be loaded by the system.

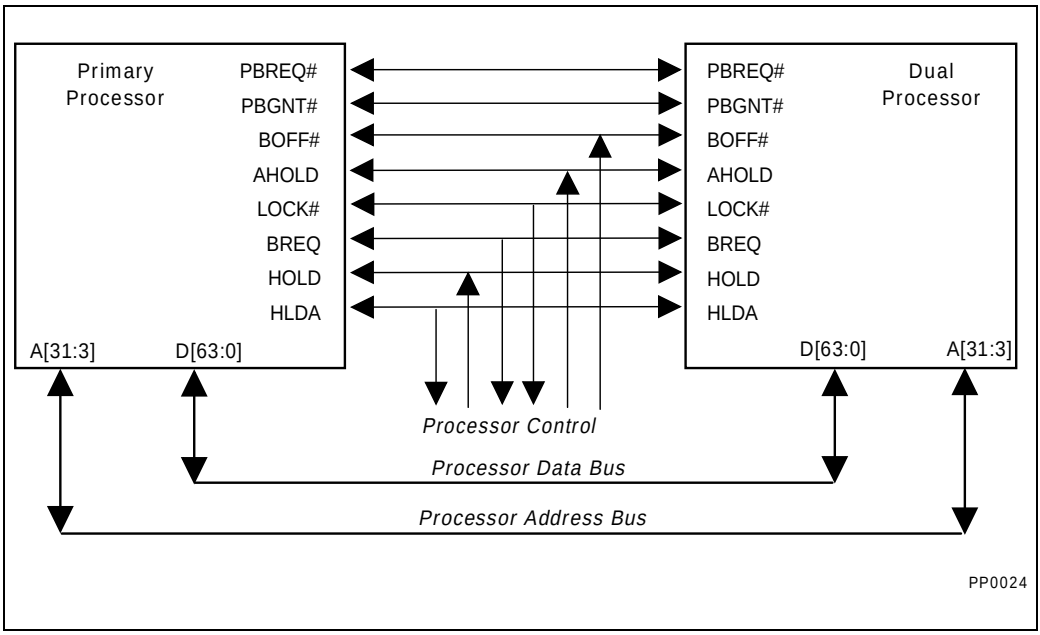


Figure 20-2. Dual Processor Arbitration Interface

Figure 20-3 shows a typical arbitration exchange.

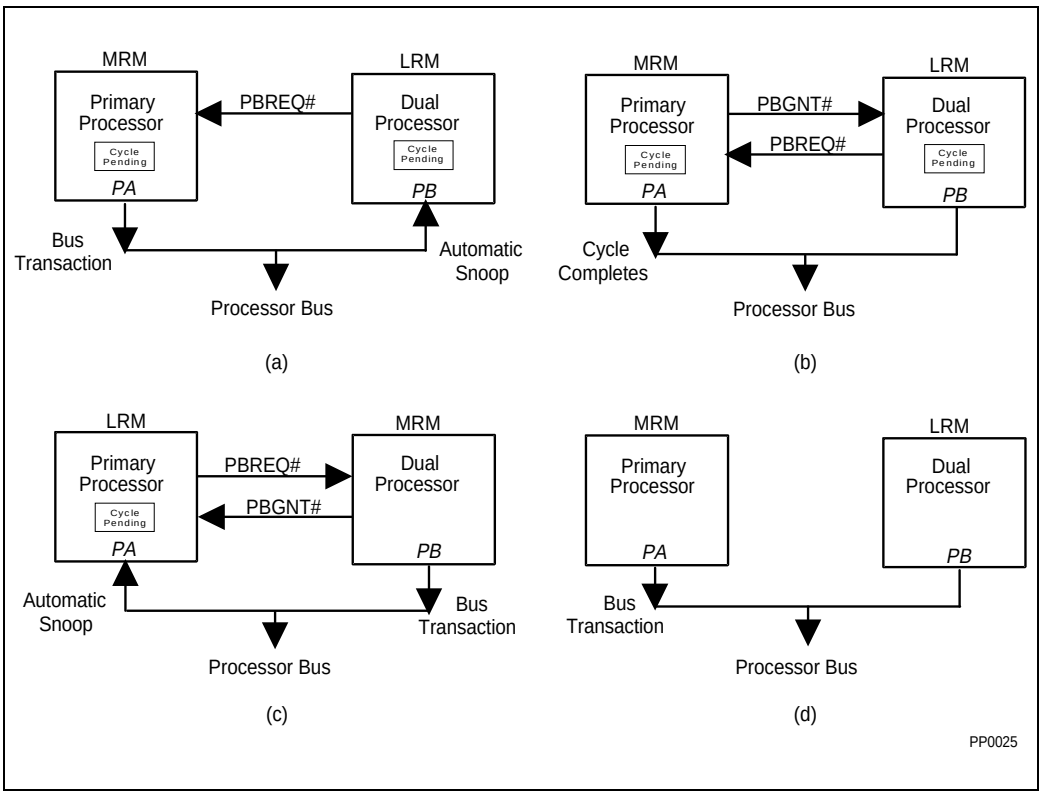


Figure 20-3. Typical Dual Processor Arbitration Example

Diagram (a) of Figure 20-3 shows *PA* running a cycle on the processor bus with a transaction pending. At the same time, *PB* has a cycle pending and has asserted the PBREQ# pin to notify *PA* that *PB* needs the bus.

Diagram (b) of Figure 20-3 shows *PA's* cycle completing with an **NA#** or the last **BRDY#**. Note here that *PA* does not run the pending cycle, instead, *PA* grants the bus to *PB* to allow *PB* to run its pending cycle.

In Diagram (c) of Figure 20-3, *PB* is running the pending transaction on the processor bus, and *PA* asserts a request for the bus to *PB*. The bus is granted to *PA*, and Diagram (d) of Figure 20-3 shows *PA* running the last pending cycle on the bus.

20.1.2.3. DP ARBITRATION FROM A PARKED BUS

When both processors are idle on the CPU bus, and the LRM wants to issue an ADS#, there is an arbitration delay in order that it may become the MRM. The following figure shows how the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) DP arbitration mechanism handles this case.

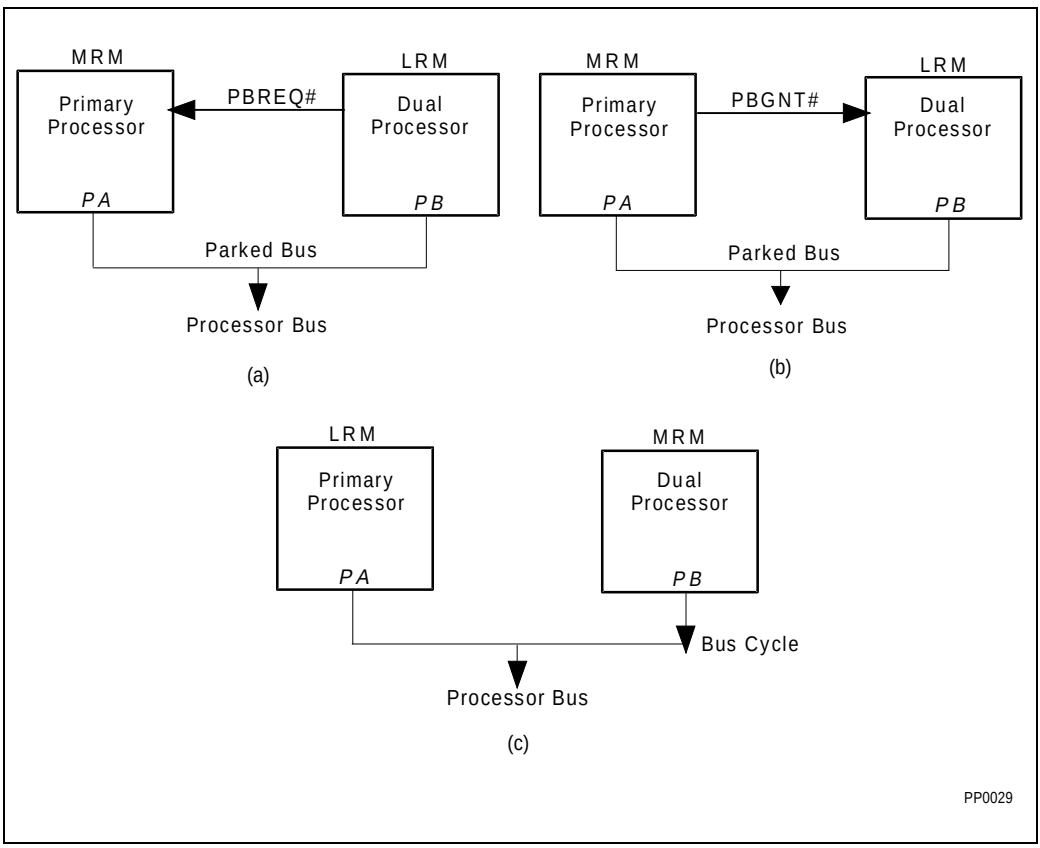


Figure 20-4. Arbitration from LRM to MRM When Bus is Parked

This example shows the arbitration necessary for the LRM to gain control of the idle CPU bus in order to drive a cycle. In this example, *PA* is the Primary processor, and *PB* is the Dual processor.

Diagram (a) of Figure 20-4 shows *PB* requesting the bus from the MRM (*PA*). Diagram (b) of Figure 20-4 shows *PA* granting control of the bus to *PB*. Diagram (c) of Figure 20-4 shows *PB*, now the MRM, issuing a cycle.

20.1.3. Dual Processor Cache Consistency

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) incorporates a mechanism to maintain cache coherency with the Dual processor. The mechanism allows a dual processor to be inserted into the upgrade socket without special consideration to the system hardware or software. The presence or absence of the dual processor is totally transparent to the system.

20.1.3.1. BASIC CACHE CONSISTENCY MECHANISM

A private snoop interface has been added to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). The interface consists of two pins (PHIT#, PHITM#) that only connect between the two sockets. The dual processors will arbitrate for the system bus via two private arbitration pins (PBREQ#, PBGNT#).

The LRM processor will initiate a snoop sequence for all ADS# cycles to memory that are initiated by the MRM. The LRM processor will assert the private hit indication (PHIT#) if the data accessed (read or written) by the MRM matches a valid cache line in the LRM. In addition, if the data requested by the MRM matches a valid cache line in the LRM that is in the modified state, the LRM will assert the PHITM# signal. The system snooping indication signals (HIT#, HITM#) will not change state as a result of a private snoop.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will support system snooping via the EADS# pin in the same manner that the Pentium processor (510\60, 567\66) supports system snooping.

The private snoop interface is bi-directional. The processor that is currently the MRM will sample the private snoop interface, while the processor that is the LRM will drive the private snoop signals.

The MRM will initiate a self backoff sequence if the MRM detects an assertion of the **PHITM#** signal while running a bus cycle. The self backoff sequence will involve the following steps:

1. The MRM will allow the cycle that was requested on the bus to finish. However, the MRM will ignore the data returned by the system.
2. The MRM-LRM will exchange ownership of the bus (as well as MRM-LRM state) to allow the LRM to write the modified data back to the system.
3. The bus ownership will exchange one more time to allow the original bus master ownership of the bus. At this point the MRM will retry the cycle, receiving the fresh data from the system or writing the data once again.

The MRM will use an assertion of the PHIT# signal as an indication that the requested data is being shared with the LRM. Independent of the WB/WT# pin, a cache line will be placed in the cache in the shared state if PHIT# is asserted. This will make all subsequent writes to that line externally visible until the state of the line becomes exclusive (E or M states). In a uniprocessor system, the line may have been placed in the cache in the E state. In this situation, all subsequent writes to that line will not be visible on the bus until the state is changed to I.

20.1.3.2. CACHE CONSISTENCY INTERFACE

The following diagram details the hardware cache consistency interface**.**

NOTE

For proper operation, PHIT# and PHITM# must not be loaded by the system.

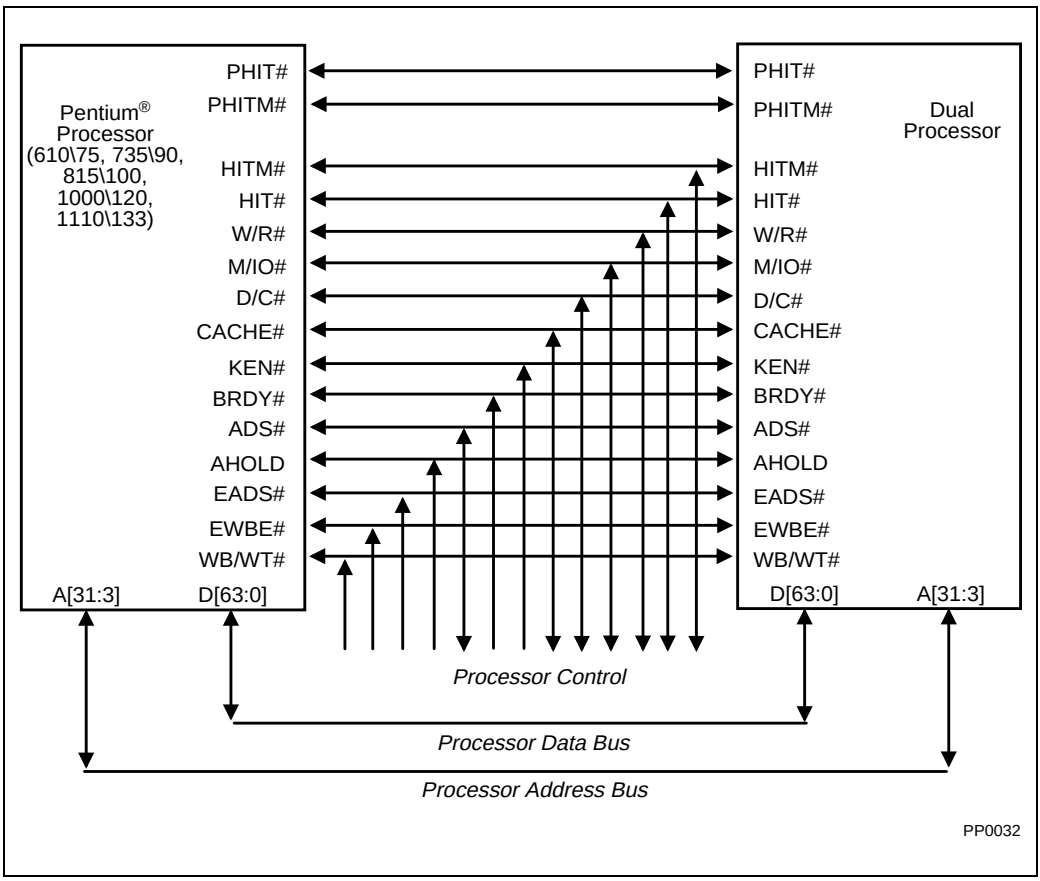


Figure 20-5. Cache Consistency Interface

20.1.3.3. PIN MODIFICATIONS DUE TO THE DUAL PROCESSOR

20.1.3.4. LOCKED CYCLES

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) implements atomic bus transactions by asserting the LOCK# pin. Atomic transactions can be initiated explicitly in software by using a LOCK prefix on specific instructions. In addition, atomic cycles may be initiated implicitly for instructions or transactions that perform locked read-modify-write cycles. By asserting the LOCK# pin, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) indicates to the system that the bus transaction in progress can not be interrupted.

20.1.3.4.1. Locked Cycle Cache Consistency

Lock cycles adhere to the following sequence:

1. Like the Pentium processor (510\60, 567\66), an unlocked write back will occur if a cache line is in the modified state in the MRM processor. Two unlocked write back cycles may be required if the locked item spans two cache lines that are both in the modified state.
2. A locked read to a cache line that is in the shared, exclusive or invalid state is always run on the system bus. The cache line will always be moved to the invalid state at the completion of the cycle. A locked read cycle that is run by the MRM could hit a line that is in the modified state in the LRM. In this case, the LRM will assert the PHITM# signal indicating that the requested data is modified in the LRM data cache. The MRM will complete the locked read, but will ignore the data returned by the system. The components will exchange ownership of the bus, allowing the Modified cache line to be written back with LOCK# still active. The sequence will complete with the original bus owner re-running the locked read followed by a locked write. The sequence would be as shown in Figure 20-6 below.

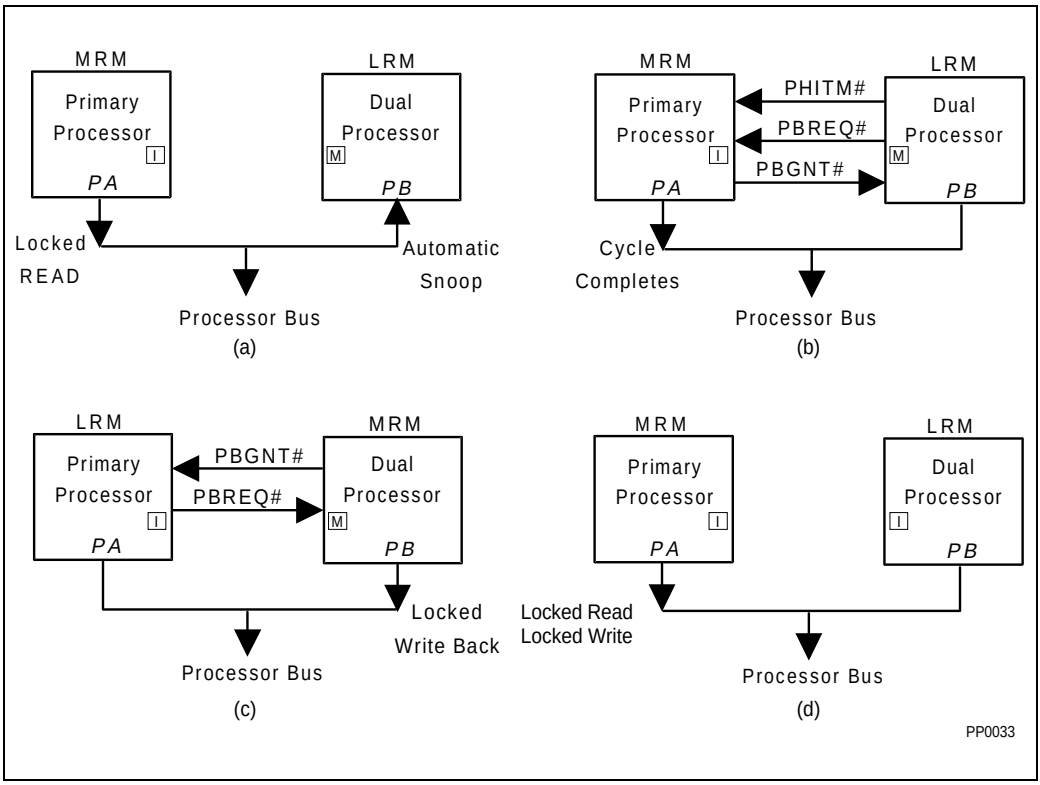


Figure 20-6. DP Cache Consistency for Locked Accesses

In Figure 20-6, the small box inside each CPU indicates the state of an individual cache line in the sequence shown above. Diagram (c) of Figure 20-6 shows the locked write back occurring as a result of the inter-processor snoop hit to the M state line.

20.1.3.5. EXTERNAL SNOOP EXAMPLES

20.1.3.5.1. Example 1: During a Write to an M State Line

The following set of diagrams illustrates the actions performed when one processor attempts a write to a line that is contained in the cache of the other processor. In this situation, the cached line is in the M state in the LRM processor. The external snoop and the write are to the same address in this example.

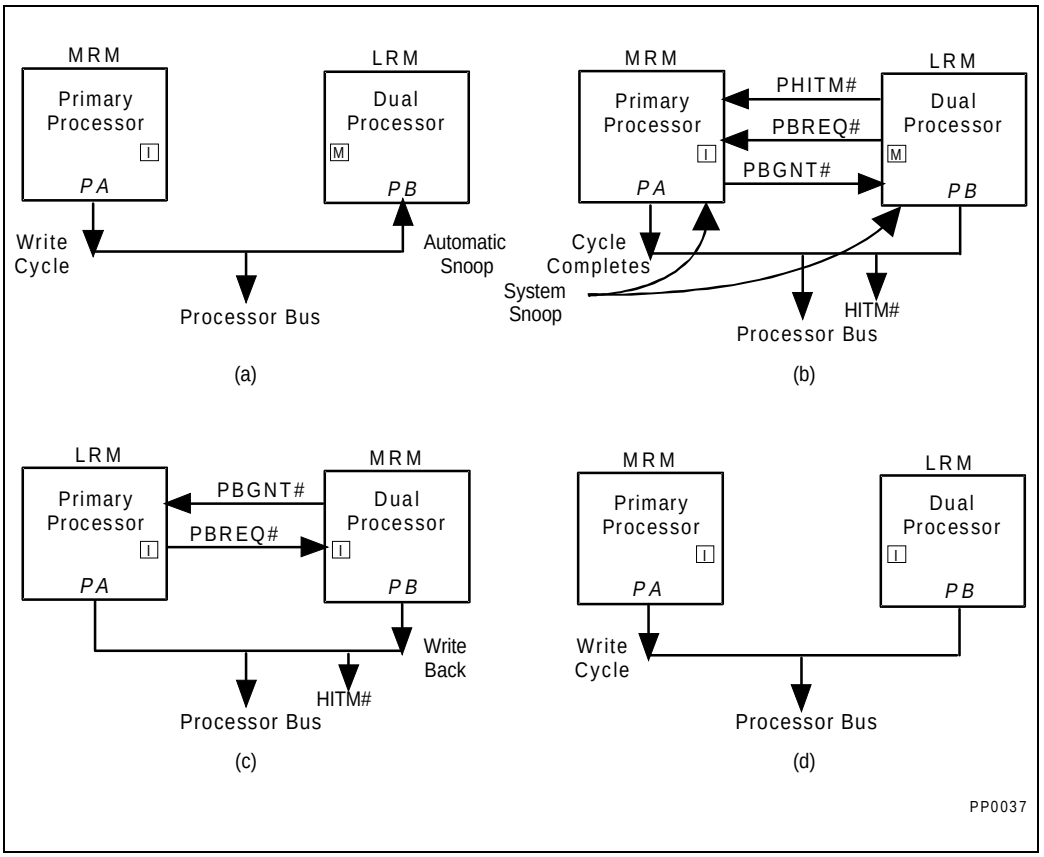


Figure 20-7. DP Cache Consistency for External Snoops

In this example, *PA* is the Primary processor, and *PB* is the Dual processor.

In diagram (a) of Figure 20-7, processor *PA* starts a write cycle on the bus to a line that is in the M state in processor *PB*. Processor *PB* notifies *PA* that the write transaction has hit an M state line in diagram (B) of Figure 20-7 by asserting the PHITM# signal. The MRM (*PA*) completes the write cycle on the bus as if the LRM processor did not exist.

In this example, an external snoop happens just as the write cycle completes on the bus, but before *PB* has a chance to write the modified data back to the system memory. Diagram (b) of Figure 20-7 shows *PB* asserting the HITM# signal, informing the system that the snoop address is cached in the dual processing pair and is in the modified state. The external snoop in this example is hitting the same line that caused the PHITM# signal to be asserted.

Diagram (c) of Figure 20-7 shows that an arbitration exchange has occurred on the bus, and *PB* is now the MRM. Processor *PB* writes back the M state line, and it will appear to the system as if a single processor was completing a snoop transaction.

Finally, diagram (d) of Figure 20-7 shows processor *PA* re-running the original write cycle after *PB* has granted the bus back to *PA*.

20.1.3.5.2. Example 2: During an MRM Self-Backoff

The following diagrams show an example where an external snoop hits an M state line during a self backoff sequence.

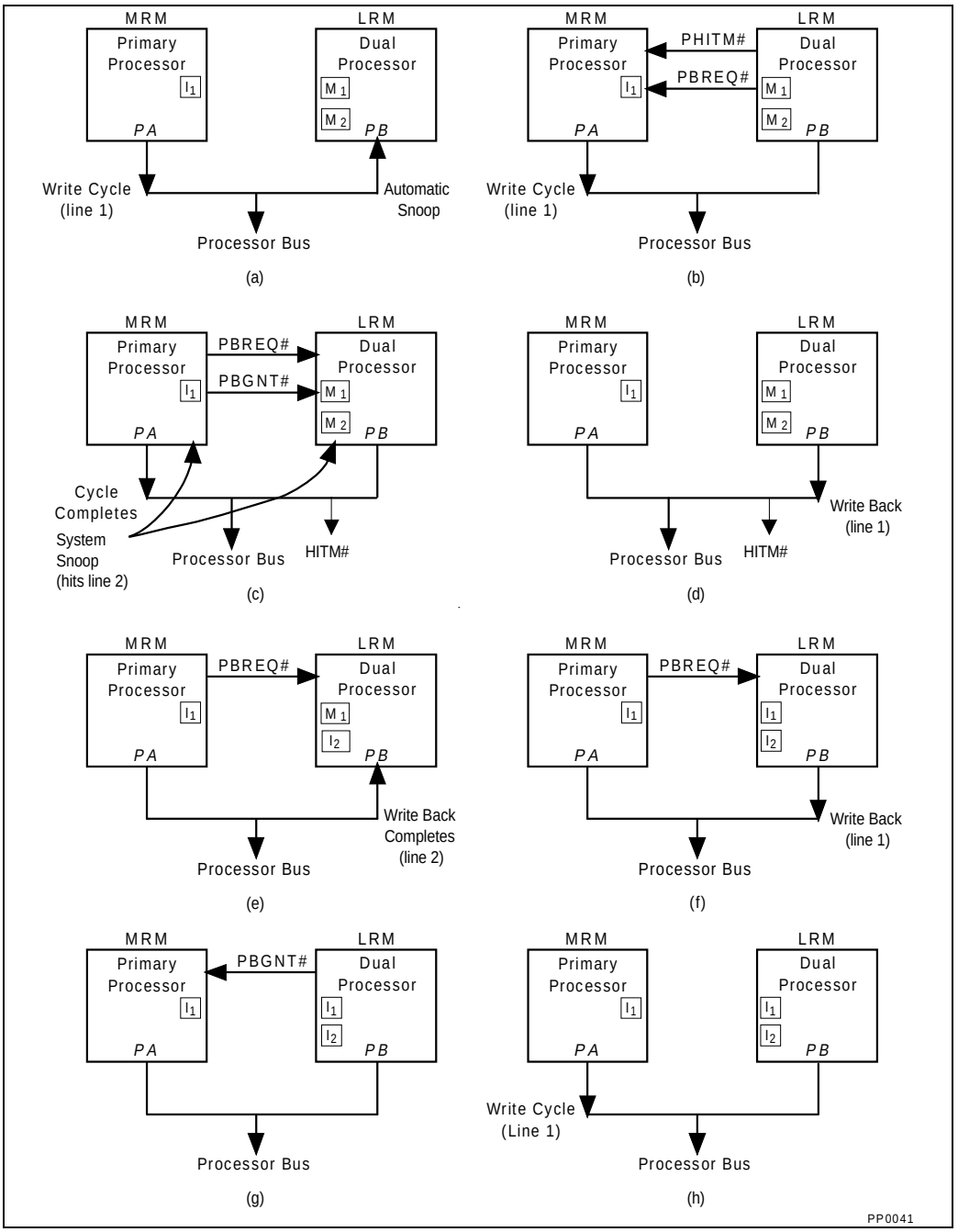


Figure 20-8. DP Cache Consistency for External Snoops

In this example, *PA* is the Primary processor, and *PB* is the Dual processor.

In diagram (a) of Figure 20-8 processor *PA* initiates a write cycle that hits a line that is modified in processor *PB*. In diagram of (b) of Figure 20-8, processor *PB* notifies *PA* that the line is modified in its cache by asserting the PHITM# signal.

Diagram (c) of Figure 20-8 shows an external snoop occurring just as the bus arbitration has exchanged ownership of the bus. Processor *PB* asserts the HITM# signal to notify the system that the external snoop has hit a line in the cache. In this example, the external snoop hits a different line that was just hit on the private snoop.

In diagram (d) of Figure 20-8, processor *PB* takes ownership of the processor bus from *PA*. Processor *PB* initiates a writeback of the data just hit on the external snoop even though a writeback due to the private snoop is pending. The external snoop causes processor *PB* to delay the write back that was initiated by the private snoop (to line 1).

Diagram (f) of Figure 20-8 shows the write back of the modified data hit during the initial private snoop. Processor *PA* then restarts the write cycle for the second time, and completes the write cycle in Diagram (h) of Figure 20-8.

20.1.3.6. STATE TRANSITIONS DUE TO DP CACHE CONSISTENCY

The following tables outline the state transitions that a cache line can encounter during various conditions.

Table 20-1. Read Cycle State Transitions Due to DP

Present State	Pin Activity	Next State	Description
M	n/a	M	Read hit. Data is provided to the processor core by the cache. No bus activity.
E	n/a	E	Read hit. Data is provided to the processor core by the cache. No bus activity.
S	n/a	S	Read hit. Data is provided to the processor core by the cache. No bus activity.
I	CACHE#(L) & KEN#(L) & WB/WT#(H) & PHIT#(H) & PWT(L)	E	Cache miss. The cacheability information indicates that the data is cacheable. A bus cycle is requested to fill the cache line. PHIT#(H) indicates that the data is not shared by the LRM processor.
I	CACHE#(L) & KEN#(L) & [WB/WT#(L) + PHIT#(L) + PWT(H)]	S	Cache miss. The line is cacheable and a bus cycle is requested to fill the cache line. In this case, either the system or the LRM is sharing the requested data.
I	CACHE#(H) + KEN#(h)	I	Cache miss. The system or the processor indicates that the line is not cacheable.

NOTE:

The assertion of PHITM# would cause the requested cycle to complete as normal, with the requesting processor ignoring the data returned by the system. The LRM processor would write the data back and the MRM would retry the cycle. This is called a self backoff cycle.

Table 20-2. Write Cycle State Transitions Due to DP

Present State	Pin Activity	Next State	Description
M	n/a	M	Write hit. Data is written directly to the cache. No bus activity.
E	n/a	M	Write hit. Data is written directly to the cache. No bus activity.
S	PWT(L) & WB/WT#(H)	E	Write hit. Data is written directly to the cache. A write-through cycle will be generated on the bus to update memory and invalidate the contents of other caches. The LRM will invalidate the line if it is sharing the data. The state transition from S to E occurs AFTER the write completes on the processor bus.
S	PWT(H) + WB/WT#(L)	S	Write hit. Data is written directly to the cache. A write-through cycle will be generated on the bus to update memory and invalidate the contents of other caches. The LRM will invalidate the line if it is sharing the data.
I	n/a	I	Write miss (the Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) does not support write allocate). The LRM will invalidate the line if it is sharing the data.

Table 20-3. Inquire Cycle State Transitions Due to External Snoop

Present State	Next State (INV=1)	Next State (INV=0)	Description
M	I	S	Snoop hit to an M state line. HIT# and HITM# will be asserted, followed by a write-back of the line.
E	I	S	Snoop hit. HIT# will be asserted.
S	I	S	Snoop hit. HIT# will be asserted.
I	I	I	Snoop miss.

Table 20-4. State Transitions in the LRM Due to DP "Private" Snooping

Present State	Next State (MRM Write)	Next State (MRM Read)	Description
M	I	S	Snoop hit to an M state line. PHIT# and PHITM# will be asserted, followed by a write-back of the line. Note that HIT# and HITM# will NOT be asserted.
E	I	S	Snoop hit. PHIT# will be asserted.
S	I	S	Snoop hit. PHIT# will be asserted.
I	I	I	Snoop miss.

20.2. DESIGNING WITH SYMMETRICAL DUAL PROCESSORS

Figure 20-9 shows how a typical system might be configured to support the Dual processor.

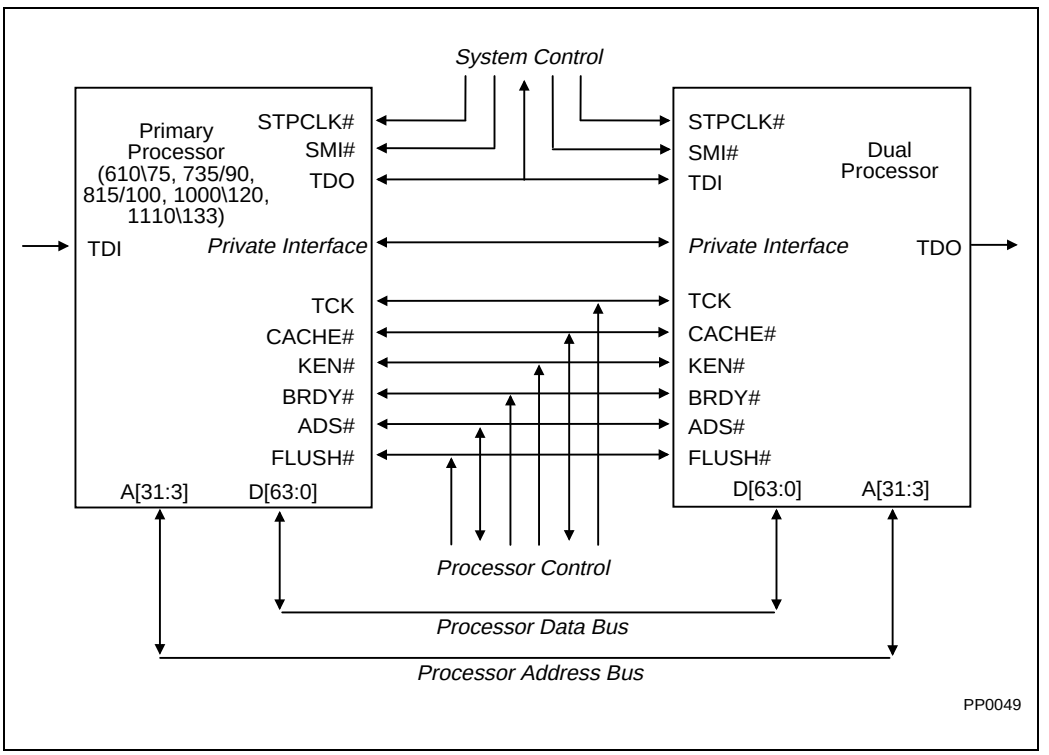


Figure 20-9. Dual Processor Configuration

Refer to Table 20-8 for a complete list of dual processor signal connection requirements.

20.2.1. Dual Processor Bus Interface

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in the dual processor configuration is designed to have an identical bus interface to a standard Pentium processor (510\60, 567\66). In addition, it supports core clock frequency to bus clock frequency ratios of 1/2 and 2/3. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in dual processor mode has the capability to run the following types bus of cycles:

- Single reads and writes from one processor.
- Burst reads and writes from one processor.
- Address pipe-lining with up to two outstanding bus cycles from one processor.
- Inter-processor address pipe-lining with up to two outstanding bus cycles, one from each processor.

All cycles run by the two processors are clock accurate to corresponding Pentium processor (510\60, 567\66) bus cycles.

20.2.1.1. INTRA- AND INTER-PROCESSOR PIPE-LINING

In uni-processor mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports bus pipe-lining with the use of the NA# pin. The bus pipe-lining concept has been extended to the dual processor pair by allowing inter-CPU pipe-lining. This mechanism allows an exchange between LRM and MRM on assertions of NA#.

When NA# is sampled low, the current MRM processor may drive one more cycle onto the bus or it may grant the address bus and the control bus to the LRM. The MRM will give the bus to the LRM only if its current cycle can have another cycle pipe-lined into it.

The cacheability (KEN#) and cache policy (WB/WT#) indicators for the current cycle are sampled either in the same clock that NA# is sampled or with the first BRDY# of the current cycle, whichever comes first.

There are no restrictions on NA# due to dual processing mode.

Inter-CPU pipe-lining will not be supported in some situations as shown in Table 20-5.

Table 20-5. Primary and Dual Processor Pipe-lining

		Primary and Dual Processor Pipe-lining		
Cycle Types		Inter-CPU	Intra-CPU	
First Cycle	Pipe-lined Cycle	Primary<>Dual	Primary<>Primary	Dual<>Dual
Write Back	X	No	No	No
LOCK#	X	No	No	No
X	Write Back	No	No	No
X	LOCK#	No	No	No
Write	***Write***	***No***	***Yes***	***Yes***
Write	Read	Yes	Yes	Yes
Read	Write	Yes	Yes	Yes
Read	Read	Yes	Yes	Yes
I/O	***I/O****	***Yes***	***No***	***No***

NOTE:

*I/O write cycles may not be inter-processor pipe-lined into I/O write cycles.

The table indicates that, unlike the Pentium processor (510\60, 567\66) or a uni-processor Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system, back-to-back write cycles will never be pipe-lined between the two processors.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) alone may pipe-line I/O cycles into non-I/O cycles, non-I/O cycles into I/O cycles, and I/O cycles into I/O cycles only for OUTS or INS (e.g. string instructions). I/O cycles may be pipe-lined in any combination (barring writes into writes) between the Primary and Dual processors.

20.2.1.2. FLUSH# CYCLES

The on-chip caches can be flushed by asserting the FLUSH# pin. The FLUSH# pin must be connected together to both the Primary and Dual processor parts. All cache lines in the instruction cache as well as all lines in the data cache that are not in the modified state will be invalidated when the FLUSH# pin is asserted. All modified lines in the data cache will be written back to system memory and then marked as invalid in the data cache. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will run a special bus cycle indicating that the flush process has completed.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) incorporates the following mechanism to present a unified view of the cache flush operation to the system when used with a Dual processor part:

1. FLUSH# is asserted by the system.
2. The Dual processor requests the bus (if it is not already MRM when FLUSH# is recognized). The Dual processor will always perform the cache flush operation first, but will not run a flush special cycle on the system bus.

3. The Dual processor completes write backs of modified cache lines, and invalidates all others.
4. Once the Dual processor caches are completely invalid, the processor grants the bus to the Primary processor.
5. The Primary processor completes any pending cycles. The Primary processor may have outstanding cycles if the Dual processor initiated its flush operation prior to the Primary processor completing pending operations.
6. Primary processor flushes both of its internal caches and runs the cache flush special cycle. The Primary processor maintains its status of MRM. The Dual processor halts all code execution while the Primary processor is flushing its caches, and does not begin executing code until it recognizes the flush acknowledge special cycle.

The atomic flush operation assumes that the system can tolerate potentially longer interrupt latency during flush operations. The interrupt latency in a dual processor system can be double the interrupt latency in a single processor system during flush operations.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) primary cache can be flushed using the *WBINVD* instruction. In a dual processor system, the *WBINVD* instruction only flushes the cache in the processor that executed the instruction. The other processor's cache will be intact.

If the FLUSH# signal is de-asserted before the corresponding FLUSH ACK cycle, the FLUSH# signal **must** not be asserted again until the FLUSH ACK cycle is generated. This requirement does not apply to a uniprocessor system. In a dual processor system, a single FLUSH ACK cycle is generated after the caches in both processors have been flushed.

WARNING

If FLUSH# is recognized active a second time by the Primary and Dual processors prior to the flush acknowledge special cycle, the private bus arbitration state machines will be corrupted.

If the FLUSH# signal is asserted in dual processing mode, it must be deasserted at least one clock prior to BRDY# of the FLUSH Acknowledge cycle to avoid DP arbitration problems.

20.2.1.3. ARBITRATION EXCHANGE — WITH BUS PARKING

The dual processor pair supports a number of different types of bus cycles. Each processor can run single-transfer cycles or burst-transfer cycles similar to the Pentium processor (510\60, 567\66). A processor can only initiate bus cycles if it is the MRM. To gain ownership of the bus, the LRM processor will request the bus from the MRM processor by asserting PBREQ#.

In response to PBREQ# the MRM will grant the address and the control buses to the LRM by asserting PBGNT#. If NA# is not asserted or if the current cycle on the bus is not capable of being pipe-lined, the MRM will wait until the end of the active cycle before granting the bus to the LRM. Once PBGNT# is asserted, since the bus is idling, the LRM will immediately

become the MRM. While the MRM, the processor owns the address and the control buses and can therefore start a new cycle.

20.2.1.4. BOFF#

If BOFF# is asserted, the dual processor pair will immediately (in the next clock) float the address, control, and data buses. 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#, Primary and Dual processors will float the same pins as it does when HOLD is active.

The Primary and Dual processors may reorder cycles after a BOFF#. The reordering will occur if there is inter-CPU pipe-lining at the time of the BOFF#, but the system cannot change the cacheability of the cycles after the BOFF#. Note that there could be a change of bus ownership transparent to the system while the processors are in the backed-off state. Table 20-6 illustrates the flow of events which would result in cycle re-ordering due to BOFF#:

Table 20-6. Cycle Reordering Due to BOFF#

Time*	Processor A	System	Processor B
0	ADS# driven	--	--
1	--	NA# active	--
2	--	--	ADS# driven
3	Bus float	BOFF# active	Bus float
4	--	EADS# active	--
5	--	--	HITM# driven
6	--	BOFF# inactive	--
7	--	--	Write back 'M' data
8	--	BRDY#s	--
9	--	--	Restart ADS#
10	Restart ADS#	--	--

NOTE:

*Time is merely sequential, NOT measured in CLKs.

20.2.1.5. BUS HOLD

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports a bus hold/hold acknowledge protocol using the HOLD and HLDA signals. When the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) completes all outstanding bus cycles, it will release the bus by floating the external bus, and driving HLDA active. HLDA will normally be driven two clocks after the later of the last BRDY# or HOLD being asserted, but may be up to six clocks due to active internal APIC cycles. Because of this, it

is possible that an additional cycle may begin after HOLD is asserted but before HLDA is driven. Therefore, asserting HOLD does not prevent a DP arbitration from occurring before HLDA is driven out. Even if an arbitration switch occurs, no new cycles will be started after HOLD has been active for two clocks.

20.2.2. Dual Processing Power Management

20.2.2.1. STPCLK#

The Primary and Dual processor STPCLK# signals may be tied together or left separate. Refer to Chapter 30 for more information on stop clock and Autohalt.

20.2.2.2. SYSTEM MANAGEMENT MODE

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports system management mode (SMM) with a processor inserted in the upgrade socket. SMM provides a means to implement power management functions as well as operating system independent functions. SMM in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) consists of an interrupt (SMI), an alternate address space and an instruction (RSM). SMM is entered by asserting the SMI# pin or delivering the SMI interrupt via the local APIC.

Although SMM functions the same when a Dual processor is inserted in Socket 5, the dual processor operation of the system must be carefully considered. The SMI# pins may be tied together or not, depending upon the power management features supported.

In order to ensure proper SMM operation when a Future Pentium OverDrive processor upgrade is installed in the system, it is recommended that the SMI# and SMIACT# signals be connected together. Refer to Chapter 30 for more details.

20.2.3. Other Dual Processor Considerations

20.2.3.1. STRONG WRITE ORDERING

The ordering of write cycles in the processor can be controlled with the EWBE# pin. During uniprocessor operation, the EWBE# pin is sampled by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) with each BRDY# assertion during a write cycle. The processor will stall all subsequent write operations to E or M state lines if EWBE# is sampled inactive. If the EWBE# pin is sampled inactive, it will continue to be sampled on every clock until it is found to be active.

In dual processing mode, each processor will track EWBE# independently of bus ownership. EWBE# is sampled and handled independently between the two processors. Only the processor which owns the bus (MRM) samples EWBE#. Once sampled inactive, the CPU will stall subsequent write operations.

20.2.3.2. BUS SNARFING

The dual processor pair does not support cache-to-cache transfers (bus snarfing). If a processor *PB* requires data that is modified in processor *PA*, processor *PA* will write the data back to memory. After *PA* has completed the data transfer, *PB* will run a read cycle to memory. Where *PA* is either the Primary or the Dual processor, and *PB* is the other processor.

20.2.3.3. INTERRUPTS

A processor may need to arbitrate for the use of the bus as a result of an interrupt. However, from the simple arbitration model used by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), an interrupt is not a special case. There is no interaction between dual processor support and the interrupt model in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

20.2.3.4. INIT SEQUENCES

The INIT operation in dual-processor mode is exactly the same as in uni-processor mode. The two INIT pins must be tied together. However, in dual processor mode, the Primary processor must send an IPI and a starting vector to the Dual processor via the local APIC modules.

20.2.3.5. BOUNDARY SCAN

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports the full IEEE JTAG specification. The system designer is responsible to configure an upgrade ready system in such a way that the addition of a Dual processor in Socket 5 allows the boundary scan chain to functional as normal. This could be implemented with a jumper in Socket 5 that connects the TDI and TDO pins. The jumper would then be removed when the dual processor is inserted.

Alternatively, Socket 5 could be placed near the end of the boundary scan chain in the system. A multiplexer in the system boundary scan logic could switch between the TDO of the Primary and the dual processors as a Dual processor part is inserted. An illustration of this approach is shown in Figure 20-10.

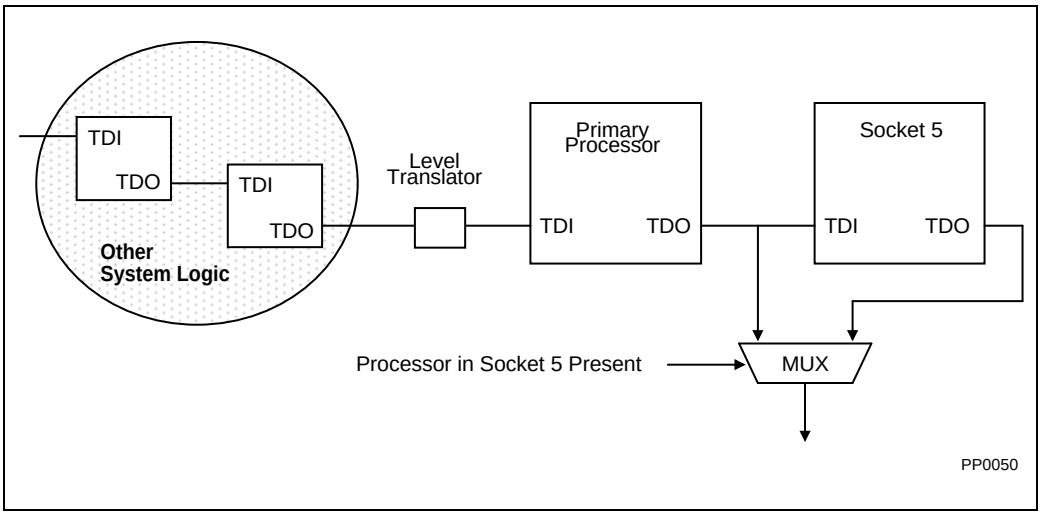


Figure 20-10. Dual Processor Boundary Scan Connections

20.2.3.6. PRESENCE OF A PROCESSOR IN SOCKET 5

The Dual processor or future Pentium OverDrive processor drives the DPEN# signal low during RESET to indicate to the Primary processor that a processor is present in Socket 5. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) samples this line during RESETs falling edge.

DPEN# shares a pin with the APIC PICD0 signal.

20.2.3.7. MRM PROCESSOR INDICATION

In a DP system, the D/P# (Dual processor/Primary processor Indication) signal indicates which processor is running a cycle on the bus. Table 20-7 shows how the external hardware can determine which CPU is the MRM.

Table 20-7. Using D/P# to Determine MRM

D/P#	Bus Owner
0	Primary processor is MRM
1	Dual processor is MRM

D/P# can be sampled by the system with ADS# to determine which processor is driving the cycle on the bus.

D/P# is driven only by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) when operating as the Primary processor. Because of this, this signal is never driven by the

Dual processor and does not exist on the fFuture Pentium OverDrive processor. When the future Pentium OverDrive processor is installed, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) continues to drive the D/P# signal high despite being "shut down."

20.2.4. Dual Processor Pin Functions

Refer to Chapter 17 for brief descriptions of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signals added to support new features.

All the inputs pins described in section Chapter 17 are sampled with bus clock or test clock, and therefore, must meet setup and hold times with respect to the rising edge of the appropriate clock. In the DP configuration, the RESET and FLUSH# pins have been changed to be synchronous (i.e. meet setup and hold times). There have been no changes to the other existing input pins.

If the FLUSH# signal is deasserted before the corresponding FLUSH ACK cycle, the FLUSH# signal must not be asserted again until the FLUSH ACK cycle is generated. This requirement does not apply to a uni-processor system. In a dual processor system, a single FLUSH ACK cycle is generated after the caches in both processors have been flushed.

All system output pins will be driven from the rising edge of the bus clock and will meet maximum and minimum valid delays with respect to the bus clock. TDO is driven with respect to the rising edge of TCK and PICD0-1 are driven with respect to the rising edge of PICCLK.

The following table summarizes the functional changes of all the pins in DP mode.

Table 20-8. DP Pin Functions vs. Pentium® Processor (510\60, 567\66)

Pin Name	I/O	Load (Note 1)	Same? (Note 2)	Tied Together? (Note 3)	Comments
A[31:3]	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates these signals for one CLK.
A20M#	I	Y	Y	Yes	Used in virtual mode and possibly in real mode by DOS and DOS extenders. Internally masked by the Dual processor. It is necessary to connect this signal to Socket 5 in order for proper Future Pentium® OverDrive® processor operation.

Table 20-8. DP Pin Functions vs. Pentium® Processor (510\60, 567\66) (Contd.)

Pin Name	I/O	Load (Note 1)	Same? (Note 2)	Tied Together? (Note 3)	Comments
ADS#, ADSC#	I/O O	Y	N	Yes	ADS# and ADSC# are tristated by the LRM processor in order to allow the MRM processor to begin driving them. There are no system implications.
AHOLD	I	Y	Y	Yes	
AP	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
APCHK#	O	N	Y	No	Requires a system OR function.
BE[7:5]# BE[4:0]#	O I/O	Y Y	N N	Yes Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates these signals for one CLK. BE[3:0]# are used by the local APIC modules to load the APIC_ID at RESET. BE[3:0]# will be tristated by the Primary and Dual processors during RESET.
BF	I	Y	n/a	Yes	
BOFF#	I	Y	Y	Yes	
BP[3:0]	O	N	N	No	BP[3:0] will now only indicate breakpoint match in the I/O clock. Each processor must have different breakpoints. Note that BP[1:0] are mux'd with PM[1:0].
BRDY#, BRDYC#	I	Y	Y	Yes	
BREQ	O	Y	N	Yes	The MRM drives this signal as a combined bus cycle request for itself and the LRM.
BUSCHK#	I	Y	Y	Yes	
CACHE#	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.

Table 20-8. DP Pin Functions vs. Pentium® Processor (510\60, 567\66) (Contd.)

Pin Name	I/O	Load (Note 1)	Same? (Note 2)	Tied Together? (Note 3)	Comments
CLK	I	Y	Y	Yes	Both processors must use the same system clock.
CPUTYP	I	Y	n/a	No	
D/C#	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
D/P#	O	n/a	n/a	No	The Primary processor always drives this signal. This output is not defined on the Dual processor or Future Pentium OverDrive processor.
D[63:0]	I/O	Y	Y	Yes	
DP[7:0]	I/O	Y	Y	Yes	
EADS#	I	Y	Y	Yes	
EWBE#	I	Y	Y	Yes	This signal is sampled active with BRDY#, but inactive asynchronously. For optimized performance (minimum number of write E/M stalls) the chip set/platform should allow a dead clock between buffer going empty to buffer going full. This will allow this signal to be completely independent between the two processors and not have one stall internal cache writes due to the other filling the external buffer.
FERR#	O	Y	Y	Yes	Used for DOS floating point compatibility. The Primary processor will drive this signal. The Dual processor will never drive this signal.
FLUSH#	I	Y	Y	Yes	In a DP system, the flush operation will be atomic with a single flush acknowledge bus cycle. Therefore, FLUSH# must not be re-asserted until the corresponding FLUSH ACK cycle is generated.
FRCMC#	I	N	Y	Yes	Both processors must be in Master mode. A processor in the Socket 5 cannot be used as a Checker.

Table 20-8. DP Pin Functions vs. Pentium® Processor (510\60, 567\66) (Contd.)

Pin Name	I/O	Load (Note 1)	Same? (Note 2)	Tied Together? (Note 3)	Comments
HIT#	I/O	Y	N	Yes	This signal is asserted by the MRM based on the combined outcome of the inquire cycle between the two processors.
HITM#	I/O	Y	N	Yes	See HIT#.
HLDA	I/O	Y	N	Yes	Driven by the MRM.
HOLD	I	Y	Y	Yes	
IERR#	O	N	Y	No	
IGNNE#	I	Y	Y	Yes	The Dual processor will ignore this signal.
INIT	I	N	N	Yes	In DP mode, the Dual processor requires an IPI during initialization.
INTR/LINT0	I	N	N	May Be	If the APIC is enabled, then this pin is a local interrupt. If the APIC is hardware disabled, this pin function is not changed.
INV	I	Y	Y	Yes	
KEN#	I	Y	Y	Yes	
LOCK#	I/O	Y	N	Yes	The LRM samples the value of LOCK#, and drives the sampled value in the clock it gets the ownership of the DP bus. If sampled active, then the LRM will keep driving the LOCK# signal until ownership changes again.
M/IO#	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
NA#	I	Y	Y	Yes	
NC	n/a	N	Y	No	
NMI/LINT1	I	N	Y	May Be	If the APIC is enabled, then this pin is a local interrupt. If the APIC is hardware disabled, this pin function is not changed.
PBGNT#	I/O	n/a	n/a	Yes	This signal is always driven by one of the processors.

Table 20-8. DP Pin Functions vs. Pentium® Processor (510\60, 567\66) (Contd.)

Pin Name	I/O	Load (Note 1)	Same? (Note 2)	Tied Together? (Note 3)	Comments
PBREQ#	I/O	n/a	n/a	Yes	This signal is always driven by one of the processors.
PCD	O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
PCHK#	O	N	Y	May Be	May be wire-AND'd together in the system, tied together, or the chip set may have two PCHK# inputs for DP data parity.
PEN#	I	Y	Y	Yes	
PHIT#	I/O	n/a	n/a	Yes	This signal is always driven by one of the processors.
PHITM#	I/O	n/a	n/a	Yes	This signal is always driven by one of the processors.
PICCLK	I	Y	n/a	Yes	
PICD[1:0]	I/O	Y	n/a	Yes	
PM[1:0]	O	N	N	No	Each processor may track different performance monitoring events. Note that PM[1:0] are mux'd with BP[1:0].
PRDY	O	N	Y	No	
PWT	O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
R/S#	I	N	Y	No	
RESET	I	Y	Y	Yes	In DP mode, RESET must be synchronous to the CPU CLK which goes to the Primary and Dual processors.
SCYC	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
SMI#	I	N	Y	May Be	Refer to Chapter 30.
SMIACT#	O	N	Y	Yes	Refer to Chapter 30.
STPCLK#	I	n/a	n/a	May Be	Refer to Chapter 30.
TCK	I	n/a	n/a	May Be	System dependent

Table 20-8. DP Pin Functions vs. Pentium® Processor (510\60, 567\66) (Contd.)

Pin Name	I/O	Load (Note 1)	Same? (Note 2)	Tied Together? (Note 3)	Comments
TDI	I	n/a	n/a	No	System dependent
TDO	O	n/a	n/a	No	System dependent
TMS	I	n/a	n/a	May Be	System dependent
TRST#	I	n/a	n/a	May Be	System dependent
V_{CC}	I	N	N	Yes	V_{CC} on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) must be connected to 3.3V.
$V_{CC}5$	I	N	Y	no	Two $V_{CC}5$ pins remain on the Future Pentium OverDrive processor in order to support a 5V fan/heatsink in the future.
V_{SS}	I	N	Y	Yes	
W/R#	I/O	Y	N	Yes	When the MRM becomes the LRM (and issues PBGNT#), it tristates this signal for one CLK.
WB/WT#	I	Y	Y	Yes	

NOTES:

1. "Load" indicates whether the pin would introduce a capacitive load to the system board due to the dual processor being present.
2. "N" indicates that there is a minor functional change to the pin(s) either as an enhancement to the Pentium processor (510\60, 567\66) or due to dual processor operation.
3. "Yes" means that both processors must see the same value on the pin(s) for proper DP operation. "No" means that the system must provide the signal to each processor independently. "May Be" means that the system designer can choose to provide the signal to both processors or provide independent signals to each processor.

21

Hardware Interface

CHAPTER 21
HARDWARE INTERFACE

21.1. DETAILED PIN DESCRIPTIONS

In addition to the detailed pin descriptions found in Chapter 5 for the Pentium processor family, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports additional signals and extensions to Pentium processor (510\60, 567\66) signals.

This Chapter describes the new signals added to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) over the Pentium processor (510\60, 567\66). This chapter also describes the functionality changes to the Pentium processor (510\60, 567\66) signals due to dual processing, the local APIC, and 3.3V operation.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) adds the following new signals:

- ADSC#, APICEN, BF, BRDYC#, CPUTYP, D/P#, DPEN#, PBGNT#, PBREQ#, PHIT#, PHITM#, PICCLK, PICD1, PICD0, STPCLK#

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) modifies the functionality of the following signals:

- BP, INTR/LINT0, NMI/LINT1, V_{CC}

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when operating in dual processing mode, modifies the functionality of the following signals:

- A20M#, ADS#, BE4#-BE0#, CACHE#, D/C#, FERR#, FLUSH#, HIT#, HITM#, HLDA, IGNNE#, LOCK#, M/IO#, PCHK#, RESET, SCYC, SMIACT#, W/R#

21.1.1. A20M#

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

Signal Description

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as a Dual processor, will ignore the A20M# input.

When Sampled/Driven

No new features.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
CPUTYP	When strapped to V_{CC}, the processor will ignore the A20M# input.

21.1.2. ADS#

ADS#	Address Strobe
	Indication that a new valid bus cycle is currently being driven by the processor.
	Synchronous Input/Output

Signal Description

Refer to the ADS# pin description in Chapter 5.

This signal is normally identical to the ADSC# output. When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an Input/Output when two Pentium processors (610\75, 735\90, 815\100, 1000\120) are operating together in Dual Processing Mode.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADSC#	ADS# is identical to the ADSC# output.
APCHK#	When operating in dual processing mode, APCHK# is driven in response to ADS# for a private snoop.
D/P#	When operating in dual processing mode, D/P# should be sampled with an active ADS#.
SMIACT#	When operating in dual processing mode, SMIACT# should be sampled with an active ADS# and qualified by D/P#.

21.1.3. ADSC#

ADSC#	Additional Address Strobe
	Indicates that a new valid bus cycle is currently being driven by the processor.
	Synchronous Output

Signal Description

This signal is identical to the ADS# output. This signal can be used to relieve tight board timings by easing the load on the address strobe signal.

When Sampled/Driven

Refer to the ADS# signal description.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	ADSC# is identical to the ADS# output.

21.1.4. APCHK#

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

Signal Description

Refer to the APCHK# pin description in Chapter 5.

Address parity is checked during every private snoop between the Primary and Dual processors. Therefore, APCHK# may be asserted due to an address parity error during this private snoop. If an error is detected, APCHK# will be asserted 2 clocks after ADS# for one processor clock period. The system can choose to acknowledge this parity error indication at this time or do nothing.

When Sampled/Driven

APCHK# is valid for one clock and should be sampled two clocks following ADS# and EADS# assertion.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	When operating in dual processing mode, APCHK# is driven in response to a private snoop.
EADS#	APCHK# is driven in response to an external snoop.

21.1.5. APICEN

APICEN	APIC Enable
	This pin enables the APIC on the processor.
	Synchronous Configuration Input
	Needs external pull-up resistors.

Signal Description

APICEN, if sampled high at the falling edge of RESET, enables the on-chip APIC. If it is sampled low, then the on-chip APIC is not enabled and the processor uses the interrupts as if the APIC was not present (Bypass mode).

APICEN must be driven by the system.

When Sampled/Driven

APICEN should be valid and stable two clocks before and after the falling edge of RESET.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BE3#-BE0#	When APICEN is sampled active, BE3#-BE0# are used to sample the APIC ID.
INTR/LINT0	When APICEN is sampled active, this input becomes the APIC local interrupt 0.
NMI/LINT1	When APICEN is sampled active, this input becomes the APIC local interrupt 1.
PICCLK	PICCLK must be tied or driven high when APICEN is sampled low at the falling edge of RESET.
PICD1	APICEN shares a pin with PICD1.
RESET	APICEN is sampled at the falling edge of RESET.

21.1.6. BE4#-BE0#

BE4#-BE0#	Byte Enable Outputs / APIC ID Inputs
	When operating in dual processing mode, BE4# is used to transfer information between the Dual and Primary processors during the atomic Flush operation.
	At RESET, the BE3#-BE0# pins read the APIC ID bits for the Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133).
	After RESET, these pins are byte enables and help define the physical area of memory to I/O accessed.
	BE4#: Synchronous Input/Output, Dual Processing Mode. BE3#-BE0#: Synchronous Configuration Inputs, During RESET. BE3#-BE0#: Synchronous Outputs, Following RESET.

Signal Description

The local APIC module on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) loads its 4-bit APIC ID value from the four least significant byte-enable pins at the falling edge of RESET. The following table shows the four pins that comprise the APIC ID.

APIC ID

APIC ID Register Bit	Pin Latched at RESET
bit 24	BE0#
bit 25	BE1#
bit 26	BE2#
bit 27	BE3#

Loading the APIC ID should be done with external logic that drives the proper address at reset. If the BE3#-BE0# signals are not driven, the APIC ID value will default to 0000 for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and 0001 for the Dual processor.

WARNING

An APIC ID of all 1s is an APIC special case (i.e., a broadcast) and must not be used. Since the Dual processor inverts the lowest order bit of the APIC ID placed on the lowest four BE pins, the value "1110" must not be used when operating in Dual Processing mode.

In a dual-processor configuration, the OEM and Socket 5/Socket 7 should have the four BE pairs tied together. The OEM processor will load the value seen on these four pins at RESET. The dual processor will load the value seen on these pins and automatically invert bit 24 of the APIC ID Register. Thus the two processors will have unique APIC ID values.

The Primary and Dual processors incorporate a mechanism to present an atomic view of the cache flush operation to the system when in dual processing mode. The Dual processor performs the cache flush operation and grants the bus to the Primary processor by PBREQ#/PBGNT# arbitration exchange. The Primary processor then flushes both of its internal caches and runs cache flush acknowledge special cycle by asserting BE4#, to indicate to the external system that the cache line entries have been invalidated. The Dual processor halts all code execution while the Pentium processor is flushing its caches, and does not begin executing code until it recognizes the flush acknowledge special cycle. Please refer to Table 6-10 and 6-11 of this volume for more detailed operation.

When Sampled/Driven

The four least significant byte-enable bits are sampled for APIC ID at the falling edge of RESET. These pins should be valid and stable two clocks before and after the falling edge of RESET. As outputs, refer to the BE7#-BE0# pin description in Chapter 5.

NOTE

Asserting the APIC ID is not specified for the rising edge of RESET. In a FRC system, the BE3#-BE0# pins must not be driven for the 2 clocks following the rising edge of RESET. The system design should drive these signals on the third clock or later.

There are strong pull down resistors on the BE pins internally that make it impractical to use pullup circuits to drive the APIC ID. When not using the default APIC ID of the component, the value of the pullup resistors would have to be 50ohms or less. For this reason it is suggested to use active drivers on these lines that would drive the APIC ID to the component during the falling edge of RESET; passive pullups should be avoided.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	When APICEN is sampled active, BE3#-BE0# are used to sample the APIC ID.
RESET	During reset the BE3#-BE0# pins are sampled to determine the APIC ID. Following RESET, they function as byte-enable outputs.

21.1.7. BF1-0

BF1-0	Bus to Core Frequency Ratio
	Used to configure processor bus to core frequency ratio as 1/2 or 2/3.
	Asynchronous Input

Signal Description

The BF pin determines whether the processor will operate at a 1/2 or 2/3 I/O bus to core frequency ratio. This pin must be strapped to either V_{CC} or V_{SS}. When strapped to V_{CC}, the processor will operate at a 2/3 bus/core frequency ratio. When strapped to V_{SS}, the processor will operate at a 1/2 bus/core frequency ratio. The available Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) bus frequency selections are shown in the table below.

Bus Frequency Selections

BF1*	BF0	Clock and Bus Frequency	Bus/Core Ratio	Core Frequency
1	0	50 MHz	1/2	100 MHz
1	0	60 MHz	1/2	120 MHz
1	0	66 MHz	1/2	133 MHz
1	1	50 MHz	2/3	75 MHz
1	1	60 MHz	2/3	90 MHz
1	1	66 MHz	2/3	100 MHz

* Intel reserves other combinations of BF1 and BF0 for future usage.

If BF0 and BF1 are left unconnected, the bus-to-core ratio defaults to 2/3.

When Sampled/Driven

BF is sampled at RESET and cannot be changed until another non-warm (1 ms) assertion of RESET. BF must meet a 1 ms setup time to the falling edge of RESET.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
RESET	BF is sampled at the falling edge of RESET.

21.1.8. BP3-BP0

BP3-BP0	Breakpoint signals
	BP3-BP0 externally indicate a breakpoint match.
	Synchronous Output

Signal Description

Refer to the BP3-BP0 pin description in Chapter 5.

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.

Because of the fractional-speed bus, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) breakpoint pins BP0-3 are defined differently from the Pentium processor (510\60, 567\66). Each assertion of a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) BP pin indicates that one or more BP matches occurred. The maximum number of matches per assertion is 2 when using the 1/2 bus/core fraction. This is different from the Pentium processor (510\60, 567\66) which will only get a single match per clock.

When Sampled/Driven

The BP3-BP0 pins are driven in every clock and are not floated during bus HOLD of BOFF#.

Relation to Other Signals

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

21.1.9. BRDYC#

BRDYC#	Burst Ready
	Transfer complete indication.
	Synchronous Input

Signal Description

This signal is identical to the BRDY# input. This signal can be used to relieve tight board timings by easing the load on the address strobe signal.

When Sampled/Driven

Refer to the BRDY# signal description.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
BRDY#	BRDYC# is identical to the BRDY# input.

21.1.10. CACHE#

CACHE#	Cacheability
	External indication of internal cacheability.
	Synchronous Input/Output

Signal Description

Refer to the CACHE# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an Input/Output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.11. CPUTYP

CPUTYP	CPU Type Definition Pin
	Used to configure the Pentium® processor (610\75, 735\90, 815\100, 1000\120) as a Dual processor.
	Asynchronous Input

Signal Description

The CPUTYP pin is used to determine whether the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will function as a Primary or Dual processor. CPUTYP must be strapped to either V_{CC} or V_{SS}. When CPUTYP is strapped to V_{CC}, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will function as a Dual processor. When CPUTYP is strapped to V_{SS}, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will function as a Primary processor. In a single socket system design, CPUTYP pin should be strapped to V_{SS}.

When Sampled/Driven

CPUTYP is sampled at RESET and cannot be changed until another non-warm (1 ms) assertion of RESET. CPUTYP must meet a 1 ms setup time to the falling edge of RESET. It is recommended that CPUTYP be strapped to V_{CC} or V_{SS}.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A20M#	When CPUTYP is strapped to V_{CC}, the processor will ignore the A20M# input.
BE4#-BE0#	The BE3#-BE0# input values are sampled during RESET to determine the APIC ID. The Dual processor uses BE4# to indicate to the Primary processor that it has completed it's cache flush operation. Refer to the BE4#-BE0# pin description.
D/P#	D/P# is driven by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) only when the CPUTYP signal is strapped to V_{SS}.
DPEN#	When CPUTYP is strapped to V_{CC}, DPEN# is driven active to indicate that the second socket is occupied.
FERR#	When CPUTYP is strapped to V_{CC}, the FERR# output is undefined.
FLUSH#	When operating in dual processing mode, the FLUSH# inputs become Synchronous to the CPU clock.
IGNNE#	When CPUTYP is strapped to V_{CC}, the processor will ignore the IGNNE# input.
RESET	CPUTYP is sampled at the falling edge of RESET. When operating in dual processing mode, the RESET inputs become Synchronous to the CPU clock.

NOTE

It is common practice to put either a pullup or pulldown resistor on a net. If a pullup resistor is connected to the CPUTYP pin in order to operate in a Dual Processing mode, the value of this resistor must be 100 Ohms or less to override the internal pulldown that is normally used to put the part into a primary CPU mode of operation. In the absence of an external pullup, the internal pulldown will sufficiently pulldown the CPUTYP pin, therefore the pin can be left floating.

21.1.12. D/C#

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

Signal Description

Refer to the D/C# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an Input/Output when two Pentium processors (610\75, 735\90, 815\100, 1000\120) are operating together in Dual Processing Mode.

Relation to Other Signals

No new features.

21.1.13. D/P#

D/P#	Dual Processor / Primary Processor
	Indicates whether the Dual processor or the Primary processor is driving the bus.
	Synchronous Output

Signal Description

The D/P# pin is driven LOW when the Primary processor is driving the bus. Otherwise, the Primary processor drives this pin high to indicate that the Dual processor owns the bus. The D/P# pin can be sampled for the current cycle with ADS#. This pin is defined only on the Primary processor. In a single socket system design, D/P# pin should be left NC.

When Sampled/Driven

The D/P# pin is always driven by the Primary processor and should be sampled with ADS# of the current cycle.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	D/P# is valid for the current cycle with ADS# (like a status pin).
CPUTYP	D/P# is driven by the Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) only when the CPUTYP signal is strapped to V_{SS}.
SMIACT#	When operating in dual processing mode, D/P# qualifies the SMIACT# SMM indicator.

21.1.14. DPEN#

DPEN#	Second Socket Occupied
	Configuration signal which indicates that the second socket in a dual socket system is occupied.
	Synchronous Input (to the Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) processor) Synchronous Output (from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as a Dual processor)

Signal Description

DPEN# is driven during RESET by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) when configured as a Dual processor to indicate to the Primary processor in the first socket that there is a Dual processor present in the system. This signal is also driven by the fFuture Pentium OverDrive processor to indicate to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) that it must execute the bootup procedure.

When Sampled/Driven

DPEN# is driven during RESET by the Dual processor, and sampled at the falling edge of RESET by the Primary processor. This pin becomes PICD0 following the falling edge of RESET. This pin should be valid and stable two clocks before and after the falling edge of RESET.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
CPUTYP	When CPUTYP is strapped to V_{CC}, DPEN# is driven active to indicate that the second socket is occupied.
RESET	DPEN# is valid during the falling edge of RESET.
PICD0	DPEN# shares a pin with PICD0.

21.1.15. FERR#

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

Signal Description

Refer to the FERR# pin description in Chapter 5.

This signal is undefined when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is configured as a Dual processor.

When Sampled/Driven

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as a Dual processor, will not drive this signal to valid levels.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
CPUTYP	When CPUTYP is strapped to V_{CC}, the FERR# output is undefined.

21.1.16. FLUSH#

FLUSH#	Cache Flush
	Writes all modified lines in the data cache back and flushes the code and data caches.
	Asynchronous Input (Normal, Uni-processor, mode) Synchronous Input (Dual Processor mode)

Signal Description

Refer to the FLUSH# pin description in Chapter 5.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when operating with a second Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in dual processing mode, incorporates a mechanism to present an atomic cache flush operation to the system. The Dual processor performs the cache flush operation first, then grants the bus to the Primary processor. The Primary processor flushes its internal caches, and then runs the cache flush special cycle. This could cause the total flush latency of two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) in dual processor mode to be up to twice that of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in uni-processor mode.

The flush latency of the Future Pentium OverDrive processor may also be up to twice that of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) due to the implementation of larger on-chip caches.

When Sampled/Driven

When operating in a dual processing system, FLUSH# must be sample synchronously to the rising CLK edge to ensure both processors recognize an active FLUSH# signal in the same clock.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
CPUTYP	When operating in dual processing mode, the FLUSH# inputs become Synchronous to the CPU clock.

21.1.17. HIT#

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

Signal Description

Refer to the HIT# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an Input/Output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.18. HITM#

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

Signal Description

Refer to the HITM# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an input/output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.19. HLDA

HLDA	Bus Hold Acknowledge
	External indication that the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) outputs are floated.
	Synchronous Input/Output

Signal Description

Refer to the HLDA pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an input/output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.20. IGNNE#

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

Signal Description

Refer to the IGNNE# pin description in Chapter 5.

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as a Dual processor, will ignore the IGNNE# input.

When Sampled/Driven

No new features.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
CPUTYP	When strapped to V_{CC}, the processor will ignore the IGNNE# input.

21.1.21. INTR

INTR	External Interrupt
	Indicates that an external interrupt has been generated.
	Asynchronous Input

Signal Description

Refer to the INTR pin description in Chapter 5.

When the local APIC is hardware enabled, this pin becomes the programmable interrupt LINT0. It can be programmed in software in any of the interrupt modes. Since this pin is the INTR input when the APIC is disabled, it is logical to program the vector table entry for this pin as ExtINT (i.e. through local mode). In this mode, the interrupt signal is passed on to the processor through the local APIC. The Processor generates the interrupt acknowledge, INTA, cycle in response to this interrupt and receives the vector on the processor data bus.

When the local APIC is hardware disabled, this pin is the INTR input for the processor. It bypasses the local APIC in that case.

When Sampled/Driven

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 (610\75, 735\90, 815\100, 1000\120, 1110\133).

NOTE

This applies only when using the APIC in the through local (virtual wire) mode. Once INTR has been asserted (by a rising edge), it must not be asserted again until after the end of the first resulting interrupt acknowledge cycle. Otherwise, the new interrupt may not be recognized. The end of an interrupt acknowledge cycle is defined by the end of the system's BRDY# response to the CPU cycle. Note that the APIC through local mode was designed to match the protocol of an 8259A PIC, and an 8259A will always satisfy this requirement.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	When the APICEN configuration input is sampled inactive, this input becomes the INTR interrupt.
LINT0	INTR shares a pin with LINT0.

21.1.22. LINT1-LINT0

LINT1-LINT0	Local Interrupts 1 and 0
	APIC Programmable Interrupts.
	Asynchronous Inputs

Signal Description

When the local APIC is hardware enabled, these pins become the programmable interrupts (LINT1-LINT0). They can be programmed in software in any of the interrupt modes. Since these pins are the INTR and NMI inputs when the APIC is disabled, it is logical to program the vector table entry for them as ExtINT (i.e. through local mode) and NMI, respectively. In this mode, the interrupt signals are passed on to the processor through the local APIC.

When the local APIC is hardware disabled, these pins are the INTR and NMI inputs for the processor. They bypass the APIC in that case.

When Sampled

LINT1-LINT0 are sampled on every rising clock edge. LINT1-LINT0 are asynchronous inputs, but recognition of LINT1-LINT0 are guaranteed in a specific clock if they are asserted synchronously and meets the setup and hold times. To guarantee recognition if LINT1-LINT0 are asserted asynchronously they must have been deasserted for a minimum of 2 clocks before being returned active to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	When the APICEN configuration input is sampled inactive, these inputs become the INTR and NMI interrupts.
INTR	INTR shares a pin with LINT0.
NMI	NMI shares a pin with LINT1.

21.1.23. LOCK#

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

Signal Description

Refer to the LOCK# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an input/output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.24. M/IO#

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

Signal Description

Refer to the M/IO# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an input/output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.25. NMI

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

Signal Description

Refer to the NMI pin description in Chapter 5.

When the local APIC is hardware enabled, this pin becomes the programmable interrupt LINT1. It can be programmed in software in any of the interrupt modes. Since this pin is the NMI input when the APIC is disabled, it is logical to program the vector table entry for this pin as NMI. In this mode, the interrupt signal is passed on to the processor through the local APIC.

When the local APIC is hardware disabled, this pin is the NMI input for the processor. It bypasses the APIC in that case.

When Sampled

NMI is sampled on every rising clock edge. NMI is an asynchronous input, but 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 an minimum of 2 clocks before being returned active to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	When the APICEN configuration input is sampled inactive, this input becomes the NMI interrupt.
LINT1	NMI shares a pin with LINT1.

21.1.26. PBGNT#

PBGNT#	Dual Processor Bus Grant
	Indicates to the LRM processor that it will become the MRM in the next clock.
	Synchronous Input (to the Least Recent Master, LRM, processor) Synchronous Output (of the Most Recent Master, MRM, processor)

Signal Description

Two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as dual processors, will arbitrate for the system bus via two private arbitration pins (PBREQ#, PBGNT#). The processor that currently owns the system bus is referred to as the MRM processor. The processor that does not own the bus is referred to as the LRM processor.

PBGNT# is used by the dual processing private arbitration mechanism to indicate that bus ownership will change in the next clock. The LRM processor will request ownership of the processor bus by asserting the private arbitration request pin, PBREQ#. The processor that is currently the MRM and owns the bus, will grant the bus to the LRM as soon as any pending bus transactions have completed. The MRM will notify that the LRM can assume ownership by asserting the private arbitration grant pin, PBGNT#. The PBGNT# pin is always the output of the MRM and an input to the LRM.

NOTE

In a single socket system design, PBGNT# pin should be left NC. For proper operation, PBGNT# must not be loaded by the system.

When Sampled/Driven

PBGNT# is driven by the MRM processor in response to the PBREQ# signal from the LRM processor. It is asserted following the completion of the current cycle on the processor bus, or in the clock following the request if the bus is idle.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
PBREQ#	PBGNT# is asserted in response to a bus request, PBREQ#, by the LRM processor.
A[31:3], AP, BE[7:0]#, CACHE#, D/C#, M/IO#, PCD, PWT, SCYC, W/R#	These signals are tristated for one CLK in response to PBGNT# (when the MRM becomes the LRM).

21.1.27. PBREQ#

PBREQ#	Dual Processor Bus Request
	Indicates to the MRM processor that the LRM processor requires ownership of the bus.
	Synchronous Input (to the Most Recent Master, MRM, processor) Synchronous Output (of the Least Recent Master, LRM, processor)

Signal Description

Two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133), when configured as dual processors, will arbitrate for the system bus via two private arbitration pins (PBREQ#, PBGNT#). The processor that currently owns the system bus is referred to as the MRM processor. The processor that does not own the bus is referred to as the LRM processor.

PBREQ# is used by the dual processing private arbitration mechanism to indicate that the LRM processor requests bus ownership. The processor that is currently the MRM and owns the bus, will grant the bus to the LRM as soon as any pending bus transactions have completed. The MRM will notify that the LRM can assume ownership by asserting the private arbitration grant pin, PBGNT#. The PBREQ# pin is always the output of the LRM and an input to the MRM.

NOTE

In a single socket system design, PBREQ# pin should be left NC. For proper operation, PBREQ# must not be loaded by the system.

When Sampled/Driven

PBREQ# is driven by the LRM processor, and sampled by the MRM processor.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
PBGNT#	PBGNT# is asserted in response to a bus request, PBREQ#, by the LRM processor.

21.1.28. PCHK#

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

Signal Description

Refer to the PCHK# pin description in Chapter 5.

When operating in dual processing mode, the PCHK# signal can be asserted either 2 OR 3 CLKs following incorrect parity being detected on the data bus. When operating in Dual Processing mode the PCHK# pin circuit is implemented as a weak driving high output that operates similar to an open drain output. This implementation allows connection of the two processor PCHK# pins together in a dual processing system with no ill effects. Nominally this circuit acts like a 360 Ohm resistor tied to V_{CC}.

When Sampled/Driven

No new features.

Relation to Other Signals

No new features.

21.1.29. PHIT#

PHIT#	Private Inquire Cycle Hit/Miss Indication
	Indicates whether a private, dual processor, inquire cycle resulted in a hit or miss.
	Synchronous Input (to the Most Recent Master, MRM, processor) Synchronous Output (of the Least Recent Master, LRM, processor)

Signal Description

A private snoop interface has been added to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) for use in dual processing. The interface consists of two pins (PHIT#, PHITM#) that connect between the OEM socket and Socket 5/Socket 7.

The LRM processor will initiate a snoop sequence for all ADS# cycles that are initiated by the MRM. The LRM processor will assert the private hit indication (PHIT#) if the data requested by the MRM matches a valid cache line in the LRM. In addition, if the data requested by the MRM matches a valid cache line in the LRM that is in the modified state, the LRM will assert the PHITM# signal. The system snooping indication signals (HIT#, HITM#) will not change state as a result of a private snoop.

The MRM will use an assertion of the PHIT# signal as an indication that the requested data is being shared with the LRM. Independent of the WB/WT# pin, a cache line will be placed in the cache in the shared state if PHIT# is asserted. This will make all subsequent writes to that line externally visible until the state of the line becomes exclusive (E or M states). In a uniprocessor system, the line may have been placed in the cache in the E state. In this situation, all subsequent writes to that line will not be visible on the bus until the state is changed to I.

PHIT# will also be driven by the LRM during external snoop operations (e.g., following EADS#) to indicate the private snoop results.

NOTE

In a single socket system, PHIT# pin should be left NC. For proper operation, PHIT# must not be loaded by the system.

When Sampled/Driven

PHIT# is driven by the LRM processor, and sampled by the MRM processor. It is asserted within two clocks following an assertion of ADS# or EADS#.

The PHIT# signal operates at the processor core frequency.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A[31:5]	PHIT# is driven to indicate whether the private inquire address driven on A[31:5] is valid in the LRM's on-chip cache.
ADS#	PHIT# is driven within 2 clocks after ADS# is sampled asserted to indicate the outcome of the private inquire cycle.
EADS#	PHIT# is driven within 2 clocks after EADS# is sampled asserted to indicate the outcome of the external inquire cycle.
PHITM#	PHITM# is never asserted without PHIT# also being asserted.
WB/WT#	The state of the WB/WT# pin will be ignored by the MRM if the PHIT# pin is sampled active, and the cache line placed in the shared state.

21.1.30. PHITM#

PHITM#	Private Inquire Cycle Hit/Miss to a Modified Line Indication
	Indicates whether a private, dual processor, inquire cycle resulted in a hit or miss to a Modified line.
	Synchronous Input (to the Most Recent Master, MRM, processor) Synchronous Output (of the Least Recent Master, LRM, processor)

Signal Description

A private snoop interface has been added to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) for use in dual processing. The interface consists of two pins (PHIT#, PHITM#) that connect between the OEM socket and Socket 5/Socket 7.

The LRM processor will initiate a snoop sequence for all ADS# cycles that are initiated by the MRM. The LRM processor will assert the private hit to a modified line indication (PHITM#) if the data requested by the MRM matches a modified cache line in the LRM. In addition, if the data requested by the MRM matches a valid cache line in the LRM, the LRM will also assert the PHIT# signal. The system snooping indication signals (HIT#, HITM#) will not change state as a result of a private snoop.

PHITM# will also be driven by the LRM during external snoop operations (e.g. following EADS#) to indicate the private snoop results.

NOTE

In a single socket system, PHITM# pin should be left NC. For proper operation, PHITM# must not be loaded by the system.

When Sampled/Driven

PHITM# is driven by the LRM processor, and sampled by the MRM processor. It is asserted within two clocks following an assertion of ADS# or EADS#.

The PHITM# signal operates at the processor core frequency.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A[31:5]	PHITM# is driven to indicate whether the private inquire address driven on A[31:5] is modified in the LRM's on-chip cache.
ADS#	PHITM# is driven within 2 clocks after ADS# is sampled asserted to indicate the outcome of the private inquire cycle.
EADS#	PHITM# is driven within 2 clocks after EADS# is sampled asserted to indicate the outcome of the external inquire cycle.
PHIT#	PHITM# is never asserted without PHIT# also being asserted.

21.1.31. PICCLK

PICCLK	Processor Interrupt Controller Clock
	This pin drives the clock for the APIC serial data bus operation.
	Input

Signal Description

This pin provides the clock timings for the on chip APIC unit of the processor. This clock input controls the frequency for the APIC operation and data transmission on the 2-wire APIC serial data bus. All the timings on APIC bus are referenced to this clock.

When hardware disabled, PICCLK must be tied high.

When Sampled

PICCLK is a clock signal and is used as a reference for sampling the APIC data signals.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	PICCLK must be tied or driven high when APICEN is sampled low at the falling edge of RESET.
PICD0-1	External timing parameters for the PICD0-1 pins are measured with respect to this clock.

21.1.32. PICD1-PICD0

PICD1-PICD0	Processor Interrupt Controller Data
	These are the data pins for the 3-wire APIC bus.
	Synchronous Input/Output to PICCLK
	Needs external pull-up resistors.

Signal Description

The PICD1-PICD0 are bi-directional pins which comprise the data portion of the 3-wire APIC bus.

When Sampled/Driven

This signal is sampled with the rising edge of PICCLK.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	PICD1 shares a pin with APICEN.
DPEN#	PICD0 shares a pin with DPEN#.

21.1.33. RESET

RESET	Reset
	Forces the Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to begin execution at a known state.
	Asynchronous Input (Normal, Uni-processor, mode) Synchronous Input (Dual Processor mode)

Signal Description

Refer to the RESET pin description in Chapter 5.

No new features.

When Sampled/Driven

When operating in a dual processing system, RESET must be sample synchronously to the rising CLK edge to ensure both processors recognize the falling edge in the same clock.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
APICEN	APICEN is sampled at the falling edge of RESET.
BE3#-BE0#	During reset the BE3#-BE0# pins are sampled to determine the APIC ID. Following RESET, they function as Byte Enable outputs.
BF	BF is sampled at the falling edge of RESET.
CPUTYP	CPUTYP is sampled at the falling edge of RESET.
DPEN#	DPEN# is valid during RESET.

21.1.34. SCYC

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

Signal Description

Refer to the SCYC pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an input/output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

21.1.35. SMIACT#

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

Signal Description

Refer to the SMIACT# pin description in Chapter 5.

When the system is operating in dual processing mode, the D/P# signal toggles based upon whether the Primary or Dual processor owns the bus (MRM). The SMIACT# pins may be tied together or be used separately to insure SMRAM access by the correct processor.

CAUTION

If SMIACT# is used separately: the SMIACT# signal is only driven by the CPUs when the processor is the MRM, so this signal must be qualified with the D/P# signal).

Connecting the SMIACT# signals on the Primary and Dual processors together is strongly recommended for operation with the Dual processor and upgradability with the future Pentium OverDrive processor.

In dual processing systems, SMIACT# may not remain low (e.g., may toggle) if both processors are not in SMM mode. The SMIACT# signal is asserted by either the Primary or Dual processor based on two conditions: the processor is in SMM mode and is the bus master (MRM). If one processor is executing in normal address space, the SMIACT# signal will go inactive when that processor is MRM. The LRM processor, even if in SMM mode, will not drive the SMIACT# signal low.

When Sampled/Driven

When operating in dual processing mode, the SMIACT# output must be sampled with an active ADS# and qualified with the D/P# signal to determine which Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) (e.g., the Primary or Dual) is driving the SMM cycle.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
ADS#	SMIACT# should be sampled with an active ADS# during dual processing operation.
D/P#	When operating in dual processing mode, D/P# qualifies the SMIACT# SMM indicator.

21.1.36. STPCLK#

STPCLK#	Stop Clock Pin
	Used to stop the internal Pentium® processor (610\75, 735\90, 815\100, 1000\120) clock and consume less power.
	Asynchronous Input

Signal Description

Assertion of STPCLK# causes the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to stop its internal clock and consume less power while still responding to interprocessor and external snoop requests. This low-power state is called the stop grant state. When the CPU recognizes a STPCLK# interrupt, the CPU will do the following:

1. Wait for all instructions being executed to complete.
2. Flush the instruction pipeline of any instructions waiting to be executed.
3. Wait for all pending bus cycles to complete and EWBE# to go active.
4. Drive a special bus cycle (stop grant bus cycle) to indicate that the clock is being stopped.
5. Enter low power mode.

The stop grant bus cycle consists of the following signal states: M/IO# = 0, D/C# = 0, W/R# = 1, Address Bus = 0000 0010H (A_4 = 1), BE7#-BE0# = 1111 1011, Data bus = undefined.

STPCLK# must be driven high (not floated) to exit the stop grant state. The rising edge of STPCLK# will tell the CPU that it can return to program execution at the instruction following the interrupted instruction.

When Sampled/Driven

STPCLK# is treated as a level triggered interrupt to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and is prioritized below all of the external interrupts. When the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) recognizes the STPCLK# interrupt, the processor will stop execution on the instruction boundary following the STPCLK# assertion.

Relation to Other Signals

Pin Symbol	Relation to Other Signals
A4, Cycle Control signals (M/IO#, D/C#, W/R#, BE7#-BE0#, D/P#)	The Stop Grant Special Bus Cycle is driven on these pins in response to an assertion of the STPCLK# signal. M/IO# = 0, D/C# = 0, W/R# = 1. Address Bus = 0000 0010H (A4 = 1), BE7#-BE0# = 1111 1011.
EWBE#	After STPCLK# has been recognized, all pending cycles must be completed and EWBE# must go active before the internal clock will be disabled.
External Interrupt signals (FLUSH#, INIT, INTR, NMI, R/S#, SMI#)	While in the Stop Grant state, the CPU will latch transitions on the external interrupt signals. All of these interrupts are taken after the de-assertion of STPCLK#. The CPU requires that INTR be held active until the CPU issues an interrupt acknowledge cycle in order to guarantee recognition.
HLDA	The CPU will not respond to a STPCLK# request from a HLDA state because it cannot generate a Stop Grant cycle.

21.1.37. VCC

V_{CC}	Supply Voltage
	V_{CC} is used to supply power to the Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and Socket 5.
	Power Input

Signal Description

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and future Pentium OverDrive processor require 3.3V V_{CC} inputs.

When Sampled/Driven

No new features.

Relation to Other Signals

No new features.

21.1.38. W/R#

W/R#	Write/Read
	Distinguishes a Write cycle from a Read cycle.
	Synchronous Input/Output

Signal Description

Refer to the W/R# pin description in Chapter 5.

When operating in dual processing mode, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses this signal for private snooping.

When Sampled/Driven

This signal becomes an input/output when two Pentium processors (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating together in dual processing mode.

Relation to Other Signals

No new features.

22

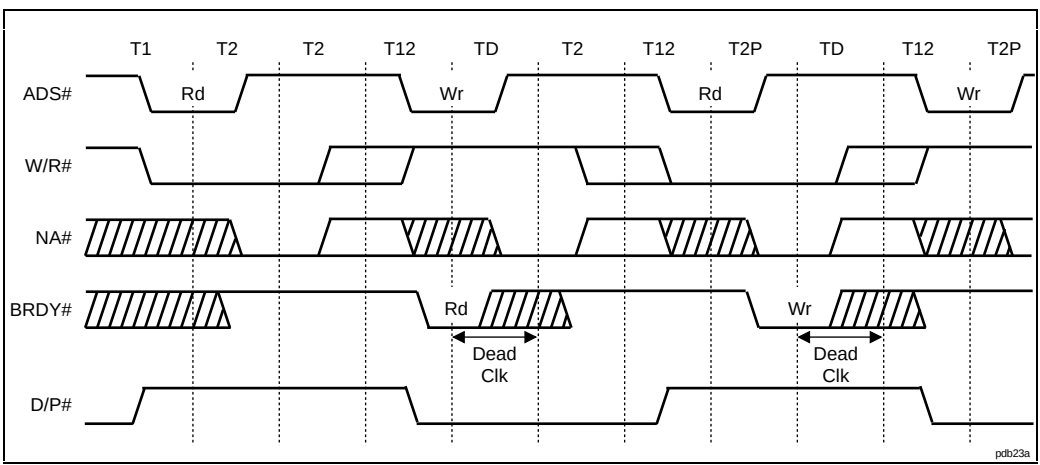
Bus Functional Description

CHAPTER 22
BUS FUNCTIONAL DESCRIPTION

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports the same bus functional features as found in Chapter 6 for the Pentium processor family.

NOTE

According to the *Pentium® Processor Family Developer's Manua*l, Volume 1 (Section 6.6.2), there is a dead clock defined for the bus switching from a pipelined write to read cycle or a pipelined read to write cycle. The processor ignores BRDY# during this dead clock, but in dual processing mode BRDY# may be falsely recognized in an inter-CPU pipelined cycle. The dual processing system design must not drive BRDY# low during this dead clock.



23

Electrical Specifications

CHAPTER 23
ELECTRICAL SPECIFICATIONS

This section describes the electrical differences between the Pentium processor (510\60, 567\66) and the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), as well as the DC and AC specifications. Unless otherwise stated, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and Dual processor have the same electrical characteristics.

23.1. ELECTRICAL DIFFERENCES OF PENTIUM® PROCESSOR (610\75, 735\90, 815\100, 1000\120, 1110\133) FROM PENTIUM PROCESSOR (510\60, 567\66)

When designing a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system from a Pentium processor (510\60, 567\66) system, there are a number of electrical differences that require attention. If these differences are understood during Pentium processor (510\60, 567\66) system design, the transition from a Pentium processor (510\60, 567\66) to Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will require little or no redesign.

Pentium® Processor (510\60, 567\66) Electrical Characteristics	**Pentium Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Electrical Characteristics**
5V Power Supply	3.3V Power Supply*
5V TTL Inputs/Outputs	3.3V Inputs/Outputs
Pentium processor Buffer Models	3.3V Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Buffer Models

NOTE:

* The upgrade socket specifies two 5V supply pins (Chapter 32).

The sections that follow will briefly point out some ways to overcome these electrical differences.

23.1.1. 3.3V Power Supply

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has all V_{CC} 3.3V inputs. By connecting all Pentium processor (510\60, 567\66) V_{CC} inputs to a common and dedicated power plane, that plane can be converted to 3.3V for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

The CLK and PICCLK inputs can tolerate a 5V input signal. This allows the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to use 5V or 3.3V clock drivers.

23.1.2. 3.3V Inputs and Outputs

The inputs and outputs of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are 3.3V JEDEC standard levels. Both inputs and outputs are also TTL-compatible, although the inputs cannot tolerate voltage swings above the $V_{IN}3$ max.

For Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) outputs, if your Pentium processor (510\60, 567\66) system support components use TTL-compatible inputs, they will interface to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) without extra logic. This is because the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) drives according to the 5V TTL specification (but not beyond 3.3V).

For Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) inputs, the voltage must not exceed the 3.3V $V_{IH}3$ maximum specification. System support components can consist of 3.3V devices or open-collector devices. 3.3V support components may interface to the Pentium processor (510\60, 567\66) since they typically meet 5V TTL specifications. In an open-collector configuration, the external resistor may be biased with the CPU V_{CC}; as the CPU's V_{CC} changes from 5V to 3.3V, so does this signal's maximum drive.

The CLK and PICCLK inputs of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are 5V tolerant, so they are electrically identical to the Pentium processor (510\60, 567\66) clock input. This allows a Pentium processor clock driver to drive the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

All pins, other than the CLK and PICCLK inputs, are 3.3V only. If an 8259A interrupt controller is used, for example, the system must provide level converters between the 8259A and the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

23.1.3. 3.3V Buffer Models

The structure of the buffer models of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are the same as those of the Pentium processor (510\60, 567\66), but the values of the components change since the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) buffers are 3.3V buffers on a different process.

Despite this difference, the simulation results of Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) buffers and Pentium processor (510\60, 567\66) buffers look nearly identical. Since the AC specifications of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are derived from the Pentium processor (510\60, 567\66) specifications, the system should see little difference between the AC behavior of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and the Pentium processor (510\60, 567\66).

To meet specifications, simulate the AC timings with Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) buffer models. Pay special attention to the new signal quality restrictions imposed by 3.3V buffers.

23.2. ABSOLUTE MAXIMUM RATINGS

The values listed below are stress ratings only. Functional operation at the maximums is not implied or guaranteed. Functional operating conditions are given in the AC and DC specification tables.

Extended exposure to the maximum ratings may affect device reliability. Furthermore, although the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) contains protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.

Table 23-1. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit	Notes
	Storage Temperature	-65	150	° C	
	Case Temperature Under Bias	-65	110	° C	
$V_{CC}3$	3V Supply Voltage with respect to V_{SS}	-0.5	4.6	V	
$V_{IN}3$	3V Only Buffer DC Input Voltage	-0.5	$V_{CC}3$+0.5 (not to exceed $V_{CC}3$ Max)	V	(2)
V_{INSB}	5V Safe Buffer DC Input Voltage	-0.5	6.5	V	(1) (3)

NOTES:

1. Applies to the CLK and PICCLK.
2. Applies to all Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) inputs except CLK and PICCLK.
3. See Table 23-3.

WARNING

Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

23.3. DC SPECIFICATIONS

Tables 23-2, and 23-3 list the DC specifications which apply to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is a 3.3V part internally. The CLK and PICCLK inputs may be 3.3V or 5V. Since the 3.3V (5V safe) input levels defined in Table 23-3 are the same as the 5V TTL levels, the CLK and PICCLK inputs are compatible with existing 5V clock drivers. The power dissipation specification in Table 23-5 is provided for design of thermal

solutions during operation in a sustained maximum level. This is the worst case power the device would dissipate in a system for a sustained period of time. This number is used for design of a thermal solution for the device.

Table 23-2. 3.3V DC Specifications

T_{CASE} = 0 to 70° C; V_{CC} = 3.135V - 3.6V

Symbol	Parameter	Min	Max	Unit	Notes
$V_{IL}3$	Input Low Voltage	–0.3	0.8	V	TTL Level (3)
$V_{IH}3$	Input High Voltage	2.0	V_{CC}+0.3	V	TTL Level (3)
$V_{OL}3$	Output Low Voltage		0.4	V	TTL Level (1) (3)
$V_{OH}3$	Output High Voltage	2.4		V	TTL Level (2) (3)
$I_{CC}3$	Power Supply Current		3400	mA	@ 133 MHz (4)
			3730	mA	@ 120 MHz (4), (5)
			3250	mA	@ 100 MHz (4)
			2950	mA	@ 90 MHz (4)
			2650	mA	@ 75 MHz (4)

NOTES:

1. Parameter measured at 4 mA.
2. Parameter measured at 3 mA.
3. 3.3V TTL levels apply to all signals except CLK and PICCLK.
4. This value should be used for power supply design. It was determined using a worst case instruction mix and V_{CC} += 3.6V. Power supply transient response and decoupling capacitors must be sufficient to handle the instantaneous current changes occurring during transitions from stop clock to full active modes. For more information, refer to section 23.4.3.
5. Please also check the *Pentium® processor Specification Update*.

Table 23-3. 3.3V (5V Safe) DC Specifications

Symbol	Parameter	Min	Max	Unit	Notes
$V_{IL}5$	Input Low Voltage	–0.3	0.8	V	TTL Level (1)
$V_{IH}5$	Input High Voltage	2.0	5.55	V	TTL Level (1)

NOTES:

1. Applies to CLK and PICCLK only.

Table 23-4. Input and Output Characteristics

Symbol	Parameter	Min	Max	Unit	Notes
C_{IN}	Input Capacitance		15	pF	(4)
C_O	Output Capacitance		20	pF	(4)
$C_{I/O}$	I/O Capacitance		25	pF	(4)
C_{CLK}	CLK Input Capacitance		15	pF	(4)
C_{TIN}	Test Input Capacitance		15	pF	(4)
C_{TOUT}	Test Output Capacitance		20	pF	(4)
C_{TCK}	Test Clock Capacitance		15	pF	(4)
I_{LI}	Input Leakage Current		$\pm$ 15	uA	$0 \le V_{IN} \le V_{CC}3$ (1)
I_{LO}	Output Leakage Current		$\pm$ 15	uA	$0 \le V_{IN} \le V_{CC}3$ (1)
I_{IH}	Input Leakage Current		200	uA	V_{IN} = 2.4V (3)
I_{IL}	Input Leakage Current		-400	uA	V_{IN} = 0.4V (2)

NOTES:

1. This parameter is for input without pull up or pull down.
2. This parameter is for input with pull up.
3. This parameter is for input with pull down.
4. Guaranteed by design.

Table 23-5. Power Dissipation Requirements for Thermal Design

Parameter	Typical(1)	Max(2)	Unit	Notes
Active Power Dissipation	4.3	11.2	Watts	@ 133 MHz
	5.0	12.81	Watts	@ 120 MHz (6)
	3.9	10.1	Watts	@ 100 MHz
	3.5	9.0	Watts	@ 90 MHz
	3.0	8.0	Watts	@ 75 MHz SPGA (7)
Stop Grant and AutoHalt Powerdown Power Dissipation		1.7	Watts	@ 133 MHz (3)
		1.76	Watts	@ 120 MHz (3), (6)
		1.55	Watts	@ 100 MHz (3)
		1.40	Watts	@ 90 MHz (3)
		1.2	Watts	@ 75 MHz (3)
Stop Clock Power Dissipation	.02	< 0.3	Watts	(4) (5)

NOTES:

1. This is the typical power dissipation in a system. This value was the average value measured in a system using a typical device at V_{CC} = 3.3V running typical applications. This value is highly dependent upon the specific system configuration.
2. Systems must be designed to thermally dissipate the maximum active power dissipation. It is determined using a worst case instruction mix with V_{CC} = 3.3V. The use of nominal V_{CC} in this measurement takes into account the thermal time constant of the package.
3. Stop Grant/Auto Halt Powerdown Power Dissipation is determined by asserting the STPCLK# pin or executing the HALT instruction.
4. Stop Clock Power Dissipation is determined by asserting the STPCLK# pin and then removing the external CLK input.
5. Complete characterization of this specification was still in process at the time of print. Please contact Intel for the latest information. The final specification will be less than 0.1 W.
6. Please also check the *Pentium® Processor Specification Update*.
7. For power dissipation requirements of Pentium processor (610\75) TCP package, please refer to Chapter 34 of this document.

23.4. AC SPECIFICATIONS

The AC specifications of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) consist of setup times, hold times and valid delays at 0 pF.

WARNING

Do not exceed the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) internal maximum frequency of 100 MHz by selecting the wrong bus fraction.

23.4.1. Private Bus

When two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are operating in dual processor mode, a "private bus" exists to arbitrate for the CPU bus and maintain local cache coherency. The private bus consists of two pinout changes:

1. Five pins are added: PBREQ#, PBGNT#, PHIT#, PHITM#, D/P#.
2. Ten output pins become I/O pins: ADS#, D/C#, W/R#, M/IO#, CACHE#, LOCK#, HIT#, HITM#, HLDA, SCYC.

The new pins are given AC specifications of valid delays at 0 pF, setup times and hold times. Simulate with these parameters and their respective I/O buffer models to guarantee that proper timings are met.

The AC specification gives input setup and hold times for the ten signals that become I/O pins. These setup and hold times must only be met when a Dual processor is present in the system.

23.4.2. Power and Ground

For clean on-chip power distribution, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) has 53 V_{CC} (power) and 53 V_{SS} (ground) inputs. Power and ground connections must be made to all external V_{CC} and V_{SS} pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). On the circuit board all V_{CC} pins must be connected to a 3.3V V_{CC} plane. All V_{SS} pins must be connected to a V_{SS} plane.

23.4.3. Decoupling Recommendations

Liberal decoupling capacitance should be placed near the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) driving its large address and data buses at high frequencies can cause transient power surges, particularly when driving large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening circuit board traces between the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and decoupling capacitors as much as possible.

These capacitors should be evenly distributed around each component on the 3.3V plane. Capacitor values should be chosen to ensure they eliminate both low and high frequency noise components.

For the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), the power consumption can transition from a low level of power to a much higher level (or high to low power) very rapidly. A typical example would be entering or exiting the Stop Grant state. Another example would be executing a HALT instruction, causing the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to enter the Auto HALT Powerdown state, or

transitioning from HALT to the Normal state. All of these examples may cause abrupt changes in the power being consumed by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). Note that the Auto HALT Powerdown feature is always enabled even when other power management features are not implemented.

Bulk storage capacitors with a low ESR (Effective Series Resistance) in the 10 to 100 uf range are required to maintain a regulated supply voltage during the interval between the time the current load changes and the point that the regulated power supply output can react to the change in load. In order to reduce the ESR, it may be necessary to place several bulk storage capacitors in parallel.

These capacitors should be placed near the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) (on the 3.3V plane) to ensure that the supply voltage stays within specified limits during changes in the supply current during operation.

23.4.4. Connection Specifications

All NC and INC pins must remain unconnected.

For reliable operation, always connect unused inputs to an appropriate signal level. Unused active low inputs should be connected to V_{CC}. Unused active high inputs should be connected to ground.

23.4.5. AC Timing Tables

23.4.5.1. AC TIMING TABLE FOR A 50-MHZ BUS

The AC specifications given in Tables 23-6 and 23-7 consist of output delays, input setup requirements and input hold requirements for a 50-MHz external bus. All AC specifications (with the exception of those for the TAP signals and APIC signals) are relative to the rising edge of the CLK input.

All timings are referenced to 1.5V for both "0" and "1" logic levels unless otherwise specified. Within the sampling window, a synchronous input must be stable for correct Pentium processor (610\75, 815\100) operation. For (610\75) TCP AC timing specifications, see Chapter 34.

Table 23-6. Pentium® Processor (610\75, 815\100) AC Specifications for 50-MHz Bus Operation

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	Frequency	25.0	50.0	MHz		Max Core Freq = 100 MHz@1/2
t_{1a}	CLK Period	20.0	40.0	nS	23-1	
t_{1b}	CLK Period Stability		±250	pS		(1), (25)
t_2	CLK High Time	4.0		nS	23-1	@2V, (1)
t_3	CLK Low Time	4.0		nS	23-1	@0.8V, (1)
t_4	CLK Fall Time	0.15	1.5	nS	23-1	(2.0V-0.8V), (1), (5)
t_5	CLK Rise Time	0.15	1.5	nS	23-1	(0.8V-2.0V), (1), (5)
t_{6a}	ADS#, ADSC#, PWT, PCD, BE0-7#, M/IO#, D/C#, CACHE#, SCYC, W/R# Valid Delay	1.0	7.0	nS	23-2	
t_{6b}	AP Valid Delay	1.0	8.5	nS	23-2	
t_{6c}	A3-A31, LOCK# Valid Delay	1.1	7.0	nS	23-2	
t_7	ADS#, ADSC#, AP, A3-A31, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10.0	nS	23-3	(1)
t_8	APCHK#, IERR#, FERR#, PCHK# Valid Delay	1.0	8.3	nS	23-2	(4)
t_{9a}	BREQ, HLDA, SMIACT# Valid Delay	1.0	8.0	nS	23-2	(4)
t_{10a}	HIT# Valid Delay	1.0	8.0	nS	23-2	
t_{10b}	HITM# Valid Delay	1.1	6.0	nS	23-2	
t_{11a}	PM0-1, BP0-3 Valid Delay	1.0	10.0	nS	23-2	
t_{11b}	PRDY Valid Delay	1.0	8.0	nS	23-2	
t_{12}	D0-D63, DP0-7 Write Data Valid Delay	1.3	8.5	nS	23-2	
t_{13}	D0-D63, DP0-3 Write Data Float Delay		10.0	nS	23-3	(1)
t_{14}	A5-A31 Setup Time	6.5		nS	23-4	(26)
t_{15}	A5-A31 Hold Time	1.0		nS	23-4	
t_{16a}	INV, AP Setup Time	5.0		nS	23-4	

Table 23-6. Pentium® Processor (610\75, 815\100) AC Specifications for 50-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{16b}	EADS# Setup Time	6.0		nS	23-4	
t_{17}	EADS#, INV, AP Hold Time	1.0		nS	23-4	
t_{18a}	KEN# Setup Time	5.0		nS	23-4	
t_{18b}	NA#, WB/WT# Setup Time	4.5		nS	23-4	
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.0		nS	23-4	
t_{20}	BRDY#, BRDYC# Setup Time	5.0		nS	23-4	
t_{21}	BRDY#, BRDYC# Hold Time	1.0		nS	23-4	
t_{22}	BOFF# Setup Time	5.5		nS	23-4	
t_{22a}	AHOLD Setup Time	6.0		nS	23-4	
t_{23}	AHOLD, BOFF# Hold Time	1.0		nS	23-4	
t_{24}	BUSCHK#, EWBE#, HOLD, PEN# Setup Time	5.0		nS	23-4	
t_{25}	BUSCHK#, EWBE#, PEN# Hold Time	1.0		nS	23-4	
t_{25a}	HOLD Hold Time	1.5		nS	23-4	
t_{26}	A20M#, INTR, STPCLK# Setup Time	5.0		nS	23-4	(12), (16)
t_{27}	A20M#, INTR, STPCLK# Hold Time	1.0		nS	23-4	(13)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5.0		nS	23-4	(12), (16), (17)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.0		nS	23-4	(13)
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width, Async	2.0		CLKs		(15), (17)
t_{31}	R/S# Setup Time	5.0		nS	23-4	(12), (16), (17)
t_{32}	R/S# Hold Time	1.0		nS	23-4	(13)
t_{33}	R/S# Pulse Width, Async.	2.0		CLKs		(15), (17)
t_{34}	D0-D63, DP0-7 Read Data Setup Time	3.8		nS	23-4	
t_{35}	D0-D63, DP0-7 Read Data Hold Time	1.5		nS	23-4	

Table 23-6. Pentium® Processor (610\75, 815\100) AC Specifications for 50-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{36}	RESET Setup Time	5.0		nS	23-5	(11), (12), (16)
t_{37}	RESET Hold Time	1.0		nS	23-5	(11), (13)
t_{38}	RESET Pulse Width, V_{CC} & CLK Stable	15		CLKs	23-5	(11), (17)
t_{39}	RESET Active After V_{CC} & CLK Stable	1.0		mS	23-5	Power up
t_{40}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time	5.0		nS	23-5	(12), (16), (17)
t_{41}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time	1.0		nS	23-5	(13)
t_{42a}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time, Async.	2.0		CLKs	23-5	To RESET falling edge (16)
t_{42b}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#, BRDYC#, BUSCHK#) Hold Time, Async.	2.0		CLKs	23-5	To RESET falling edge (27)
t_{42c}	Reset Configuration Signals (BRDYC#, BUSCHK#) Setup Time, Async.	3.0		CLKs	23-5	To RESET falling edge (27)
t_{42d}	Reset Configuration Signal BRDYC# Hold Time, RESET driven synchronously	1.0		nS		To RESET falling edge (1), (27)
t_{43a}	BF, CPUTYP Setup Time	1.0		mS	23-5	(22) to RESET falling edge
t_{43b}	BF, CPUTYP Hold Time	2.0		CLKs	23-5	(22) to RESET falling edge
t_{43c}	APICEN, BE4# Setup Time	2.0		CLKs	23-5	To RESET falling edge
t_{43d}	APICEN, BE4# Hold Time	2.0		CLKs	23-5	To RESET falling edge
t_{44}	TCK Frequency	—	16.0	MHz		
t_{45}	TCK Period	62.5		nS	23-1	
t_{46}	TCK High Time	25.0		nS	23-1	@2V, (1)

Table 23-6. Pentium® Processor (610\75, 815\100) AC Specifications for 50-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{47}	TCK Low Time	25.0		nS	23-1	@0.8V, (1)
t_{48}	TCK Fall Time		5.0	nS	23-1	(2.0V–0.8V), (1), (8), (9)
t_{49}	TCK Rise Time		5.0	nS	23-1	(0.8V–2.0V), (1), (8), (9)
t_{50}	TRST# Pulse Width	40.0		nS	23-7	(1), Asynchronous
t_{51}	TDI, TMS Setup Time	5.0		nS	23-6	(7)
t_{52}	TDI, TMS Hold Time	13.0		nS	23-6	(7)
t_{53}	TDO Valid Delay	3.0	20.0	nS	23-6	(8)
t_{54}	TDO Float Delay		25.0	nS	23-6	(1), (8)
t_{55}	All Non-Test Outputs Valid Delay	3.0	20.0	nS	23-6	(3), (8), (10)
t_{56}	All Non-Test Outputs Float Delay		25.0	nS	23-6	(1), (3), (8), (10)
t_{57}	All Non-Test Inputs Setup Time	5.0		nS	23-6	(3), (7), (10)
t_{58}	All Non-Test Inputs Hold Time	13.0		nS	23-6	(3), (7), (10)
APIC AC Specifications						
t_{60a}	PICCLK Frequency	2.0	16.66	MHz		
t_{60b}	PICCLK Period	60.0	500.0	nS	23-1	
t_{60c}	PICCLK High Time	15.0		nS	23-1	
t_{60d}	PICCLK Low Time	15.0		nS	23-1	
t_{60e}	PICCLK Rise Time	0.15	2.5	nS	23-1	
t_{60f}	PICCLK Fall Time	0.15	2.5	nS	23-1	
t_{60g}	PICD0-1 Setup Time	3.0		nS	23-4	To PICCLK
t_{60h}	PICD0-1 Hold Time	2.5		nS	23-4	To PICCLK
t_{60i}	PICD0-1 Valid Delay (LtoH)	4.0	38.0	nS	23-2	From PICCLK, (28)
t_{60j}	PICD0-1 Valid Delay (HtoL)	4.0	22.0	nS	23-2	From PICCLK, (28)
t_{61}	PICCLK Setup Time	5.0		nS		To CLK (31)
t_{62}	PICCLK Hold Time	2.0		nS		To CLK (31)
t_{63}	PICCLK Ratio (CLK/PICCLK)	4.0		nS		(32)

Table 23-7. Pentium® Processor (610\75, 815\100) Dual Processor Mode AC Specifications for 50-MHz Bus Operation

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{80}	PBREQ#, PBGNT#, PHIT#, PHITM# Flight Time	0	2.0	nS	23-8	(29)
t_{83a}	A5-A31 Setup Time	6.5		nS	23-4	(18), (21), (26)
t_{83b}	D/C#, W/R#, CACHE#, LOCK#, SCYC Setup Time	6.0		nS	23-4	(18), (21)
t_{83c}	ADS#, M/IO# Setup Time	8.0		nS	23-4	(18), (21)
t_{83d}	HIT#, HITM# Setup Time	8.0		nS	23-4	(18), (21)
t_{83e}	HLDA Setup Time	6.0		nS	23-4	(18), (21)
t_{84}	ADS#, D/C#, W/R#, M/IO#, CACHE#, LOCK#, A5-A31, HLDA, HIT#, HITM#, SCYC Hold Time	1.0		nS	23-4	(18), (21)
t_{85}	DPEN# Valid Time		10.0	CLKs		(18), (19), (23)
t_{86}	DPEN# Hold Time	2.0		CLKs		(18), (20), (23)
t_{87}	APIC ID (BE0#-BE3#) Setup Time	2.0		CLKs	23-5	(23) to falling Edge of RESET
t_{88}	APIC ID (BE0#-BE3#) Hold Time	2.0		CLKs	23-5	(23) from Falling Edge of RESET
t_{89}	D/P# Valid Delay	1.0	8.0	nS	23-2	Primary processor Only

23.4.5.2. AC TIMING TABLES FOR A 60-MHz BUS

The AC specifications given in Tables 23-8 and 23-9 consist of output delays, input setup requirements and input hold requirements for a 60-MHz external bus. All AC specifications (with the exception of those for the TAP signals and APIC signals) are relative to the rising edge of the CLK input.

All timings are referenced to 1.5V for both "0" and "1" logic levels unless otherwise specified. Within the sampling window, a synchronous input must be stable for correct Pentium processor (735\90, 1000\120) operation.

Table 23-8. Pentium® Processor (735\90, 1000\120) AC Specifications for 60-MHz Bus Operation

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	Frequency	30.0	60.0	MHz		
t_{1a}	CLK Period	16.67	33.33	nS	23-1	
t_{1b}	CLK Period Stability		±250	pS		(1), (25)
t_2	CLK High Time	4.0		nS	23-1	@2V, (1)
t_3	CLK Low Time	4.0		nS	23-1	@0.8V, (1)
t_4	CLK Fall Time	0.15	1.5	nS	23-1	(2.0V-0.8V), (1), (5)
t_5	CLK Rise Time	0.15	1.5	nS	23-1	(0.8V-2.0V), (1), (5)
t_{6a}	ADS#, ADSC#, PWT, PCD, BE0-7#, M/IO#, D/C#, CACHE#, SCYC, W/R# Valid Delay	1.0	7.0	nS	23-2	
t_{6b}	AP Valid Delay	1.0	8.5	nS	23-2	
t_{6c}	LOCK# Valid Delay	1.1	7.0	nS	23-2	
t_{6e}	A3-A31 Valid Delay	1.1	6.3	nS	23-2	
t_7	ADS#, ADSC#, AP, A3-A31, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10.0	nS	23-3	(1)
t_{8a}	APCHK#, IERR#, FERR# Valid Delay	1.0	8.3	nS	23-2	(4)
t_{8b}	PCHK# Valid Delay	1.0	7.0	nS	23-2	(4)
t_{9a}	BREQ,HLDA Valid Delay	1.0	8.0	nS	23-2	(4)
t_{9b}	SMIACT# Valid Delay	1.0	7.6	nS	23-2	
t_{10a}	HIT# Valid Delay	1.0	8.0	nS	23-2	
t_{10b}	HITM# Valid Delay	1.1	6.0	nS	23-2	
t_{11a}	PM0-1, BP0-3 Valid Delay	1.0	10.0	nS	23-2	
t_{11b}	PRDY Valid Delay	1.0	8.0	nS	23-2	
t_{12}	D0-D63, DP0-7 Write Data Valid Delay	1.3	7.5	nS	23-2	
t_{13}	D0-D63, DP0-3 Write Data Float Delay		10.0	nS	23-3	(1)
t_{14}	A5-A31 Setup Time	6.0		nS	23-4	(26)

Table 23-8. Pentium® Processor (735\90, 1000\120) AC Specifications for 60-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{15}	A5-A31 Hold Time	1.0		nS	23-4	
t_{16a}	INV, AP Setup Time	5.0		nS	23-4	
t_{16b}	EADS# Setup Time	5.5		nS	23-4	
t_{17}	EADS#, INV, AP Hold Time	1.0		nS	23-4	
t_{18a}	KEN# Setup Time	5.0		nS	23-4	
t_{18b}	NA#, WB/WT# Setup Time	4.5		nS	23-4	
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.0		nS	23-4	
t_{20}	BRDY#, BRDYC# Setup Time	5.0		nS	23-4	
t_{21}	BRDY#, BRDYC# Hold Time	1.0		nS	23-4	
t_{22}	AHOLD, BOFF# Setup Time	5.5		nS	23-4	
t_{23}	AHOLD, BOFF# Hold Time	1.0		nS	23-4	
t_{24}	BUSCHK#, EWBE#, HOLD, PEN# Setup Time	5.0		nS	23-4	
t_{25}	BUSCHK#, EWBE#, PEN# Hold Time	1.0		nS	23-4	
t_{25a}	HOLD Hold Time	1.5		nS	23-4	
t_{26}	A20M#, INTR, STPCLK# Setup Time	5.0		nS	23-4	(12), (16)
t_{27}	A20M#, INTR, STPCLK# Hold Time	1.0		nS	23-4	(13)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5.0		nS	23-4	(12), (16), (17)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.0		nS	23-4	(13)
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width, Async	2.0		CLKs		(15), (17)
t_{31}	R/S# Setup Time	5.0		nS	23-4	(12), (16), (17)
t_{32}	R/S# Hold Time	1.0		nS	23-4	(13)
t_{33}	R/S# Pulse Width, Async.	2.0		CLKs		(15), (17)
t_{34}	D0-D63, DP0-7 Read Data Setup Time	3.0		nS	23-4	

Table 23-8. Pentium® Processor (735\90, 1000\120) AC Specifications for 60-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{35}	D0-D63, DP0-7 Read Data Hold Time	1.5		nS	23-4	
t_{36}	RESET Setup Time	5.0		nS	23-5	(11), (12), (16)
t_{37}	RESET Hold Time	1.0		nS	23-5	(11), (13)
t_{38}	RESET Pulse Width, V_{CC} & CLK Stable	15		CLKs	23-5	(11), (17)
t_{39}	RESET Active After V_{CC} & CLK Stable	1.0		mS	23-5	Power up
t_{40}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time	5.0		nS	23-5	(12), (16), (17)
t_{41}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time	1.0		nS	23-5	(13)
t_{42a}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time, Async.	2.0		CLKs	23-5	To RESET falling edge (16)
t_{42b}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#, BRDYC#, BUSCHK#) Hold Time, Async.	2.0		CLKs	23-5	To RESET falling edge (27)
t_{42c}	Reset Configuration Signals (BRDYC#, BUSCHK#) Setup Time, Async.	3.0		CLKs	23-5	To RESET falling edge (27)
t_{42d}	Reset Configuration Signal BRDYC# Hold Time, RESET driven synchronously	1.0		nS		To RESET falling edge (1), (27)
t_{43a}	BF, CPUTYP Setup Time	1.0		mS	23-5	(22) to RESET falling edge
t_{43b}	BF, CPUTYP Hold Time	2.0		CLKs	23-5	(22) to RESET falling edge
t_{43c}	APICEN, BE4# Setup Time,	2.0		CLKs	23-5	To RESET falling edge
t_{43d}	APICEN, BE4# Hold Time,	2.0		CLKs	23-5	To RESET falling edge
t_{44}	TCK Frequency	—	16.0	MHz		
t_{45}	TCK Period	62.5		nS	23-1	
t_{46}	TCK High Time	25.0		nS	23-1	@2V, (1)
t_{47}	TCK Low Time	25.0		nS	23-1	@0.8V, (1)

Table 23-8. Pentium® Processor (735\90, 1000\120) AC Specifications for 60-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70 °C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{48}	TCK Fall Time		5.0	nS	23-1	(2.0V–0.8V), (1), (8), (9)
t_{49}	TCK Rise Time		5.0	nS	23-1	(0.8V–2.0V), (1), (8), (9)
t_{50}	TRST# Pulse Width	40.0		nS	23-6	(1), Asynchronous
t_{51}	TDI, TMS Setup Time	5.0		nS	23-6	(7)
t_{52}	TDI, TMS Hold Time	13.0		nS	23-6	(7)
t_{53}	TDO Valid Delay	3.0	20.0	nS	23-6	(8)
t_{54}	TDO Float Delay		25.0	nS	23-6	(1), (8)
t_{55}	All Non-Test Outputs Valid Delay	3.0	20.0	nS	23-6	(3), (8), (10)
t_{56}	All Non-Test Outputs Float Delay		25.0	nS	23-6	(1), (3), (8), (10)
t_{57}	All Non-Test Inputs Setup Time	5.0		nS	23-6	(3), (7), (10)
t_{58}	All Non-Test Inputs Hold Time	13.0		nS	23-6	(3), (7), (10)
APIC AC Specifications						
t_{60a}	PICCLK Frequency	2.0	16.66	MHz		
t_{60b}	PICCLK Period	60.0	500.0	nS	23-1	
t_{60c}	PICCLK High Time	15.0		nS	23-1	
t_{60d}	PICCLK Low Time	15.0		nS	23-1	
t_{60e}	PICCLK Rise Time	0.15	2.5	nS	23-1	
t_{60f}	PICCLK Fall Time	0.15	2.5	nS	23-1	
t_{60g}	PICD0-1 Setup Time	3.0		nS	23-4	To PICCLK
t_{60h}	PICD0-1 Hold Time	2.5		nS	23-4	To PICCLK
t_{60i}	PICD0-1 Valid Delay (LtoH)	4.0	38.0	nS	23-2	From PICCLK, (28)
t_{60j}	PICD0-1 Valid Delay (HtoL)	4.0	22.0	nS	23-2	From PICCLK, (28)
t_{61}	PICCLK Setup Time	5.0		nS		To CLK (31)
t_{62}	PICCLK Hold Time	2.0		nS		To CLK (31)
t_{63}	PICCLK Ratio (CLK/PICCLK)	4.0		nS		(32)

Table 23-9. Pentium® Processor (735\90, 1000\120) Dual Processor Mode AC Specifications for 60-MHz Bus Operation

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{80}	PBREQ#, PBGNT#, PHIT#, PHITM# Flight Time	0	2.0	nS	23-8	(29)
t_{83a}	A5-A31 Setup Time	3.9		nS	23-4	(18), (21), (26)
t_{83b}	D/C#, W/R#, CACHE#, LOCK#, SCYC Setup Time	4.0		nS	23-4	(18), (21)
t_{83c}	ADS#, M/IO# Setup Time	6.0		nS	23-4	(18), (21)
t_{83d}	HIT#, HITM# Setup Time	6.0		nS	23-4	(18), (21)
t_{83e}	HLDA Setup Time	6.0		nS	23-4	(18), (21)
t_{84}	ADS#, D/C#, W/R#, M/IO#, CACHE#, LOCK#, A5-A31, HLDA, HIT#, HITM#, SCYC Hold Time	1.0		nS	23-4	(18), (21)
t_{85}	DPEN# Valid Time		10.0	CLKs		(18), (19), (23)
t_{86}	DPEN# Hold Time	2.0		CLKs		(18), (20), (23)
t_{87}	APIC ID (BE0#-BE3#) Setup Time	2.0		CLKs	23-5	(23) to falling Edge of RESET
t_{88}	APIC ID (BE0#-BE3#) Hold Time	2.0		CLKs	23-5	(23) from Falling Edge of RESET
t_{89}	D/P# Valid Delay	1.0	8.0	nS	23-2	Primary processor Only

23.4.5.3. AC TIMING TABLES FOR A 66-MHz BUS

The AC specifications given in Tables 23-10 and 23-11 consist of output delays, input setup requirements and input hold requirements for a 66-MHz external bus. All AC specifications (with the exception of those for the TAP signals and APIC signals) are relative to the rising edge of the CLK input.

All timings are referenced to 1.5V for both "0" and "1" logic levels unless otherwise specified. Within the sampling window, a synchronous input must be stable for correct Pentium processor 815\100 operation.

Table 23-10. Pentium® Processor (815\100, 1110\133) AC Specifications for 66-MHz Bus Operation

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	Frequency	33.33	66.6	MHz		
t_{1a}	CLK Period	15.0	30.0	nS	23-1	
t_{1b}	CLK Period Stability		±250	pS		(1), (25)
t_2	CLK High Time	4.0		nS	23-1	@2V, (1)
t_3	CLK Low Time	4.0		nS	23-1	@0.8V, (1)
t_4	CLK Fall Time	0.15	1.5	nS	23-1	(2.0V–0.8V), (1)
t_5	CLK Rise Time	0.15	1.5	nS	23-1	(0.8V–2.0V), (1)
t_{6a}	ADSC#, PWT, PCD, BE0-7#, D/C#, W/R#, CACHE#, SCYC Valid Delay	1.0	7.0	nS	23-2	
t_{6b}	AP Valid Delay	1.0	8.5	nS	23-2	
t_{6c}	LOCK# Valid Delay	1.1	7.0	nS	23-2	
t_{6d}	ADS# Valid Delay	1.0	6.0	nS	23-2	
t_{6e}	A3-A31 Valid Delay	1.1	6.3	nS	23-2	
t_{6f}	M/IO# Valid Delay	1.0	5.9	nS	23-2	
t_7	ADS#, ADSC#, AP, A3-A31, PWT, PCD, BE0-7#, M/IO#, D/C#, W/R#, CACHE#, SCYC, LOCK# Float Delay		10.0	nS	23-3	(1)
t_{8a}	APCHK#, IERR#, FERR# Valid Delay	1.0	8.3	nS	23-2	(4)
t_{8b}	PCHK# Valid Delay	1.0	7.0	nS	23-2	(4)
t_{9a}	BREQ Valid Delay	1.0	8.0	nS	23-2	(4)
t_{9b}	SMIACT# Valid Delay	1.0	7.3	nS	23-2	(4)
t_{9c}	HLDA Valid Delay	1.0	6.8	nS	23-2	(4)
t_{10a}	HIT# Valid Delay	1.0	6.8	nS	23-2	
t_{10b}	HITM# Valid Delay	1.1	6.0	nS	23-2	
t_{11a}	PM0-1, BP0-3 Valid Delay	1.0	10.0	nS	23-2	
t_{11b}	PRDY Valid Delay	1.0	8.0	nS	23-2	
t_{12}	D0-D63, DP0-7 Write Data Valid Delay	1.3	7.5	nS	23-2	
t_{13}	D0-D63, DP0-3 Write Data Float Delay		10.0	nS	23-3	(1)

Table 23-10. Pentium® Processor (815\100, 1110\133) AC Specifications for 66-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{14}	A5-A31 Setup Time	6.0		nS	23-4	(26)
t_{15}	A5-A31 Hold Time	1.0		nS	23-4	
t_{16a}	INV, AP Setup Time	5.0		nS	23-4	
t_{16b}	EADS# Setup Time	5.0		nS	23-4	
t_{17}	EADS#, INV, AP Hold Time	1.0		nS	23-4	
t_{18a}	KEN# Setup Time	5.0		nS	23-4	
t_{18b}	NA#, WB/WT# Setup Time	4.5		nS	23-4	
t_{19}	KEN#, WB/WT#, NA# Hold Time	1.0		nS	23-4	
t_{20}	BRDY#, BRDYC# Setup Time	5.0		nS	23-4	
t_{21}	BRDY#, BRDYC# Hold Time	1.0		nS	23-4	
t_{22}	AHOLD, BOFF# Setup Time	5.5		nS	23-4	
t_{23}	AHOLD, BOFF# Hold Time	1.0		nS	23-4	
t_{24a}	BUSCHK#, EWBE#, HOLD,Setup Time	5.0		nS	23-4	
t_{24b}	PEN# Setup Time	4.8		nS	23-4	
t_{25a}	BUSCHK#, EWBE#, PEN# Hold Time	1.0		nS	23-4	
t_{25b}	HOLD Hold Time	1.5		nS	23-4	
t_{26}	A20M#, INTR, STPCLK# Setup Time	5.0		nS	23-4	(12), (16)
t_{27}	A20M#, INTR, STPCLK# Hold Time	1.0		nS	23-4	(13)
t_{28}	INIT, FLUSH#, NMI, SMI#, IGNNE# Setup Time	5.0		nS	23-4	(12), (16), (17)
t_{29}	INIT, FLUSH#, NMI, SMI#, IGNNE# Hold Time	1.0		nS	23-4	(13)
t_{30}	INIT, FLUSH#, NMI, SMI#, IGNNE# Pulse Width, Async	2.0		CLKs		(15), (17)
t_{31}	R/S# Setup Time	5.0		nS	23-4	(12), (16), (17)
t_{32}	R/S# Hold Time	1.0		nS	23-4	(13)
t_{33}	R/S# Pulse Width, Async.	2.0		CLKs		(15), (17)
t_{34}	D0-D63, DP0-7 Read Data Setup Time	2.8		nS	23-4	

Table 23-10. Pentium® Processor (815\100, 1110\133) AC Specifications for 66-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{35}	D0-D63, DP0-7 Read Data Hold Time	1.5		nS	23-4	
t_{36}	RESET Setup Time	5.0		nS	23-5	(11), (12), (16)
t_{37}	RESET Hold Time	1.0		nS	23-5	(11), (13)
t_{38}	RESET Pulse Width, V_{CC} & CLK Stable	15.0		CLKs	23-5	(11), (17)
t_{39}	RESET Active After V_{CC} & CLK Stable	1.0		mS	23-5	Power up
t_{40}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time	5.0		nS	23-5	(12), (16), (17)
t_{41}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Hold Time	1.0		nS	23-5	(13)
t_{42a}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#) Setup Time, Async.	2.0		CLKs	23-5	To RESET falling edge (16)
t_{42b}	Reset Configuration Signals (INIT, FLUSH#, FRCMC#, BRDYC#, BUSCHK#) Hold Time, Async.	2.0		CLKs	23-5	To RESET falling edge (27)
t_{42c}	Reset Configuration Signals (BRDYC#, BUSCHK#) Setup Time, Async.	3.0		CLKs	23-5	To RESET falling edge (27)
t_{42d}	Reset Configuration Signal BRDYC# Hold Time, RESET driven synchronously	1.0		nS		To RESET falling edge (1), (27)
t_{43a}	BF, CPUTYP Setup Time	1.0		mS	23-5	(22) to RESET falling edge
t_{43b}	BF, CPUTYP Hold Time	2.0		CLKs	23-5	(22) to RESET falling edge
t_{43c}	APICEN, BE4# Setup Time	2.0		CLKs	23-5	To RESET falling edge
t_{43d}	APICEN, BE4# Hold Time	2.0		CLKs	23-5	To RESET falling edge
t_{44}	TCK Frequency	--	16.0	MHz		
t_{45}	TCK Period	62.5		nS	23-1	
t_{46}	TCK High Time	25.0		nS	23-1	@2V, (1)
t_{47}	TCK Low Time	25.0		nS	23-1	@0.8V, (1)

Table 23-10. Pentium® Processor (815\100, 1110\133) AC Specifications for 66-MHz Bus Operation (Contd.)

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{48}	TCK Fall Time		5.0	nS	23-1	(2.0V–0.8V), (1), (8), (9)
t_{49}	TCK Rise Time		5.0	nS	23-1	(0.8V–2.0V), (1), (8), (9)
t_{50}	TRST# Pulse Width	40.0		nS	23-6	(1), Asynchronous
t_{51}	TDI, TMS Setup Time	5.0		nS	23-6	(7)
t_{52}	TDI, TMS Hold Time	13.0		nS	23-6	(7)
t_{53}	TDO Valid Delay	3.0	20.0	nS	23-6	(8)
t_{54}	TDO Float Delay		25.0	nS	23-6	(1), (8)
t_{55}	All Non-Test Outputs Valid Delay	3.0	20.0	nS	23-6	(3), (8), (10)
t_{56}	All Non-Test Outputs Float Delay		25.0	nS	23-6	(1), (3), (8), (10)
t_{57}	All Non-Test Inputs Setup Time	5.0		nS	23-6	(3), (7), (10)
t_{58}	All Non-Test Inputs Hold Time	13.0		nS	23-6	(3), (7), (10)
APIC AC Specifications						
t_{60a}	PICCLK Frequency	2.0	16.66	MHz		
t_{60b}	PICCLK Period	60.0	500.0	nS	23-1	
t_{60c}	PICCLK High Time	15.0		nS	23-1	
t_{60d}	PICCLK Low Time	15.0		nS	23-1	
t_{60e}	PICCLK Rise Time	0.15	2.5	nS	23-1	
t_{60f}	PICCLK Fall Time	0.15	2.5	nS	23-1	
t_{60g}	PICD0-1 Setup Time	3.0		nS	23-4	To PICCLK
t_{60h}	PICD0-1 Hold Time	2.5		nS	23-4	To PICCLK
t_{60i}	PICD0-1 Valid Delay (LtoH)	4.0	38.0	nS	23-2	From PICCLK, (28)
t_{60j}	PICD0-1 Valid Delay (HtoL)	4.0	22.0	nS	23-2	From PICCLK, (28)
t_{61}	PICCLK Setup Time	5.0		nS		To CLK (31)
t_{62}	PICCLK Hold Time	2.0		nS		To CLK (31)
t_{63}	PICCLK Ratio (CLK/PICCLK)	4.0		nS		(32)

Table 23-11. Pentium® Processor (815\100, 1110\133) Dual Processor Mode AC Specifications for 66-MHz Bus Operation

$3.135 \le V_{CC} \le 3.6V$, T_{CASE} = 0 to 70°C, C_L = 0 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{80}	PBREQ#, PBGNT#, PHIT#, PHITM# Flight Time	0	2.0	nS	23-8	(29)
t_{83a}	A5-A31 Setup Time	3.7		nS	23-4	(18), (21), (26)
t_{83b}	D/C#, W/R#, CACHE#, LOCK#, SCYC Setup Time	4.0		nS	23-4	(18), (21)
t_{83c}	ADS#, M/IO# Setup Time	5.8		nS	23-4	(18), (21)
t_{83d}	HIT#, HITM# Setup Time	6.0		nS	23-4	(18), (21)
t_{83e}	HLDA Setup Time	6.0		nS	23-4	(18), (21)
t_{84}	ADS#, D/C#, W/R#, M/IO#, CACHE#, LOCK#, A5-A31, HLDA, HIT#, HITM#, SCYC Hold Time	1.0		nS	23-4	(18), (21)
t_{85}	DPEN# Valid Time		10.0	CLKs		(18), (19), (23)
t_{86}	DPEN# Hold Time	2.0		CLKs		(18), (20), (23)
t_{87}	APIC ID (BE0#-BE3#) Setup Time	2.0		CLKs	23-5	(23) to falling Edge of RESET
t_{88}	APIC ID (BE0#-BE3#) Hold Time	2.0		CLKs	23-5	(23) from Falling Edge of RESET
t_{89}	D/P# Valid Delay	1.0	8.0	nS	23-2	Primary processor only

NOTES:

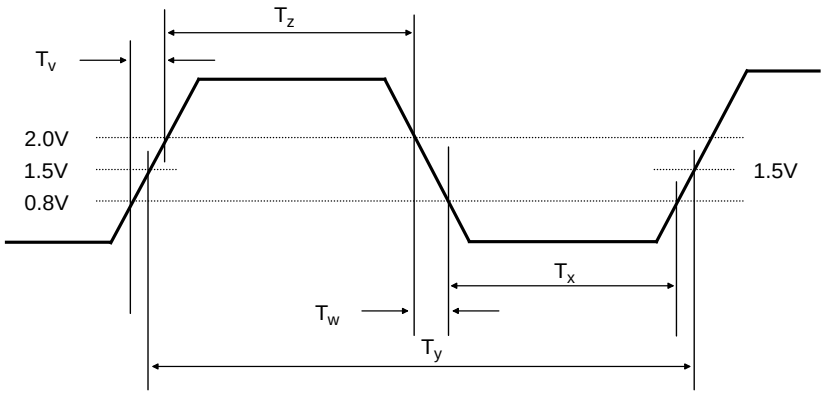
Notes 2, 6, and 14 are general and apply to all standard TTL signals used with the Pentium® Processor family.

1. Not 100% tested. Guaranteed by design.
2. TTL input test waveforms are assumed to be 0 to 3V transitions with 1V/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. 0.8V/ns ≤ CLK input rise/fall time ≤ 8V/ns.
6. 0.3V/ns ≤ Input rise/fall time ≤ 5V/ns.
7. Referenced to TCK rising edge.
8. Referenced to TCK falling edge.
9. 1 ns can be added to the maximum TCK rise and fall times for every 10 MHz of frequency below 33 MHz.
10. During probe mode operation, do not use the boundary scan timings ($t_{55\text{-}58}$).
11. FRCMC# should be tied to V_{CC} (high) to ensure proper operation of the Pentium processor (610\75, 735\90, 815\100, 1000\120) as a primary processor.

12. Setup time is required to guarantee recognition on a specific clock. Pentium processor (610\75, 735\90, 815\100, 1000\120) must meet this specification for dual processor operation for the FLUSH# and RESET signals.
13. Hold time is required to guarantee recognition on a specific clock. Pentium processor (610\75, 735\90, 815\100, 1000\120) must meet this specification for dual processor operation for the FLUSH# and RESET signals.
14. All TTL timings are referenced from 1.5V.
15. To guarantee proper asynchronous recognition, the signal must have been de-asserted (inactive) for a minimum of 2 clocks before being returned active and must meet the minimum pulse width.
16. This input may be driven asynchronously. However, when operating two processors in dual processing mode, FLUSH# and RESET must be asserted synchronously to both processors.
17. When driven asynchronously, RESET, NMI, FLUSH#, R/S#, INIT, and SMI# must be de-asserted (inactive) for a minimum of 2 clocks before being returned active.
18. Timings are valid only when Dual processor is present.
19. Maximum time DPEN# is valid from rising edge of RESET.
20. Minimum time DPEN# is valid after falling edge of RESET.
21. The D/C#, M/IO#, W/R#, CACHE#, and A5-A31 signals are sampled only on the CLK that ADS# is active.
22. BF and CPUTYP should be strapped to V_{CC} or V_{SS}.
23. RESET is synchronous in dual processing mode and functional redundancy checking mode. All signals which have a setup or hold time with respect to a falling or rising edge of RESET in UP mode, should be measured with respect to the first processor clock edge in which RESET is sampled either active or inactive in dual processing and functional redundancy checking modes.
24. The PHIT# and PHITM# signals operate at the core frequency (75, 90, 100, 120, or 133 MHz).
25. These signals are measured on the rising edge of adjacent CLKs at 1.5V. 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. The amount of jitter present must be accounted for as a component of CLK skew between devices.
26. In dual processing mode, timing t_{14} is replaced by t_{83a}. Timing t_{14} is required for external snooping (e.g. address setup to the CLK in which EADS# is sampled active) in both uniprocessor and dual processor modes.
27. BRDYC# and BUSCHK# are used as reset configuration signals to select buffer size.
28. This assumes an external pullup resistor to V_{CC} and a lumped capacitive load. The pullup resistor must be between 150 ohms and 1K ohms, the capacitance must be between 20 pF and 240 pF, and the RC product must be between 3 ns and 36 ns.
29. This is a flight time specification, that includes both flight time and clock skew. The flight time is the time from where the unloaded driver crosses 1.5V (50% of min V_{CC}), to where the receiver crosses the 1.5V level (50% of min V_{CC}). See Figure 23.8.

** 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.

31. This is for the Lock Step operation of the component only. This guarantees that APIC interrupts will be recognized on specific clocks to support 2 processors running in a Lock Step fashion including FRC mode. FRC on the APIC pins is not supported but mismatches on these pins will result in a mismatch on other pins of the CPU.
32. The CLK to PICCLK ratio for lock step operation has to be an integer and the ratio (CLK/PICCLK) cannot be smaller than 4.



$T_v = t_5, t_{49}, t_{60e}$

$T_w = t_4, t_{48}, t_{60f}$

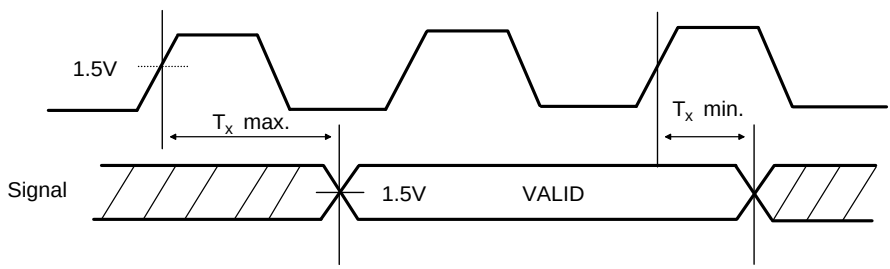
$T_x = t_3, t_{47}, t_{60e}$

$T_y = t_1, t_{45}, t_{60b}$

$T_z = t_2, t_{46}, t_{60c}$

PP0051

Figure 23-1. Clock Waveform



$T_x = t_6, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{60i}, t_{80a}, t_{89}$

PP0052

Figure 23-2. Valid Delay Timings

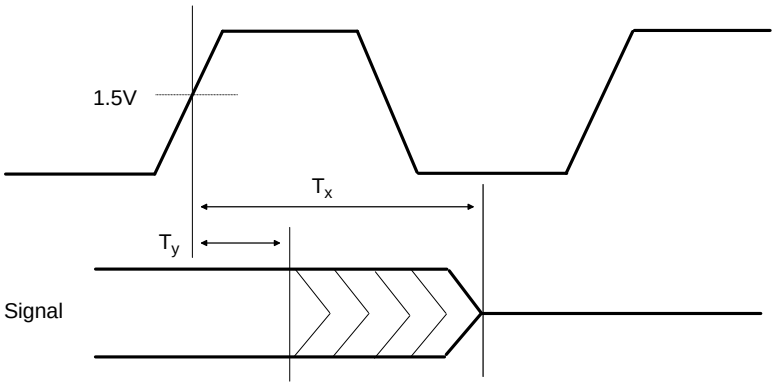


Figure 23-3. Float Delay Timings

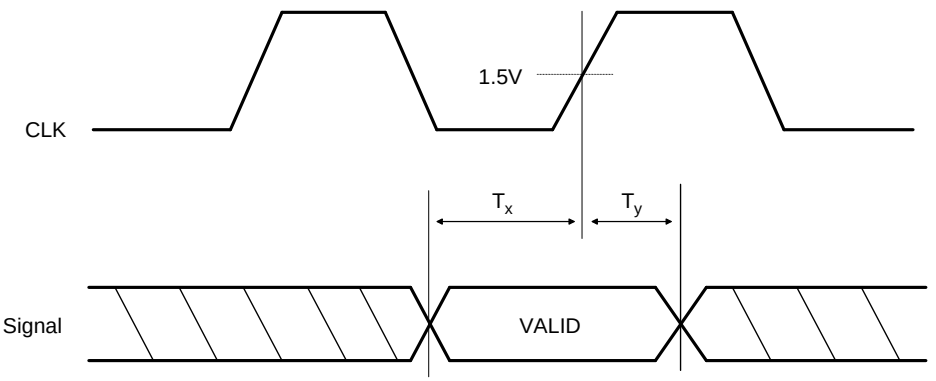
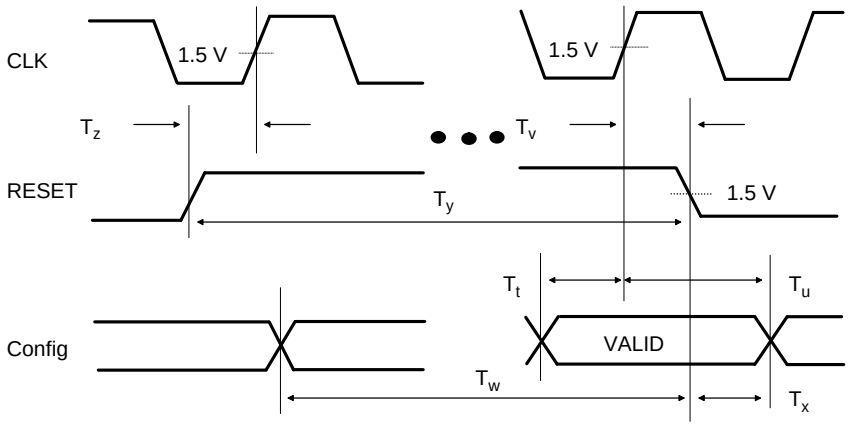


Figure 23-4. Setup and Hold Timings



$T_t = t_{40}$

$T_u = t_{41}$

$T_v = t_{37}$

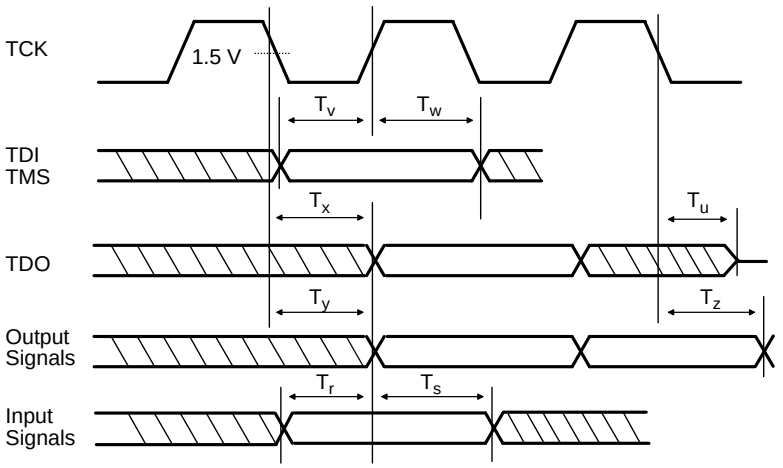
$T_w = t_{42}, t_{43a}, t_{43c}, t_{87}$

$T_x = t_{43b}, t_{43d}, t_{88}$

$T_y = t_{38}, t_{39}$

$T_z = t_{36}$

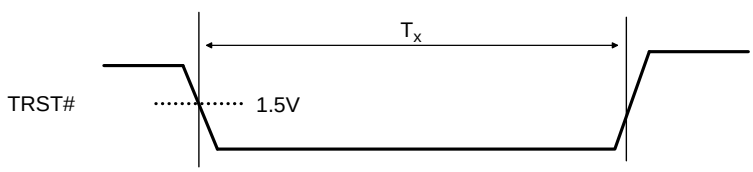
Figure 23-5. Reset and Configuration Timings



$T_r = t_{57}$
$T_s = t_{58}$
$T_u = t_{54}$
$T_v = t_{51}$
$T_w = t_{52}$
$T_x = t_{53}$
$T_y = t_{55}$
$T_z = t_{56}$

PP0056

Figure 23-6. Test Timings



$T_x = t_{50}$

PP0057

Figure 23-7. Test Reset Timings

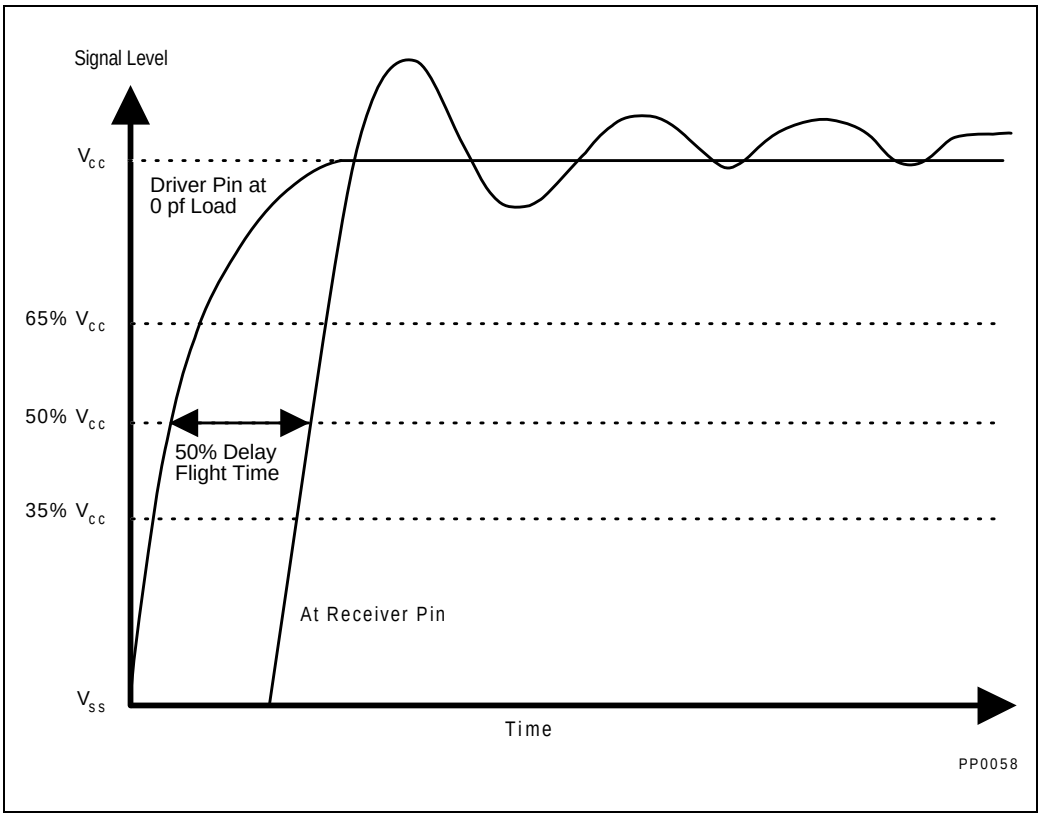


Figure 23-8. V_{cc} Measurement of Flight Time

24

I/O Buffer Models

CHAPTER 24
I/O BUFFER MODELS

The 3.3V buffers of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and the 5V buffers of the Pentium processor (510\60, 567\66) are different. This Chapter describes the I/O buffer models of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

The first order I/O buffer model is a simplified representation of the complex input and output buffers used in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). Figures 24-1 and 24-2 show the structure of the input buffer model and Figure 24-3 shows the output buffer model. Tables 24-1 and 24-2 show the parameters used to specify these models.

Although simplified, these buffer models will accurately model flight time and signal quality. For these parameters, there is very little added accuracy in a complete transistor model.

The following two models represent the input buffer models. The first model, Figure 24-1, represents all of the input buffers of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) except for a special group of input buffers. The second model, Figure 24-2, represents these special buffers. These buffers are the inputs: AHOLD, EADS#, KEN#, WB/WT#, INV, NA#, EWBE#, BOFF#, CLK, and PICCLK.

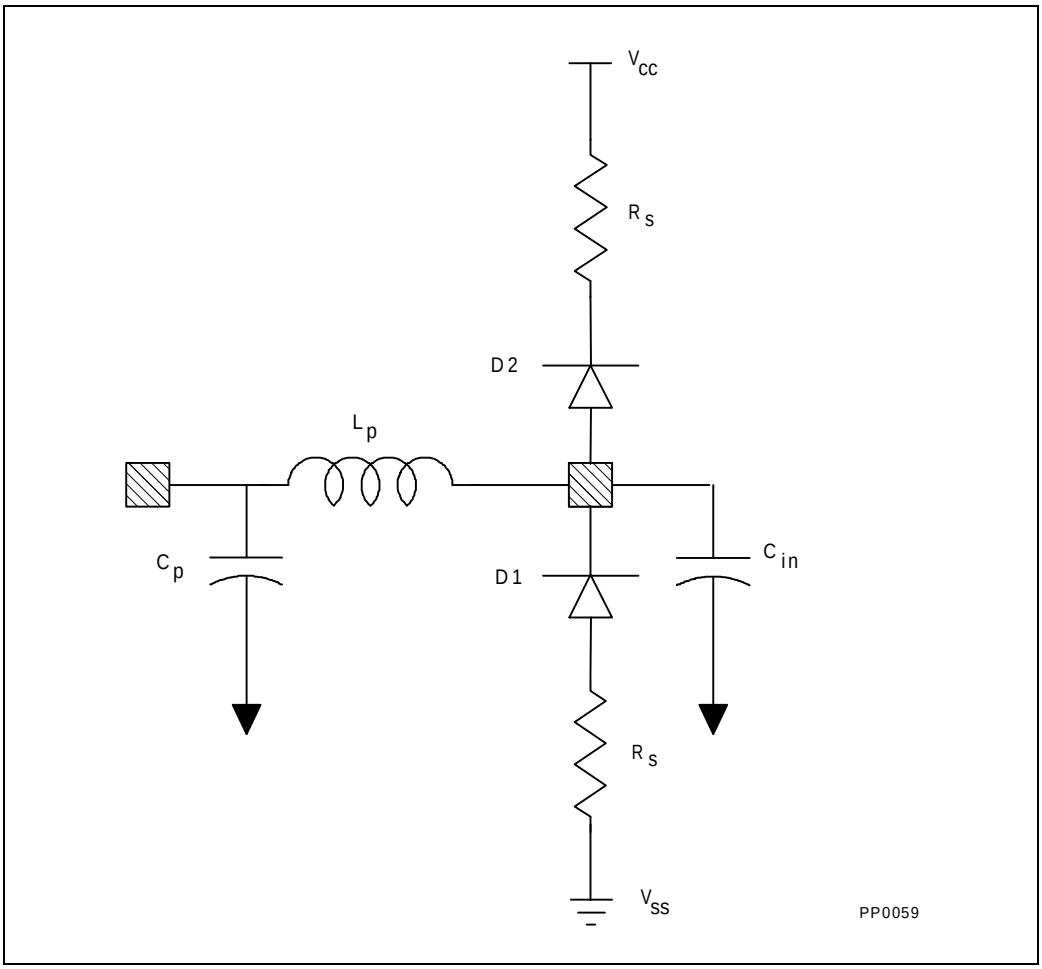


Figure 24-1. Input Buffer Model, Except Special Group

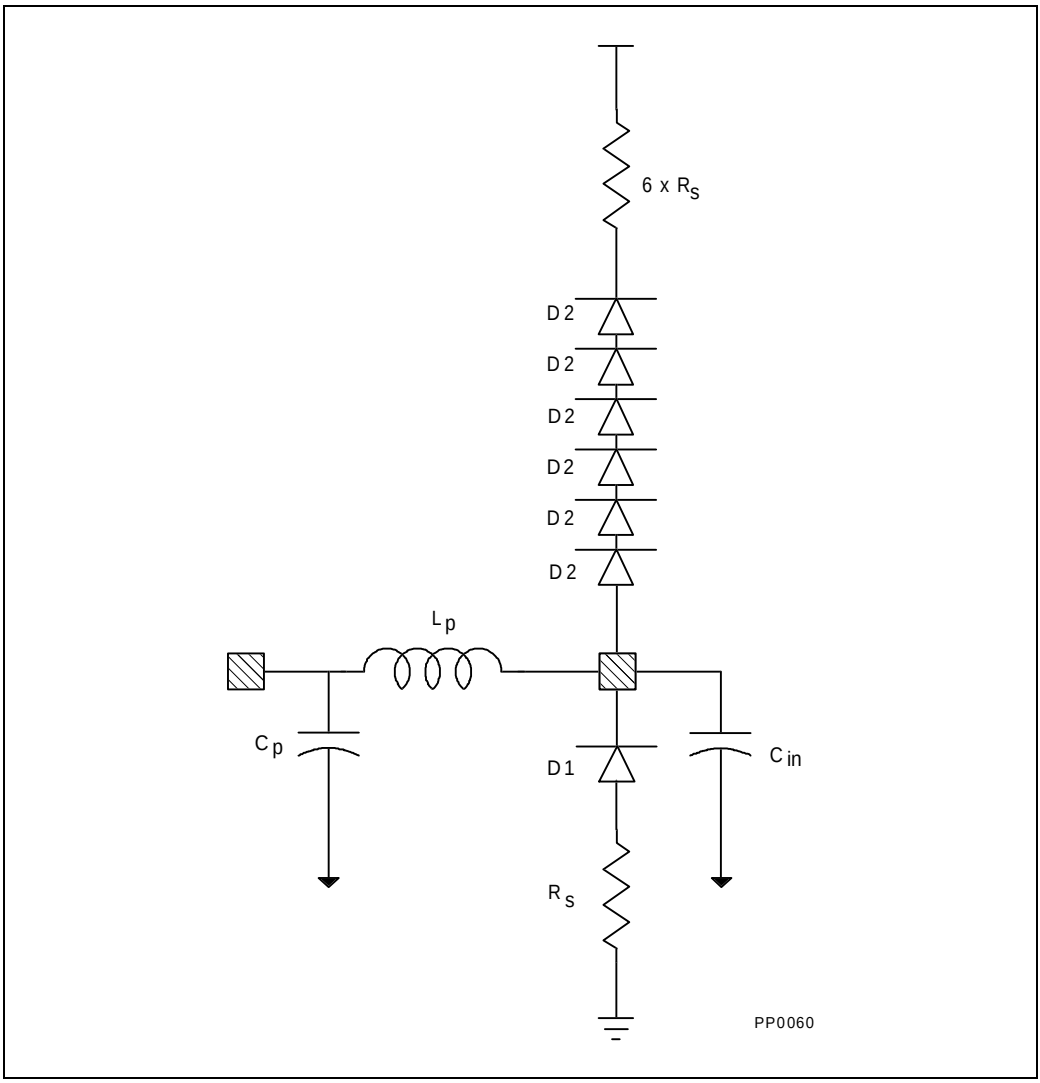


Figure 24-2. Input Buffer Model for Special Group

Table 24-1. 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.
Rs	Diode Series Resistance
D1, D2	Ideal Diodes

Figure 24-3 below shows the structure of the output buffer model. This model is used for all of the output buffers of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

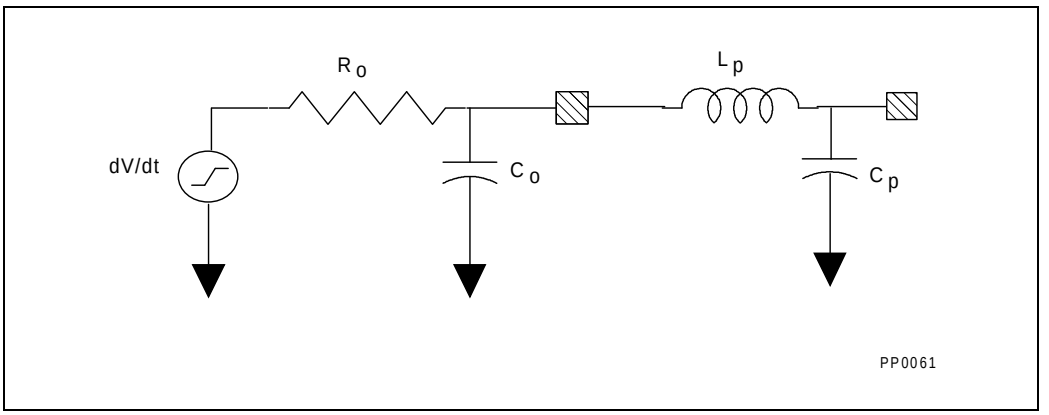


Figure 24-3. First Order Output Buffer Model

Table 24-2. 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.

In addition to the input and output buffer parameters, input protection diode models are provided for added accuracy. These diodes have been optimized to provide ESD protection

and provide some level of clamping. Although the diodes are not required for simulation, it may be more difficult to meet specifications without them.

Note however, some signal quality specifications require that the diodes be removed from the input model. The series resistors (Rs) are a part of the diode model. Remove these when removing the diodes from the input model.

24.1. BUFFER MODEL PARAMETERS

This section gives the parameters for each Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) input, output, and bidirectional signals, as well as the settings for the configurable buffers.

In dual processor mode, a few signals change from output signals to I/O signals. These signals are: ADS#, M/IO#, D/C#, W/R#, LOCK#, CACHE#, SCYC, HLDA, HIT#, and HITM#. When simulating these signals use the correct operation of the buffer while in DP mode.

Some pins on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) have selectable buffers sizes. These pins use the configurable output buffer EB2. Table 24-3 shows the drive level required at falling edge of RESET, to select the buffer strength. The buffer sizes selected should be the appropriate size required, otherwise A.C. timings might not be met, or too much overshoot and ringback may occur. There are no other selection choices, all the configurable buffers get set to the same size at the same time.

Table 24-3 shows the proper settings on BRDYC# and BUSCHK# for proper buffer size selection.

Table 24-3. Buffer Selection Chart

Environment	BRDYC#	BUSCHK#	Buffer Selection
Typical Stand Alone Component	1	X	EB2
Loaded Component	0	1	EB2A
Heavily Loaded Component	0	0	EB2B

NOTE:

X is a "DON'T CARE" (0 or 1).

Please refer to the following table for the groupings of the buffers. Due to the new package for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) these values have been simplified and the number of different buffers is reduced.

Table 24-4. Signal to Buffer Type

Signals	Type	Driver Buffer Type	Receiver Buffer Type
CLK	I		ER0
A20M#, AHOLD, BF, BOFF#, BRDY#, BRDYC#, BUSCHK#, EADS#, EWBE#, FLUSH#, FRCMC#, HOLD, IGNNE#, INIT, INTR, INV, KEN#, NA#, NMI, PEN#, PICCLK, R/S#, RESET, SMI#, STPCLK#, TCK, TDI, TMS, TRST#, WB/WT#	I		ER1
ADSC#, APCHK#, BE[7:5]#, BP[3:2], BREQ, FERR#, IERR#, PCD, PCHK#, PM0/BP0, PM1/BP1, PRDY, PWT, SMIACT#, TDO, U/O#	O	ED1	
A[31:21], AP, BE[4:0]#, CACHE#, D/C#, D[63:0], DP[8:0], HLDA, LOCK#, M/IO#, PBGNT#, PBREQ#, PHIT#, PHITM#, SCYC	I/O	EB1	EB1
A[20:3], ADS#, HITM#, W/R#	I/O	EB2/A/B	EB2
HIT#	I/O	EB3	EB3
PICD0, PICD1	I/O	EB4	EB4

The input, output and bidirectional buffers values are listed below. This table contains listings for all three types, do not get them confused during simulation. When a bidirectional pin is operating as an input, just use the Cin, Cp and Lp values, if it is operating as a driver use all the data parameters.

Table 24-5. Input, Output and Bidirectional Buffer Model Parameters

Buffer Type	Transition	dV/dt (V/nsec)		Ro (Ohms)		Cp (pF)		Lp (nH)		Co/Cin (pF)	
		min	max	min	max	min	max	min	max	min	max
ER0	Rising					3.0	5.0	4.0	6.0	0.8	1.2
(input)	Falling					3.0	5.0	4.0	6.0	0.8	1.2
ER1	Rising					1.1	5.3	7.7	15.3	0.8	1.2
(input)	Falling					1.1	5.3	7.7	15.3	0.8	1.2
ED1	Rising	3/3.0	3.7/0.9	21.6	53.1	1.1	5.8	8.1	16.3	2.0	2.6
(output)	Falling	3/2.8	3.7/0.8	17.5	50.7	1.1	5.8	8.1	16.3	2.0	2.6
EB1	Rising	3/3.0	3.7/0.9	21.6	53.1	1.3	7.0	8.2	18.4	2.0	2.6
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	1.3	7.0	8.2	18.4	2.0	2.6
EB2	Rising	3/3.0	3.7/0.9	21.6	53.1	1.3	5.4	8.5	16.0	9.1	9.7
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	1.3	5.4	8.5	16.0	9.1	9.7
EB2A	Rising	3/2.4	3.7/0.9	10.1	22.4	1.3	5.4	8.5	16.0	9.1	9.7
(bidir)	Falling	3/2.4	3.7/0.9	9.0	21.2	1.3	5.4	8.5	16.0	9.1	9.7
EB2B	Rising	3/1.8	3.7/0.7	5.5	12.9	1.3	5.4	8.5	16.0	9.1	9.7
(bidir)	Falling	3/1.8	3.7/0.7	4.6	12.3	1.3	5.4	8.5	16.0	9.1	9.7
EB3	Rising	3/3.0	3.7/0.9	21.6	53.1	1.9	4.2	10.5	14.3	3.3	3.9
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	1.9	4.2	10.5	14.3	3.3	3.9
EB4	Rising	3/3.0	3.7/0.9	100K*	100K*	2.0	4.3	10.7	14.6	5.0	7.0
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	2.0	4.3	10.7	14.6	5.0	7.0

* The buffer is an open drain, and for simmulation purposes should be a very large internal resistance with additional external pull-up.

Table 24-6. Input Buffer Model Parameters: D (Diodes)

Symbol	Parameter	D1	D2
IS	Saturation Current	1.4e-14A	2.78e-16A
N	Emission Coefficient	1.19	1.00
RS	Series Resistance	6.5 ohms	6.5 ohms
TT	Transit Time	3 ns	6 ns
VJ	PN Potential	0.983V	0.967V
CJ0	Zero Bias PN Capacitance	0.281 pF	0.365 pF
M	PN Grading Coefficient	0.385	0.376

24.2. SIGNAL QUALITY SPECIFICATIONS

Signals driven by the system into the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) must meet signal quality specifications to guarantee that the components read data properly and to ensure that incoming signals do not affect the reliability of the component. There are two signal quality parameters: Ringback and Settling Time.

24.2.1. Ringback

Excessive ringback can contribute to long-term reliability degradation of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), and can cause false signal detection. Ringback is simulated at the input pin of a component using the input buffer model. Ringback can be simulated with or without the diodes that are in the input buffer model.

Ringback is the absolute value of the maximum voltage at the receiving pin below V_{CC} (or above V_{SS}) relative to V_{CC} (or V_{SS}) level after the signal has reached its maximum voltage level. The input diodes are assumed present.

Maximum Ringback on Inputs = 0.8V

(with diodes)

If simulated without the input diodes, follow the Maximum Overshoot/Undershoot specification. By meeting the overshoot/undershoot specification, the signal is guaranteed not to ringback excessively.

If simulated with the diodes present in the input model, follow the maximum ringback specification.

Overshoot (Undershoot) is the absolute value of the maximum voltage above V_{CC} (below V_{SS}). The guideline assumes the absence of diodes on the input.

- Maximum Overshoot/Undershoot on 5V 82497 Cache Controller, 82492 Cache SRAM, and Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Inputs (CLK and PICCLK only) = 1.6V above $V_{CC}5$ (without diodes)
- Maximum Overshoot/Undershoot on 3.3V Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Inputs (not CLK and PICCLK) = 1.4V above $V_{CC}3$ (without diodes)

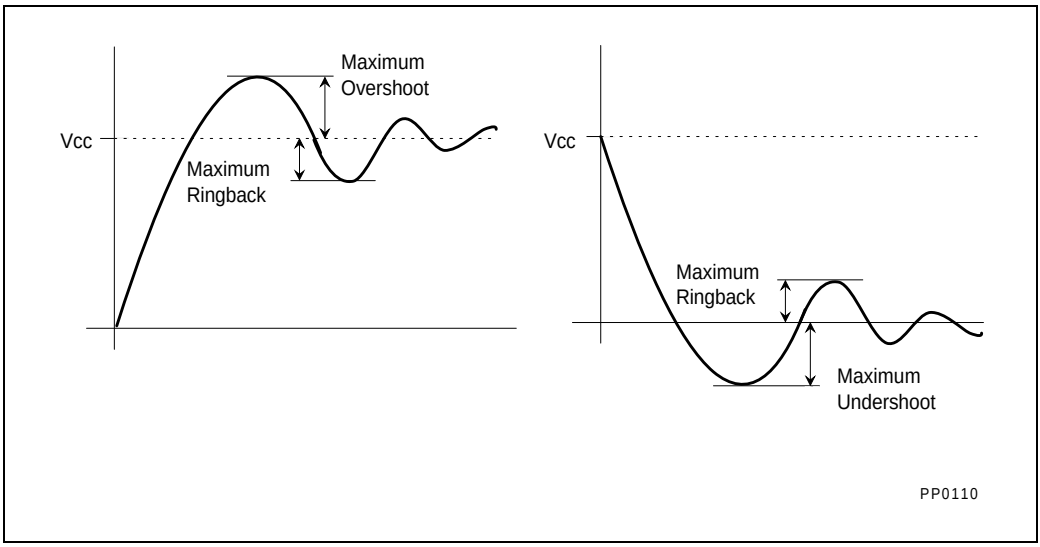


Figure 24-4. Overshoot/Undershoot and Ringback Guidelines

24.2.2. Settling Time

The settling time is defined as the time a signal requires at the receiver to settle within 10% of V_{CC} or V_{SS}. Settling time is the maximum time allowed for a signal to reach within 10% of its final value.

Most available simulation tools are unable to simulate settling time so that it accurately reflects silicon measurements. On a physical board, second-order effects and other effects serve to dampen the signal at the receiver. Because of all these concerns, settling time is a recommendation or a tool for layout tuning and not a specification.

Settling time is simulated at the slow corner, to make sure that there is no impact on the flight times of the signals if the waveform has not settled. Settling time may be simulated with the diodes included or excluded from the input buffer model. If diodes are included, settling time recommendation will be easier to meet.

Although simulated settling time has not shown good correlation with physical, measured settling time, settling time simulations can still be used as a tool to tune layouts.

Use the following procedure to verify board simulation and tuning with concerns for settling time.

1. Simulate settling time at the slow corner for a particular signal.
2. If settling time violations occur, simulate signal trace with D.C. diodes in place at the receiver pin. The D.C. diode behaves almost identically to the actual (non-linear) diode on the part as long as excessive overshoot does not occur.
3. If settling time violations still occur, simulate flight times for 5 consecutive cycles for that particular signal.
4. If flight time values are consistent over the 5 simulations, settling time should not be a concern. If however, flight times are not consistent over the 5 simulations, tuning of the layout is required.
5. Note that, for signals that are allocated 2 cycles for flight time, the recommended settling time is doubled.

Maximum Settling Time to within 10% of V_{CC} is: 12.5ns @66 MHz

14.2ns @60 MHz

17.5ns @50 MHz

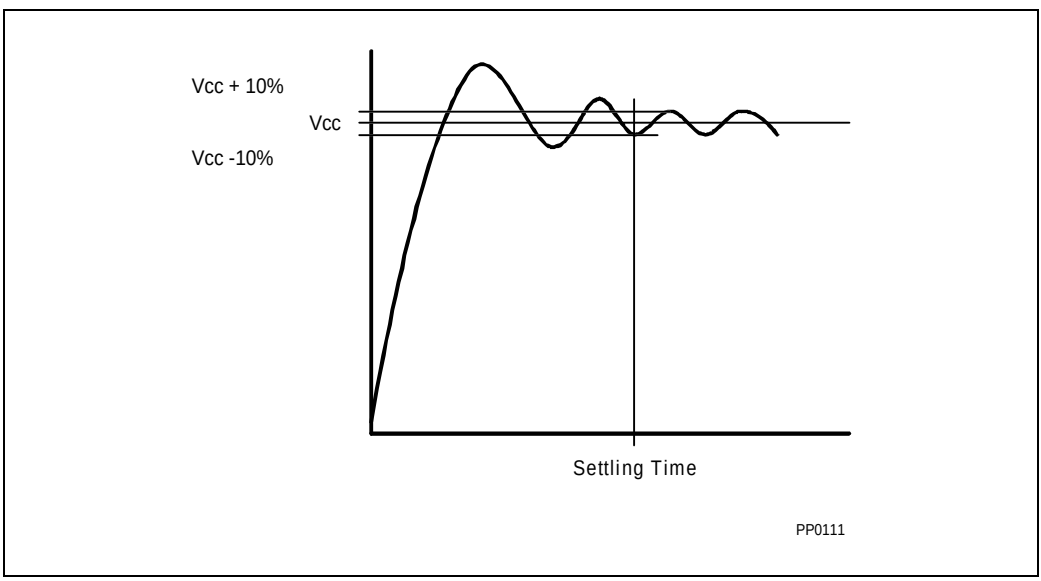


Figure 24-5. Settling Time

25

Mechanical Specifications

CHAPTER 25
MECHANICAL SPECIFICATIONS

The physical packages of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) and the Pentium processor (510\60, 567\66) are different. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses a 296-pin Staggered PGA package, while the Pentium processor (510\60, 567\66) uses a 273-pin PGA.

The pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are arranged in a 37 x 37 matrix and the package dimensions are 1.95" x 1.95" (Table 25-1). A 1.25" x 1.25" copper tungsten heat spreader may be attached to the top of the ceramic.

Intel provides two kinds of packages for Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). One has a 1.25" × 1.25" copper tungsten heat spreader brazed to the top of the ceramic, and the later package conversion removes the heat spreader from the top of the package, and replaces the metal lid covering the die with a ceramic lid. Table 25-1 summarizes the information of both packages.

Table 25-1. Package Information Summary

	Package Type	Total Pins	Pin Array	Package Size
Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133)	SPGA	296	37 x 37	1.95" x 1.95" 4.95cm x 4.95cm

Figure 25-1 and Figure 25-2 show the dimensions of the packages with the heat spreader and without the heat spreader respectively. Table 25-2 and Table 25-3 provide the mechanical specifications of the packages with the heat spreader and without the heat spreader respectively.

Table 25-2. Package (with the heat spreader) Dimensions

Family: Ceramic Pin Grid Array Package						
Symbol	**Millimeter**			**Inches**		
	Min	**Max**	**Notes**	**Min**	**Max**	**Notes**
A	3.91	4.70	Solid Lid	0.154	0.185	Solid Lid
A1	0.33	0.43	Solid Lid	0.013	0.017	Solid Lid
A2	2.62	2.97		0.103	0.117	
B	0.43	0.51		0.017	0.020	
D	49.28	49.91		1.940	1.965	
D1	45.47	45.97		1.790	1.810	
D2	31.50	32.00	Square	1.240	1.260	Square
D3	33.99	34.59		1.338	1.362	
D4	8.00	9.91		0.315	0.390	
E1	2.41	2.67		0.095	0.105	
E2	1.14	1.40		0.045	0.055	
F		0.127	Diagonal		0.005	Diagonal
L	3.05	3.30		0.120	0.130	
N	296			296		
S1	1.52	2.54		0.060	0.100	

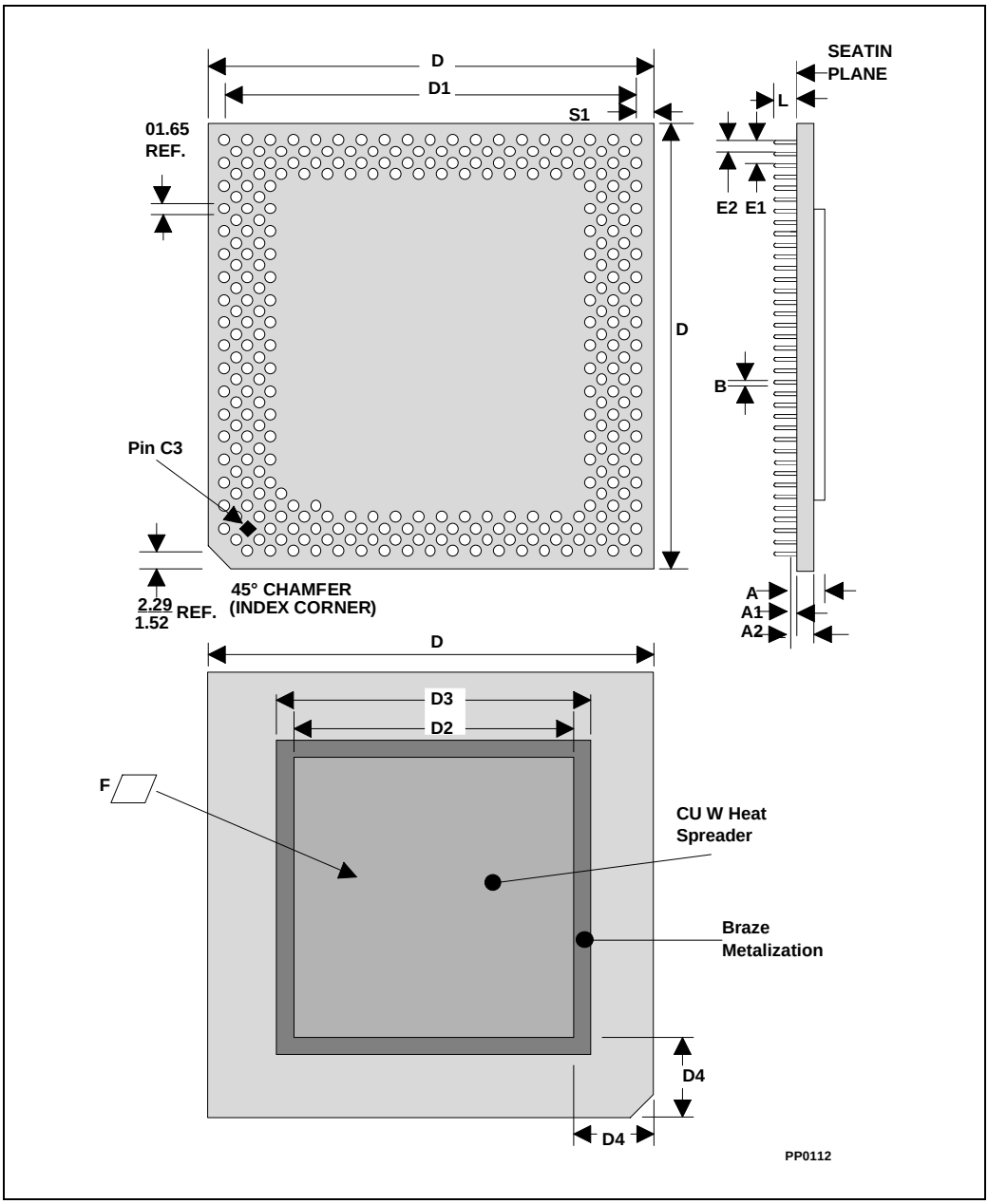


Figure 25-1. Package (with the heat spreader) Dimensions

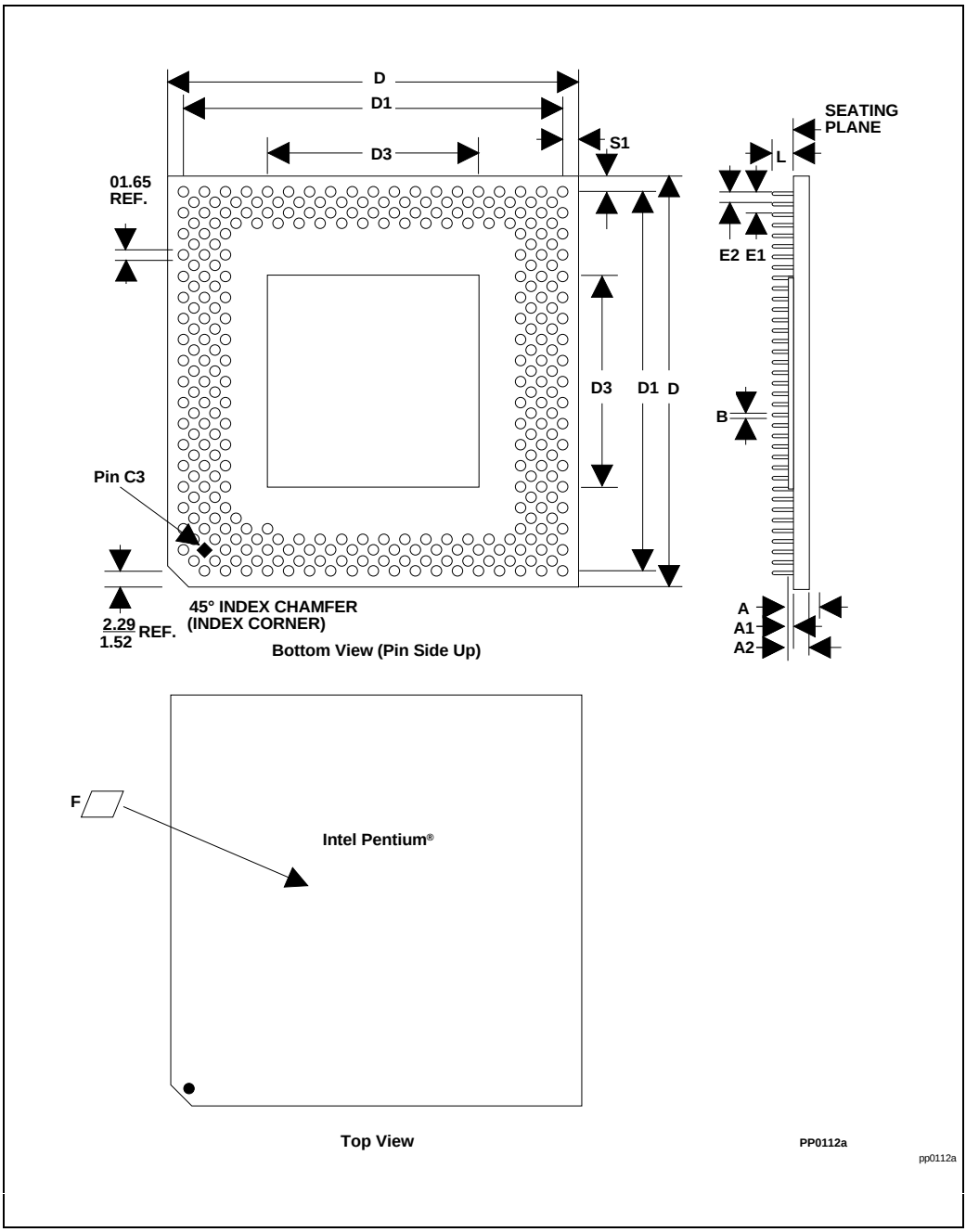


Figure 25-2. Package (without the heat spreader) Dimensions

Table 25-3. Package (without the heat spreader) Dimensions

Family: 296 Pin Ceramic Pin Grid Array Package						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	3.27	3.83	Ceramic Lid	0.129	0.151	Ceramic Lid
A1	0.66	0.86	Ceramic Lid	0.026	0.034	Ceramic Lid
A2	2.62	2.97		0.103	0.117	
B	.043	0.51		0.017	0.020	
D	49.28	49.78		1.940	1.960	
D1	45.59	45.85		1.795	1.805	
D3	24.00	24.25	Includes Fillet	0.945	0.955	Includes Fillet
e1	2.29	2.79		0.090	0.110	
F		0.127	Flatness of the top of the package, measured diagonally		0.005	Flatness of the top of the package, measured diagonally
L	3.05	3.30		0.120	1.130	
N	296		Total Pins	296		Total Pins
S1	1.52	2.54		0.060	0.100	

26

Thermal Specifications

CHAPTER 26
THERMAL SPECIFICATIONS

Due to the advanced 3.3V BiCMOS process that it is produced on, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) dissipates less power than the Pentium processor (510\60, 567\66).

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is specified for proper operation when case temperature, T_{CASE}, (T_C) is within the specified range of 0°C to 70°C.

26.1. MEASURING THERMAL VALUES

To verify that the proper T_C (case temperature) is maintained, it should be measured at the center of the package top surface on the heat spreader (opposite of the pins). The measurement is made in the same way with or without a heat sink attached. When a heat sink is attached a hole (smaller than .150" diameter) should be drilled through the heat sink to allow probing the center of the package. See Figure 26-1 for an illustration of how to measure T_C.

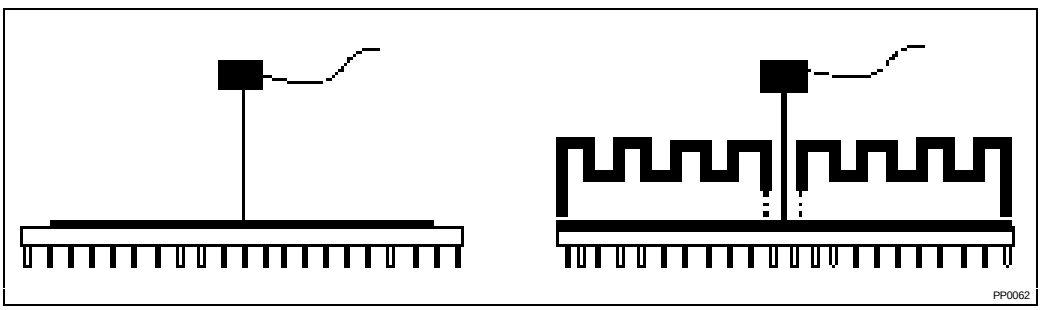


Figure 26-1. Technique for Measuring T_{C*}

* Though the figure shows the package with a heat spreader, the same technique applies to measuring TC of the package without a heat spreader.

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 26-1.
- The hole size should be smaller than .150" in diameter.

26.1.1. Thermal Equations and Data

For the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), an ambient temperature, T_A (air temperature around the processor), is not specified directly. The only restriction is that T_C is met. To calculate T_A values, the following equations may be used:

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

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

where, T_A and T_C = ambient and case temperature. (^oC)

Θ_{CA} = case-to-ambient thermal resistance. (^oC/Watt)

Θ_{JA} = junction-to-ambient thermal resistance. (^oC/Watt)

Θ_{JC} = junction-to-case thermal resistance. (^oC/Watt)

P = maximum power consumption (Watt)

(See DC specifications in Chapter 23 for detailed power consumption specifications.)

Table 26-1 lists the Θ_{CA} values for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) with heat spreader and passive heat sinks. Figure 26-2 shows Table 26-1 in graphic format.

Table 26-2 lists the ΘCA values for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) with passive heat sinks, but without a heat spreader. Figure 26-3 shows Table 26-2 in graphic format.

Table 26-1. Thermal Resistances for Packages with Heat Spreader

		Θ_{CA} (^oC/Watt) vs. Laminar Airflow (linear ft/min)					
Heat Sink in inches	**Θ_{JC} (^oC/Watt)**	**0**	**100**	**200**	**400**	**600**	**800**
1.95x1.95x0.25	0.9	8.7	7.6	6.2	4.0	3.2	2.6
1.95x1.95x0.35	0.9	8.4	7.1	5.6	3.6	2.9	2.4
1.95x1.95x0.45	0.9	8.0	6.6	4.9	3.2	2.5	2.1
1.95x1.95x0.55	0.9	7.7	6.1	4.3	2.8	2.2	1.9
1.95x1.95x0.65	0.9	7.3	5.6	3.9	2.6	2.0	1.7
1.95x1.95x0.80	0.9	6.6	4.9	3.5	2.2	1.8	1.6
1.95x1.95x1.00	0.9	5.9	4.2	3.2	2.2	1.7	1.4
1.95x1.95x1.20	0.9	5.5	3.9	2.9	2.0	1.6	1.4
1.95x1.95x1.40	0.9	5.0	3.5	2.6	1.8	1.5	1.3
Without Heat Sink	1.4	11.4	10.5	8.7	5.7	4.5	3.8

Table 26-2. Thermal Resistances for Packages without Spreader

Heat Sink in Inches	ΘJC (°C/Watt)	Θ_{CA} (^{O}C/Watt) vs. Laminar Airflow (linear ft/min)					
		0	100	200	400	600	800
0.25	0.8	9.1	8.0	6.6	4.4	3.6	3.0
0.35	0.8	8.8	7.5	6.0	4.0	3.3	2.8
0.45	0.8	8.4	7.0	5.3	3.6	2.9	2.5
0.55	0.8	8.1	6.5	4.7	3.2	2.6	2.3
0.65	0.8	7.7	6.0	4.3	3.0	2.4	2.1
0.80	0.8	7.0	5.3	3.9	2.8	2.2	2.0
1.00	0.8	6.3	4.6	3.6	2.6	2.1	1.8
1.20	0.8	5.9	4.3	3.3	2.4	2.0	1.8
1.40	0.8	5.4	3.9	3.0	2.2	1.9	1.7
Without Heat Sink	1.3	14.4	13.1	11.7	8.8	7.4	6.5

NOTES:

Heat sinks are omni directional pin aluminum alloy.

Features were based on standard extrusion practices for a given height

Pin size ranged from 50 to 129 mils

Pin spacing ranged from 93 to 175 mils

Based thickness ranged from 79 to 200 mils

Heat sink attach was 0.005" of thermal grease.

Attach thickness of 0.002" will improve performance approximately 0.3^{O}C/Watt

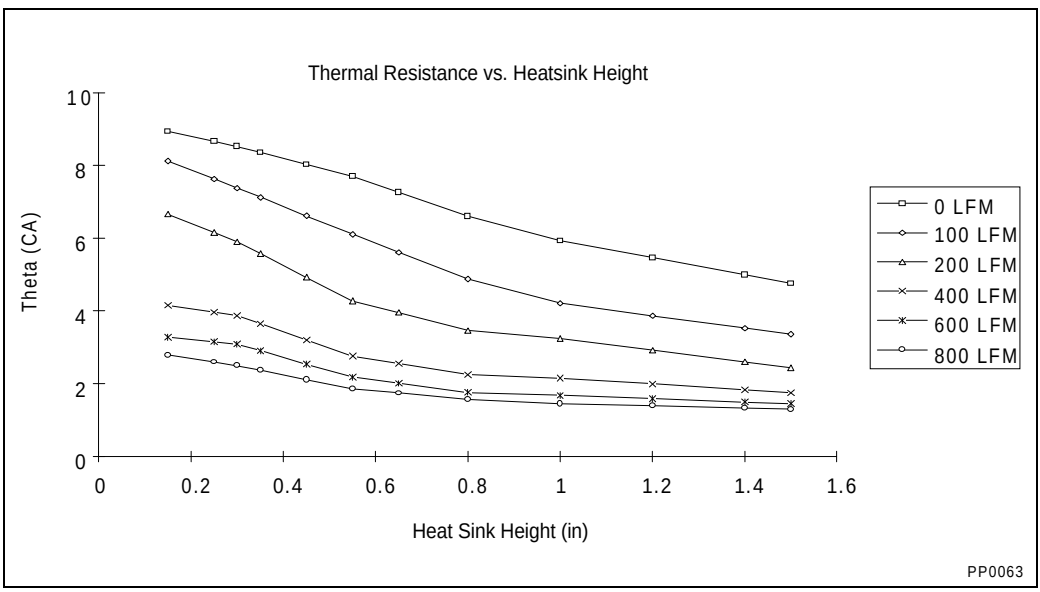


Figure 26-2. Thermal Resistance vs. Heat Sink Height

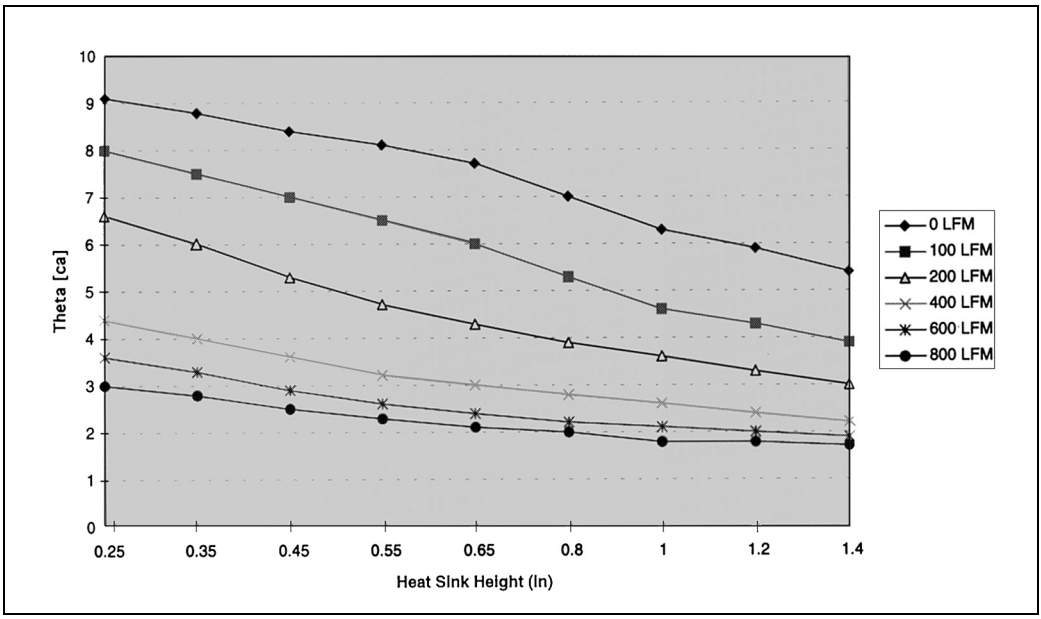


Figure 26-3. Thermal Resistance vs. Heatsink Height (Non-Spreader Package)

27

Testability

CHAPTER 27
TESTABILITY

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports the same testability features as found in Chapter 11 for the Pentium processor family. Since the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) operates at 3.3V and the Pentium processor (510\60, 567\66) operates at 5V, the following section describes the boundary scan considerations to handle this change.

27.1. BOUNDARY SCAN 3.3V CONSIDERATIONS

The boundary scan pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are all 3.3V signals (including TCK).

Intel recommends that the system design use separate scan chains for its 5V portions and 3.3V portions.

Contact your local Intel representative for BSDL files describing the implementation of the boundary scan chain.

If a single scan chain must be used:

1. The system TCK must be a 3.3V-level clock (unlike CLK and PICCLK).
2. TDO from 5V components must have a level translator before connecting to the TDI input of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

27.2. BOUNDARY SCAN CHAIN

The Boundary Scan Register for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) contains a cell for each pin. The following is the bit order of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) boundary scan register (left to right, top to bottom):

TDI -> Disapsba*, PICD1, PICD0, Reserved, PICCLK, D0, D1, D2, D3, D4, D5, D6, D7, DP0, D8, D9, D10, D11, D12, D13, D14, D15, DP1, D16, D17, D18, D19, D20, D21, D22, D23, DP2, D24, D25, D26, D27, D28, D29, D30, D31, DP3, D32, D33, D34, D35, D36, D37, D38, D39, DP4, D40, D41, D42, D43, D44, D45, D46, Diswr*, D47, DP5, D48, D49, D50, D51, D52, D53, D54, D55, DP6, D56, D57, D58, D59, D60, D61, D62, D63, DP7, IERR#, FERR#, PM0/BP0, PM1/BP1, BP2, BP3, M/IO#, CACHE#, EWBE#, INV, AHOLD, KEN#, BRDYC#, BRDY#, BOFF#, NA#, Disbus*, Dismisch*, Disbus1*, Dismisc*, Disua2bus*, Disua1bus*, Dismisca*, Dismiscfa*, WB/WT#, HOLD, PHITM#, PHIT#, PBREQ#, PBGNT#, SMIACT#, PRDY, PCHK#, APCHK#, BREQ, HLDA, AP, LOCK#, ADSC#, PCD, PWT, D/C#, EADS#, ADS#, HITM#, HIT#, W/R#, BUSCHK#, FLUSH#, A20M#,

BE0#, BE1#, BE2#, BE3#, BE4#, BE5#, BE6#, BE7#, SCYC, CLK, RESET, Disabus*, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A31, A30, A29, A28, A27, A26, A25, A24, A23, A22, A21, D/P#, NMI, R/S#, INTR, SMI#, IGNNE#, INIT, PEN#, FRCMC#, Reserved, Reserved, Reserved, Reserved, BF, STPCLK#, Reserved, Reserved, Reserved, Reserved, Reserved, Reserved -> TDO

"Reserved" includes the "NC" signals on the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).

The cells marked with an "*" 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
Dismiscfa:	D/P#, FERR#
Dismisca:	IERR#
Disua1bus:	PBREQ#, PHIT#, PHITM#
Disua2bus:	PBGNT#
Dismisc:	APCHK#, PHCK#, PRDY#, BP3, BP2, PM1/BP1, PM0/BP0
Disbus1:	ADS#, ADSC#, LOCK#
Dismisch:	HIT#, HITM#, HLDA, BREQ#, SMIACT#
Disbus:	SCYC, BE7#-BE0#, W/R#, D/C#, PWT, PCD, CACHE#, M/IO#
Diswr:	DP7-DP0, D63-D0
Disapsba:	PICD0, PICD1

27.3. PRIVATE INTERFACE PINS

In a dual-processor system, the private interface pins are not floated in Tri-state Test mode. These pins are PBREQ#, PBGNT#, PHIT#, and PHITM#.

NOTE

When Tri-state Test mode is entered, by holding FLUSH# low at the falling edge of RESET, all output pins of the Pentium processor are set into a Tri-state Test mode. There are several pins that have internal pullups or pulldowns attached that show these pins going high or low, respectively, during Tri-state Test mode. There is one pin, PICD1, that has an internal pulldown attached that shows this pin going low during Tri-state Test mode. The five pins that have pullups are PHIT#, PHITM#, PBREQ#, PBGNT#, and PICD0. There are two other pins that have pullups attached during dual processor mode, HIT# and HITM#. The pullups on these pins (except HIT#) have a value of about 30K ohms, HIT# is about 2K ohms.

28

Error Detection

CHAPTER 28
ERROR DETECTION

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) supports the same error detection features as found in Chapter 12 for the Pentium processor family.

29

Execution Tracing

CHAPTER 29
EXECUTION TRACING

Due to pin and frequency changes in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), the execution tracing operation described in Chapter 13 for the Pentium processor family has been modified. This section describes these changes in detail.

29.1. EXECUTION TRACING CHANGES

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) uses special bus cycles to support execution tracing. These bus cycles, which are optional, have a significant impact on overall performance. Execution tracing allows the external hardware to track the flow of instructions as they execute inside the processor. Unlike the Pentium processor (510\60, 567\66), the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) only uses the special bus cycles, and does not have pins IU, IV, and IBT.

The special bus cycles generated by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are Branch Trace Messages. The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) format of these messages is different from the Pentium processor (510\60, 567\66) format. Also, because of physical limitations, the maximum number of outstanding taken branches allowed is two. Once the second taken branch reaches the last stage of the pipeline, execution is stalled until the first branch message is sent on the bus.

Branch trace messages may be enabled by setting the Execution Tracing bit, TR, of TR12 (bit 1) to a 1. Once enabled, there are two forms of branch trace messages: normal and fast. Normal messages produce two cycles, one for the linear address of the instruction causing the taken branch, and one for the branch target linear address. Fast messages only produce the second of these two cycles. Fast execution tracing is enabled by setting bit 8 of TR12 to 1.

Normal and fast branch trace messages may be delayed by 0 or more clocks after the cycle in which the branch was taken depending on the bus activity. Also, higher priority cycles may be run between the first and second cycles of a normal branch trace message. In dual-processor mode, branch trace message cycles may be interleaved with cycles from the other processor. Branch trace message cycles are buffered so they do not normally stall the processor.

Branch trace messages, normal and fast, may be identified by the following cycle:

M/IO# = 0

D/C# = 0

W/R# = 1

BE [7:0]# = 0DFh

The address and data bus fields for the two bus cycles associated with a branch trace message are defined below:

First Cycle (Normal)

A31 - A4	Bits 31 - 4 of the branch target linear address
A3	"1" if the default operand size is 32 bits
	"0" if the default operand size is 16 bits
D63 - D60	Bits 3 - 0 of the branch target linear address
D59	"0" - indicating the first of the two cycles
D58-D00	Reserved. Driven to a valid state, but must be ignored

Second Cycle (Normal)

A31 - A4	Bits 31 - 4 of the linear address of the instruction causing the taken branch
A3	"1" if the default operand size is 32 bits
	"0" if the default operand size is 16 bits
D63 - D60	Bits 3 - 0 of the linear address of the instruction causing the taken branch
D59	"1" - indicating the second of the two cycles
D58-D00	Reserved. Driven to a valid state, but must be ignored

Fast Cycle

A31 - A4	Bits 31 - 4 of the linear address of the instruction causing the taken branch
A3	"1" if the default operand size is 32 bits
	"0" if the default operand size is 16 bits
D63 - D60	Bits 3 - 0 of the linear address of the instruction causing the taken branch
D59	Driven to a "1"
D58-D00	Reserved. Driven to a valid state, but must be ignored

In addition to conditional branches, jumps, calls, returns, software interrupts, and interrupt returns, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) treats the following operations as causing taken branches:

- serializing instructions
- segment descriptor loads
- hardware interrupts
- exceptions that invoke a trap or fault handler

With execution tracing enabled, these operations will also cause a corresponding branch trace message cycle.

Note that in the Pentium processor (510\60, 567\66), the data bus was undefined for all special bus cycles. However, in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), the data bus is valid during branch trace message special cycles.

30

Power Management

CHAPTER 30
POWER MANAGEMENT

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) implements Intel's System Management Mode (SMM) architecture. This chapter describes the hardware interface to SMM and Clock Control. For a detailed architectural description, refer to the Power Management chapter in the *Pentium® Processor Family Developer's Manual, Volume 3.*

30.1. PENTIUM® PROCESSOR (610\75, 735\90, 815\100, 1000\120, 1110\133) POWER MANAGEMENT DIFFERENCES vs. PENTIUM PROCESSOR (510\60, 567\66)

- System Management Interrupt can be delivered through the SMI# signal or through the local APIC using the SMI# message. The Pentium processor (510\60, 567\66) supported only the SMI# pin interface. The addition of the APIC SMI# message enhances the SMI interface, and provides for SMI delivery in APIC-based Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) dual processing systems.
- In dual processing systems, SMIACT# from the bus master (MRM) behaves the same as in Pentium processor (510\60, 567\66) systems. If the LRM processor is the CPU in SMM mode, SMIACT# will be inactive and remain so until that processor becomes the MRM.
- The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is capable of supporting an SMM I/O instruction restart feature (not supported in the Pentium processor (510\60, 567\66)). This feature is automatically disabled following RESET. To enable the I/O instruction restart feature, set bit 9 of the TR12 register to "1".
- The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) default SMM revision identifier changes to 2 when the SMM I/O instruction restart feature is enabled. The Pentium processor (510\60, 567\66) revision ID is 0.
- SMI# is NOT recognized by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) in the shutdown state.
- Clock control was not implemented in the Pentium processor (510\60, 567\66).

30.2. SYSTEM MANAGEMENT INTERRUPT VIA APIC

When SMI# is asserted (SMI# pin asserted low or APIC SMI# message) it causes the processor to invoke SMM.

30.3. I/O INSTRUCTION RESTART

For additonal information, refer to Chapter 20 of the *Pentium® Processor Family Developer's Manual, Volume 3*.

30.3.1. SMI# Synchronization for I/O Instruction Restart

The SMI# signal is synchronized internally and must be asserted at least three (3) CLK periods prior to asserting the BRDY# signal in order to guarantee recognition on a specific I/O instruction boundary. This is important for servicing an I/O trap with an SMI# handler. Due to the asynchronous nature of SMI# delivery with the APIC, it is impossible to synchronize the assertion of BRDY#. As a result, the SMM I/O instruction restart feature cannot be used when an SMI is delivered via the local APIC.

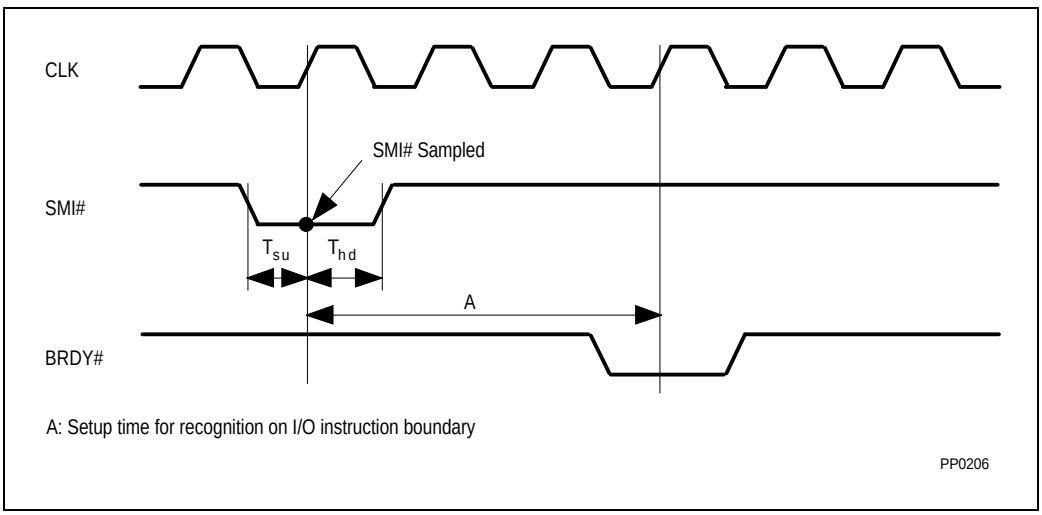


Figure 30-1. SMI# Timing for Servicing an I/O Trap

30.4. SMM — DUAL PROCESSING CONSIDERATIONS

Although the SMM functions the same when the Dual processor is inserted into Socket 5, the dual processor operation of the system must be carefully considered. The issues have to be addressed with the following: SMI# delivery, SMRAM and SMIACT#.

30.4.1. SMI# Delivery

Four options are possible depending on the SMM applications (mainly power management) the system has to support. There are implications to system design and the SMM handler.

Note that for operation with the Dual processor and upgradability with the Future Pentium OverDrive processor, Option #3 is strongly recommended.

	SMI# Pins Tied Together	SMI# Pins NOT Tied Together
SMI# pins delivering SMI	Option #1 Both CPUs enter SMM.	Option #2 One CPU enters SMM.
APIC delivering SMI	**Option #3** **One or Both CPUs enter SMM.**	Option #4 One or Both CPUs enter SMM

NOTE:

The I/O Instruction Restart Power Management feature should not be used when delivering the system management interrupt via the local APIC. Refer to Chapter 20 of the *Pentium® Processor Family Developer's Manual, Volume 3* for additional details on I/O instruction restart.

Implications

1. SMI# pin delivery of SMI and SMI# pins are tied together: Any assertion of the SMI# pin will cause both the Primary and Dual processors to interrupt normal processing, enter SMM mode and start executing SMM code in their respective SMRAM spaces. In this case, using the I/O Instruction restart feature in DP mode will require additional system hardware (D/P# pin) and software (detection of which processor was the MRM when the SMI# pin was asserted) considerations. This option will work for the Future Pentium OverDrive processor.
2. SMI# pin delivery of SMI and SMI# pins are NOT tied together: Only the processor whose SMI# pin is asserted will handle SMM processing. It is possible that both the Primary and Dual processor will be doing SMM processing at the same time, especially if the I/O Instruction restart feature is being used. If I/O instruction restart is not supported, then it is possible to dedicate only one processor for SMM handling at any time. This option is not recommended for Future Pentium OverDrive processor compatibility.
3. APIC SMI# delivery of SMI and SMI# pins are tied together: This option is strongly recommended for operation with the Dual processor and upgradability with the Future Pentium OverDrive processor. System Management Interrupts should be delivered via the APIC for DP systems, and may be delivered either via the APIC or the SMI# pin for turbo-upgraded systems. Either the Primary or Dual processor can be the assigned target for SMI# delivery and hence SMM handling. The SMM I/O instruction restart feature may be used in a uniprocessor system or in a system with a Future Pentium OverDrive processor (with SMI# pin delivery of the interrupt), but the system must not use this feature when operating in dual processing mode (with APIC delivery of the interrupt).
4. APIC SMI# delivery of SMI and SMI# pins are NOT tied together: I/O Instruction Restart feature is not recommended when delivering SMI via the local APIC. Either the Primary or Dual processor can be the assigned target for SMI# delivery and hence SMM handling. This option is not recommended for Future Pentium OverDrive processor compatibility.

30.4.2. SMIACT#

When the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) is the only CPU present, then it always drives the D/P# signal low. SMIACT# is asserted when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) enters SMM and is de-asserted only when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) exits SMM.

When the Dual processor is also present, the D/P# signal toggles depending upon whether the Primary or Dual processor owns the bus (MRM). The SMIACT# pins may be tied together or be used separately to insure SMRAM access by the correct processor.

CAUTION

If SMIACT# is used separately: the SMIACT# signal is only driven by the Primary or Dual processor when it is the MRM, so this signal must be qualified with the D/P# signal.

In a dual socket system, connecting the SMIACT# signals together on the Primary and Dual processor sockets is strongly recommended for both dual processing operation and upgradability with the Future Pentium OverDrive processor.

In dual processing systems, SMIACT# may not remain low (e.g., may toggle) if both processors are not in SMM mode. The SMIACT# signal is asserted by either the Primary or Dual processor based on two conditions: the processor is in SMM mode and is the bus master (MRM). If one processor is executing in normal address space, the SMIACT# signal will go inactive when that processor is MRM. The LRM processor, even if in SMM mode, will not drive the SMIACT# signal low.

30.4.3. Cache Flushes

Cache flushing during SMM exit is not possible while both the Primary and Dual processors are present due to the fact that it is not possible to clearly predict when the CPU in SMM has exited. This is because the SMIACT# is not a static status indicator but only a bus cycle indicator for SMRAM accesses.

30.5. CLOCK CONTROL

30.5.1. Clock Generation

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), although running internally at frequencies up to 2/3 the bus speed, allows its interface to the CPU bus to be at multiple system frequencies. The 1/2 or the 2/3 bus/core frequency clock allows simpler system design by lowering the clock speeds required in the external system. The 1/2 or 2/3 clock relies on an internal Phase Lock Loop (PLL) to generate the two internal clock phases, "phase one" and "phase two". Most external timing parameters are specified with respect to

the rising edge of CLK. The PLL requires a constant frequency CLK input, and therefore the CLK input cannot be changed dynamically.

On the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), CLK provides the fundamental timing reference for the bus interface unit. The internal clock converter enhances all operations functioning out of the internal cache and/or not blocked by external bus accesses. This mode uses a Phase Lock Loop (PLL) and therefore the CLK input must be maintained at a constant frequency.

30.5.2. Stop Clock

The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) provides an interrupt mechanism, STPCLK#, that allows system hardware to control the power consumption of the CPU by stopping the internal clock (output of the PLL) to the CPU core in a controlled manner. This low-power state is called the Stop Grant state. The target for low-power mode supply current in the Stop Grant state is ~15% of normal I_{cc}.

When the CPU recognizes a STPCLK# interrupt, the processor will stop execution on the next instruction boundary (unless superseded by a higher priority interrupt), stop the pre-fetch unit, complete all outstanding writes, generate a Stop Grant bus cycle, and then stop the internal clock. At this point, the CPU is in the Stop Grant state.

NOTE

If STPCLK# is asserted during RESET and continues to be held active after RESET is deasserted, the processor will execute one instruction before the STPCLK# interrupt is recognized. Execution of instructions will therefore stop on the second instruction boundary after the falling edge of RESET.

The CPU cannot respond to a STPCLK# request from a HLDA state because it cannot generate a Stop Grant cycle.

The rising edge of STPCLK# will tell the CPU that it can return to program execution at the instruction following the interrupted instruction.

Unlike the normal interrupts, INTR and NMI, the STPCLK# interrupt does not initiate interrupt table reads. Among external interrupts, STPCLK# is the lowest priority.

30.5.2.1. DUAL PROCESSING CONSIDERATIONS

The Primary and Dual processors may or may not tie their STPCLK# signals together. The decision is dependent on system specific CPU power conservation needs. Connecting the STPCLK# signals on the Primary and Dual processors together is strongly recommended for operation with the Dual processor and upgradability with the Future Pentium OverDrive processor.

Tying the STPCLK# signals together causes both the Primary and Dual processors to eventually enter the Stop Grant state on assertion of STPCLK#. The system ceases

processing until the STPCLK# signal is deasserted. In DP mode with the STPCLK# pins tied together, independent STPCLK# control of each processor is not possible. Both the Primary processor and Dual processor will go into the Stop Grant state independently, and will each generate a Stop Grant special bus cycle.

NOTE

In a dual processing system where STPCLK# is tied to both the primary and dual processors, the system expects to see two Stop Grant Bus Cycles after STPCLK# is asserted. FLUSH# should not be asserted between the time STPCLK# is asserted and the completion of the second Stop Grant Bus Cycle. If FLUSH# is asserted during this interval, the system may not see the second Stop Grant Bus Cycle until after STPCLK# is deasserted.

Not tying the STPCLK# signals together gives the flexibility to control either or both the processors' power consumption based on the system performance required. External logic would be required to control this signal to each processor in a DP system. In order to be upgradable with the Future Pentium OverDrive processor, system-level logic would be required (and be end-user invisible) to allow the STPCLK# signal to operate properly with both the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) (in a non-upgraded system) and with the Future Pentium OverDrive processor (in an upgraded system).

30.5.2.2. STPCLK# PIN

STPCLK# is treated as a level triggered interrupt to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). This interrupt may be asserted asynchronously and is prioritized below all of the external interrupts. If asserted, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will recognize STPCLK# on the next instruction boundary, and then do the following:

1. Flush the instruction pipeline of any instructions waiting to be executed.
2. Wait for all pending bus cycles to complete and EWBE# to go active.
3. Drive a special bus cycle (Stop Grant bus cycle) to indicate that the clock is being stopped.
4. Enter low power mode.

STPCLK# is active LOW. To ensure STPCLK# recognition, the system must keep this signal active until the appropriate special cycle has been issued by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). If STPCLK# does not remain active until the Stop Grant bus cycle has been issued, it may not be recognized. In the event that the first assertion of STPCLK# is recognized, a subsequent reassertion of STPCLK# before the completion of the first Stop Grant bus cycle may not generate a second Stop Grant bus cycle. Though STPCLK# is asynchronous, setup and hold times (Refer to Chapter 23) may be met to ensure recognition on a specific clock.

The STPCLK# input must be driven high (not floated) in order to exit the Stop Grant state. Once STPCLK# is deasserted and the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) resumes execution, the Pentium processor (610\75, 735\90, 815\100,

1000\120, 1110\133) is guaranteed to execute at least one instruction before STPCLK# is recognized again. To return to normal state, external hardware must deassert STPCLK#.

30.5.3. Stop Grant Bus Cycle

A special Stop Grant bus cycle will be driven to the bus after the CPU recognizes the STPCLK# interrupt. The definition of this bus cycle is the same as the HALT cycle definition for the standard Intel486 microprocessor architecture, with the exception that the Stop Grant bus cycle drives the value 0000 0010H on the address pins. In a DP system, with both STPCLK# signals tied together, two stop grant cycles will occur in a row. The system hardware must acknowledge the Stop Grant cycle by returning BRDY#. The CPU will not enter the Stop Grant state until BRDY# has been returned.

The Stop Grant Bus cycle consists of the following signal states: M/IO# = 0, D/C# = 0, W/R# = 1, Address Bus = 0000 0010H (A4 = 1), BE7#-BE0# = 1111 1011, Data bus = undefined.

NOTE

When operating in dual processing mode, and the STPCLK# signals are tied together, both the Primary processor and Dual processor will go into the Stop Grant state independently, and will each generate a Stop Grant special bus cycle. The system must return BRDY# for both of the special bus cycles.

The latency between a STPCLK# request and the Stop Grant bus cycle is dependent on the current instruction, the amount of data in the CPU write buffers, and the system memory performance. Refer to Figure 30-2.

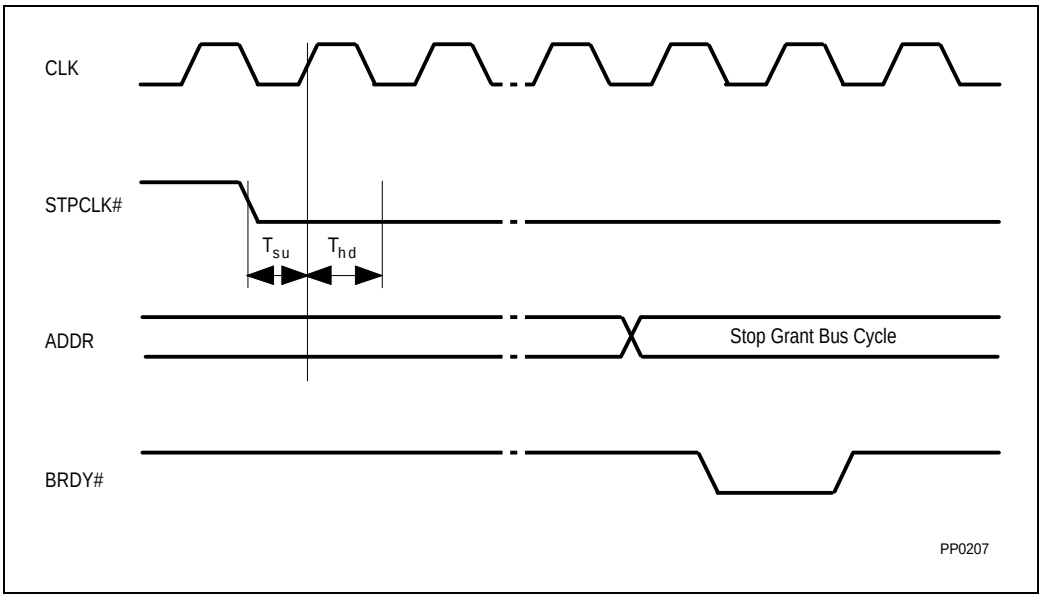


Figure 30-2. Entering Stop Grant State

30.5.4. Pin State during Stop Grant

During the Stop Grant state, most output and input/output signals of the microprocessor will be held at their previous states (the level they held when entering the Stop Grant state). However, the data bus and data parity pins will be floated. In response to HOLD being driven active during the Stop Grant state (when the CLK input is running), the CPU will generate HLDA and tri-state all output and input/output signals that are tri-stated during the HOLD/HLDA state. After HOLD is de-asserted, all signals will return to their states prior to the HOLD/HLDA sequence.

Table 30-1. Pin State During Stop Grant Bus State

Signal	Type	State
A31-A3	I/O	Previous State
D63-D0	I/O	Floated
BE7# - BE0#	O	Previous State
DP7 - DP0	I/O	Floated
W/R#, D/C#, M/IO#	O	Previous State
ADS#, ADSC#	O	Inactive
LOCK#	O	Inactive
BREQ	O	Previous State
HLDA	O	As per HOLD
FERR#	O	Previous State
PCHK#	O	Previous State
PWT, PCD	O	Previous State
SMIACT#	O	Previous State

In order to achieve the lowest possible power consumption during the Stop Grant state, the system designer must ensure the input signals with pull-up resistors are not driven LOW and the input signals with pull-down resistors are not driven HIGH. (Refer to Chapter 17, tables 17-3 to 17-6 in this document for signals with internal pull-up and pull-down resistors).

All inputs, except data bus pins, must be driven to the power supply rails to ensure the lowest possible current consumption during Stop Grant or Stop Clock modes. Data pins should be driven low to achieve the lowest power consumption. Pull down resistors or bus keepers are needed to minimize the leakage current.

30.5.4.1. CLOCK CONTROL STATE DIAGRAM

The following state descriptions and diagram show the state transitions for the clock control architecture.

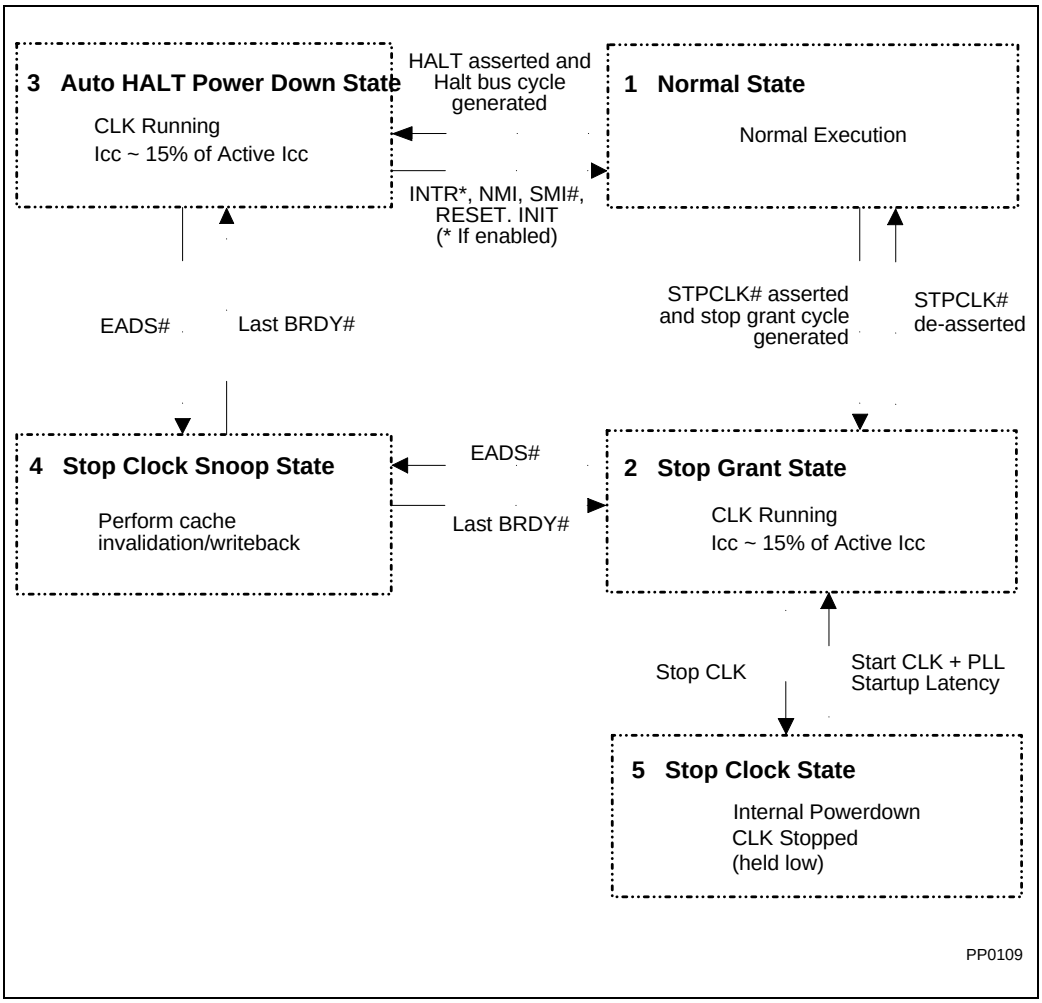


Figure 30-3. Stop Clock State Machine

A Flush State can be entered from states 1, 2 and 3 by asserting the FLUSH# input signal. The flush state is exited (e.g., the CPU returns to the state from which it came) when the Flush Acknowlege Special Bus Cycle is issued by the CPU.

A Probe Mode State can be entered from states 1, 2 and 3 by asserting the R/S# input signal. The Probe Mode State is exited (e.g., the CPU returns to the state from which it came) when the R/S# signal is deasserted by the system.

30.5.4.2. NORMAL STATE — STATE 1

This is the normal operating state of the CPU.

30.5.4.3. STOP GRANT STATE — STATE 2

The Stop Grant state (~15% of normal state I_{cc}) provides a fast wake-up state that can be entered by simply asserting the external STPCLK# interrupt pin. Once the Stop Grant bus cycle has been placed on the bus, and BRDY# is returned, the CPU is in this state. The CPU returns to the normal execution state in approximately 10 clock periods after STPCLK# has been deasserted.

For minimum CPU power consumption, all other input pins should be driven to their inactive level while the CPU is in the Stop Grant state. A RESET will bring the CPU from the Stop Grant state to the normal state (note: unless STPCLK# is also deasserted, an active RESET will only bring the CPU out of the Stop Grant state for a few cycles). The CPU will recognize the inputs required for maintaining cache coherency (e.g., HOLD, AHOLD, BOFF#, and EADS# for cache invalidations and snoops) as explained later in this section. The CPU will not recognize any other inputs while in the Stop Grant state. Input signals to the CPU will not be recognized until 1 CLK after STPCLK# is de-asserted.

While in the Stop Grant state, the CPU will latch transitions on the external interrupt signals (SMI#, NMI, INTR, FLUSH#, R/S#, and INIT). All of these interrupts are taken after the deassertion of STPCLK# (e.g. upon re-entering the normal state). The Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) requires INTR to be held active until the CPU issues an interrupt acknowledge cycle in order to guarantee recognition.

The CPU will generate a Stop Grant bus cycle only when entering that state from the normal state. When the CPU enters the Stop Grant state from the Stop Clock Snoop state, the CPU will not generate a Stop Grant bus cycle.

30.5.4.4. AUTO HALT POWERDOWN STATE — STATE 3

The execution of a HALT instruction will also cause the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) to automatically enter the Auto HALT Power Down state where Icc will be ~15% of I_{cc} in the Normal state. The CPU will issue a normal HALT bus cycle when entering this state. The CPU will transition to the normal state upon the occurrence of INTR, NMI, SMI#, RESET, or INIT.

A FLUSH# event during the Auto HALT power down state will be latched and acted upon while in this state.

STPCLK# is not recognized by the CPU while in the Auto HALT Powerdown state. The system can generate a STPCLK# while the CPU is in the Auto HALT Powerdown state, but the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) will only service this interrupt if the STPCLK# pin is still asserted when the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) returns to the normal state.

While in Auto HALT Powerdown state, the CPU will only recognize the inputs required for maintaining cache coherency (e.g., HOLD, AHOLD, BOFF#, and EADS# for cache invalidations and snoops) as explained later in this section.

30.5.4.5. STOP CLOCK SNOOP STATE (CACHE INVALIDATIONS) — STATE 4

When the CPU is in the Stop Grant state or the Auto HALT Powerdown state, the CPU will recognize HOLD, AHOLD, BOFF# and EADS# for cache invalidation/writebacks. When the system asserts HOLD, AHOLD, or BOFF#, the CPU will float the bus accordingly. When the system then asserts EADS#, the CPU will transparently enter the Stop Clock Snoop state and perform the required cache snoop cycle. It will then re-freeze the clock to the CPU core and return to the previous state. The CPU does not generate the Stop Grant bus cycle or HALT special cycle when it returns to the previous state.

30.5.4.6. STOP CLOCK STATE — STATE 5

Stop Clock state is entered from the Stop Grant state by stopping the CLK input. Note: for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133), the CLK must be held at a logic low while stopped. This is different than the Intel486 family of processors which allow the CLK to be held at either logic high or logic low while stopped. None of the CPU input signals should change state while the CLK input is stopped. Any transition on an input signal (with the exception of INTR) before the CPU has returned to the Stop Grant state will result in unpredictable behavior. If INTR is driven active while the CLK input is stopped, and held active until the CPU issues an interrupt acknowledge bus cycle, it will be serviced in the normal manner once the clock has been restarted. The system design must ensure the CPU is in the correct state prior to asserting cache invalidation or interrupt signals to the CPU.

While the processor is in Stop Clock state, all pins with static pullups or pulldowns must be driven to their appropriate values as specified in Chapter 17, tables 17-3 to 17-6.

If, while in the Stop Grant state, the CPU clock input frequency is changed, the CPU will not return to that state until the CLK input has been running at a constant frequency for the time period necessary for the PLL to stabilize. This constant frequency must be within the specified operating frequency range of the CPU. When the CLK input is stopped, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) requires the CLK input to be held at a constant frequency for a minimum of 1 ms before deasserting STPCLK#. This 1-ms time period is necessary so that the PLL can stabilize, and it must be met before the CPU will return to the Stop Grant state. The CLK input can be restarted to any frequency between the minimum and maximum frequency listed in the A.C. timing specifications.

In order to realize the maximum power reduction while in the Stop Clock state, PICCLK and TCK should also be stopped. These clock inputs have the same restarting restrictions as CLK. The local APIC cannot be used while in the Stop Clock state since it also uses the system clock, CLK.

WARNING

The Stop Clock state feature cannot be used in dual processing mode because there is no way to re-synchronize the internal clocks of the two processors.

31

Pentium® Processor (610\75, 735\90, 815\100, 1000\120, 1110\133) Debugging

CHAPTER 31
PENTIUM® PROCESSOR (610\75, 735\90, 815\100, 1000\120, 1110\133) DEBUGGING

31.1. INTRODUCTION

Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system designers intending to use integration tools to debug their prototype systems can interface to the CPU debug hooks using two methods: (1) insert an emulator probe into the CPU socket, or (2) include some simple logic on their board that implements a debug port connection. Inserting an emulator probe into the CPU socket will allow access to all bus signals, but capacitive loading issues may affect high speed operations. In contrast, the debug port connection will allow a debugger access to thePentium processor's (610\75, 735\90, 815\100, 1000\120, 1110\133) probe mode interface and a few other signals without affecting any high speed CPU signals. This will ensure that the system can operate at full speed with the debugger attached. Intel recommends that all Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133)-based system designs include a debug port .

31.2. TWO LEVELS OF SUPPORT

Two levels of support are defined for the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) debug port , the second level being a superset of first. The system designer should choose the level of support that is appropriate for the particular system design and implement that level. Samples of each level of implementation are given in section 31.6 of this document.

31.2.1. Level 1 Debug Port (L1)

The Level 1 debug port supports systems with a single Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) CPU. L1 is equivalent to the debug port described for the Pentium processor (510\60, 567\66) and uses a 20-pin connector to allow a debugger to access the CPU's boundary scan and probe mode interface.

31.2.2. Level 2 Debug Port (L2)

L2 extends the 20-pin debug port connector to 30 pins. The extra ten pins include a second set of boundary scan signals as well as additional R/S# and PRDY signals. The additional R/S# and PRDY signals are added to support the Pentium processor (610\75, 735\90,

815\100, 1000\120, 1110\133) in the dual-processor configuration. This enables a debugger to provide separate control over the two CPUs during debug.

Signals on pins 1 through 20 of the L2 debug port are identical to the signals on the L1 debug port.

31.3. DEBUG PORT CONNECTOR DESCRIPTIONS

A debugger for Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) designs can have a 30-pin connector on its probe that supports both levels of the debug port (as described previously, L1 or L2). Two cables can be provided, each cable having a 30-pin connector at one end (to mate with the debugger's probe connector) and the appropriate size connector at the other end to mate with the debug port in the system under debug. (For example, the L1 debug port Cable can be a 20-conductor cable with a 20-pin connector at one end and a 30-pin connector at the other end, leaving pins 21 to 30 unconnected.)

Intel recommended connectors to mate with debug port Cables are available in either a vertical or right-angle configuration. Use the one that fits best in your design. The connectors are manufactured by AMP Incorporated and are in their AMPMODU System 50 line. Following are the AMP part numbers for the various connectors:

	Vertical	Right-Angle
20-pin shrouded header	104068-1	104069-1
30-pin shrouded header	104068-3	104069-5

NOTE:

These are high density through hole connectors with pins on 0.050" by 0.100" centers. Do not confuse these with the more common 0.100" by 0.100" center headers.

The following is an example of the pinout of the connector footprint as viewed from the connector side of the circuit board. This is just an example. Contact your tools representative to determine the correct implementation for the tool you will use. Note that the 30-pin connector is a logical extension of the 20-pin connector with the key aligned with pin 15.

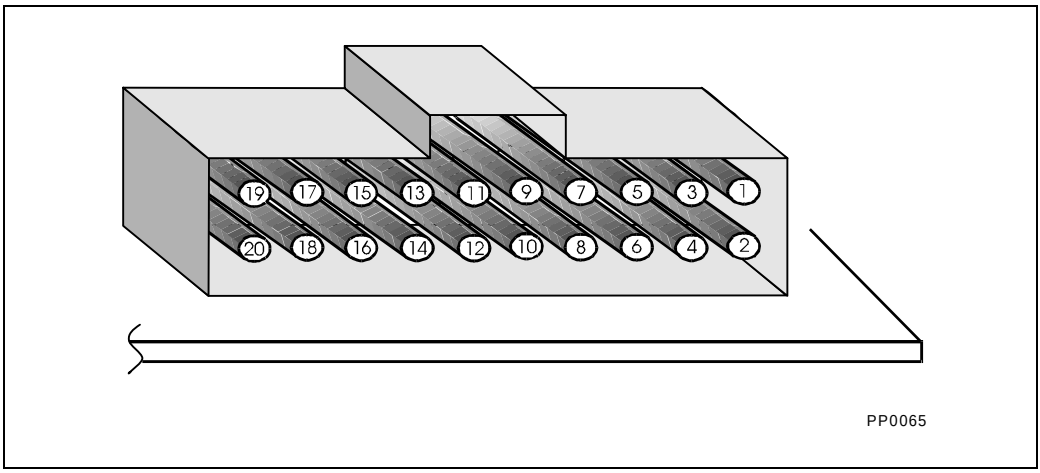


Figure 31-1. Debug Port Connector

31.4. SIGNAL DESCRIPTIONS

Following are the debug port signals. Direction is given as follows: O = output from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) board to a debugger; I = input to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) board from a debugger. These are 3.3 V signals, compatible with the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) DC specifications. For the L1 debug port , ignore signals on pins 21 through 30.

NOTE

Target systems should be sure to provide a way for debugging tools like emulators, in-target probes and logic analyzers to reset the entire system, including uprgrade processor, chip sets, etc. For example, if you follow the debug port implementation described below, the DBRESET signal provides this functionality. If you are not implementing the debug port , make sure that your system has a test point connected into the system reset logic to which a debug tool can connect.

Table 31-1. Debug Port Signals

Signal Name	Dir	Pin	Description
INIT	O	1	(Pentium® processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal). A debugger may use INIT to support emulating through the CPU INIT sequence while maintaining breakpoints or breaking on INIT.
DBRESET	I	2	Debugger Reset output. A debugger may assert DBRESET (high) while performing the "RESET ALL" and "RESET TARGET" debugger commands. DBRESET should be connected to the system reset circuitry such that the system and processor(s) are reset when DBRESET is asserted. This is useful for recovering from conditions like a "ready hang". This signal is asynchronous.
RESET	O	3	(Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal). A debugger may use RESET to support emulating through the reset while maintaining breaking on RESET.
GND		4	Signal ground.
NC		5	No connect. Leave this pin unconnected.
V_{CC}		6	V_{CC} from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system. A debugger uses this signal to sense that system power is on. Connect this signal to V_{CC} through a 1 K-ohm (or smaller) resistor.
R/S#	I	7	Connect to the R/S# pin of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).
GND		8	Signal ground.
NC		9	No connect. Leave this pin unconnected.
GND		10	Signal ground.
PRDY	O	11	From the PRDY pin of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).
TDI	I	12	Boundary scan data input (Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal). This signal connects to TDI of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133). For dual processor operation, TDI of the Dual Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) would connect to TDO of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133).
TDO	O	13	Boundary scan data output (Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal). This signal connects to TDO from the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) for a single processor design, or to TDO from the Dual Pentium Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) for dual processor operation.
TMS	I	14	Boundary scan mode select (Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal).
GND		15	Signal ground.
TCK	I	16	Boundary scan clock (Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal).
GND		17	Signal ground.

Table 31-1. Debug Port Signals (Contd.)

Signal Name	Dir	Pin	Description
TRST#	I	18	Boundary scan reset (Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) signal).
DBINST#	I	19	DBINST# is asserted (connected to GND) while the debugger is connected to the debug port . DBINST# can be used to control the isolation of signals while the debugger is installed.
BSEN#	I	20	Boundary scan enable. This signal can be used by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system to control multiplexing of the boundary scan input pins (TDI, TMS, TCK, and TRST# signals) between the debugger and other boundary scan circuitry in the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system. The debugger asserts (low) BSEN# when it is driving the boundary scan input pins. Otherwise, the debugger drivers are high impedance. If the boundary scan pins are actively driven by the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system, then BSEN# should control the system drivers/multiplexers on the boundary scan input pins. See example 2 in section 31.6
PRDY2	O	21	From the PRDY pin of the Dual Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) (for dual processor operation).
GND		22	Signal ground.
R/S#2	I	23	Connect to the R/S# pin of the Dual Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) (for dual processor operation).
NC		24	
NC		25	
NC		26	
NC		27	
NC		28	
GND		29	Signal ground.
NC		30	

31.5. SIGNAL QUALITY NOTES

Since debuggers can connect to the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system via cables of significant length (e.g., 18 inches), care must be taken in Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system design with regard to the signals going to the debug port . If system outputs to the debug port (i.e. TDO, PRDY, INIT and RESET) are used elsewhere in the system they should have dedicated drivers to the debug port . This will isolate them from the reflections from the end of the debugger cable. Series termination is recommended at the driver output. If the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) boundary scan signals are used elsewhere in the

system, then the TDI, TMS, TCK, and TRST# signals from the debug port should be isolated from the system signals with multiplexers.

31.6. IMPLEMENTATION EXAMPLES

31.6.1. Example 1: Single CPU, Boundary Scan Not Used by System

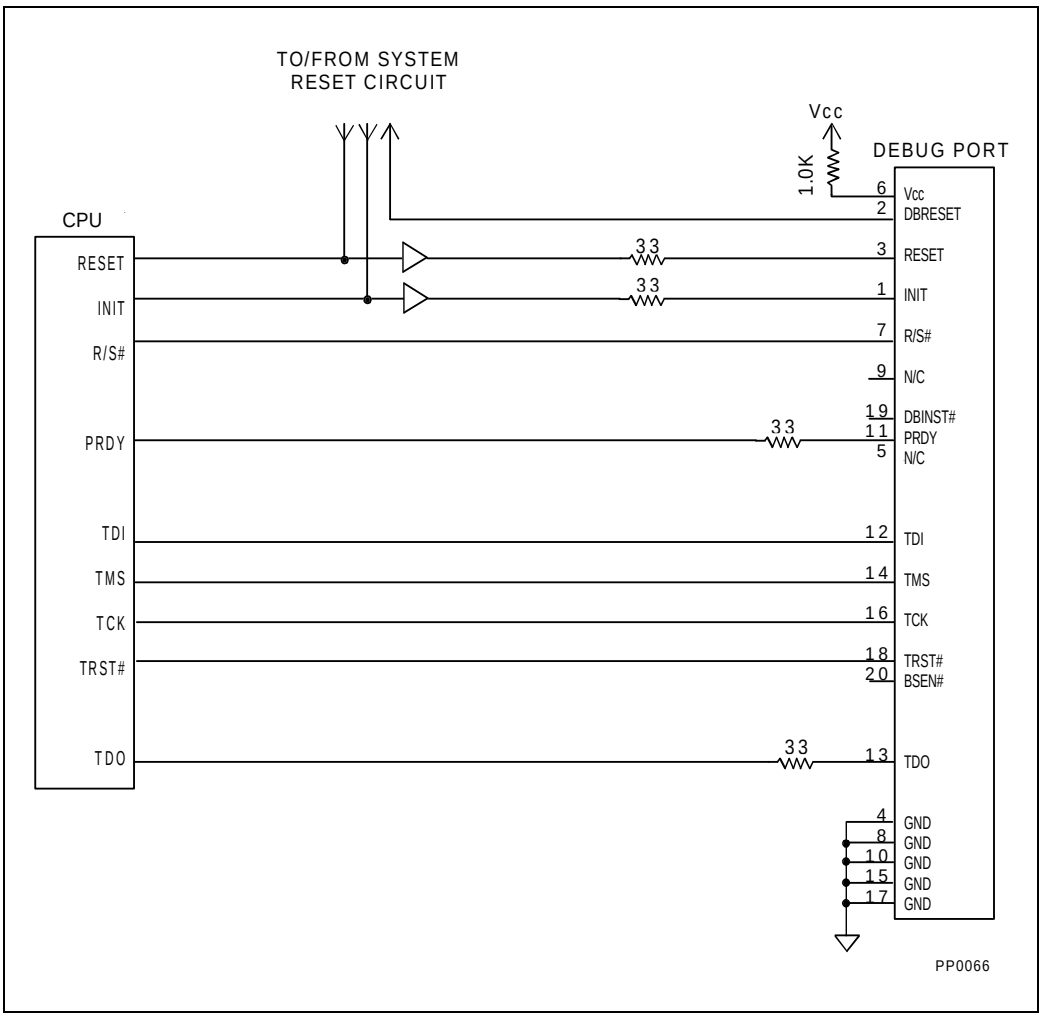


Figure 31-2. Single CPU — Boundary Scan Not Used

Figure 31-2 shows a schematic of a minimal Level 1 debug port implementation for a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) single processor system in which the boundary scan pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are not used in the system.

31.6.2. Example 2: Single CPU, Boundary Scan Used by System

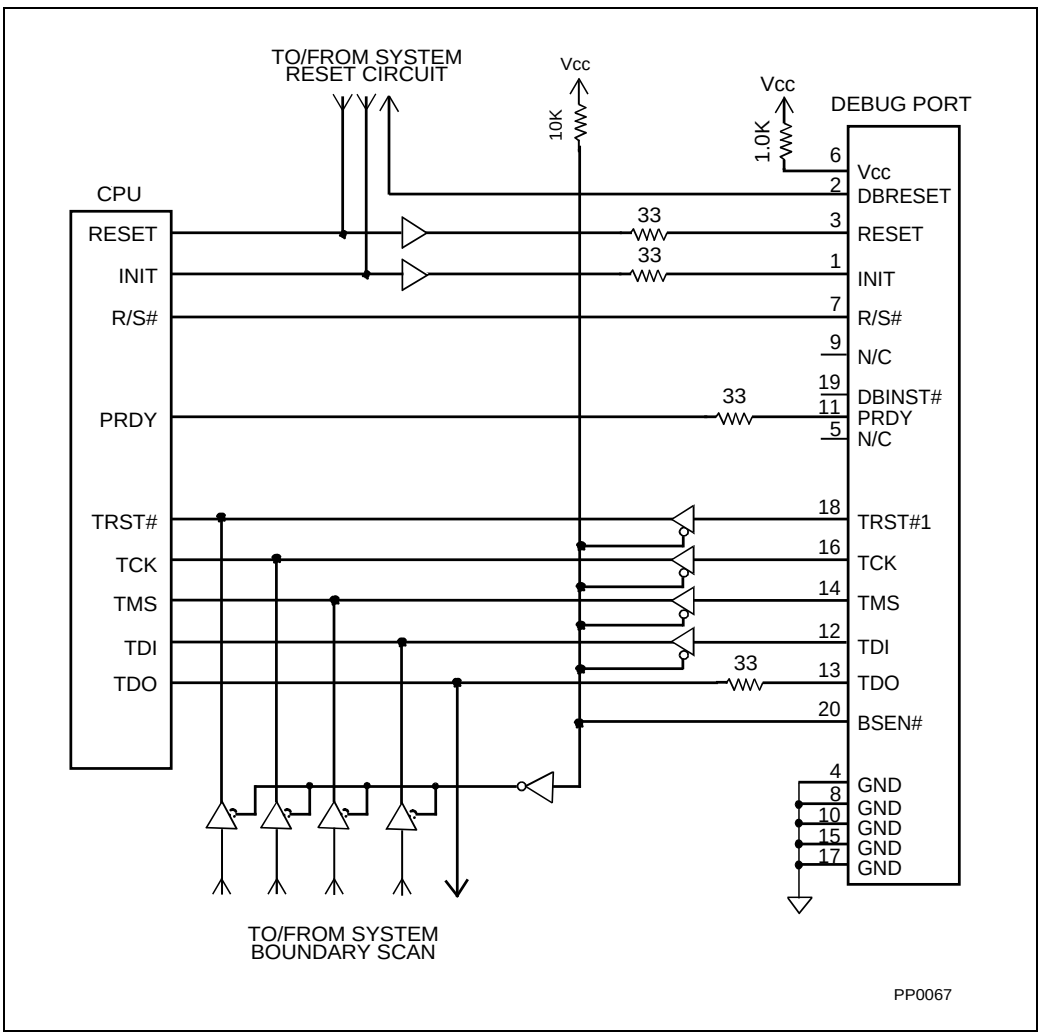


Figure 31-3. Single CPU — Boundary Scan Used

Figure 31-3 shows a schematic of a Level 1 debug port implementation for a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) single processor system in which

the boundary scan pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are used in the system. Note that the BSEN# signal controls the multiplexing of the boundary scan signals. With this implementation, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system could use the boundary scan (through the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133)) while the debugger is "emulating", but could not while the debugger is "halted" (because the chain is broken).

31.6.3. Example 3: Dual CPUs, Boundary Scan Not Used by System

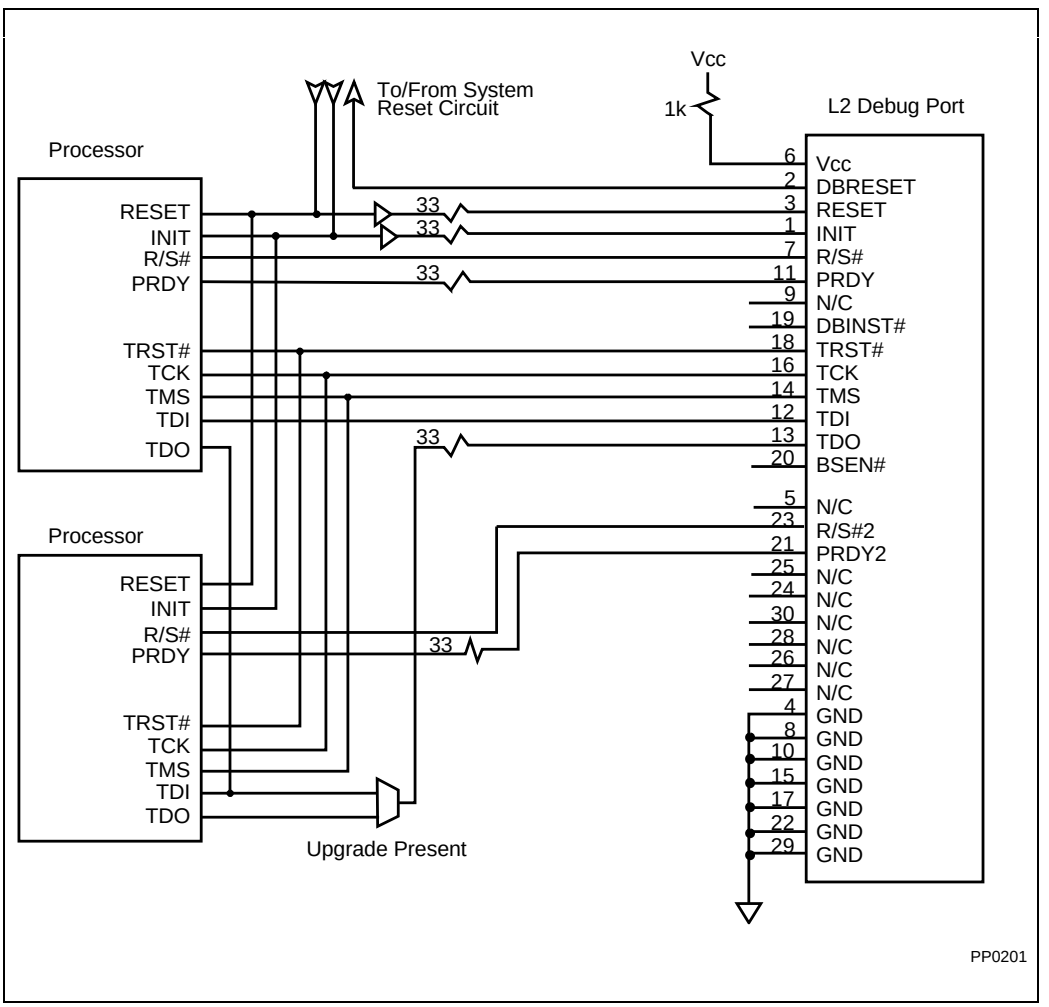


Figure 31-4. Dual CPUs — Boundary Scan Not Used

Figure 36-4 shows a schematic of a typical Level 2 debug port implementation for a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) dual processor system in which the boundary scan pins of the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) are not used in the system. The multiplexer circuit for use with the "upgrade socket" concept is shown, but could be replaced with a jumper.

31.6.4. Example 4: Dual CPUs, Boundary Scan Used by System

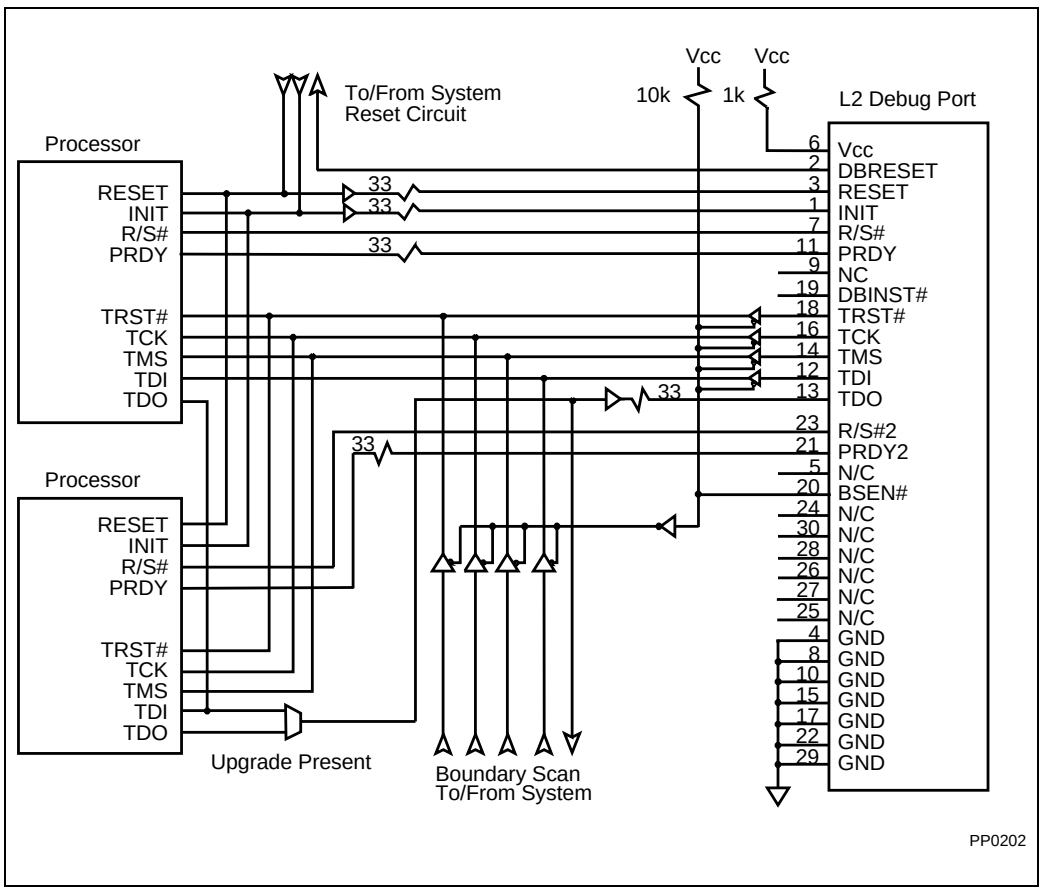


Figure 31-5. Dual CPUs — Boundary Scan Used

Figure 31-5 shows a schematic of a Level 2 debug port implementation for a Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) dual processor system that uses boundary scan. Note that the BSEN# signal controls the multiplexing of the boundary scan signals. With this implementation, the Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) system could use the boundary scan (through the Pentium processor

(610\75, 735\90, 815\100, 1000\120, 1110\133)) while the debugger is "emulating", but could not while the debugger is "halted" (because the chain is broken).

31.7. IMPLEMENTATION DETAILS

31.7.1. Signal Routing Note

The debugger software communicates with the CPU through the debug port using the boundary scan signals listed above. Typically, the debugger expects the CPU to be the first and only component in the scan chain (from the perspective of the debug port). That is, it expects TDI to go directly from the debug port to the TDI pin of the CPU, and the TDO pin to go directly from the CPU to the debug port (see Figure 31-6, below). If you have designed your system so that this is not the case (for instance, see Figure 31-7, below), you will need to provide the debugger software with the following information: (1) position of the CPU in the scan chain, (2) the length of the scan chain, (3) instruction register length of each device in the scan chain. Without this information the debugger will not be able to establish communication with the CPU.

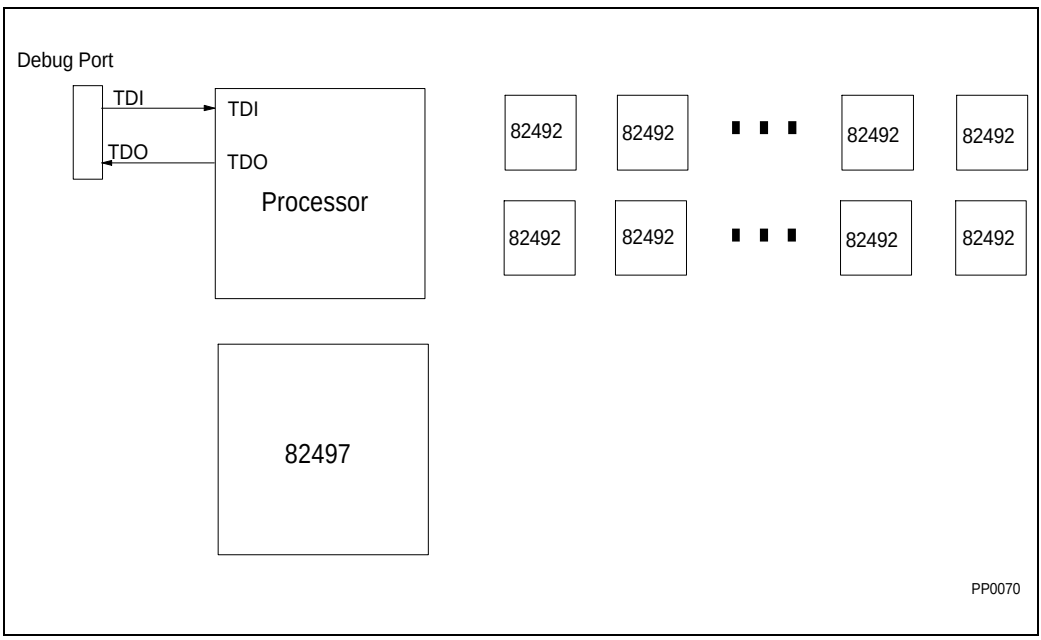


Figure 31-6. Example of CPU Only in Scan Chain

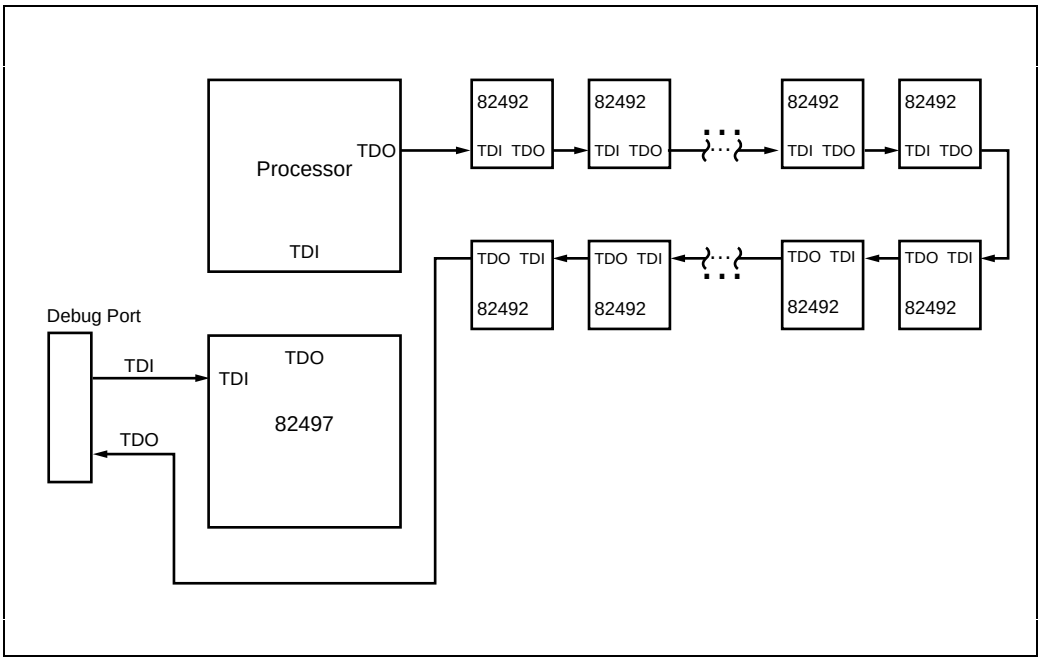


Figure 31-7. Example of Multiple Components in Scan Chain

31.7.2. Special Adapter Descriptions

For those designs where board real estate is a concern or where the design is finished and it is too late to implement the debug port, it may be possible to use a special "debug port adapter" to replace the on-board debug port described in the previous sections. The purpose of the adapter is to provide easy access to the boundary scan signals of the CPU(s). For simplicity, the adapter should make the boundary scan signals accessible to the debug tool while at the same time preventing the target system from accessing them. Two debug port adapters are described: (1) for uni-processor debug, (2) for dual processor debug.

31.7.2.1. UNI-PROCESSOR DEBUG

A debug port adapter for use in uni-processor systems, or dual processor systems where only one processor will be debugged at a time, can be built by reworking two Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) SPGA sockets as follows (see diagram below):

NOTE

This adapter can be used only when the CPU is NOT included in the target system boundary scan string. In addition, when used in dual processor systems you will only be able to debug the CPU to which the adapter is connected.

Connect lines of appropriate 20- or 30-wire cable to the following pins on the top socket:

Cable wire #	SPGA Pin#	Signal
1	AA33	INIT
2	NC	DRESET
3	AK20	RESET
4	AD36(V_{SS})	GND
5	NC	NC
6	U37	Vcc
7	AC35	R/S#
8	AB36(V_{SS})	GND
9	NC	NC
10	Z36(V_{SS})	GND
11	AC05	PRDY
12	N35	TDI
13	N33	TDO
14	P34	TMS
15	X36(V_{SS})	GND
16	M34	TCK
17	R36(V_{SS})	GND
18	Q33	TRST#
19	NC	DBINST#
20	NC	BSEN#

NOTE

You may connect the GND pins to any pin marked V_{SS} on the SPGA pinout diagram. The NC pins are no connects. You may simply cut those wires.

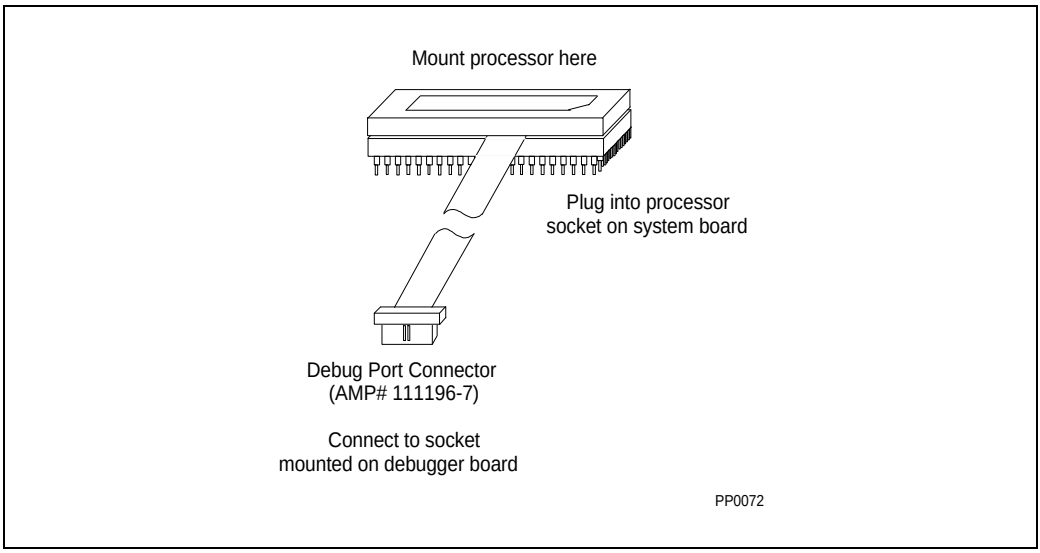


Figure 31-8. Uni-Processor Debug

Connect a double-row receptacle (AMP# 111196-7) to the debug port connector end of the cable. This is a 30-pin connector, so that it fits into the socket on the debugger buffer board.

Remove the following pins from the bottom socket:

R/S#	AC35
PRDY	AC05
TDI	N35
TDO	N33
TMS	P34
TCK	M34
TRST#	Q33

Connect the two sockets together. Make sure not to crush the wires between the pins.

31.7.2.2. DUAL PROCESSOR DEBUG

A debug port adapter for use in dual processor debugging can be built by reworking four Pentium processor (610\75, 735\90, 815\100, 1000\120, 1110\133) SPGA sockets. (See diagram below).

NOTE

This adapter can be used only when the CPUs are NOT included in the target system boundary scan string.

You will need to use two SPGA sockets per processor location. For this discussion, assume that the startup processor is called processor 1 and that the upgrade processor is called processor 2. Thus, you will use two SPGA sockets to connect to processor 1 and two SPGA sockets to connect to processor 2. Certain debug port signals must be shared by Processor 1 and Processor 2. These signals must be connected from the debug port connector end of the cable (on which you will place a double-row receptacle: AMP# 111196-7) to both double SPGA sockets.

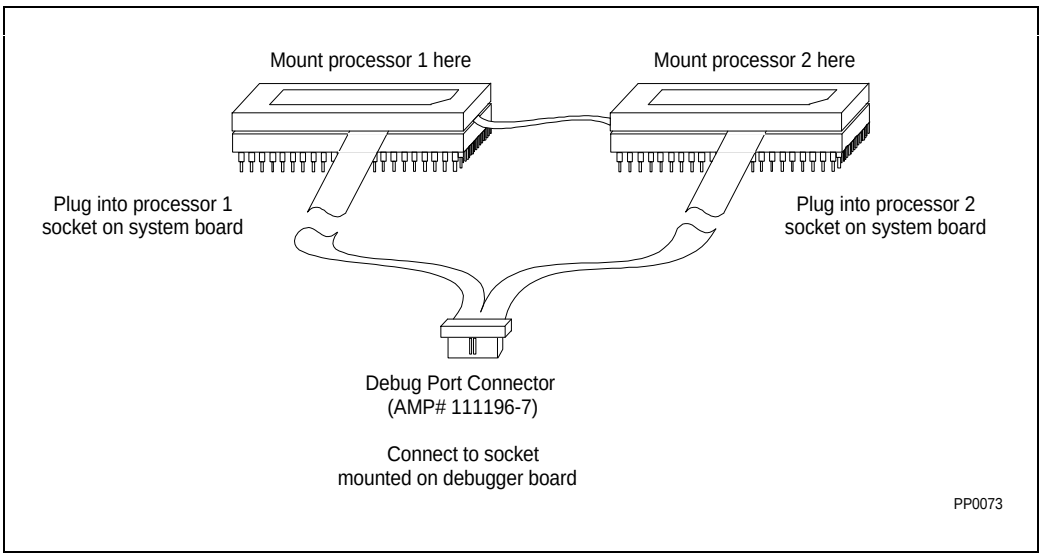


Figure 31-9. Dual Processor Debug Port Adapter

Connect lines of 30-wire cable to the pins on the top SPGA sockets for both processor 1 and 2. Following are the signals which should be connected to each processor socket. Make sure to connect the shared lines to both top sockets.

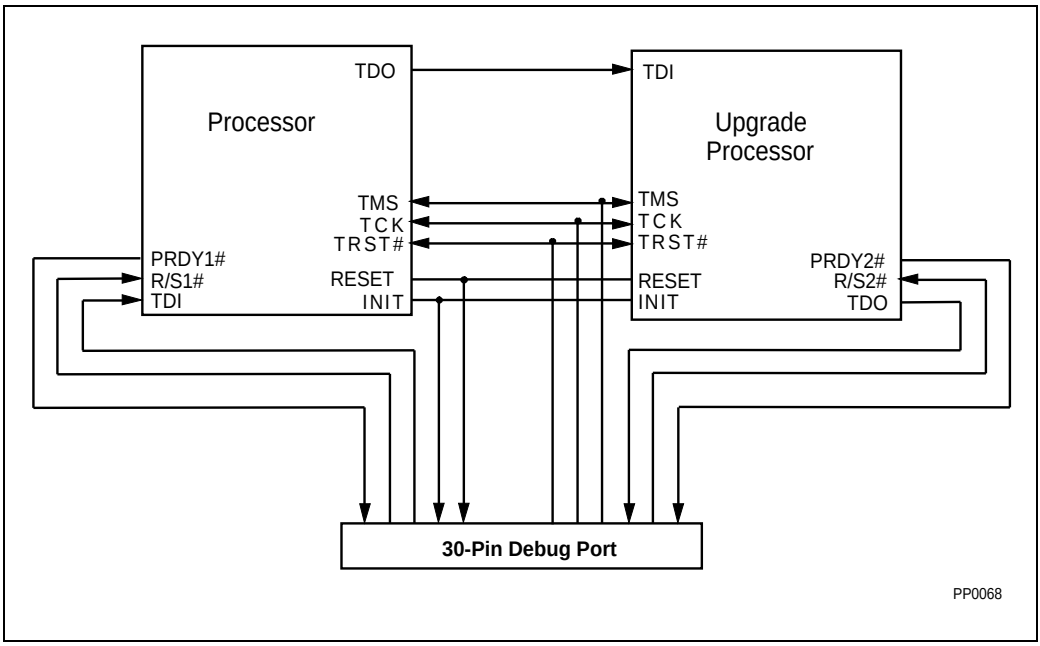


Figure 31-10. Shared Pins for Dual Processor Adapter

Table 31-2. Debug Port Connector Pinout

Cable wire #	SPGA Pin#	Processor Socket	Signal
1	AA33	1,2	INIT
2	NC		DBRESET
3	AK20	1,2	RESET
4	V_{SS}	1	GND
5	NC		NC
6	V_{CC}	1	V_{cc}
7	AC35	1	R/S1#
8	V_{SS}	1	GND
9	NC		NC
10	V_{SS}	1	GND
11	AC05	1	PRDY1
12	N35	1	TDI
13	N33	2	TDO
14	P34	1,2	TMS
15	V_{SS}	1	GND
16	M34	1,2	TCK
17	V_{SS}	1	GND
18	Q33	1,2	TRST#
19	NC		DBINST#
20	NC		BSEN#
21	AC05	2	PRDY2
22	V_{SS}	2	GND
23	AC35	2	R/S2#
24	NC		NC
25	NC		NC
26	NC		NC
27	NC		NC
28	NC		NC
29	V_{SS}	2	GND
30	NC		NC

NOTE: You may connect the V_{CC} and GND pins to any convenient power or ground pin.

Connect a double-row receptacle (AMP# 111196-7) to the debug port end of the cable. This is a 30-pin connector, so that it fits into the socket on the debugger buffer board.

Remove the following pins from the bottom of both double sockets:

R/S#	AC35
PRDY	AC05
TDI	N35
TDO	N33
TMS	P34
TCK	M34
TRST#	Q33

Connect each set of two sockets together. Make sure not to crush the wires between the pins.

32

Future OverDrive® Processor for Pentium® Processor (610\75, 735\90, 815\100, 1000\120)-Based Systems Socket Specification

CHAPTER 32
FUTURE OverDrive® PROCESSOR FOR PENTIUM® PROCESSOR (610\75, 735\90, 815\100, 1000\120, 1110\133)-BASED SYSTEMS SOCKET SPECIFICATION

32.1. INTRODUCTION

The future OverDrive processors are end-user single-chip CPU upgrade products for Pentium processor-based systems. The future OverDrive processors will speed up most software applications by 40% to 70% and are binary compatible with the Pentium processor.

Two upgrade sockets have been defined for the Pentium processor-based as part of the processor architecture. Socket 5 has been defined for the Pentium processor (610\75, 735\90, 815\100, 1000\120)-based systems. Upgradability can be supported by implementing either a single socket or a dual socket strategy. A single socket system will include a 320-pin SPGA Socket 5. When this system configuration is upgraded, the Pentium processor is simply replaced by the future OverDrive processor. A dual socket system will include a 296-pin SPGA socket for the Pentium processor and a 320-pin SPGA Socket 5 for the second processor. In dual socket systems, Socket 5 can be filled with either the Dual processor or the future OverDrive processors. The rest of this chapter describes the Socket 5 specifications.

Socket 7 has been defined as the upgrade socket for the Pentium processor (1110\133) in addition to the Pentium processor (610\75, 735\90, 815\100, 1000\120). The flexibility of the Socket 7 definition makes it backward compatible with Socket 5 and should be used for all new Pentium processor-based system designs. The Socket 7 support requires key changes from Socket 5 designs; split voltage planes, voltage regulator module header, 3.3 volt clocks, BIOS updates, additional decoupling, etc. This information is not provided in this databook. Contact Intel for further information regarding the Socket 7 specifications.

32.1.1. Upgrade Objectives

Systems using the Pentium processor (610\75, 735\90, 815\100, 1000\120), and equipped with only one processor socket, must use Socket 5 to accept the future OverDrive processor. Systems equipped with two processor sockets must use Socket 5 as the second socket to contain either the Pentium processor Dual processor or the future OverDrive processor.

Inclusion of Socket 5 in Pentium processor systems provides the end-user with an easy and cost-effective way to increase system performance. The majority of upgrade installations which take advantage of Socket 5 will be performed by end users and resellers. Therefore, it is important that the design be "end-user easy," and that the amount of training and technical

expertise required to install the upgrade processors be minimized. Upgrade installation instructions should be clearly described in the system user's manual. In addition, by making installation simple and foolproof, PC manufacturers can reduce the risk of system damage, warranty claims and service calls. Three main characteristics of end user easy designs are:

- Accessible socket location
- Clear indication of upgrade component orientation
- Minimization of insertion force

The future OverDrive processor will support all Intel chip sets that are supported by the Pentium processor, including 82430NX PCIset and 82430FX PCIset.

32.1.2. Intel Platform Support Labs

The Intel Platform Support Labs ensures that Pentium processor (610\75, 735\90, 815\100, 1000\120) and Pentium processor (1110\133)-based personal computers meet design criteria for reliable and straightforward CPU upgradability with the future OverDrive processor. Evaluation performed at the Intel Platforms Support Labs confirms that Pentium processor and future OverDrive processor specifications for mechanical, thermal, electrical, functional, and end-user installation attributes have been met.

The OEM submits motherboard and system designs to one of Intel's worldwide Platform Support Labs for evaluation. The OEM benefits from engineering feedback on Pentium processor and future OverDrive processor support. Contact your local Intel representative for more information on the Intel Platform Support Labs.

32.2. FUTURE OverDrive® PROCESSOR (SOCKET 5) PINOUT

This section contains pinouts of the future OverDrive processor socket (Socket 5) when used as a single-socket turbo upgrade.

32.2.1. Pin Diagrams

32.2.1.1. SOCKET 5 PINOUT

For systems with a single socket for the Pentium processor (610\75, 735\90, 815\100, 1000\120) and future OverDrive processor, the following pinout ***must*** be followed for the single socket location. The socket footprint contains V_{CC}, $V_{CC}5$, and V_{SS} pins that are internal no connects on the Pentium processor. These pins ***must*** be connected to the appropriate PCB power and ground layers to ensure future OverDrive processor compatibility.

OEMs should contact Intel for the most current list of Intel-qualified socket vendors.

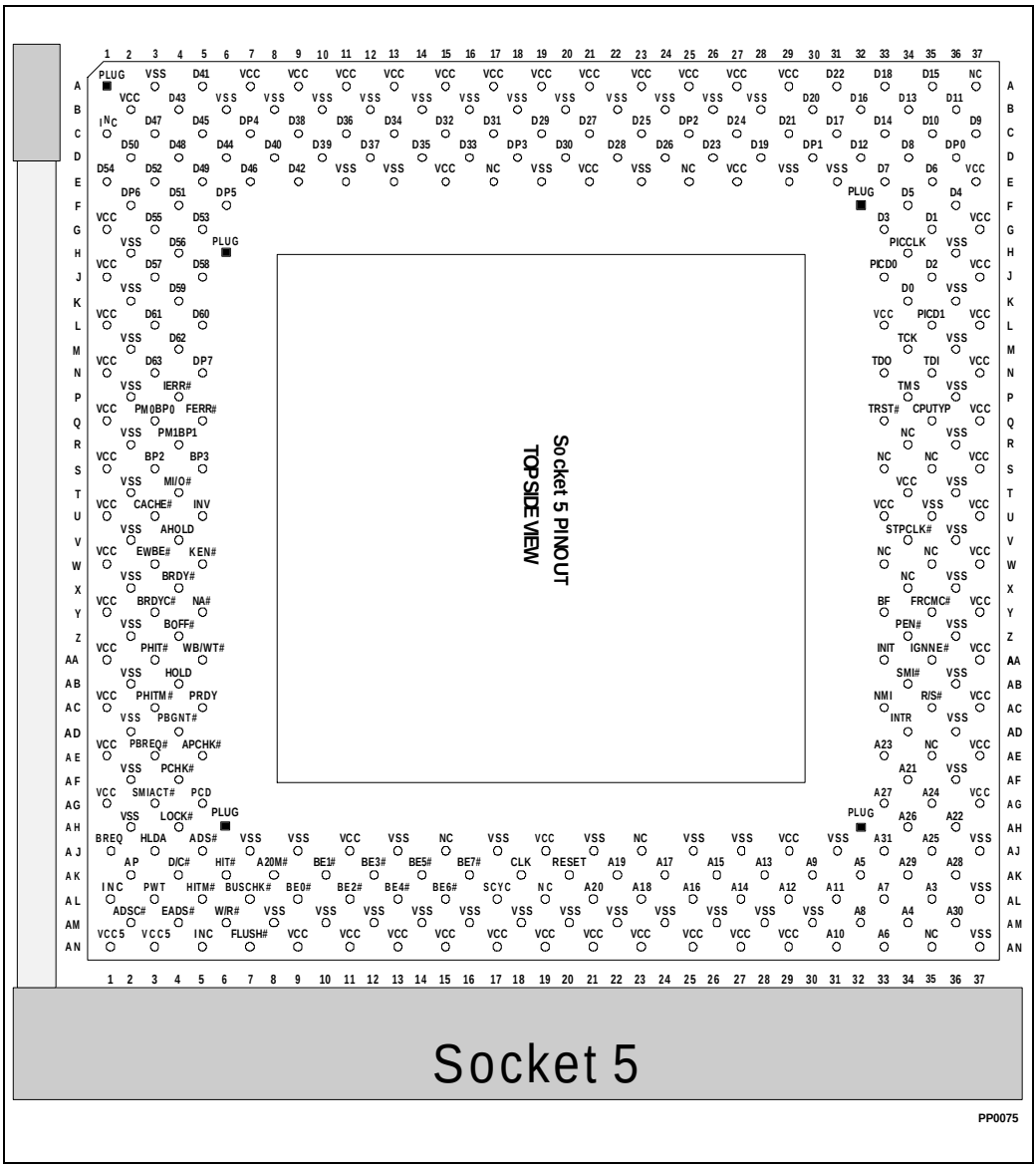


Figure 32-1. Socket 5 Pinout — Top Side View

NOTE:

The "Socket 5 PINOUT TOP SIDE VIEW" text orientation on the top side view drawing in this section represents the orientation of the ink mark on the actual packages. (Note that the text shown in this section is not the actual text which will be marked on the packages).

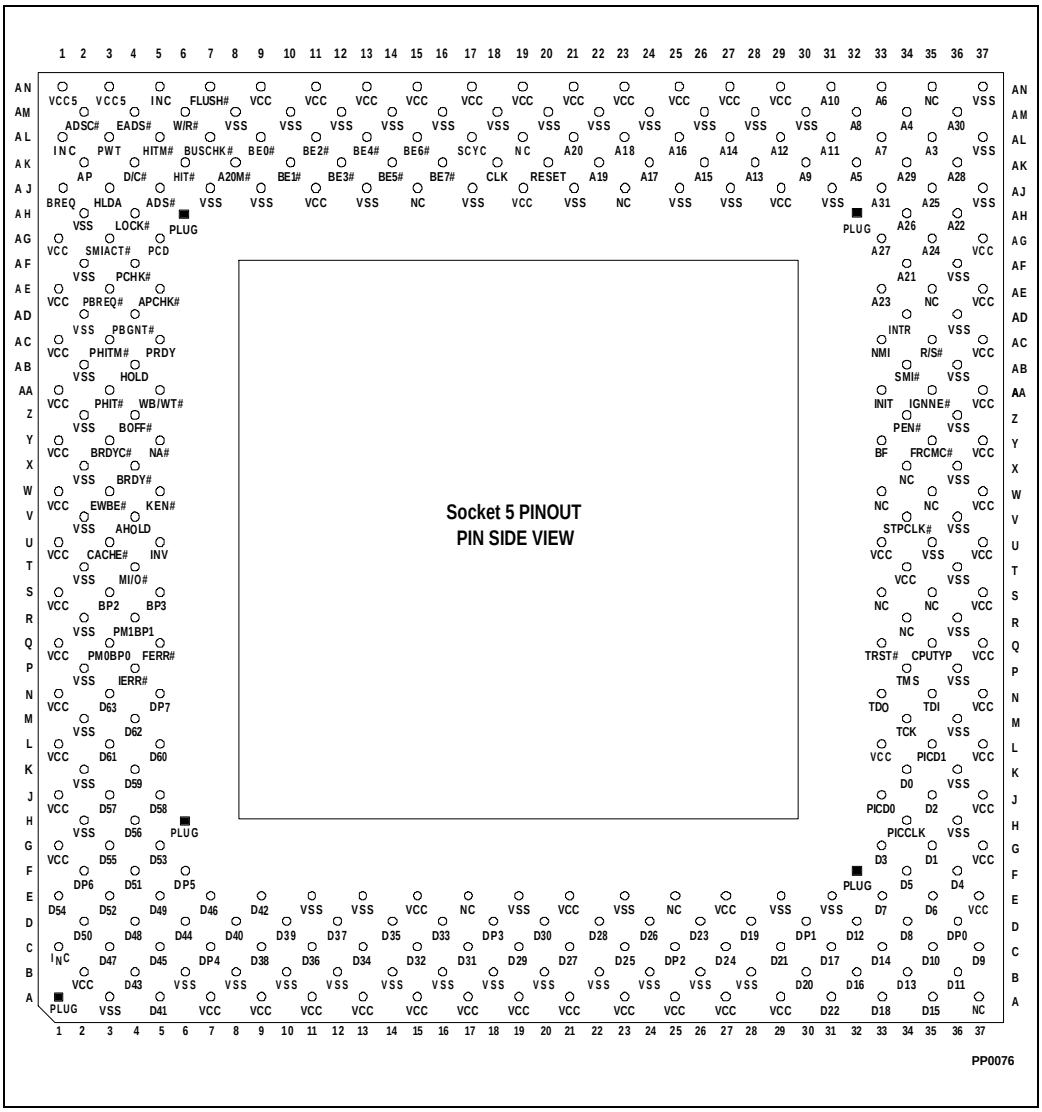


Figure 32-2. Socket 5 Pinout — Pin Side View

32.2.2. Socket 5 Pin Cross Reference Table

Note that the shaded signals in the following tables have different pin definitions for the Pentium processor (610\75, 735\90, 815\100, 1000\120) as compared to the future OverDrive processor (refer to Table 32-1 for details):

Table 32-1. Socket 5 Pin Cross Reference by Pin Name

Address									
A3	AL35	A9	AK30	A15	AK26	A21	AF34	A27	AG33
A4	AM34	A10	AN31	A16	AL25	A22	AH36	A28	AK36
A5	AK32	A11	AL31	A17	AK24	A23	AE33	A29	AK34
A6	AN33	A12	AL29	A18	AL23	A24	AG35	A30	AM36
A7	AL33	A13	AK28	A19	AK22	A25	AJ35	A31	AJ33
A8	AM32	A14	AL27	A20	AL21	A26	AH34		
Data									
D0	K34	D13	B34	D26	D24	D39	D10	D52	E03
D1	G35	D14	C33	D27	C21	D40	D08	D53	G05
D2	J35	D15	A35	D28	D22	D41	A05	D54	E01
D3	G33	D16	B32	D29	C19	D42	E09	D55	G03
D4	F36	D17	C31	D30	D20	D43	B04	D56	H04
D5	F34	D18	A33	D31	C17	D44	D06	D57	J03
D6	E35	D19	D28	D32	C15	D45	C05	D58	J05
D7	E33	D20	B30	D33	D16	D46	E07	D59	K04
D8	D34	D21	C29	D34	C13	D47	C03	D60	L05
D9	C37	D22	A31	D35	D14	D48	D04	D61	L03
D10	C35	D23	D26	D36	C11	D49	E05	D62	M04
D11	B36	D24	C27	D37	D12	D50	D02	D63	N03
D12	D32	D25	C23	D38	C09	D51	F04		

Table 32-1. Socket 5 Pin Cross Reference by Pin Name (Contd.)

Control							
A20M#	AK08	BRDYC#	Y03	FRCMC#	Y35	PM0/BP0	Q03
ADS#	AJ05	BREQ	AJ01	HIT#	AK06	PM1/BP1	R04
ADSC#	AM02	BUSCHK#	AL07	HITM#	AL05	PRDY	AC05
AHOLD	V04	CACHE#	U03	HLDA	AJ03	PWT	AL03
AP	AK02	CPUTYP	Q35	HOLD	AB04	R/S#	AC35
APCHK#	AE05	D/C#	AK04	IERR#	P04	RESET	AK20
BE0#	AL09	DP0	D36	IGNNE#	AA35	SCYC	AL17
BE1#	AK10	DP1	D30	INIT	AA33	SMI#	AB34
BE2#	AL11	DP2	C25	INTR/LINT0	AD34	SMIACT#	AG03
BE3#	AK12	DP3	D18	INV	U05	TCK	M34
BE4#	AL13	DP4	C07	KEN#	W05	TDI	N35
BE5#	AK14	DP5	F06	LOCK#	AH04	TDO	N33
BE6#	AL15	DP6	F02	M/IO#	T04	TMS	P34
BE7#	AK16	DP7	N05	NA#	Y05	TRST#	Q33
BOFF#	Z04	EADS#	AM04	NMI/LINT1	AC33	W/R#	AM06
BP2	S03	EWBE#	W03	PCD	AG05	WB/WT#	AA05
BP3	S05	FERR#	Q05	PCHK#	AF04		
BRDY#	X04	FLUSH#	AN07	PEN#	Z34		

APIC		Clock Control		Dual Processor Private Interface	
PICCLK	H34	CLK	AK18	PBGNT#	AD04
PICD0	J33	BF	Y33	PBREQ#	AE03
[DPEN#]		STPCLK#	V34	PHIT#	AA03
PICD1 [APICEN]	L35			PHITM#	AC03

Table 32-1. Socket 5 Pin Cross Reference by Pin Name (Contd.)

V_{CC}									
A07	A19	B02	G37	N01	T34	Y01	AE01	AJ29	AN19
A09	A21	E15	J01	N37	U01	Y37	AE37	AN09	AN21
A11	A23	E21	J37	Q01	U33	AA01	AG01	AN11	AN23
A13	A25	E27	L01	Q37	U37	AA37	AG37	AN13	AN25
A15	A27	E37	L33	S01	W01	AC01	AJ11	AN15	AN27
A17	A29	G01	L37	S37	W37	AC37	AJ19	AN17	AN29

V_{SS}								
A03	B20	E23	M36	V02	AD02	AJ17	AM10	AM26
B06	B22	E29	P02	V36	AD36	AJ21	AM12	AM28
B08	B24	E31	P36	X02	AF02	AJ25	AM14	AM30
B10	B26	H02	R02	X36	AF36	AJ27	AM16	AN37
B12	B28	H36	R36	Z02	AH02	AJ31	AM18	
B14	E11	K02	T02	Z36	AJ07	AJ37	AM20	
B16	E13	K36	T36	AB02	AJ09	AL37	AM22	
B18	E19	M02	U35	AB36	AJ13	AM08	AM24	

NC/INC								
A37	E25	S33	W33	X34	AJ23	AN35	AN05	C01
E17	R34	S35	W35	AJ15	AL19	AE35*	AL01	

NOTE:

*This is the D/P# signal in the Pentium® processor (610\75, 735\90, 815\100, 1000\120)

$V_{CC}5$:	AN01	AN03

Table 32-2. Pentium® Processor (610\75, 735\90, 815\100, 1000\120) vs. Socket 5 Pins

Pentium® Processor (610\75, 735\90, 815\100, 1000\120) Signal	Socket 5 Signal	Pin Number	Pentium® Processor (610\75, 735\90, 815\100, 1000\120) Signal	Socket 5 Signal	Pin Number
INC	$V_{CC}5$	AN01	D/P#	NC	AE35
INC	$V_{CC}5$	AN03	NO PIN	V_{SS}	AJ07
INC	V_{CC}	B02	NO PIN	V_{SS}	AJ09
NO PIN	V_{SS}	E11	NO PIN	V_{CC}	AJ11
NO PIN	V_{SS}	E13	NO PIN	V_{SS}	AJ13
NO PIN	V_{CC}	E15	NO PIN	NC	AJ15
NO PIN	NC	E17	NO PIN	V_{SS}	AJ17
NO PIN	V_{SS}	E19	NO PIN	V_{CC}	AJ19
NO PIN	V_{CC}	E21	NO PIN	V_{SS}	AJ21
NO PIN	V_{SS}	E23	NO PIN	NC	AJ23
NO PIN	NC	E25	NO PIN	V_{SS}	AJ25
NO PIN	V_{CC}	E27	NO PIN	V_{SS}	AJ27
NO PIN	V_{SS}	E29	NO PIN	V_{CC}	AJ29
NO PIN	V_{SS}	E31	NO PIN	V_{SS}	AJ31
INC	V_{SS}	A03			

NOTES:

See highlighted cells in previous table.

All INCs are internal no connects. These signals are guaranteed to remain internally not connected in the Pentium processor processor.

32.3. SYSTEM DESIGN CONSIDERATIONS

32.3.1. CPU Type Pin Definition

The CPUTYP pin is a new configuration signal which, when sampled by the Pentium processor at the falling edge of RESET, indicates the type of OEM processor which will be placed in each socket site. Refer to Table 32-3 for a detailed description of the CPUTYP signal.

Table 32-3. CPUTYP Definition

Symbol	Type	Function
CPUTYP	Input	The **Processor Type** input signal is used to configure the Pentium processor for a single or dual processor system. The Pentium processor and the future OverDrive® processors will be configured as standalone processors when CPUTYP is connected to V_{SS}. The Pentium processor will be configured as a Dual processor when CPUTYP is connected to V_{CC}. In a two socket system, CPUTYP is connected to V_{SS} in the 296 pin SPGA site, and to V_{CC} in the 320 pin SPGA site. In a single socket system, CPUTYP is connected to V_{SS}. For the future OverDrive processor, CPUTYP will be used to determine whether the bootup handshake protocol will be used (in a dual socket system) or not (in a single socket system).

32.3.2. Single Socket System Considerations

32.3.2.1. SINGLE SOCKET PIN CONNECTIONS

Use the 320-pin SPGA Socket 5 pinout for single-socket systems. The following table shows how to connect the dual processing signals which remain unused in a single-socket system design.

Table 32-4. Dual Processing Signal Connections in a Single Socket System

Private Interface Signal	Connection
CPUTYP	V_{SS}
PBGNT#	NC
PBREQ#	NC
PHIT#	NC
PHITM#	NC
D/P#	NC

32.4. DUAL SOCKET SYSTEM CONSIDERATIONS

32.4.1. Dual Socket Power Considerations

In a dual socket system, the Pentium processor (610\75, 735\90, 815\100, 1000\120) will have a nominal power dissipation when non-operational due to the presence of a future OverDrive

processor in Socket 5. This power should not exceed 1Watt (this value is *preliminary* and an *approximation* only).

There will also be a period of time in which BOTH the Pentium processor (610\75, 735\90, 815\100, 1000\120) and the future OverDrive processor may be operating simultaneously. This is during RESET, and for a brief period of time following the falling edge of RESET due to the bootup handshake protocol between the two processors. The power dissipation during this time will not exceed the maximum power of a dual processing system. Therefore, dual socket Pentium processor (610\75, 735\90, 815\100, 1000\120) systems must be able to handle the full power of a Primary and Dual processor operating simultaneously.

32.4.1.1. DUAL SOCKET PIN CONNECTIONS

In dual socket systems, the 296 pin SPGA Pentium processor pinout should be used for the primary OEM processor site. The second processor site should be occupied by the 320-pin SPGA Socket 5 pinout.

Table 32-5 lists all Pentium processor future OverDrive processor signals which should be connected together in order to operate in either dual processing mode with the Dual processor present, or upgraded with the future OverDrive processor.

Table 32-5. Signal Connections in a Dual Socket System

CONNECT TOGETHER				
A[31:3]	A20M#	ADS#	ADSC#	AHOLD
AP	BE[7:0]#	BF	BOFF#	BRDY#
BRDYC#	BREQ	BUSCHK#	CACHE#	CLK
D/C#	D[63:0]	DP[7:0]	EADS#	EWBE#
FERR#	FLUSH#		HIT#	HITM#
HLDA	HOLD	IGNNE#	INIT	INV
KEN#	LOCK#	M/IO#	NA#	PBGNT#
PBREQ#	PCD	PEN#	PHIT#	PHITM#
PICCLK	PICD[1:0]	PWT	RESET	SCYC
	V_{CC}	V_{SS}	W/R#	WB/WT#

Table 32-6 lists the signals which must not be connected together in a dual socket Pentium processor system. These functions, if used in the system, must be handled by the system design individually for each socket. If any of these features are used in uni-processing mode with only the OEM processor present, then they *MUST* be handled for the case of a future OverDrive processor in the second socket.

NOTES

CPUTYP should be strapped to V_{SS} in the 296 pin SPGA Pentium processor (610\75, 735\90, 815\100, 1000\120) site. CPUTYP should be strapped to V_{CC} in the 320 pin SPGA Socket 5 (future OverDrive processor) site.

D/P# is only a Pentium processor signal, and is defined as a no connect in Socket 5.

If boundary scan is used in the design, TDO of one socket is connected to TDI of the other to make a serial chain with other boundary scan components in the system. If boundary scan is not used in the design, TDI and TDO should remain disconnected.

Table 32-6. Signal Connections in a Dual Socket System

DO NOT CONNECT TOGETHER				
APCHK#	BP[3:0]	CPUTYP	D/P#	IERR#
PM/BP[1:0]	PRDY	R/S#	TDI	TDO

Table 32-7 lists the signals which may be connected together or not. It is the system designer's choice as to whether they want to handle these signals individually for each socket, or connect them together. Intel recommends that they be connected together for ease of using both the Dual processor and the future OverDrive processor in the same second socket. If any of these signals are not connected together, and their functions are used in uni-processing mode with only the OEM processor present, then they *MUST* be handled for the case of a future OverDrive processor in the second socket. Refer to earlier chapters in this document for additional details on the functionality of these signals.

Table 32-7. Signal Connections in a Dual Socket System

MAY BE CONNECTED TOGETHER				
FRCMC#	INTR/LINT0	NMI/LINT1	PCHK#	SMI#
SMIACT#	STPCLK#	TCK	TMS	TRST#

NOTE

NC signals should not be connected to anything. $V_{CC}5$ signals can be connected together since pins AN01 and AN03 are Internal No Connects (INC) on the Pentium processor (610\75, 735\90, 815\100, 1000\120).

32.5. ELECTRICAL SPECIFICATIONS

The future OverDrive processor will have the same power and ground specifications, decoupling recommendations, and connection specifications as the Pentium processor.

32.5.1. V_{CC5} Pin Definition

The future OverDrive processor pinout contains two 5V V_{CC} pins ($V_{CC}5$) used to provide power to the fan/heatsink. These pins should be connected to +5 volts ±5% regardless of the system design. Failure to connect $V_{CC}5$ to 5V may cause the component to shut down.

32.5.2. Absolute Maximum Ratings of Upgrade

The on-chip Voltage Regulation and fan/heatsink devices included on the future OverDrive processor require different stress ratings than the Pentium processor (610\75, 735\90, 815\100, 1000\120). The voltage regulator is surface mounted on the future OverDrive processor and is,therefore, an integral part of the assembly. The future OverDrive processor storage temperature ratings are tightened as a result. The fan is a detachable unit, and the storage temperature is stated separately in the table below. Functional operation of the future OverDrive processor remains 0°C to 70°C.

Table 32-8. Absolute Maximum Ratings

Future Pentium® OverDrive® Processor and Voltage Regulator Assembly:					
	Parameter	**Min**	**Max**	**Unit**	**Notes**
	Storage Temperature	-30	100	°C	
	Case Temperature Under Bias	-30	100	°C	
Fan:					
	Parameter	**Min**	**Max**	**Unit**	**Notes**
	Storage Temperature	-30	75	°C	
	Case Temperature Under Bias	-30	75	°C	

WARNING

Stressing the devices beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.

32.5.3. DC Specifications

The future OverDrive processor will have compatible DC specifications to the Pentium processor, except that I_{CC} (Power Supply Current), $I_{CC}5$ (Fan/Heatsink Current), and V_{CC} are the following:

Table 32-9. I_{cc} Specification

Symbol	Parameter	Min	Max	Unit
I_{CC}	Power Supply Current		4330	mA
$I_{CC}5$	Fan/Heatsink Current		200	mA

NOTE:

V_{CC} = 3.135V to 3.6V, $V_{CC}5$ = 5V ±5%

Refer to Chapter 23 for a listing of the remaining DC Specifications.

32.5.3.1. AC SPECIFICATIONS

The future OverDrive processor will have the same AC specifications as the Pentium processor. Refer to Chapter 23 for a listing of the remaining AC Specifications.

32.6. MECHANICAL SPECIFICATIONS

The future OverDrive Processor will be packaged in a 320-pin ceramic staggered pin grid array (SPGA). The pins will be arranged in a 37 x 37 matrix and the package dimensions will be 1.95″ x 1.95″ (4.95cm x 4.95cm).

Table 32-10. Future Pentium® OverDrive® Processor Package Summary

	Package Type	Total Pins	Pin Array	Package Size
Future Pentium® OverDrive® Processor	SPGA	320	37 x 37	1.95" x 1.95" 4.95cm x 4.95cm

Table 32-11. Future Pentium® OverDrive® Processor Package Dimensions

Family: Ceramic Staggered Pin Grid Array Package						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A*		33.88	Solid Lid		1.334	Solid Lid
A1	0.33	0.43	Solid Lid	0.013	0.017	Solid Lid
A2	2.62	2.97		0.103	0.117	
A4		20.32			0.800	
A5	10.16		Air Space	0.400		Air Space
B	0.43	0.51		0.017	0.020	
D	49.28	49.91		1.940	1.965	
D1	45.47	45.97		1.790	1.810	
E1	2.41	2.67		0.095	0.105	
E2	1.14	1.40		0.045	0.055	
L	3.05	3.30		0.120	0.130	
N	320		SPGA pins	320		SPGA pins
S1	1.52	2.54		0.060	0.100	

NOTES:

* Assumes the minimum air space above the fan/heatsink

A 0.2″ clearance around three of four sides of the package is also required to allow free airflow through the fan/heatsink.

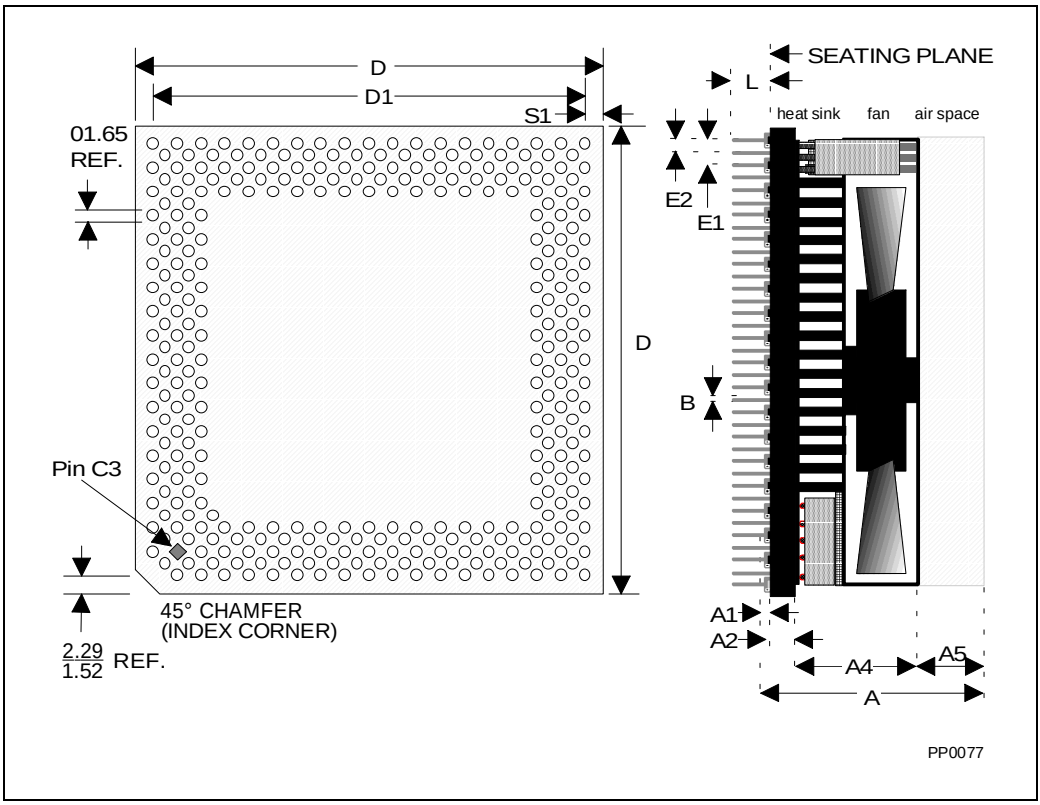


Figure 32-3. Future OverDrive® Processor Package Dimensions

32.6.1. Socket 5 Mechanical Specifications

Mechanical specifications for Socket 5 may vary among socket vendors. OEMs should contact Intel for the most current list of Intel-qualified socket vendors and should directly contact the socket vendors for the most current socket information. For a complete list of qualified sockets and vendor order numbers, contact Intel or call the Intel Faxback number for your geographical area and have document number 7209 automatically faxed to you.

32.7. THERMAL SPECIFICATIONS

Section 32.7.1 outlines the thermal specifications for the future OverDrive processor. Section 32.7.2 follows with information specific to the future OverDrive processor cooling solution. Section 32.7.3 contains information regarding thermal failure protection

32.7.1. Thermal Information

32.7.1.1. THERMAL SPECIFICATIONS

The future OverDrive processor will be cooled with a fan/heatsink cooling solution. The future OverDrive processor with a fan/heatsink is specified for proper operation when T_A (air temperature entering the fan/heatsink) is a maximum of 45°C. When the $T_A(max) \leq 45°C$ specification is met, the fan/heatsink will keep T_{CASE} within the specified range of 0°C to 70° C provided airflow through the fan/heatsink is unimpeded.

32.7.1.2. THERMAL EQUATIONS AND DATA

The future OverDrive processor fan/heatsink cooling solution requires that the T_A does not exceed 45°C. To calculate T_A values, the following equations may be used:

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

where, T_A and T_C = ambient and case temperature, respectively (°C)

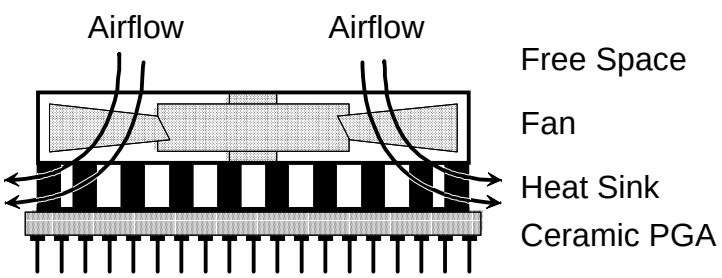
Θ_{CA} = case-to-ambient thermal resistance (°C/Watt)

P = maximum power consumption (Watt)

32.7.2. Upgrade Processor Cooling Requirements

32.7.2.1. THERMAL AND PHYSICAL SPACE REQUIREMENTS

Figure 32-4 illustrates the thermal and physical space requirements for the future OverDrive processor.



* For the Future Pentium® OverDrive® processor with a fan/heatsink, the specification is T <= 45°C with T_A measured where the air enters the fan/heatsink unit. The air will enter the fan/heatsink from the top, and exit from the sides.

Figure 32-4. Thermal and Physical Space Requirements

- Physical Requirements
  - 1.4" vertical clearance above the surface (opposite pin side) of Socket 5. Note that the actual total space is slightly less than this value (1.334"). This requirement is rounded in order to remain consistent over other upgrade processor products.
  - 0.2" clearance around three of four sides of the package. The one side exempt from the 0.2" is determined by the OEMs custom layout. This is to add flexibility into board layouts.
- Thermal Requirements
  - For the future OverDrive processor, a maximum air temperature entering the fan/heatsink of 45°C is specified. T_A is measured where the air enters the fan/heatsink unit. The air will enter the fan/heatsink from the top, and exit from the sides.
- Other Important Considerations
  - Adequate airflow for the future OverDrive processor with a fan/heatsink.
  - Two external 5V power connections via package pins.

32.7.2.2. FAN/HEATSINK COOLING SOLUTION

The future OverDrive processor will utilize a fan/heatsink cooling solution. Intel's fan/heatsink cooling solution requires that the air temperature entering the fan/heatsink (T_A) does not exceed 45°C under worst case conditions. When the air temperature requirement is met, the fan/heatsink will keep the case temperature, T_C, within the specified range, provided airflow through the fan/heatsink is unimpeded. The 45°C maximum air temperature entering the fan/heatsink was chosen to provide a reliable and acceptable fan life, and adequately cool the future OverDrive processor.

Although the thermal performance of fan/heatsink cooling solutions does not significantly increase with increased airflow over the processor, adequate airflow through the PC chassis is required in order to prevent localized heating around the processor. A clear air path from the PC vents to the power supply fan, as shown in the system on the left in Figure 32-5, will enable the warm air from the future OverDrive processor to be pulled out of the system by the power supply fan. If no air path exists across the processors, as shown in the system on the right in Figure 32-5, the warm air from the future OverDrive processor will not be removed from the system, possibly resulting in localized heating ("hot spots") around the processor. Figure 32-5 shows examples of air exchange through a PC chassis.

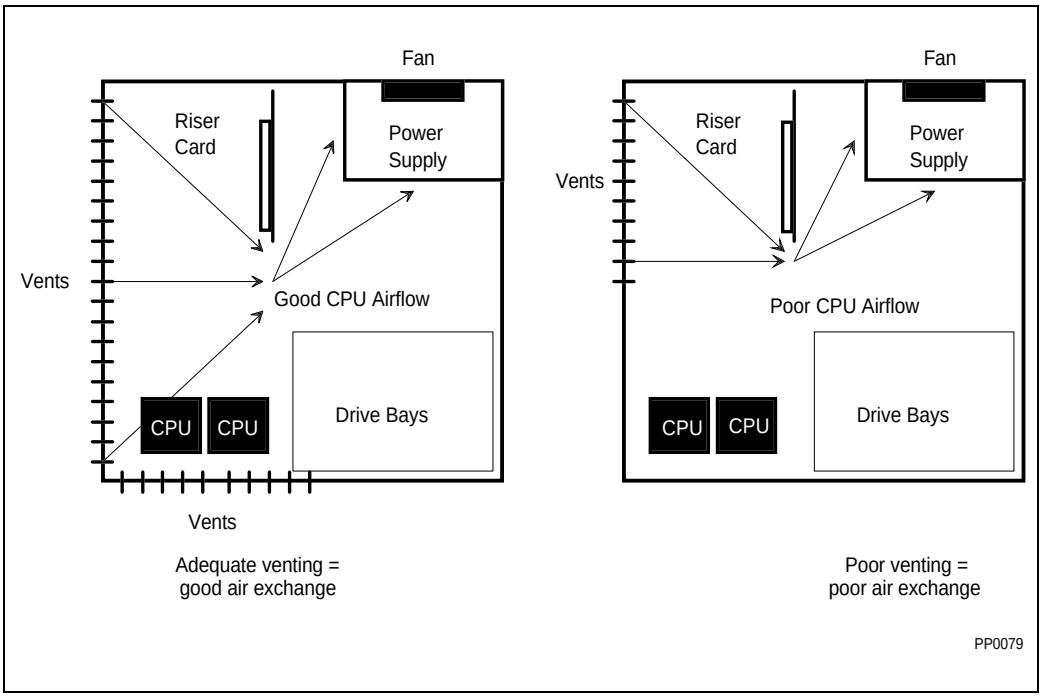


Figure 32-5. Examples of Air Exchange Through PC Chassis

A height of 0.4" airspace is REQUIRED above the fan/heatsink unit to ensure that the airflow through the fan/heatsink is not blocked. Blocking the airflow to or from the fan/heatsink reduces the cooling efficiency and decreases the fan lifetime. Figure 32-6 shows unacceptable blocking of the airflow.

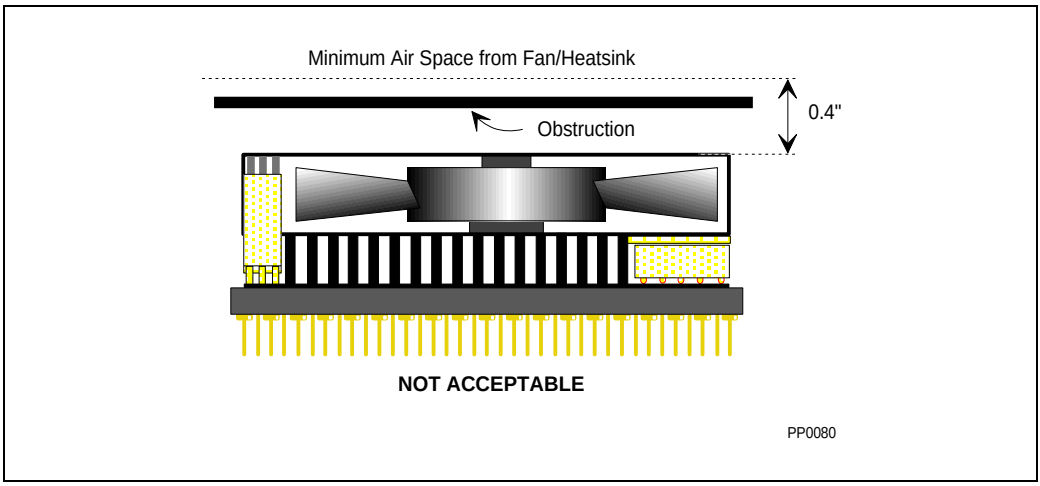


Figure 32-6. Fan/Heatsink Unacceptable Airflow Blockage

The fan/heatsink will reside within the boundaries of the surface of the chip (1.95" x 1.95"). However, there are also free airspace clearance requirements around the ceramic package to ensure that the airflow is not blocked to or from the fan/heatsink. Figure 32-7 details the minimum space needed around the chip package to ensure proper airflow through the fan/heatsink.

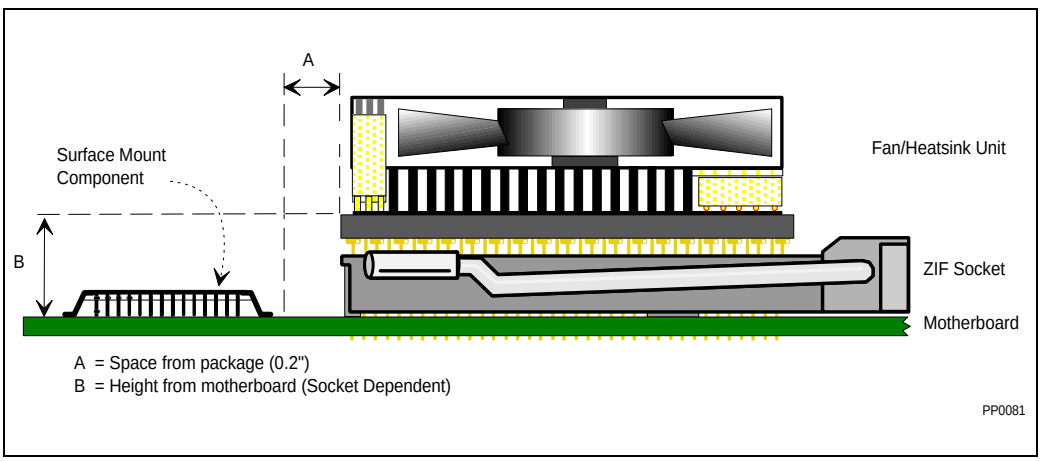


Figure 32-7. Required Free Space from Sides of Chip Package

As shown in Figure 32-7, it is acceptable to allow any device (i.e. add-in cards, surface mount device, chassis etc.) to enter within the free space distance of 0.2" from the chip package if it is not taller than the level of the heat sink base. In other words, if a component is taller than height "B", it cannot be closer to the chip package than distance "A". The 0.2” clearance "A" must be maintained on three of four sides of the chip package.

32.7.3. Thermal Failure Protection

The future OverDrive processor includes a fan/heatsink cooling solution. In order to protect the end-user from thermal failures due to a failure of the fan/heatsink device, Intel has incorporated a protection mechanism into the upgrade product which will prevent overheating in the event of a fan failure.

32.7.3.1. FAN FAILURE

The fan/heatsink connection to the future OverDrive processor includes a FANFAIL signal as an input to the processor. This signal is generated by the circuitry of the fan/heatsink device to indicate the status of the fan RPM (Rotations Per Minute). The FANFAIL signal goes active to indicate that the fan RPM has degraded, and will not sufficiently cool the upgrade component. While the signal remains inactive, it indicates that the fan RPM (and hence, the heatsink temperature) is acceptable. The FANFAIL signal to the upgrade processor, once active, will not return to an inactive state until the fan returns to an acceptable operating RPM range and the fan has been powered down and then restarted. If the system power is turned off and then turned on again, the FANFAIL signal will be reset to its inactive state, and will not be re-asserted until the fan has had an opportunity to return to the normal operating RPM range but has failed to do so.

The FANFAIL signal is a direct connection from the fan to the future OverDrive processor core. This signal is not bonded to a processor pin for system use. The next section describes how the system can use the fan failure indication via an internal register for thermal protection.

The future OverDrive processor has an internal pull-up resistor to simulate a fan failure if the fan is not present.

32.7.3.2. THERMAL ERROR RECOGNITION BY THE PROCESSOR

The Machine Check Type Register (MCT) in the future OverDrive processor can be used by the system to monitor the value of the FANFAIL input. A new bit, THermal ERRor (THERR, bit 5) has been defined in the MCT register to indicate that a thermal failure has occurred. Following RESET, the THERR bit is cleared to a value of 0.

If the FANFAIL signal goes active or fails to clear after reset, the future OverDrive processor will set the THERR bit (5) in the MCT register to a 1. Note that the Check bit (0) of the MCT register will not be set, nor will an Interrupt 18 be generated from the setting of the THERR bit if Machine Check Interrupts are enabled. The future OverDrive processor will have no mechanism to automatically inform the system that a thermal error has occurred. In order for the system to detect that a thermal failure has occurred, software must poll the THERR bit in the MCT register. Once the THERR bit in the MCT register has been set, it will be cleared only by RESET.

32.7.3.3. DETECTION OF A THERMAL ERROR VIA SOFTWARE

Two instructions, RDMSR (Read from Model Specific Register) and WRMSR (Write to Model Specific Register), may be used by software to poll the value of the THERR bit in the MCT register. Note that RDMSR clears bit 0 of the MCT register; therefore, steps must be taken to preserve the integrity of the contents of the register following a RDMSR instruction.

Intel plans to ship a diagnostic utility diskette with the future OverDrive processor which can be used by end-users to detect a thermal failure. This utility will include error detection software as well as instructions to the end-user on how to replace the fan.

32.8. TESTABILITY

32.8.1. Boundary Scan

The future OverDrive processor supports the IEEE Standard 1149.1 boundary scan using the Test Access Port (TAP) and TAP Controller as described in Chapter 11. The boundary scan register for the future OverDrive processor contains a cell for each pin. The turbo upgrade component will have a different bit order than the Pentium processor. If the TAP port on your system will be used by an end user following installation of the future OverDrive processor, please contact Intel for the bit order of the OverDrive processor boundary scan register.

33

Model Specific Registers and Functions

CHAPTER 33
MODEL SPECIFIC REGISTERS AND FUNCTIONS

This chapter introduces the model specific registers (MSR's) as they are implemented on the Pentium processor. Model specific registers are used to provide access to features that are generally tied to implementation dependent aspects of a particular processor. For example, testability features that provide test access to physical structures such as caches, and branch target buffers are inherently model specific. Features to measure the performance of the processor or particular components within the processor are also model specific.

The features provided by the model specific registers are expected to change from processor generation to processor generation and may even change from model to model within the same generation. Because these features are implementation dependent, they are not recommended for use in portable software. Specifically, software developers should not expect that the features implemented within the MSR's will be supported in an upward compatible manner across generations or even across different models within the same generation.

33.1. MODEL SPECIFIC REGISTERS

The Pentium processor implements two new instructions and several model specific registers. The RDMSR and WRMSR instructions are used to read and write the MSR's respectively. A feature bit in EDX (bit 5), reported by the CPUID instruction, indicates that the processor supports the RDMSR and WRMSR instructions.

33.1.1. Model Specific Register Usage Restrictions

Proper use of the features described in this chapter requires that the CPUID instruction be used to validate not only that the FAMILY and MODEL reported in EAX are equal to "5" and "1" or "2" respectively. Note that this requirement is significantly more restrictive than is required for new architectural features where it is sufficient to validate that the FAMILY is equal to or greater than that of the first family to implement the new feature. For more information regarding the use of the CPUID instruction, refer to Chapter 25 of Volume 3 of the *Pentium® Processor Family Developer's Manual.*

33.1.2. Model Specific Registers

Access to the model specific registers is provided through the RDMSR and WRMSR instructions. Table 33-1 lists the model specific registers that are implemented on the Pentium processor and the values to place in ECX during RDMSR and WRMSR instructions

in order to access each register. For more information regarding the use of these instructions, refer to Chapter 25 of Volume 3 of the *Pentium® Processor Family Developer's Manual.*

Table 33-1. Model Specific Registers

ECX Value (in Hex)	Register Name	Description
00	Machine Check Address*	Stores address of cycle causing the exception
01	Machine Check Type*	Stores cycle type of cycle causing the exception
02	Test Register 1	Parity Reversal Register
03	RESERVED	
04	Test Register 2	Instruction Cache End Bit
05	Test Register 3	Cache Test Data
06	Test Register 4	Cache Test Tag
07	Test Register 5	Cache Test Control
08	Test Register 6	TLB Test Linear Address
09	Test Register 7	TLB Test Control & Physical Address 31–12
0A	RESERVED	
0B	Test Register 9	BTB Test Tag
0C	Test Register 10	BTB Test Target
0D	Test Register 11	BTB Test Control
0E	Test Register 12	New Feature Control
0F	RESERVED	
10	Time Stamp Counter	Performance Monitor
11	Control and Event Select	Performance Monitor
12	Counter 0	Performance Monitor
13	Counter 1	Performance Monitor
14+	RESERVED	

NOTES:

* CR4.MCE must be 1 in order to utilize the machine check exception feature.

33.2. TESTABILITY AND TEST REGISTERS

The Pentium processor provides testability access to the on-chip caches, TLB's, BTB and internal parity checking features through model specific test registers. On the Intel486 processor, access to the test registers was provided through the dedicated MOV to/from TRx instructions. As the Pentium processor provides an expanded set of test capabilities, the functionality of the MOV to/from TRx instructions is superseded by that of the RDMSR/WRMSR instructions.

33.2.1. Cache, TLB and BTB Test Registers

The Pentium processor contains several test registers. The purpose of the test registers is to provide direct access to the Pentium processor's caches, TLBs, and BTB, so test programs can easily exercise these structures. Because the architecture of the caches, TLBs, and BTB is different, a different set of test registers (along with a different test mechanism) is required for each. Most test registers are shared between the code and data caches.

The test registers should be written to for testability purposes only. Writing to the test registers during normal operation causes unpredictable behavior. Note that when the test registers are used to read or write lines directly to or from the cache, external inquire cycles must be inhibited to guarantee predictable results when testing. This is done by setting both CR0.CD and CR0.NW to "1". In addition, the INVD, WBINVD and INVLPG instructions may be executed before and after but not during testing.

NOTE

If a memory data access occurs during a code cache testability operation using the test registers, the data chat is checked before the external memory operation in initiated. If the access is a miss in the data cache, then if the accessed line is valid in the code cache, it is invalidated through the internal snooping mechanism.

Similarly, if a code access occurs during a data cache testability operation using the test registers, the code cache is checked before the external memory operation is initiated. If the access is a miss in the code cache, then the accessed line if valid in the data cache in invalidated (or written back and then invalidated if in the M-state) through the internal snooping mechanism.

33.2.1.1. CACHE TEST REGISTERS

The registers in Figure 33-1 provide direct access to the Pentium processor's code and data caches.

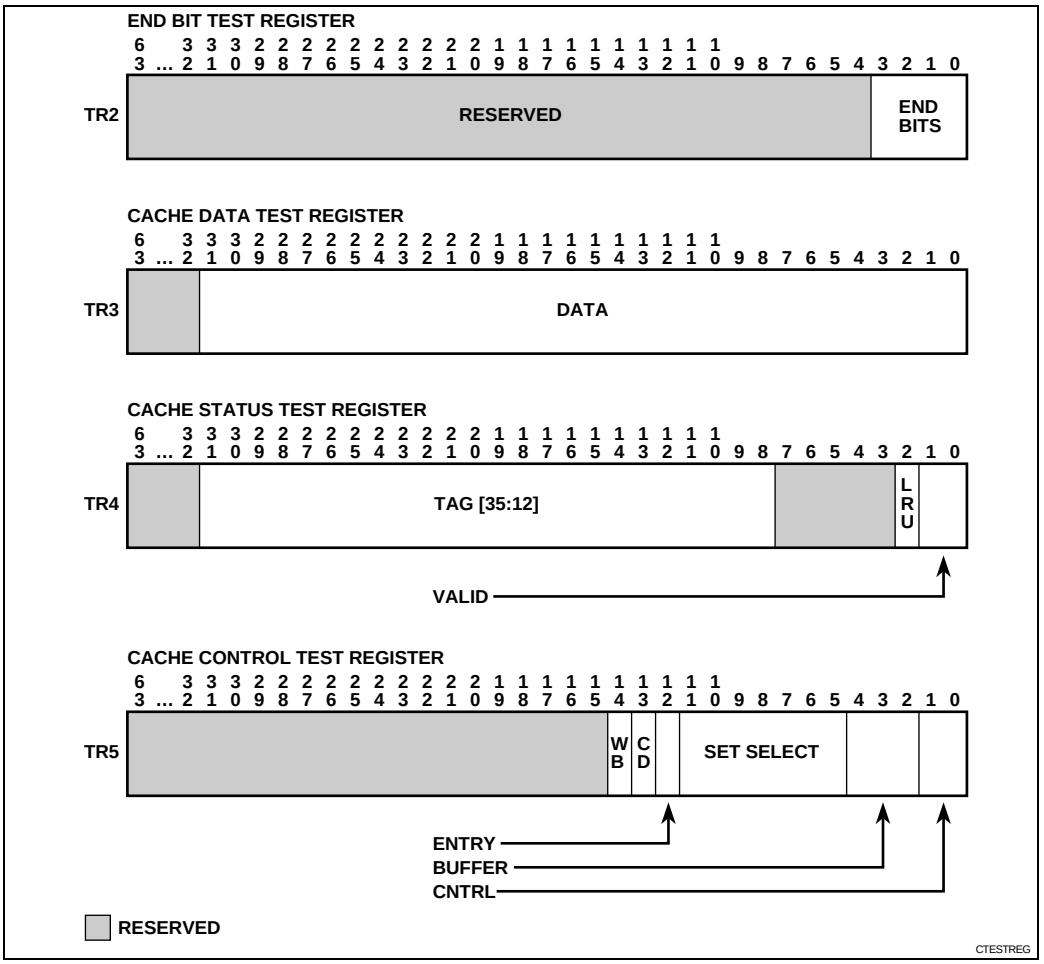


Figure 33-1. Cache Test Registers

TR2 is the End Bit Test Register for the code cache. It contains 4 end bits. Each end bit corresponds to one byte of instruction in TR3 during code cache testability access. Since a cache line 32 bytes, 8 access are needed to read or write the end bits for the entire cache line. TR2 is used for accesses to the code cache only.

The End Bits are used to indicate instruction boundaries. The end bit mechanism aids the decode of two variable length instructions per clock by providing information on where the boundary between instruction is. If a given byte is the last byte in an instruction, the corresponding end bit is set to one. When a line is written into the code cache after a miss, all end bits corresponding to the line are initialized to one. As instructions are decoded, the end bits are checked for correctness and modified if incorrect. In order for two instructions to be issued in a single clock, the end bits of the u-pipe instruction must have the correct values,

otherwise only one instruction will be issued. This does have the effect that instructions are usually not paired the first time that they are put in the code cache.

TR3 is the Cache Data Test Register. This is where the data is held on its way into or out of the cache. Prior to a cache testability write, software must load an entire cache line into the 32-byte fill buffer using TR3, 4 bytes at a time. Similarly, during a cache testability read, the Pentium processor extracts a specified 4-byte data quantity from a cache line and places the data in TR3. A 32-byte cache line may be written to or read from TR3 as eight 4-byte accesses.

TR4 is the Cache Status Test Register. It contains the tag, LRU and valid bits to be written to or read from the cache. Like TR3, TR4 must be loaded with the tag/LRU/valid bits prior to a testability write, and gets updated with the tag/LRU/valid bits as a result of a testability read. **Note that TR4[31:28] always return a zero as a result of a testability read.** The two valid bits are interpreted differently by the code and data caches, depending on the setting of TR5.CD bit. The encodings for TR4.valid are as shown in Table 33-2. The encodings for the LRU bit are shown in Table 33-3.

Table 33-2. Encoding for Valid Bits in TR4

TR5.CD=1 (Data Cache)	valid[1]	valid[0]	Meaning
	0	0	Cache line in 1 state
			Cache line in S state
			Cache line in E state
			Cache line in M state
TR5.CD=0 (Code Cache)	**valid[1]**	**valid[0]**	**Meaning**
	X	0	Cache line invalid
	X	1	Cache line valid

Table 33-3. Encoding of the LRU Bit in TR4

LRU	Points to WAY
0	0
1	1

NOTE

The LRU bits for the instruction cache change state when an entry is read using the test registers, with CR0.CD=1. The LRU bit for this data cache, however, do not change their state during testability reads with CR0.CD=1.

TR5 is the Cache Control Test Register. it contains the writeback bit, the CD bit, the cache entry, the set address, the buffer select, and a two-bit control field, cntl.

The writeback bit determines whether that particular line is configured for writethrough or allows the possibility of writeback. It is used by the data cache only. The CD bit distinguishes between the code and data cache. The entry field selects one of the two ways in the cache. The set address field selects one of 128 sets within the cache to be accessed. The buffer field selects one of the eight portions of a cache line to be visible through TR3. The control field selects one of the four possible operation modes. The encodings for the TR5 fields are shown in Table 33-4, Table 33-5, Table 33-6 and Table 33-7.

Table 33-4. Encoding of the WB Bit in TR5

WB	Writeback or Writethrough
0	Writethrough
1	Writeback

Table 33-5. Encoding of the Code/Data Cache Bit in TR5

CD	Cache
0	Code cache
1	Data cache

Table 33-6. Encoding of the Entry Bit in TR5

Entry	Way
0	Way 0
1	Way 1

Table 33-7. Encoding of the Control Bits in TR5

Cntl1	Cntl0	Command
0	0	Normal operation
0	1	Testability write
1	0	Testability read
1	1	Flush

33.2.1.1.1. Direct Cache Access

To access the cache for testing, the programmer specifies a set address and entry and requests a testability read or write. No tag comparison is done; the programmer can directly read/write a particular entry in a particular set.

To write down an entry into the cache:

- Disable replacements by setting CR0.CD=1.
- For each 4-byte access:
    1) Write address into TR5.buffer. Here, TR5.cntl=00.
    2) Write data into TR3.
    3) Write end bits into TR2 (for instruction cache only).
- Write the desired tag, LRU and valid bits into TR4. Note that the contents of TR4 completely overwrites the previous tag, LRU and valid bits in the cache.
- Perform a testability write by loading TR5 with the appropriate CD, entry, set address, and cntl fields. Here, TR5.cntl=01.

To read an entry from the cache:

- For each 4-byte access:
    1) Write the appropriate CD, entry, set address, buffer and cntl fields into TR5. Here, TR5.cntl=10.
    2) Read data from TR3.
    3) Read end bits from TR2[3:0] (for instruction cache only).
    4) Read the tag, LRU, and valid bits from TR4. No hit/comparison is performed. Whatever was in that entry in that set is read into TR4, TR3, and TR2.

To invalidate the cache or invalidate an entry:

- When TR5.cntl=11 (flush), and CD=0 (code cache), the entire code cache is invalidated. However, if TR5.cntl=11 and CD=1 (data cache), the user can specify through the TR5.WB bit whether to invalidate the entire data cache, or invalidate and writeback only the cache line specified by TR5 (see Table 33-8).

Note that TR2, TR3, and TR4 permit both reads and writes, whereas TR5 is a write-only register. The test registers should be written to for testability accesses only. Writing to the test registers during normal operation may cause unpredictable behavior. For example, inadvertent cache hits can be created.

During cache testability operations, the internal snooping mechanism functions similar to that described in section 6.4.3 of the *Pentium® Processor Data Book*. If a memory data access occurs during a code cache testability operation using the test registers, the date cache is checked before the external memory operation is initiated. If the access is a miss in the data cache, then the accessed line if valid in the code cache is invalidated through the internal snooping mechanism.

Similarly, if a code access occurs during a data cache testability operation using the test registers, the code cache is checked before the external memory operation is initiated. If the access is a miss in the code cache, then the accessed line if valid in the data cache is invalidated (or written back and then invalidated if in the M-state) through the internal snooping mechanism.

Table 33-8. Definition of the WB Bit in TR5

TR5.cntl=11	TR5.WB	Meaning
CD=0	X	Invalidate the entire code cache.
CD=1	0	Invalidate entire data cache. Modified lines are not written back.
CD=1	1	Invalidate line. Writeback if modified.

33.2.1.2. TLB TEST REGISTERS

The registers in Figure 33-2 provide access to the code and data cache translation lookaside buffers (TLBs). Note that the data cache has two TLBs: a 64-entry TLB for 4K-byte data pages and an 8-entry TLB for 4 MByte data pages. The code cache contains only one 32-entry TLB for both 4 KByte code pages and 4 MByte code pages. The 4 MByte code pages are cached in 4 KByte increments (the PS bit in TR6 is ignored).

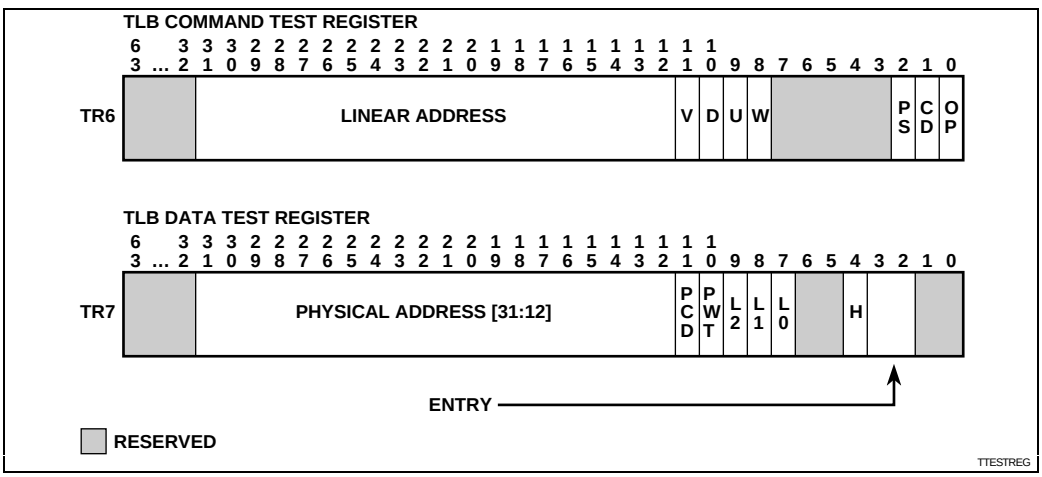


Figure 33-2. TLB Test Registers

TR6 is the TLB Command Test Register. It contains the linear address, code/data TLB select (CD), operation (Op) bits and the following status bits: valid (V), dirty (D), user (U), writeable (W), and page size (PS) bits.

The status bits are inputs to the TLB entry during testability writes, and outputs from the TLB entry during testability reads. The V bit indicates whether a TLB entry is valid or invalid during testability writes. The D bit indicates whether or not a write across was made to the page. The U bit indicates the privilege level that the Pentium processor must be in to access the page. The W bit is one of the factors in determining the read/write protection of the page. The PS (page size) bit specifies the page size for the TLB entry. The CD bit determines if the code or data TLB is being accessed. The Op bit distinguished between a read and write cycle.

The W-bit, D-bit, and PS-bit are defined only for the data TLB.

Table 33-9, Table 33-10, Table 33-11, Table 33-12, Table 33-13, Table 33-14 and Table 33-15 list the encodings for the fields in the TR6 register.

Table 33-9. Encoding for the Valid Bit in TR6

Valid	Valid/Invalid TLB Entry
0	Invalid
1	Valid

Table 33-10. Encoding for the Dirty Bit in TR6

D-bit	Write access made to page?
0	Write access was not made
1	Write access was made

Table 33-11. Encoding for the User Bit in TR6

U-bit	Privilege Level Access Allowed
0	PL=0,1,2,3
1	PL=0

Table 33-12. Encoding for the Writeable Bit in TR6

W-bit	Writes Allowed?
0	No writes, read only
1	Allows writes

Table 33-13. Encoding for the Page Size Bit in TR6

PS-bit	Page Size
0	4 KByte Page
1	4 MByte

NOTE

Normally the user should not allocate a page entry in both the TLBs; during testability however if a match is found in both, then the processor reports that it found it for the 4 MByte page size (PS=1).

Table 33-14. Encoding for the Operation Bit TR6

Op	Command
0	TLB write
1	TLB read

Table 33-15. Encoding for the Code/Data TLB Bit in TR6

CD	Cache
0	Code TLB
1	Data TLB

TR7 is the TLB Data Test Register. It contains bits 31:12 of the physical address, the hit indicator H, a two-bit entry pointer, and the status bits. The status bits include the two paging attribute bits PCD and PWT, and three LRU bits (L0, L1, and L2). PCD is the page level cache disable bit. PWT is the page level write through bit. The LRU bits determine which entry is to be replaced according to the pseudo-LRU algorithm. TLB reads which result in hits and TLB writes can change the LRU bits. The LRU bits reported for a test read are the value before the TLB read. The LRU bits are then changed according to the pseudo-LRU replacement algorithm.

The H is the hit indicator. This bit needs to be set to 1 during testability writes. During testability reads, if the input linear address matches a valid entry in the TLB, the H bit is set to 1. The two entry bits determine in which one of the four ways to write to the TLB during testability writes. During testability reads, they indicate the way that resulted in a read bit.

TR6, and TR7 are read/write registers. The test registers should be written to for testability accesses only. Writing to the test registers during normal operation causes unpredictable behavior.

When reading from the code cache TLB (TR5.cd = 0), the TR6 register zeros out bits [31:12] (corresponding to the linear address) at the end of the TLB testability read cycle. This does

not mean that an incorrect linear address was used. All operations happen normally (with whatever linear address was written into TR6 before the testability read operation).

33.2.1.2.1. TLB Access

Unlike the caches, the TLB is structured as a CAM cell and, thus, can only be searched (rather than directly read). In other words, the programmer can directly read/write a particular entry in a particular set of the code or data caches, however the TLB only reports a hit or a miss in the Hit bit in TR7. Dumping the TLB requires the programmer to step through the entire linear address space one page at a time.

To write an entry into the TLB:

- Write the physical address bits [31:12], attribute bits, LRU bits and replacement entry into TR7, setting TR7.H=1.
- Write the linear address, protection bits, and page size bit into TR6, setting TR6.Op=0.

To read an entry from the TLB:

- Write the linear address, CD, and OP bits into TR6, setting TR6.Op=1.
- If TR7.H is set to 1, the read resulted in a hit. Read the translated physical address, attribute bits, and entry from TR7. Read the V, D, U, and W bits from TR6. If TR7.H is cleared to 0, the read was a miss and the physical address is undefined.

Note that when reading from the TLB, the PS bit in the TR6 register does not have to be set; the PS bit is actually written by the processor at the end of the TLB (testability) lookup. Based on the PS bit the user is supposed to infer whether the linear address found in the TLB corresponds to the 4 KByte or 4 MByte page size. Normally the user should not allocate a page entry in both the TLBs; during testability however if a match is found in both, then the processor reports that it found it for the 4 MByte page size (PS=1).

Also note that when reading from the code cache TLB (TR5.CD=0), the TR6 register zeros out bits 12-31 (corresponding to the linear address) at the end of the TLB testability read cycle. This does not mean that an incorrect linear address was used. All operations happen normally (with whatever linear address was written into TR6 before the testability read operation).

33.2.1.3. BTB TEST REGISTERS

The test registers in Figure 33-3 provide direct access to the branch target buffer. Note that the branch prediction mechanism should be disabled through test register 12 before doing any BTB testability access.

TR9 is the BTB Tab Test Register. Before writing any entry into the BTB, software must first load TR9 with a valid tag address and history information. After reading any entry in the BTB, the hardware places the retrieved tag and history bits in TR9.

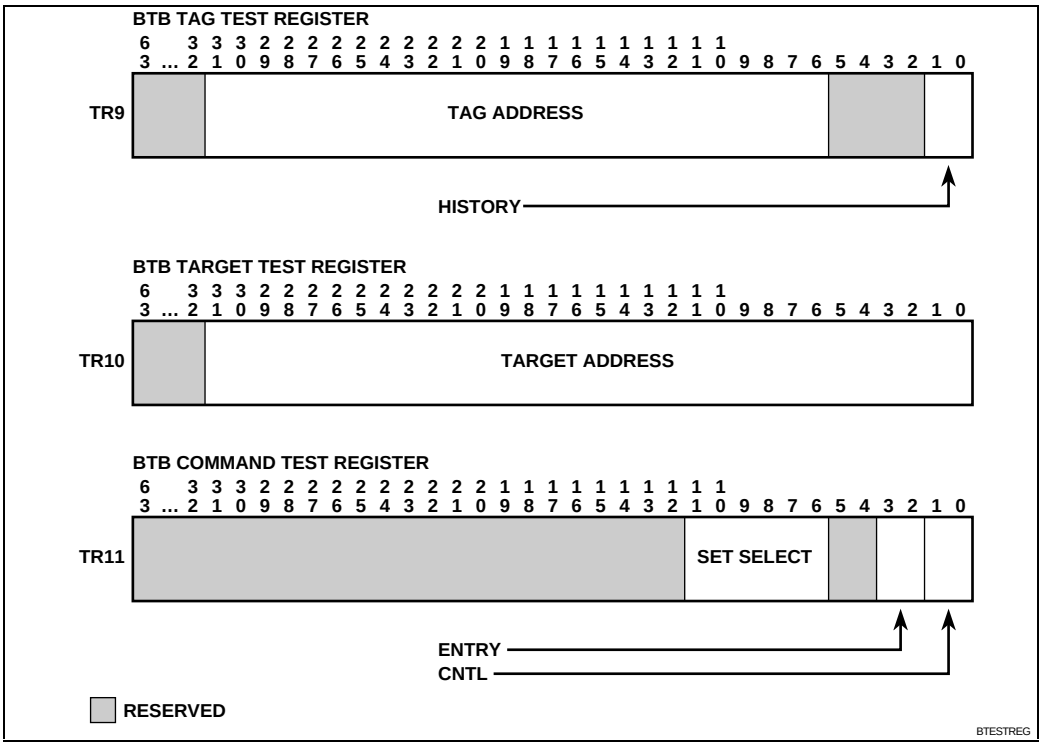


Figure 33-3. BTB Test Registers

TR10 is the BTB Target Test Register. Like TR9, TR10 must be loaded with the target address before a testability write. After a BTB testability read, the target address is placed in this register.

TR11 is the BTB Command Test Register. This register is used to issue read and write commands to the BTB. the set address field selects one of 64 sets to access. The entry field selects one of four ways within the set. A BTB testability cycle is initiated by loading TR11 with TR11.cntl=01 or TR11.cntl=10. The format for the control field is shown in Table 33-16

Table 33-16. Format for TR11 Control Field

Cntl1	Cntl0	Command
0	0	Normal operation
0	1	Testability write
1	0	Testability read
1	1	Flush

TR9, TR10 and TR11 are all read/write registers. The test registers should be written to for testability accesses only. Writing to the test registers during normal operation causes unpredictable behavior.

33.2.1.3.1. Direct BTB Access

The BTB contents are directly accessible, in a manner similar to the code/data caches. Note that the branch prediction mechanism should be disabled before doing any BTB testability access.

To write an entry into the BTB:

- Disable BTB entry allocation by setting TR12.NBP=1 (see next section)
- Write the tag address and history information in TR9
- Write the target address in TR10
- Perform a testability write to TR11 with the appropriate set address and entry fields. TR11.cntl is set to 01.

To read an entry from the BTB:

- Perform a testability read by writing to TR11 with the appropriate set address and entry fields. TR11.cntl is set to 10.
- Read the tag address and history information from TR9.
- Read the target address from TR10.

33.2.1.4. TEST PARITY CHECK (TR1)

A model specific register, TR1, the Parity Reversal Register (PRR), allows the parity check mechanism to be tested. Figure 33-4 shows the format of the PRR.

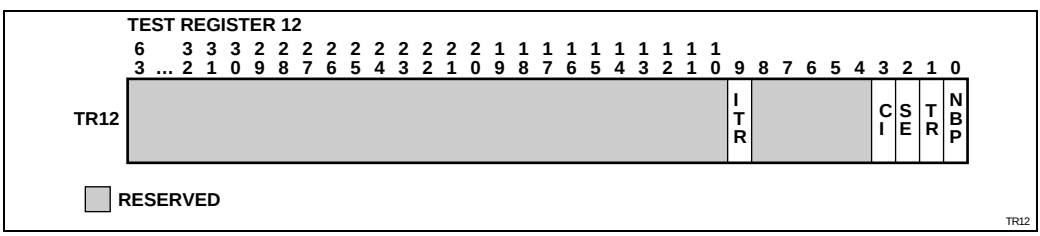


Figure 33-4. Test Register TR12

Table 33-17 lists each of the bits in the parity reversal register and their function.

Table 33-17. Parity Reversal Register Bit Definition

Bit Name	Description
PES	Parity Error Summary, set on any parity error
NS	0 = set PRR.PES, assert IERR#, and shutdown on parity error 1 = set PRR.PES, and assert IERR# on parity error
IT	code (instruction) cache tag
ID0	code cache data even bits 126, 124 ... 2,0
ID1	code cache data odd bits 127, 125 ... 3,1
ID2	code cache data even bits 254, 252 ... 130,128
ID3	code cache data odd bits 255, 253 ... 131, 129
ITT	code TLB tag
ITD	code TLB data
DT	data cache tag
DD	data cache data, use byte writes for individual access
DTT	data TLB tag
DTD	data TLB data
MC	microcode, reverse parity on read

Writing a one into bits 2-12 reverses the sense of the parity generation for any write into the corresponding array. This includes both normal cache replacements as well as testability writes and data writes. Parity is checked during both normal reads and testability reads.

To test parity error detection, software should write a one into the appropriate bit of the parity reversal register (PRR), perform a testability write into the array, and then perform a testability read. Upon successful detection of the parity error, the Pentium processor asserts the IERR# pin and may shutdown. Alternatively, after writing a one into the appropriate bit of the PRR, software may perform a normal write and read of the array by creating a cache miss and doing a read.

As an option, software may mask the shutdown by setting PRR.NS to 1 if the system is unable to recover from a shutdown. To determine if a parity error has occurred, software may read the parity error summary bit, PRR.PES. Hardware sets this bit on any parity error, and it remains set until cleared by software.

For the microcode, bad parity may be forced on a read by setting the PRR.MC bit to 1.

Bit 0 of TR1 is read/write. The remaining bits are write only. The test registers should be written to for testability accesses only. Writing to the test registers during normal operation causes unpredictable behavior.

33.3. NEW FEATURE CONTROL (TR12)

The new features of branch prediction, execution tracing, and instruction pairing in the Pentium processor can be selectively enabled or disabled through individual bits in test register TR12 (Figure 33-5). In addition, on chip caching can be disabled without affecting the PCD output to allow testing of a second level cache.

Table 33-18. New Feature Controls

Name	Position	Function
NBP	0	**No Branch Prediction** controls the allocation of new entries in the BTB. When TR12.NBP is clear, the code cache allocates entries in the BTB. When TR12.NBP is set, no new entry is allocated in the BTB, however, entries already in the BTB may continue to cause a BTB hit and result in the pipeline being reloaded from the predicted branch target. To completely disable branch prediction, first set TR12.NBP to 1 and then flush the entire BTB by loading CR3.
TR	1	**Execution 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 is still asserted, however the branch trace message special cycle is not driven by the Pentium® processor.
SE	2	**Single Pipe Execution** controls instruction pairing. When TR12.SE is cleared to zero, instructions are issued to both the u and v pipes contingent on pairing restrictions. When TR12.SE is set to one, the v pipe is disabled and instructions are issued only to the u pipe. Microcoded instructions are designed to utilize both pipes concurrently, independent of the state of TR12.SE. Note that all instructions requiring microcode are not pairable. The ability to utilize the v pipe in Probe Mode is also not affected by the state of TR12.SE.
CI	3	**Cache Inhibit** controls line fill behavior When TR12.CI is reset to 0, the on chip data and instruction caches operate normally. When TR12.CI is set to 1, all cache line fills are inhibited and all bus cycles due to cache misses are run as single transfer cycles (CACHE# is not asserted). Unlike CR0.CD, TR12.CI does not affect the state of the PCD output pin. This allows the first level cache to be disabled while the second level cache is still active and can be tested. Note that the contents of the instruction and data caches are not affected by the state of TR12.CI, e.g., they are not flushed. The second level cache test sequence should be: set TR12.CI to 1; flush the internal caches, run the second level cache tests.
	4-8	**Reserved**
ITR	9	**IO Trap Restart** enables proper interrupt pritoritization to support restarting IO accesses trapped by System Management Mode. Please refer to the “Component Operation” chapter in the *Pentium® Processor Data Book*
	10-63	**Reserved**

TR12.NBP, TR12.TR, TR12.SE, and TR12.CI are initialized to zero on reset. This register is write only and the reserved bits should be written with zeros.

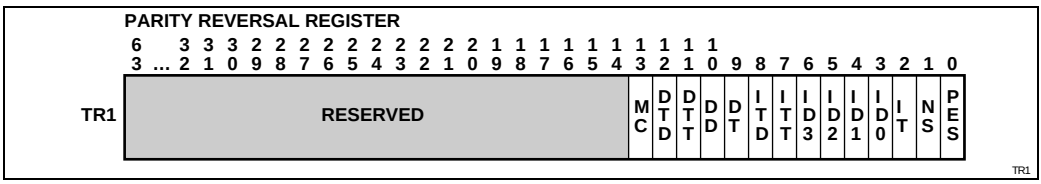


Figure 33-5. Parity Reversal Register (TR1)

33.4. PERFORMANCE MONITORING

The Pentium processor includes features to measure and monitor various parameters that contribute to the performance of the processor. This information can be then used for compiler and memory system tuning. For memory system tuning, it is possible to measure data and instruction cache hit rates, and time spent waiting for the external bus. The performance monitor allows compiler writers to gauge the effectiveness of instruction scheduling algorithms by measuring address generation interlocks and parallelism.

While the performance monitoring features that are provided by the Pentium processor are generally model specific and available only to privileged software, the Pentium processor also provides an *architectural* Time Stamp Counter that is available at the user. With this notable exception, the performance monitor features and the events they monitor are otherwise implementation dependent, and consequently, they are not considered part of the Pentium processor *architecture*. The performance monitor features are expected to change in future implementations. *It is essential that software abide by the usage restrictions for accessing model specific registers discussed in section 33.1.1 above.*

33.4.1. Performance Monitoring Feature Overview

Pentium processor performance monitoring features include:

Table 33-19. Architectural Performance Monitoring Features

RDTSC	Read Time Stamp Counter - a user level instruction to provide read access to a 64-bit free-running counter
CPUID (EDX.TSC)	Time Stamp Counter Feature Bit - Bit 4 of EDX is set to 1 to indicate that the processor implements the TSC and RDTSC instruction
CR4.TSD	Time Stamp Disable - A method for a supervisor program to disable user access to the time stamp counter in secure systems. When bit 2 of CR4 is set to 1, an attempt to execute the RDTSC instruction generates an general protection exception (#GP).

Table 33-20. Model Specific Performance Monitoring Features

CTR0, CTR1	Counter 0, Counter 1 - two programmable counters
CESR	Control and Event Select Register - programs CTR0, CTR1
TSC	Time Stamp Counter - provides read and write access to the architectural 64-bit counter in a manner that is model specific.
PM0/BP0, PM1/BP1	Event Monitoring Pins - These pins allow external hardware to monitor the activity in CTR0 and CTR1.

33.4.2. Time Stamp Counter - TSC

A dedicated, free-running, 64-bit time stamp counter is provided on-chip. Note that on the Pentium processor, this counter increments on every clock cycle, although it is not guaranteed that this will be true on future processors. As a time stamp counter, the RDTSC instruction reports values that are guaranteed to be unique and monotonically increasing. Portable software should not expect that the counter reports absolute time or clock counts. The user level RDTSC (Read Time Stamp Counter) instruction is provided to allow a program of any privilege level to sample its value. A bit in CR4, TSD (Time Stamp Disable) is provided to disable this instruction in secure environments. Supervisor mode programs may sample this counter using the RDMSR instruction or reset/preset this counter with a WRMSR instruction. The counter is cleared after reset.

While the user level RDTSC instruction and a corresponding 64-bit time stamp counter will be provided in all future Pentium CPU compatible processors, access to this counter via the RDMSR/WRMSR instructions is dependent upon the particular implementation.

33.4.3. Programmable Event Counters - CTR0, CTR1

Two programmable 40-bit counters CTR0 and CTR1 are provided. Each counter may be programmed to count any event from a pre-determined list of events. These events, which are described in the *Events* section of this chapter, are selected by programming the Control and Event Select Register (CESR). The counters are not affected by writes to CESR and must be cleared or pre-set when switching to a new event. The counters are undefined after RESET.

Associated with each counter is an event pin (PM1/BP1, PM0/BP0) which externally signals the occurrence of the selected event.

Note that neither the CTR0/CTR1 nor CESR are part of the processor state that is automatically saved and restored during a context switch. If it is desired to coordinate the use of the programmable counters in a multiprocessing system, it is the software's responsibility to share or restrict the use of these counters through a semaphore or other appropriate mechanism.

33.4.4. Control and Event Select Register - CESR

A 32-bit Control and Event Select Register (CESR) is used to control operation of the programmable counters and their associated pins. Figure 33-6 depicts the CESR. For each counter, the CESR contains a 6-bit Event Select field (ES), a Pin Control bit (PC), and a three bit control field (CC). It is not possible to selectively write a subset of the CESR. If only one event needs to be changed, the CESR must first be read, the appropriate bits modified, and all bits must be written back. At reset, all bits in the Control and Event Select Register are cleared.

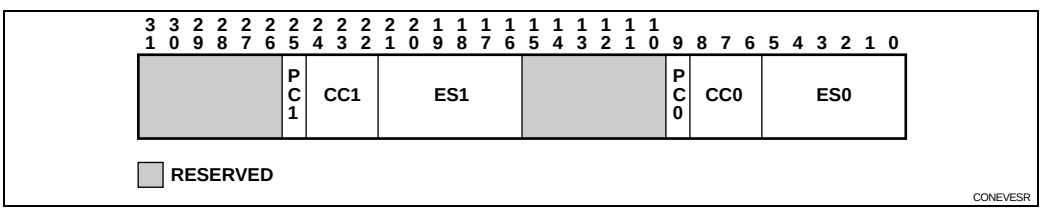


Figure 33-6. Control and Event Select Register

33.4.4.1. EVENT SELECT - ES0, ES1

Up to two independent events, may be monitored by placing the appropriate event code in the Event Select field. The events and codes are listed in the Events section of this chapter.

33.4.4.2. COUNTER CONTROL - CC0, CC1

A three bit field is used to control the operation of the counter. the highest order bit selects between counting events and counting clocks. The middle bit enables counting when the CPL=3. The low order bit enables counting when the CPL=0, 1 or 2.

CC	Meaning
000	Count Nothing (Disable Counter)
001	Count the selected Event while the CPL=0, 1 or 2
010	Count the selected Event while the CPL=3
011	Count the selected Event regardless of the CPL
100	Count Nothing (Disable Counter)
101	Count Clocks while the CPL=0, 1 or 2
110	Count Clocks while the CPL=3
111	Count Clocks regardless of the CPL

While a counter need not be stopped to sample its contents, it must be stopped and cleared or pre-set before switching to a new event.

33.4.4.3. PIN CONTROL - PC0, PC1

Associated with CTR0 and CTR1 are two pins, PM0 and PM1 (PM0/BP0, PM1/BP1), and two bits which control their operation, PC0 and PC1. These pins may be individually programmed by the PC0/PC1 bits in the CESR to indicate either that the associated counter has incremented or that it has overflowed. Note that the external signalling of the event on the pins will lag the internal event by a “few” clocks as the signals are latched and buffered.

PC	PM pin signals when the corresponding counter:
0	has incremented
1	has overflowed

When the pins are configured to signal that a counter has incremented, it should be noted that although the counters may increment by 1 or 2 in a single clock, the pins can only indicate that the event occurred. Moreover, since the internal clock frequency may be higher than the external clock frequency, a single external clock may correspond to multiple internal clocks.

A "count up to" function may be provided when the event pin is programmed to signal an overflow of the counter. Because the counters are 40 bits, a carry out of bit 39 indicates an overflow. A counter may be preset to a specific value less than $2^{40} - 1$. After the counter has been enabled and the prescribed number of events has transpired, the counter will overflow.

Approximately 5 clocks later, the overflow is indicated externally and appropriate action, such as signaling an interrupt, may then be taken.

When the performance monitor pins are configured to indicate when the performance monitor counter has incremented and an "occurrence event" is being counted, the associated PM pin is asserted (high) each time the event occurs. When a "duration event" is being counted the associated PM pin is asserted for the entire duration of the event. When the performance monitor pins are configured to indicate when the counter has overflowed, the associated PM pin is not asserted until the counter has overflowed.

The PM0/BP0, PM1/BP1 pins also serve to indicate breakpoint matches during in Circuit Emulation, during which time the counter increment or overflow function of these pins is not available. After RESET, the PM0/BP0, PM1/BP1 pins are configured for performance monitoring, however a hardware debugger may re-configure these pins to indicate breakpoint matches.

33.4.5. Events

Events may be considered to be of two types: those that count OCCURRENCES, and those that count DURATION. Each of the events listed below is classified accordingly.

Occurrences events are counted each time the event takes place. If the PM0 or PM1 pins are configured to indicate when a counter increments, they are asserted each clock the counter increments. Note that if an event can happen twice in one clock the counter increments by 2, however the PM0/1 pins are asserted only once.

For Duration events, the counter counts the total number of clocks that the condition is true. When configured to indicate when a counter increments, the PM0 and PM1 pins are asserted for the duration of the event.

Table 33-21 lists the events that can be counted, and their encodings for the Control and Event Select Register.

Table 33-21. Performance Monitoring Events

Encoding	Performance Monitoring Event	Occurrence or Duration?
000000	Data Read	OCCURRENCE
000001	Data Write	OCCURRENCE
101000	Data Read or Data Write	OCCURRENCE
000010	Data TLB Miss	OCCURRENCE
000011	Data Read Miss	OCCURRENCE
000100	Data Write Miss	OCCURRENCE
101001	Data Read Miss or Data Write Miss	OCCURRENCE
000101	Write (hit) to M or E state lines	OCCURRENCE
000110	Data Cache Lines Written Back	OCCURRENCE
000111	External Snoops	OCCURRENCE
001000	Data Cache Snoop Hits	OCCURRENCE
001001	Memory Accesses in Both Pipes	OCCURRENCE
001010	Bank Conflicts	OCCURRENCE
001011	Misaligned Data Memory or I/O References	OCCURRENCE
001100	Code Read	OCCURRENCE
001101	Code TLB Miss	OCCURRENCE
001110	Code Cache Miss	OCCURRENCE
001111	Any Segment Register Loaded	OCCURRENCE
010010	Branches	OCCURRENCE
010011	BTB Hits	OCCURRENCE
010100	Taken Branch or BTB Hit	OCCURRENCE
010101	Pipeline Flushes	OCCURRENCE
010110	Instruction Executed	OCCURRENCE
010111	Instructions Executed in the v-pipe e.g., parallelism/pairing	OCCURRENCE
011000	Clocks while a bus cycle is in progress (bus utilization)	DURATION
011001	Number of clocks stalled due to full write buffers	DURATION
011010	Pipeline stalled waiting for data memory read	DURATION
011011	Stall on write to an E or M state line	DURATION

Table 33-21. Performance Monitoring Events (Contd)

Encoding	Performance Monitoring Event	Occurrence or Duration?
011100	Locked Bus Cycle	OCCURRENCE
011101	I/O Read or Write Cycle	OCCURRENCE
011110	Non-cacheable memory reads	OCCURRENCE
011111	Pipeline stalled because of an address generation interlock	DURATION
100010	FLOPs	OCCURRENCE
100011	Breakpoint match on DR0 Register	OCCURRENCE
100100	Breakpoint match on DR1 Register	OCCURRENCE
100101	Breakpoint match on DR2 Register	OCCURRENCE
100110	Breakpoint match on DR3 Register	OCCURRENCE
100111	Hardware interrupts	OCCURRENCE

The following descriptions clarify the events. The event codes are provided in parenthesis.

Data Read (000000), Data Write (000001), Data Read or Data Write (101000):

These are memory data reads and/or writes (internal data cache hit and miss combined), I/O is not included. The individual component reads and writes for split cycles are counted individually. Data Memory Reads that are part of TLB miss processing are not included. These events may occur at a maximum of two per click.

Data TLB Miss (000010):

This event counts the number of misses to the data cache translation look-aside buffer.

Data Read Miss (000011), Data Write Miss (000100), Data Read Miss or Data Write Miss (101001):

These are memory read and/or write accesses that miss the internal data cache whether or not the access is cacheable or non-cacheable. Data accesses that are part of TLB miss processing are not included. Accesses directed to I/O space are not included.

Write (hit) to M or E state lines (000101):

This measures the number of write hits to exclusive or modified lines in the data cache. (These are the writes which may be held up if EWBE# is inactive.) This event may occur at a maximum of two per clock.

Data Cache Lines Written Back (000110):

This counts ALL Dirty lines that are written back, regardless of the cause. Replacements and internal and external snoops can all cause writeback and are counted.

External Snoops (000111), Data Cache Snoop Hits (001000):

The first event counts accepted external snoops whether they hit in the code cache or data cache or neither. Assertions of EADS# outside of the sampling interval are not counted. No internal snoops are counted. The second event applies to the data cache only. Snoop hits to a valid line in either the data cache, the data line fill buffer, or one of the write back buffers are all counted as hits.

Memory Accesses in Both Pipes (001001):

Data memory reads or writes which are paired in the pipeline. Note that these accesses are not necessarily run in parallel due to cache misses, bank conflicts, etc.

Bank Conflicts (001010):

These are the number of actual bank conflicts.

Misaligned Data Memory or I/O References (001011):

Memory or I/O reads or writes that are misaligned. A two or four byte access is misaligned when it crosses a four byte boundary; an eight byte access is misaligned when it crosses an eight byte boundary. Ten byte accesses are treated as two separate accesses of eight and two bytes each.

Code Read (001100), Code TLB Miss (001101), Code Cache Miss (001110):

Total instruction reads and reads that miss the code TLB or miss the internal code cache whether or not the read is cacheable or non-cacheable. Individual eight byte non-cacheable instruction reads are not counted.

Any Segment Register Loaded (001111):

Writes into any segment register in real or protected mode including the LDTR, GDTR, IDTR, and TR. Segment loads are caused by explicit segment register load instructions, far control transfers, and task switches. Far control transfers and task switches causing a privilege level change will signal this event twice. Note that interrupts and exceptions may initiate a far control transfer.

Branches (010010):

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. Both Taken and Not Taken Branches are counted. The pipe is not necessarily flushed. The number of branches actually executed is measured, not the number of predicted branches.

BTB Hits (010011):

Hits are counted only for those instructions that are actually executed.

Taken Branch or BTB Hit (010100):

This is a logical OR of taken branch and BTB hit. It represents an event that may cause a hit in the BTB. Specifically, it is either a candidate for a space in the BTB, or it is already in the BTB.

Pipeline Flushes (010101):

BTB Misses on taken branches, mis-predictions, exceptions, interrupts, and some segment descriptor loads all cause pipeline flushes.

Instructions Executed (010110):

Up to two per clock. Invocations of a fault handler are considered instructions.

Instructions Executed in the v-pipe e.g. parallelism/pairing (010111):

This event counts the number of instructions actually executed in the v-pipe. It indicates the number of instructions that were paired.

Clocks while a bus is in progress (bus utilization) (011000):

Including HLDA, AHOLD, BOFF# clocks.

Number of clocks stalled due to full write buffers (011001):

This event counts the number of clocks that the internal pipeline is stalled due to full write buffers. Full write buffers stall data memory read misses, data memory write misses, and data memory write hits to S state lines. Stalls on I/O accesses are not included.

Pipeline stalled waiting for data memory read (011010):

Data TLB Miss processing is also included. The pipeline stalls while a data memory read is in progress including attempts to read that are not bypassed while a line is being filled.

Locked Bus Cycle (011100):

LOCK prefix or LOCK instruction, Page Table Updates, and Descriptor Table Updates. Only the Read portion of the Locked Read-Modify-Write is counted. Split Locked cycles (SCYC active) count as two separate accesses. Cycles restarted due to BOFF# are not re-counted.

I/O Read or Write Cycle (011101):

Bus cycles directed to I/O space. Misaligned I/O accesses will generate two bus cycles. Bus cycles restarted due to BOFF# are not re-counted.

Non-cacheable memory reads (011110):

Non-cacheable instruction or data memory read bus cycles. Includes read cycles caused by TLB misses; does not include read cycles to I/O space. Cycles restarted due to BOFF# are not re-counted.

Pipeline stalled because of an address generation interlock (011111):

Number of address generation interlocks (AGIs). An AGI occurring in both the U and V pipelines in the same clock signals this event twice. An AGI occurs when the instruction in the execute stage of either of U or V pipelines is writing to either the index or base address register of an instruction in the D2 (address generation) stage of either the U or V pipelines.

FLOPs (100010);

Number of floating point adds, subtracts, multiplies, divides, remainders, and square roots. The transcendental instructions consist of multiple adds and multiplies and will signal this event multiple times. Instructions generating the divide by zero, negative square root, special operand, or stack exceptions will not be counted. Instructions generating all other floating point exceptions will be counted. As the integer multiply instructions share the floating point multiplier, they will signal this counter.

Breakpoint match on DR0 Register (100011),

Breakpoint match on DR1 Register (100100),

Breakpoint match on DR2 Register (100101),

Breakpoint match on DR3 Register (100110):

If programmed for one of these breakpoint match events, the performance monitor counters will be incremented in the event of a breakpoint match whether or not breakpoints are enabled. These events correspond to the signals driven on the BP[3:0] pins. Please refer to the Debugging chapter in Volume 3 of the *Pentium® Processor Family Developer's Manual.*

Hardware Interrupts (100111):

Number of taken INTR and NMI only.

34

Pentium® Processor (610\75) for Mobile Systems

CHAPTER 34
Pentium® Processor (610\75) for Mobile Systems

34.1. INTRODUCTION

Intel is now manufacturing a version of the Pentium processor family that is designed specifically for mobile systems, with a core frequency of 75 MHz and a bus frequency of 50 MHz. The Pentium processor (610\75) is provided in the TCP (Tape Carrier Package) and SPGA packages, and has all of the advanced features of the Pentium processor (735\90, 815\100).

The new Pentium processor (610\75) TCP package has several features which allow high-performance notebooks to be designed with the Pentium processor, including the following:

- TCP package dimensions are ideal for small form-factor designs.
- The TCP package has superior thermal resistance characteristics.
- 3.3V V_{CC} reduces power consumption by half (in both the TCP and SPGA packages).
- The SL Enhanced feature set, which was initially implemented in the Intel486™ CPU.

The Pentium processor family consists of the new Pentium processor at iCOMP rating 610\75 MHz, described in this document, the original Pentium processor (510\60, 567\66), and the New Device. The name "Pentium processor (610\75)" will be used in this document to refer to the Pentium processor at iCOMP rating 610\75 MHz. "Pentium Processor" will be used in this document to refer to the entire Pentium processor family in general.

The architecture and internal features of the Pentium processor (610\75) TCP and SPGA packages are identical to those of the New Device, although several features have been eliminated for the Pentium processor (610\75) TCP.

34.1.1. Pentium® Processor (610\75) SPGA Specifications and Differences from the TCP Package

This section provides references to the Pentium processor (610\75) SPGA specifications and describes the major differences between the Pentium processor (610\75) SPGA and TCP packages.

All Pentium processor (610\75) **SPGA specifications, with the exception of those listed in Part III, are identical to the Pentium processor (735\90, 815\100) specifications provided in the *Pentium® Processor Family Developer's Manual, Volume 1: Pentium® Processors*.**

The following features have been eliminated for the Pentium processor (610\75) TCP: the Upgrade feature, the Dual Processing (DP) feature, and the Master/Checker functional redundancy feature. Table 34-1 lists the corresponding pins which exist on the Pentium processor (610\75) SPGA but have been removed on the Pentium processor (610\75) TCP.

Table 34-1. SPGA Signals Removed in TCP

Signal	Function
ADSC#	Additional Address Status. This signal is mainly used for large or standalone L2 cache memory subsystem support required for high-performance desktop or server models.
BRDYC#	Additional Burst Ready. This signal is mainly used for large or standalone L2 cache memory subsystem support required for high-performance desktop or server models.
CPUTYP	CPU Type. This signal is used for dual processing systems.
D/P#	Dual/Primary processor identification. This signal is only used for an Upgrade processor.
FRCMC#	Functional Redundancy Checking. This signal is only used for error detection via processor redundancy, and requires two Pentium processors (master/checker).
PBGNT#	Private Bus Grant. This signal is only used for dual processing systems.
PBREQ#	Private Bus Request. This signal is used only for dual processing systems.
PHIT#	Private Hit. This signal is only used for dual processing systems.
PHITM#	Private Modified Hit. This signal is only used for dual processing systems.

The buffer model capacitance (C_p) and inductance (L_p) parameters differ between the TCP and SPGA packages. Also, the thermal parameters, T_{CASE} max and θ_{CA}, differ between the TCP and SPGA packages. For Pentium processor (610\75) SPGA values, refer to Chapters 24 and 26 of the *Pentium® Processor Family Developer's Manual, Volume 1: Pentium® Processor*.

In addition, the AC specifications for the TCP device are slightly different than those for the SPGA devices.

34.2. TCP PINOUT

34.2.1. TCP Pinout and Pin Descriptions

34.2.1.1. PENTIUM® PROCESSOR (610\75) TCP PINOUT

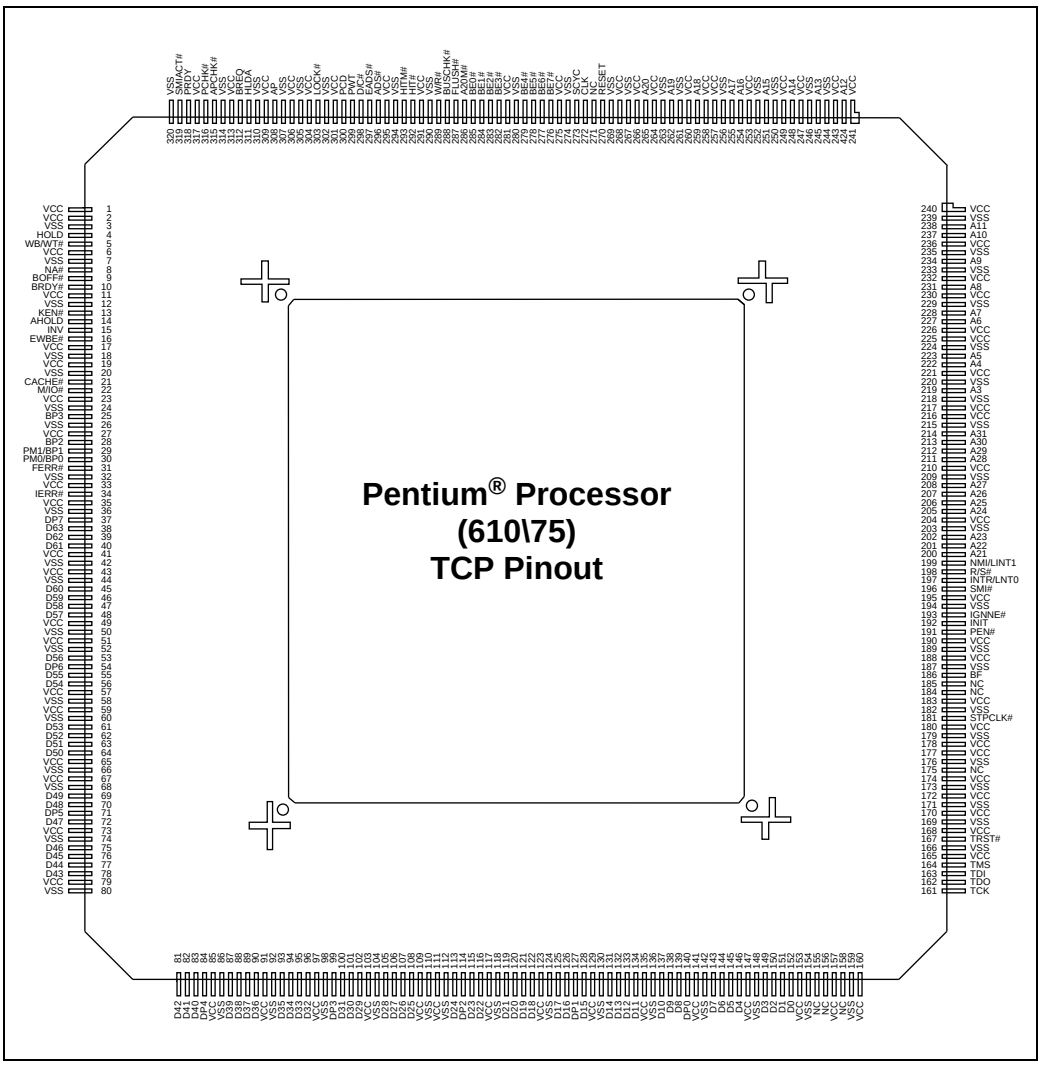


Figure 34-1. Pentium® Processor (610\75) TCP Pinout

34.2.1.2. PIN CROSS REFERENCE TABLE FOR PENTIUM® PROCESSOR (610\75) TCP

Table 34-2. TCP Pin Cross Reference by Pin Name

Address									
A3	219	A9	234	A15	251	A21	200	A27	208
A4	222	A10	237	A16	254	A22	201	A28	211
A5	223	A11	238	A17	255	A23	202	A29	212
A6	227	A12	242	A18	259	A24	205	A30	213
A7	228	A13	245	A19	262	A25	206	A31	214
A8	231	A14	248	A20	265	A26	207		
Data									
D0	152	D13	132	D26	107	D39	87	D52	62
D1	151	D14	131	D27	106	D40	83	D53	61
D2	150	D15	128	D28	105	D41	82	D54	56
D3	149	D16	126	D29	102	D42	81	D55	55
D4	146	D17	125	D30	101	D43	78	D56	53
D5	145	D18	122	D31	100	D44	77	D57	48
D6	144	D19	121	D32	96	D45	76	D58	47
D7	143	D20	120	D33	95	D46	75	D59	46
D8	139	D21	119	D34	94	D47	72	D60	45
D9	138	D22	116	D35	93	D48	70	D61	40
D10	137	D23	115	D36	90	D49	69	D62	39
D11	134	D24	113	D37	89	D50	64	D63	38
D12	133	D25	108	D38	88	D51	63		

Table 34-2. TCP Pin Cross Reference by Pin Name (Contd.)

Control							
A20M#	286	BREQ	312	HITM#	293	PM1/BP1	29
ADS#	296	BUSCHK#	288	HLDA	311	PRDY	318
AHOLD	14	CACHE#	21	HOLD	4	PWT	299
AP	308	D/C#	298	IERR#	34	R/S#	198
APCHK#	315	DP0	140	IGNNE#	193	RESET	270
BE0#	285	DP1	127	INIT	192	SCYC	273
BE1#	284	DP2	114	INTR/LINT0	197	SMI#	196
BE2#	283	DP3	99	INV	15	SMIACT#	319
BE3#	282	DP4	84	KEN#	13	TCK	161
BE4#	279	DP5	71	LOCK#	303	TDI	163
BE5#	278	DP6	54	M/IO#	22	TDO	162
BE6#	277	DP7	37	NA#	8	TMS	164
BE7#	276	EADS#	297	NMI/LINT1	199	TRST#	167
BOFF#	9	EWBE#	16	PCD	300	W/R#	289
BP2	28	FERR#	31	PCHK#	316	WB/WT#	5
BP3	25	FLUSH#	287	PEN#	191		
BRDY#	10	HIT#	292	PM0/BP0	30		
APIC				**Clock Control**			
PICCLK	155	PICD1 [APICEN]	158	BF	186	STPCLK#	181
PICD0 [DPEN#]	156			CLK	272		

Table 34-2. TCP Pin Cross Reference by Pin Name (Contd.)

V_{cc}								
1*	35	73	123	168*	190*	230	257*	295
2	41*	79	129	170*	195*	232*	258	301
6*	43	85	135	172*	204	236	260*	304*
11*	49*	91	141	174*	210	240*	264	306
17*	51	97	147	177*	216	241	266*	309*
19	57*	103	153*	178	217*	243*	268*	313
23	59	109	157*	180*	221	247	275	317*
27*	65*	111*	160	183*	225*	249*	281	
33*	67	117	165*	188*	226	253	291	

V_{ss}						
3	50	104	166	209	250	302
7	52	110	169	215	252	305
12	58	112	171	218	256	307
18	60	118	173	220	261	310
20	66	124	176	224	263	314
24	68	130	179	229	267	320
26	74	136	182	233	269	
32	80	142	187	235	274	
36	86	148	189	239	280	
42	92	154	194	244	290	
44	98	159	203	246	294	

NC			
175	184	185	271

NOTE:

*These V_{CC} pins are 3.3V supplies for the Pentium processor (610\75) TCP but will be lower voltage pins on future offerings of this microprocessor family. All other V_{CC} pins will remain at 3.3V.

34.2.1.3. POWER DISSIPATION REQUIREMENTS

Table 34-3. Power Dissipation Requirements for Thermal Solution Design

Parameter	Typical(1)	Max(2)	Unit	Notes
Active Power Dissipation	3	7.6	Watts	@ 75 Mhz
Stop Grant and Auto Halt Powerdown Power Dissipation		1.2	Watts	@ 75 MHz(3)
Stop Clock Power Dissipation	0.02	≤ .05	Watts	(4) (5)

NOTE:

1. This is the typical power dissipation in a system. This value was the average value measured in a system using a typical device at V_{CC} = 3.3V running typical applications. This value is highly dependent upon the specific system configuration.
2. Systems must be designed to thermally dissipate the maximum active power dissipation. It is determined using a worst-case instruction mix with V_{CC} = 3.3V. The use of nominal V_{CC} in this measurement takes into account the thermal time constant of the package.
3. Stop Grant/Auto Halt Powerdown Power Dissipation is determined by asserting the STPCLK# pin or executing the HALT instruction.
4. Stop Clock Power Dissipation is determined by asserting the STPCLK# pin and then removing the external CLK input.
5. Complete characterization of the specification was still in process at the time of print. Please contact Intel for the latest information. The final specification may be less than 50mW.

34.2.1.4. AC TIMING CHANGES

Table 34-4 lists the deviations for AC timing parameters with use of the TCP package. For all unlisted parameters, the AC timing values associated with SPGA packages should be used.

Table 34-4. AC Timing Changes for TCP Package

Symbol	Parameter	Min	Max	Unit	Figure	Notes
t_{6d}	C/C#, SCYC, LOCK# Valid Delay	0.9		nS		
t_{6e}	ADS# Valid Delay	0.8		nS		
t_{6f}	A3-A31, BE0-7# Valid Delay	0.7		nS		
t_{6g}	W/R# Valid Delay	0.5		nS		
t_{10b}	HITM# Valid Delay	0.5		nS		
t_{23}	BOFF# Hold Time	1.1		nS		
t_{29}	SMI# Hold Time	1.1		nS		
$t_{42}d$	Reset Configuration Signal BRDYC# (BRDY# on TCP) Hold Time, RESET driven synchronously	1.0		nS		To RESET falling edge (1), (27)

NOTE:

1. Not 100 percent tested. Guaranteed by design/characterization.

27. BRDYC# and BUSCHK# are used as reset configuration signals to select buffer size.

34.3. PENTIUM® PROCESSOR (610\75) TCP BUFFER MODEL PARAMETERS

Table 34-5. Input, Output and Bidirectional Buffer Model Parameters

Buffer Type	Transition	dV/dt (V/nsec)		Ro (Ohms)		Cp (pF)		Lp (nH)		Co/Cin (pF)	
		min	max	min	max	min	max	min	max	min	max
ER0	Rising					0.3	0.4	3.9	5.0	0.8	1.2
(input)	Falling					0.3	0.4	3.9	5.0	0.8	1.2
ER1	Rising					0.2	0.5	3.1	6.0	0.8	1.2
(input)	Falling					0.2	0.5	3.1	6.0	0.8	1.2
ED1	Rising	3/3.0	3.7/0.9	21.6	53.1	0.3	0.6	3.7	6.6	2.0	2.6
(output)	Falling	3/2.8	3.7/0.8	17.5	50.7	0.3	0.6	3.7	6.6	2.0	2.6
EB1	Rising	3/3.0	3.7/0.9	21.6	53.1	0.2	0.5	2.9	6.1	2.0	2.6
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	0.2	0.5	2.9	6.1	2.0	2.6
EB2	Rising	3/3.0	3.7/0.9	21.6	53.1	0.2	0.5	3.1	6.4	9.1	9.7
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	0.2	0.5	3.1	6.4	9.1	9.7
EB2A	Rising	3/2.4	3.7/0.9	10.1	22.4	0.2	0.5	3.1	6.4	9.1	9.7
(bidir)	Falling	3/2.4	3.7/0.9	9.0	21.2	0.2	0.5	3.1	6.4	9.1	9.7
EB3	Rising	3/3.0	3.7/0.9	21.6	53.1	0.2	0.4	3.2	4.1	3.3	3.9
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	0.2	0.4	3.2	4.1	3.3	3.9
EB4	Rising	3/3.0	3.7/0.9	21.6	53.1	0.3	0.4	4.0	4.1	5.0	7.0
(bidir)	Falling	3/2.8	3.7/0.8	17.5	50.7	0.3	0.4	4.0	4.1	5.0	7.0

Table 34-6. Input Buffer Model Parameters: D (Diodes)

Symbol	Parameter	D1	D2
IS	Saturation Current	1.4e-14A	2.78e-16A
N	Emission Coefficient	1.19	1.00
RS	Series Resistance	6.5 ohms	6.5 ohms
TT	Transit Time	3 ns	6 ns
VJ	PN Potential	0.983V	0.967V
CJ0	Zero Bias PN Capacitance	0.281 pF	0.365 pF
M	PN Grading Coefficient	0.385	0.376

34.4. PENTIUM® PROCESSOR (610\75) TCP MECHANICAL SPECIFICATIONS

Today's portable computers face the challenge of meeting desktop performance in an environment that is constrained by thermal, mechanical, and electrical design considerations. These considerations have driven the development and implementation of Intel's Tape Carrier Package (TCP). The Intel TCP package has been designed to offer a high pin count, low profile, reduced footprint package with uncompromised thermal and electrical performance. Intel continues to provide packaging solutions that meet our rigorous criteria for quality and performance, and this new entry into the Intel package portfolio is no exception.

Key features of the TCP package include: surface mount technology design, lead pitch of 0.25 mm, polyimide body size of 24 mm and polyimide up for pick&place handling. TCP components are shipped with the leads flat in slide carriers, and are designed to be excised and lead formed at the customer manufacturing site. Recommendations for the manufacture of this package are included in the New Device *Tape Carrier Package User's Guide*.

Figure 34-2 shows a cross-sectional view of the TCP package as mounted on the Printed Circuit Board. Figures 34-3 and 34-4 show the TCP as shipped in its slide carrier, and key dimensions of the carrier and package. Figure 34-5 shows a blow up detail of the package in cross-section. Figure 34-6 shows an enlarged view of the outer lead bond area of the package.

Tables 34-7 and 34-8 provide Pentium processor (610\75) Pentium processor (610\75) TCP package dimensions.

34.4.1. TCP Package Mechanical Diagrams

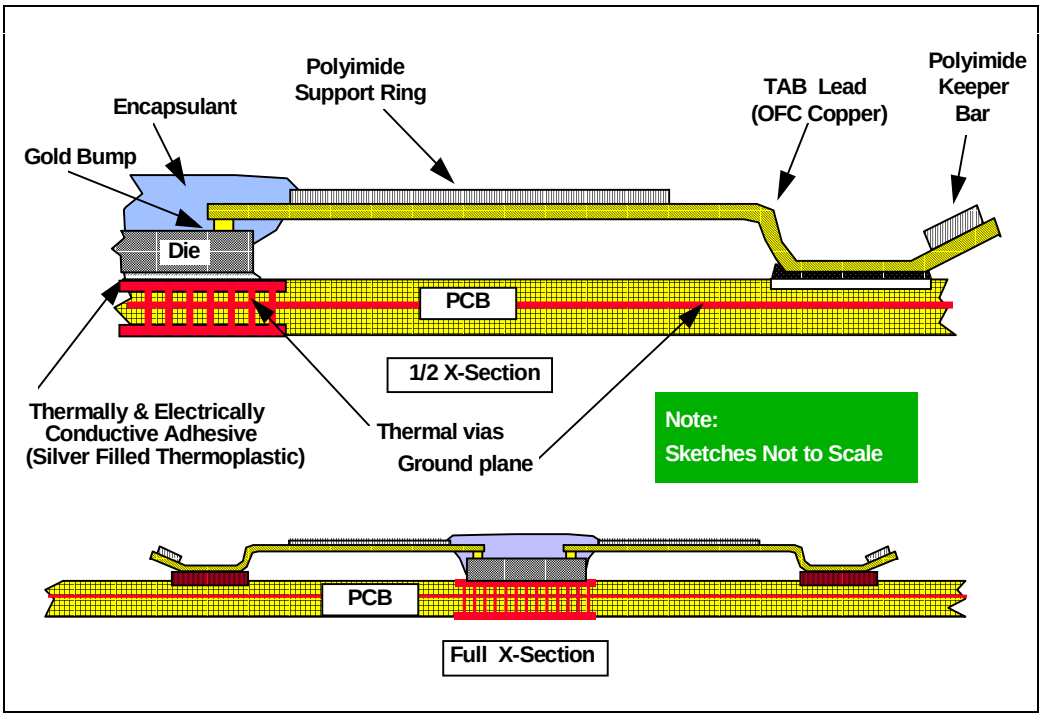


Figure 34-2. Cross-Sectional View of the Mounted TCP Package

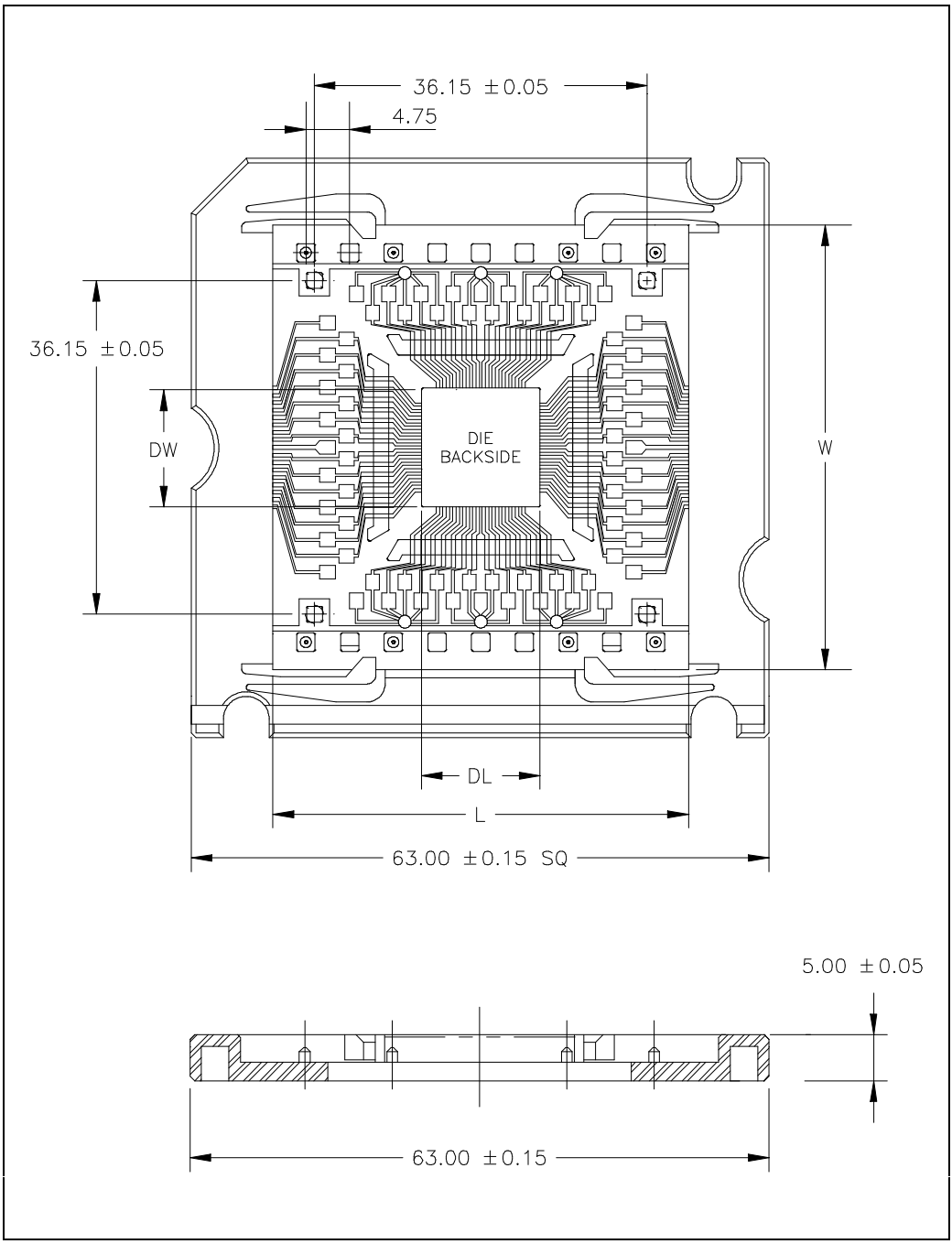


Figure 34-3. One TCP Site in Carrier (Bottom View of Die)

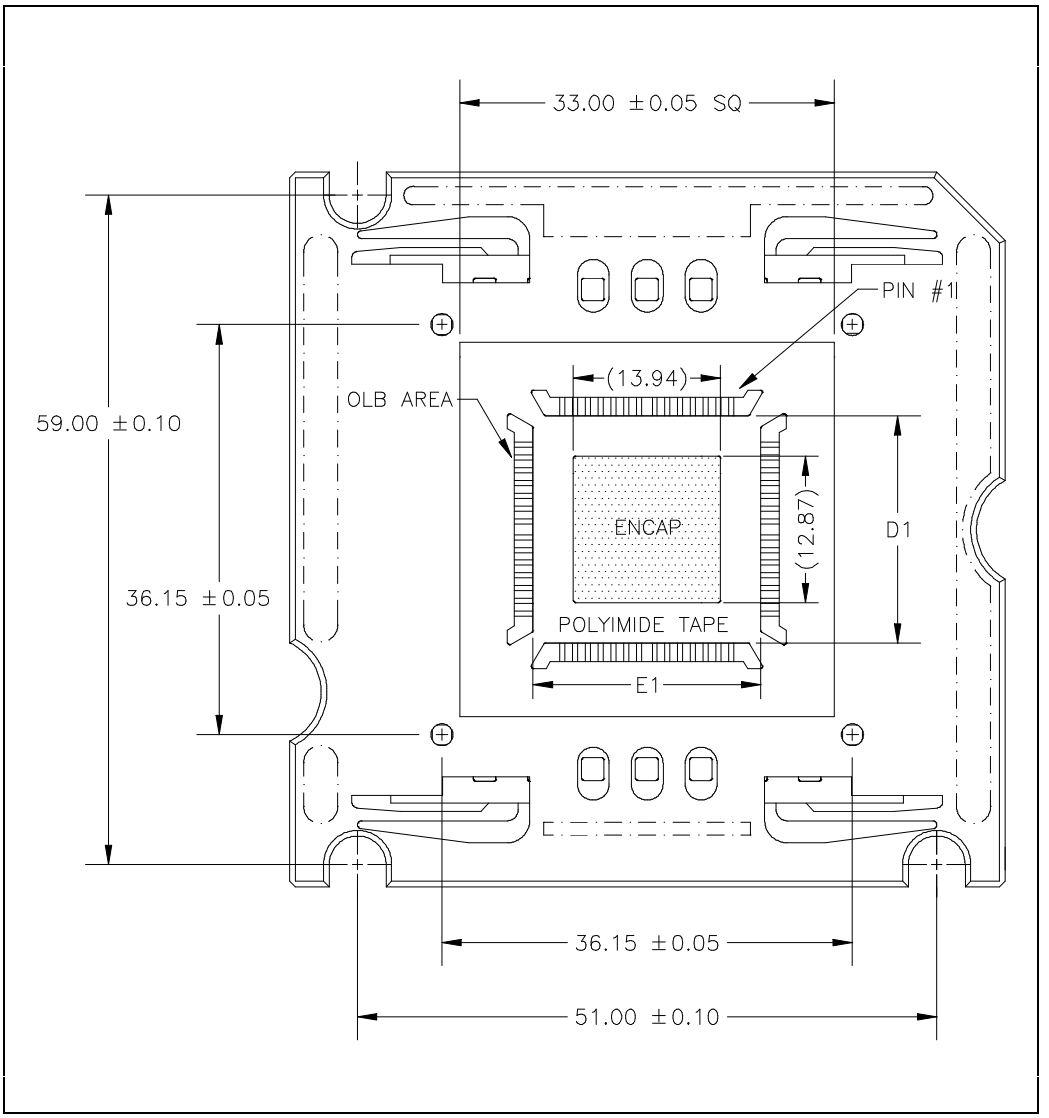


Figure 34-4. One TCP Site in Carrier (Top View of Die)

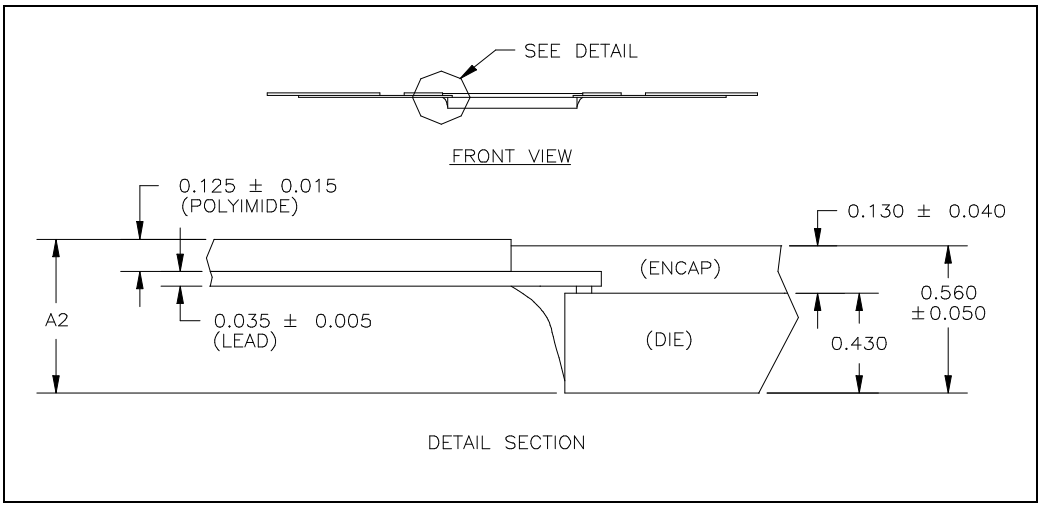


Figure 34-5. One TCP Site (Cross-Sectional Detail)

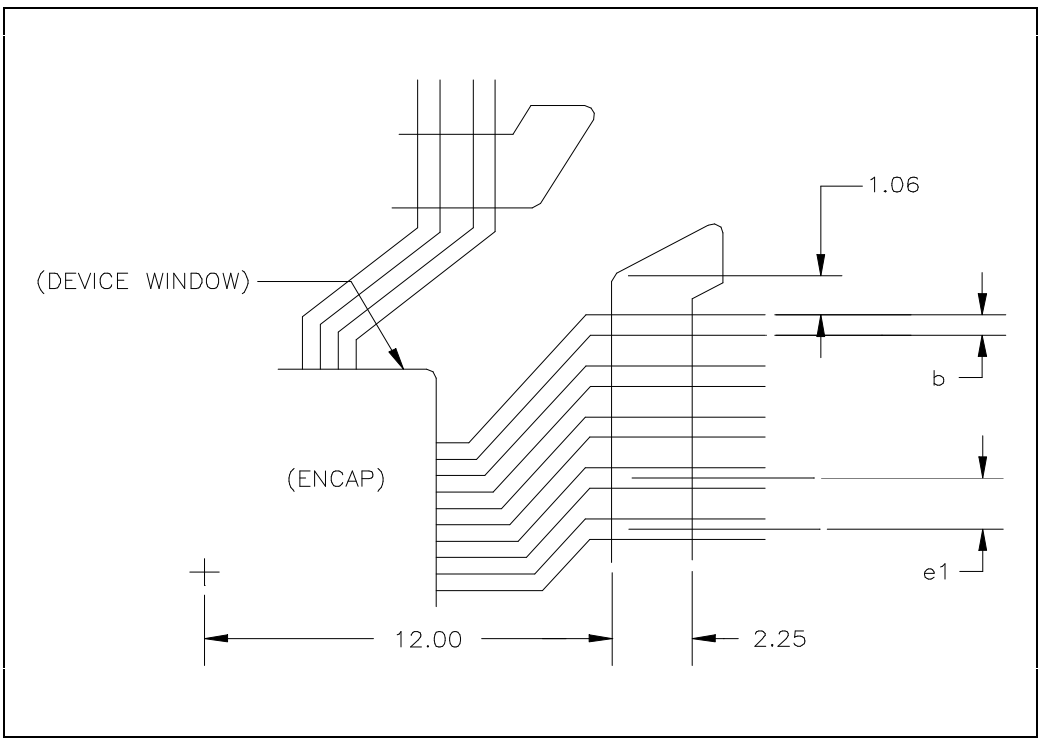


Figure 34-6. Outer Lead Bond (OLB) Window Detail

Table 34-7. TCP Key Dimensions

Symbol	Description	Dimension
N	Leadcount	320 leads
W	Tape Width	48.18 +/- 0.12
L	Site Length	(43.94) ref.
e1	Outer Lead Pitch	0.25 nom.
b	Outer Lead Width	0.10 +/- 0.01
D1,E1	Package Body Size	24.0 +/- 0.1
A2	Package Height	0.615 +/- 0.030
DL	Die Length	12.614 +/- 0.015
DW	Die Width	11.603 +/- 0.015

NOTES:

Be sure to refer to Pentium® Processor Specification Update for any changes.

Dimensions are in millimeters unless otherwise noted.

Dimensions in parentheses are for reference only.

Table 34-8. Mounted TCP Package Dimensions

Description	Dimension
Package Height	0.75 max.
Terminal Dimension	29.5 nom.
Package Weight	0.5 g max.

NOTE:

Dimensions are in millimeters unless otherwise noted.

Package terminal dimension (lead tip-to-lead tip) assumes the use of a keeper bar.

34.5. PENTIUM® PROCESSOR (610\75) TCP THERMAL SPECIFICATIONS

The Pentium processor (610\75) is specified for proper operation when the case temperature, T_{CASE}, (T_C) is within the specified range of 0 °C to 95 °C.

34.5.1. Measuring Thermal Values

To verify that the proper T_C (case temperature) is maintained for the Pentium processor (610\75), it should be measured at the center of the package top surface (encapsulant). To minimize any measurement errors, the following techniques are recommended:

- Use 36 gauge or finer diameter K, T, or J type thermocouples. Intel's 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 highly thermally conductive cements. Intel's laboratory testing was done by using Omega Bond (part number: OB-100).
- The thermocouple should be attached at a 90° angle as shown in Figure 34-7.

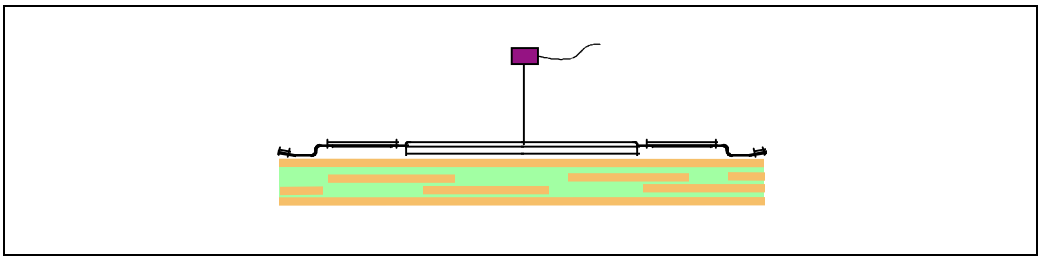


Figure 34-7. Technique for Measuring Case Temperature (T_C)

34.5.2. Thermal Equations

For the Pentium processor (610\75), an ambient temperature (T_A) is not specified directly. The only requirement is that the case temperature (T_C) is met. The ambient temperature can be calculated from the following equations:

$$T_J = T_C + P \times \theta_{JC}$$
$$T_A = T_J - P \times \theta_{JA}$$
$$T_A = T_C - (P \times \theta_{CA})$$
$$T_C = T_A + P \times [\theta_{JA} - \theta_{JC}]$$
$$\theta_{CA} = \theta_{JA} - \theta_{JC}$$

where,

T_A and T_C are ambient and case temperatures (°C)

θ_{CA} = Case-to-Ambient thermal resistance (°C/W)

θ_{JA} = Junction-to-Ambient thermal resistance (°C/W)

θ_{JC} = Junction-to-Case thermal resistance (°C/W)

P = maximum power consumption (Watts)

P (maximum power consumption) is specified in section 4.2.

34.5.3. TCP Thermal Characteristics

The primary heat transfer path from the die of the Tape Carrier Package (TCP) is through the back side of the die and into the PC board. There are two thermal paths traveling from the PC board to the ambient air. One is the spread of heat within the board and the dissipation of heat by the board to the ambient air. The other is the transfer of heat through the board and to the opposite side where thermal enhancements (e.g., heat sinks, pipes) are attached. To prevent the possibility of damaging the TCP component, the thermal enhancements should be attached to the opposite side of the TCP site — not directly mounted to the package surface.

34.5.4. PC Board Enhancements

Copper planes, thermal pads, and vias are design options that can be used to improve heat transfer from the PC board to the ambient air. Tables 34-9 and 34-10 present thermal resistance data for copper plane thickness and via effects. It should be noted that although thicker copper planes will reduce the θ_{ca} of a system without any thermal enhancements, they have less effect on the θ_{ca} of a system with thermal enhancements. However, placing vias under the die will reduce the θ_{ca} of a system with and without thermal enhancements.

Table 34-9. Thermal Resistance vs. Copper Plane Thickness with and without Enhancements

Copper Plane Thickness*	θ_{CA} (°C/W) No Enhancements	θ_{CA} (°C/W) With Heat Pipe
1 oz. Cu	18	8
3 oz. Cu	14	8

NOTES:

*225 vias underneath the die

(1 oz = 1.3 ml)

Table 34-10. Thermal Resistance vs. Thermal Vias Underneath the Die

No. of Vias Under the Die*	θ_{CA} (°C/W) No Enhancements
0	15
144	13

NOTE:

*3 oz. copper planes in test boards

34.5.4.1. STANDARD TEST BOARD CONFIGURATION

All Tape Carrier Package (TCP) thermal measurements provided in the following tables were taken with the component soldered to a 2" x 2" test board outline. This six-layer board contains 225 vias (underneath the die) in the die attach pad which are connected to two 3 oz. copper planes located at layers two and five. For the Pentium processor (610\75) TCP, the

vias in the die attach pad should be connected without thermal reliefs to the ground plane(s). The die is attached to the die attach pad using a thermally and electrically conductive adhesive. This test board was designed to optimize the heat spreading into the board and the heat transfer through to the opposite side of the board.

NOTE

Thermal resistance values should be used as guidelines only, and are highly system dependent. Final system verification should always refer to the case temperature specification.

Table 34-11. Pentium® Processor (610\75) TCP Package Thermal Resistance without Enhancements

	θ_{JC} (°C/W)	θ_{CA} (°C/W)
Thermal Resistance without Enhancements	0.8	13.9

Table 34-12. Pentium® Processor (610\75) TCP Package Thermal Resistance with Enhancements (Without Airflow)

Thermal Enhancements	θ_{CA} (°C/W)	Notes
Heat sink	11.7	1.2"×1.2"×.35"
Al Plate	8.7	4"×4"×.030"
Al Plate with Heat Pipe	7.8	0.3×1"×4"

Table 34-13. Pentium® Processor (610\75) TCP Package Thermal Resistance with Enhancements (with Airflow)

Thermal Enhancements	θ_{CA} (°C/W)	Notes
Heat sink with Fan @ 1.7 CFM	5.0	1.2"×1.2"×.35" HS 1"×1"×.4" Fan
Heat sink with Airflow @ 400 LFM	5.1	1.2"×1.2"×.35" HS
Heat sink with Airflow @ 600 LFM	4.3	1.2"×1.2"×.35" HS

HS = heat sink

LFM = Linear Feet/Minute

CFM = Cubic Feet/Minute

A

Supplemental Information

APPENDIX A
SUPPLEMENTAL INFORMATION

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 Family User's Manual* and is available with the appropriate non-disclosure agreements in place. Please contact Intel Corporation for details.

The *Supplement to the Pentium® Processor Family User's Manual* contains Intel confidential information on architecture extensions to the Pentium processor which are non-essential for standard applications. This includes low-level registers that provide access to such features as page size extensions, virtual mode extensions, testing and performance monitoring.

This information is specifically targeted at writers of the following types of software:

- operating system kernels
- virtual memory managers
- BIOS and CPU test software
- performance monitoring tools

If you are writing software that does not fall into one of these categories, this information is non-essential and all required programming details are contained in the publicly available *Pentium® Processor Family Developer's Manual* three volume set.

LITERATURE

For additional information on Intel products in the U.S. and Canada call Intel's Literature Center at (800) 548-4725.

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

To order literature outside of the U.S. and Canada contact your local international sales office.

CURRENT DATABOOKS

Product line databooks contain datasheets, application notes, article reprints, and other design information. Databooks can be ordered in the U.S. and Canada by calling TAB/McGraw-Hill at 1-800-822-8158; outside of the U.S. and Canada contact your local international sales office.

Title	Intel Order Number	ISBN
AUTOMOTIVE PRODUCTS	231792	N/A
EMBEDDED APPLICATIONS	270648	1-55512-242-6
EMBEDDED MICROCONTROLLERS	270646	1-55512-230-2
EMBEDDED MICROPROCESSORS	272396	1-55512-231-0
FLASH MEMORY (2 volume set)	210830	1-55512-232-9
Intel486™ MICROPROCESSORS and Related Products	241731	1-55512-1235-3
i960® PROCESSORS AND RELATED PRODUCTS	272084	1-55512-234-5
MILITARY AND SPECIAL PRODUCTS (2 volume set)	210461	N/A
NETWORKING	297360	1-55512-236-1
OEM BOARDS, SYSTEMS & SOFTWARE	280407	1-55512-237-X
PACKAGING	240800	1-55512-238-8
PENTIUM™ PROCESSORS and Related Products	241732	1-55512-239-6
PERIPHERAL COMPONENTS	296467	1-55512-240-X

A complete set of this information is available on CD-ROM through Intel's Data on Demand program, order number 240897. For information about Intel's Data on Demand ask for item number 240952.