AP-635

APPLICATION NOTE

Interfacing the Intel386™ EX Embedded Processor to Intel Boot Block Flash

December 1996

Order Number: 292191-001

CONTENTS

REVISION HISTORY

Number	Description
-001	Original Version

1.0 INTRODUCTION

The Intel386™ microprocessor family has gained a wide acceptance in the world of embedded applications. The Intel386 EX embedded processor is a very highly-integrated member of the Intel386 microprocessor family. There is a vast base of embedded applications developed for the 80C186 product family. When these applications require higher performance and address space, the Intel386 EX architecture provides a natural migration path to protect the code investment in the Intel architecture along with DOS compatibility. A DOS-based PC provides an easy, cost-effective means to develop, test, debug, and port embedded application code.

As embedded system designers take advantage of DOS capability in the PC platform, a revolutionary system architecture is required to meet space and power requirements:

- An architecture that is not bound by what has been done before with existing memory architecture, but free to meet the demanding requirements of embedded end-users.
- An architecture free to adopt and accommodate new technological advances in software and hardware, while protecting end-users initial base hardware investment.

Implementing this new system architecture requires an alternative to the traditional PC storage media such as ROM, DRAM, floppy disk, and hard disk. The solution is Intel's Boot Block flash memory (see architecture comparison in Figure 1).

APPLICATION	DATA MANIPULATION	CODE EXECUTION	FILE & CODE STORAGE
Desktops	DRAM	DRAM/ROM	FDD/HDD
Embedded	DRAM	FLASH	FLASH - Resident Disk - Flash Card - Flash Drive

Figure 1. Architecture Comparison

Intel Flash memory provides in-system program capabilities, along with selective block erase and program/erase automation which have gained wide acceptance in the embedded market. These features help cost-effective field updates and provide quick time-to-market solutions in most applications.

By combining flash memory with this new system architecture, completely new types of designss are now possible that fit in the palm of one's hand and replace or integrate many of the code or storage functions of other memory types. Flash memory can be used for storing eXecute-In-Place (XIP) code in the system's memory map while additionally functioning like a disk for file and program storage. Since this type of design features flash memory resident on the embedded system's motherboard and is typically arranged in an array, it is described as a Resident Flash Array (or RFA). To further differentiate the two tasks of an RFA, the file store task is called a Resident Flash Disk (RFD), while the XIP task is called Resident Flash for XIP (or RFX) code storage.

1.1 Why a New Memory Architecture?

The ideal embedded memory system is:

- Power Conscious (prolongs battery life and reduces heat)
- Dense (stores lots of code and data in a small amount of space but weighs very little)
- Updateable (allows in-system code enhancements)
- Fast (data is read and written quickly)
- Inexpensive (low cost-per-megabyte)
- Reliable (retains data when exposed to extreme temperature and mechanical shock)

While embedding the PC architecture, designers have grappled with how to construct memory systems that meet the above criteria. Embedded computing makes the system design even tougher with more stringent requirements for low power, low volume, less weight and harsh environments. The best combination available for embedded PC designs in their infancy was the same as used for the desktop: solid-state memory and magnetic storage, e.g., SRAMs, DRAMs plus magnetic hard disks. DRAMs are dense and inexpensive, yet slower than the processors they serve, and they are volatile. SRAMs, although fast enough to keep pace with processors, are relegated to caching schemes (compensating for DRAM's slowness) due to low density and high cost while also being volatile. Magnetic hard disks are dense, inexpensive on a cost-per-megabyte basis, and nonvolatile. However, they are also slow, power-hungry, susceptible to damage from

physical shock, and take up a sizable amount of volume.

Embedded computing designs cannot depend on hard drives as do desktop or portable PCs, due to size limitations. Furthermore, vitally important data such as credit card numbers or transactions, signatures, or patient monitoring information demands reliability of the highest order. The solution is Intel Flash memory.

1.2 The Flash Memory Alternative

High Density

Intel's ETOX™ IV flash memory cell is 30% smaller than equivalent DRAM cells; therefore, it will closely track DRAM density. Flash memory is more scaleable than DRAM because the flash storage cell is not sensitive to soft error failure; therefore, it can have a more simple cell structure. As density increases and process lithography continues to shrink, flash memory will pace, and ultimately overtake, the DRAM technology treadmill.

Updateable

ROMs and EPROMs may offer lower device costs, but overall system cost must be factored in, if servicing the customer or end-user is important to an OEM. Although ROMs and EPROMs are nonvolatile, changing the code within them is either very difficult (in the case of EPROMs), or entirely impossible (in the case of ROMs). Whole inventories of ROMs could be made obsolete in the event of a catastrophic bug, while an innovative design with flash memory could be updated in the factory or by end-users via networks, OEM Bulletin Board Systems, web sites, or other memory cards. Updating systems could actually become a second source of income for OEMs and Independent Software Vendors (ISVs), enhancing the quality of the product while increasing end-user satisfaction.

Power Conscious

Intel's flash memory provides a deep power-down mode, reducing power consumption to typically less than 0.2 µA. Typical read current is only 20 mA, while typical standby current (flash memory not being accessed with CE# high) is only 30 µA. Additionally, flash devices operating at as low as 2.7V are available for state-of-the-art low-power consumption designs.

Fast

Do not be misled by technology-to-technology speed comparisons. Architecting a system around flash memory bypasses the code/data bottleneck created by connecting slow mechanical serial memory (such as disks) to a high-performance, parallel bus system. For example, data seek time for a 1.8" magnetic hard disk is 20 ms, plus an 8 ms average rotational delay, while flash memory write time is less than 0.1 ms. At the chip level, read speeds for flash memory are about 70 ns. Therefore, either direct execution of code from flash memory or downloading to system RAM will dramatically enhance overall system performance.

Nonvolatile

Unlike DRAM or SRAM, flash memory requires no battery back-up. Furthermore, Intel's flash devices retain data well beyond the useful lifetime of most applications.

Rugged and Reliable

On average, today's hard disk drives can withstand up to 10 Gs of operating shock. Intel's flash memory can withstand as much as 1000 Gs. Flash components can operate beyond +70ºC while magnetic drives are limited to +55ºC. Intel's flash memory can be cycled 100,000 times per block or segment. By employing wear-leveling techniques, the cycling of a device can be minimized. For example, a 10-KB file written every 5 minutes, 24-hours a day to a 20-MB flash array takes 16 million hours, or 1826 years, before reaching the 100,000 cycle level.

1.3 Summary

Many embedded applications benefit from ROMed or XIP versions of code, particularly hand-held personal computers, vertical application pen-based clipboards, and industrial control and data accumulation equipment. These applications pose system design constraints requiring small form factor, low-power consumption, and rugged construction due to active mobile users or harsh environments. Exposure to shock, vibration, or temperature extremes is common, precluding the use of rotating media. Flash memory provides an excellent code storage choice for such system designs featuring thin TSOP packaging, low (deep power-down mode) or zero (capability to shut off power without losing data) power consumption, 1000 G shock resistance and extended temperature products. Additionally, flash memory provides remote or end-

user update capability allowing OEMs to service their products more efficiently and add new software features and applications after the sale.

Compared to RAMs and ROMs, the timing requirements for flash are slightly different. This application note explores those differences and provides a detailed analysis of the interface between the 28Fx00B5/BV Boot Block flash memory family and the Intel386 EX embedded processor. Also discussed are the issues involved with designing the required interface.

2.0 FLASH TIMING PARAMETERS

The timing parameters provided in Tables 1 through 4 are the most significant parameters involved with interfacing to Intel's Boot Block flash memory components.

Table 1. Read Timing Parameters (V_{CC} = 5V ± 5%)(1)

Timing Description	28F800BV-120	28F800BV-70 28F800B5-70	28F400BV-120	28F400BV-80 28F400B5-80	28F400BV-60 28F400B5-60
Address Valid to Data Valid, t_{AVQV} (max)	120 ns	70 ns	120 ns	80 ns	60 ns
CE# Valid to Data Valid, t_{ELQV} (max)	120 ns	70 ns	120 ns	80 ns	60 ns
OE# Valid to Output Delay, t_{GLQV} (max)	40 ns	30 ns	40 ns	40 ns	30 ns
OE# High to Data Invalid, t_{OH} (min)(3)	0	0	0	0	0
OE# High to Data Float, t_{GHQZ} (max)(3)	20 ns	20 ns	20 ns	20 ns	20 ns

Table 2. Read Timing Parameters (V_{CC} = 3.3V ± 10%)(2)

Timing Description	28F800BV-120	28F800BV-70 28F800B5-70	28F400BV-120	28F400BV-80 28F400B5-80	28F400BV-60 28F400B5-60
Address Valid to Data Valid, t_{AVQV} (max)	150 ns	120 ns	180 ns	150 ns	110 ns
CE# Valid to Data Valid, t_{ELQV} (max)	150 ns	120 ns	180 ns	150 ns	110 ns
OE# Valid to Output Delay, t_{GLQV} (max)	90 ns	65 ns	90 ns	90 ns	65 ns
OE# High to Data Invalid, t_{OH} (min)(3)	0	0	0	0	0
OE# High to Data Float, t_{GHQZ} (max)(3)	55 ns	45 ns	25 ns	25 ns	25 ns

Table 3. Write Timing Parameters (V_{CC} = 5V ± 5%)(1,4)

Timing Description	28F800BV-120	28F800BV-70 28F800B5-70	28F400BV-120	28F400BV-80 28F400B5-80	28F400BV-60 28F400B5-60
Address Valid to WE# High, t_{AVWH} (min)	50 ns	50 ns	50 ns	50 ns	50 ns
Data Valid to WE# High, t_{DVWH} (min)	50 ns	50 ns	50 ns	50 ns	50 ns
WE# Pulse Width, t_{WLWH} (min)	50 ns	50 ns	50 ns	50 ns	50 ns
Address Hold from WE# High, t_{WHAX} (min)	0	0	0	0	0

Table 4. Write Timing Parameters (V_{CC} = 3.3V ± 10%)(2,4)

Timing Description	28F800BV-120	28F800BV-70 28F800B5-70	28F400BV-120	28F400BV-80 28F400B5-80	28F400BV-60 28F400B5-60
Address Valid to WE# High, t_{AVWH} (min)	150 ns	90 ns	150 ns	120 ns	90 ns
Data Valid to WE# High, t_{DVWH} (min)	150 ns	90 ns	150 ns	120 ns	90 ns
WE# Pulse Width, t_{WLWH} (min)	150 ns	90 ns	150 ns	120 ns	90 ns
Address Hold from WE# High, t_{WHAX} (min)	0	0	0	0	0

NOTES:

1. The read and write timings provided in Tables 1 and 3 are taken from the respective component's datasheet and assume a commercial temperature range, 30 pF test load, and V_{CC} of 5V ± 5%.
2. The read and write timings provided in Tables 2 and 4 are taken from the respective component's datasheet and assume a commercial temperature range, 50 pF test load, and V_{CC} of 3.3V ± 10%.
3. Data hold times and data float times assume that OE# rises before CE#.
4. The write timing parameters assume WE#-controlled writes.

3.0 Intel386™ EX BUS SIGNALS

A successful bus transfer relies on the timing of address, data, and control signals from the CPU. This section of the application note defines the essential signals used in an Intel386 EX bus transfer.

3.1 Input Clock

The Intel386 EX operates at a frequency that is one-half of the frequency of its clock input (CLK2). The maximum operating frequencies available are 33 MHz or 25 MHz for 5V components and 25 MHz or 20 MHz for 3V components. These operating frequencies correspond to maximum CLK2 input frequencies of 66 MHz or 50 MHz for 5V components and 50 MHz or 40 MHz for 3V components. The zero wait-state memory transfer cycle is 4 CLK2 cycles. Each additional wait-state adds 2 CLK2 periods to the bus cycle.

3.2 Address Bus

The Address Bus (A[25:1]) provides address signals to an external memory device. A1 is the least significant bit in the address of a 16-bit data word. The address signals become valid after the rising edge of CLK2 at the beginning of a read or write cycle. They remain valid until the rising edge of CLK2 at the end of the bus cycle.

3.3 Data Bus

The Data Bus (D[15:0]) is a bi-directional bus which is used to transfer data to or from external memory. During read cycles, the external memory device must place valid input data on the data bus before the end of the transfer cycle. During write cycles, the CPU places output data on the bus one CLK2 cycle after the beginning of the bus cycle, and holds the data valid until the end of the bus cycle.

3.4 Control Signals

The three essential control signals that are generated by the CPU are the read indicator (RD#), the write indicator (WR#), and the chip select signals (UCS#, CS[6:0]#). These signals correspond directly to output enable (OE#), write enable (WE#), and chip enable (CE#) of the flash device.

3.4.1 READ INDICATOR

The read strobe signal (RD#) indicates to an external memory device that it should drive data information onto the data bus. It is asserted one CLK2 cycle after the beginning of a read cycle.

3.4.2 WRITE INDICATOR

During a write cycle, the write strobe signal (WR#) indicates to an external memory device that data on the data bus is available to be accessed for a write. It is asserted one CLK2 cycle after the beginning of a write cycle.

3.4.3 CHIP SELECT

Eight programmable chip select signals are available to enable external memory devices. There are seven general chip selects (CS[6:0]#) which are asserted whenever the address of a bus cycle is within the address limitations programmed by the user. The upper chip select (UCS#) is asserted whenever the address of a bus cycle is above the specified address limit. Only UCS# is active upon reset, so it must be used to select the boot device. For the analysis that follows, the chip select signals are collectively referenced by the signal name CS#.

3.5 Reset

The RESET signal is an active high input signal which causes the processor to terminate its current execution sequence. When the signal transitions low, the CPU initiates and completes an internal reset sequence and then fetches the first instruction vector from address 3FFFFF0H.

4.0 READ TIMING ANALYSIS

This section of the application note discusses the key timing issues involved with the Intel386 EX read cycle. Complete timing diagrams which include all essential microprocessor and flash device signals are shown in Figures 4 and 5, in Section 8.0.

4.1 Data Read Setup Time

The CPU requires that the data on the data bus be valid a minimum of t_{21} prior to the rising edge of CLK2 at the end of the read cycle. The time from the beginning of the read cycle to valid data appearing on the data bus is determined from the sum of CS# assertion delay (t_{34}) and data access time from CE# low (t_{ELQV}). The sum of these delays plus the required data setup time must not exceed four CLK2 periods for the transfer to be successful with no wait-states. Each additional wait-state adds two CLK2 periods to the above requirement.

4.2 Data Read Hold Time Verification

The CPU requires that the data on the data bus remain valid until RD# goes high. The address signals and CS# for the current read cycle remain valid at least until RD# goes high. Since the flash device holds the read data valid as long as the address, OE#, and CE# signals are valid, the data remains valid until RD# goes high, meeting the data hold time requirement.

4.3 Data Bus Float Time

The CPU requires that the data bus be in a high impedance state a maximum of one CLK2 (t_{50}) period from the negation of RD#. The maximum data float time (t_{GHQZ}) of a 28F400BV-60 component meets this requirement for an Intel386 EX operating at 25 MHz with V_{CC} = 5V ± 5%. However, a 28Fx00B5/BV component may violate this requirement for certain frequencies and operating voltages. This violation cannot be corrected by adding wait-states. If a violation occurs for particular operating conditions, a bi-directional buffer can be used to control access to the data bus. The buffer isolates the data bus of the CPU from the flash device and allows the CPU data bus to float to a high impedance state before the flash device has released its data bus. Refer to Tables 13 and 14 in Section 6.0 to determine if the buffer is necessary for a particular application.

The worst case timing relations for interfacing the Intel386 EX embedded processor with flash indicates bus contention on a memory read followed by a memory write. However, the Intel386 EX embedded processor specification t_{50} resolves this issue and guarantees no bus contention if the flash data float time $t_{GHQZ} \leq t_{50}$. The implications of this parameter are discussed in-depth in Appendix A.

4.4 System Reset

After the external RESET signal transitions to a low level, the CPU takes approximately 350 clock cycles before the first instruction vector is fetched from address 3FFFFF0H. Upon reset, the CPU's UCS# registers are configured for the maximum of 15 wait-states. The output delay from reset (t_{PHQV}) of all 28Fx00B5/BV components meets the reset requirements of the CPU by large margins for all operating frequencies.

4.5 Memory Requirements

Tables 5 and 6 indicate the read timing parameters required of a flash device to operate with a specific number of wait-states for the given maximum processor frequencies. These requirements are for an interface that does not use a data bus buffer as discussed in Section 7.2. Tables 7 and 8 indicate which 28Fx00B5/BV components meet the parameter requirements.

If required by timing, the addition of a data bus buffer eliminates all dependence on the t_{GHQZ} parameter. Operation for a specific number of wait-states at a given frequency is then limited by the other requirements shown in Tables 5 and 6.

Table 5. Read Timing Parameter Requirements (ns) for V_{CC} = 5V ± 5%(1)

	33 MHz				25 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVQV} (max)	32	62	93	123	49	89	129	169
t_{ELQV} (max)	29	59	90	120	43	83	123	163
t_{GLQV} (max)	20	50	81	111	31	71	111	151
t_{OH} (min)	0	0	0	0	0	0	0	0
t_{GHQZ} (max)(2)	15	15	15	15	20	20	20	20

Table 6. Read Timing Parameter Requirements (ns) for V_{CC} = 3.3V ± 10%(1)

	25 MHz				20 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVQV} (max)	39	79	119	159	55	105	155	205
t_{ELQV} (max)	38	78	118	158	47	97	147	197
t_{GLQV} (max)	21	61	101	141	32	82	132	182
t_{OH} (min)	0	0	0	0	0	0	0	0
t_{GHQZ} (max)(2)	20	20	20	20	25	25	25	25

Table 7. 28Fx00B5/BV Components Meeting Read Timing Requirements (V_{CC} = 5V ± 5%)(1,3)

	33 MHz				25 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVQV} (max)		E	BDE	ABCDE		BDE	ABCDE	ABCDE
t_{ELQV} (max)			BDE	ABCDE		BDE	ABCDE	ABCDE
t_{GLQV} (max)		ABCDE	ABCDE	ABCDE	BE	ABCDE	ABCDE	ABCDE
t_{OH} (min)	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE
t_{GHQZ} (max)(2)					ABCDE	ABCDE	ABCDE	ABCDE

Table 8. 28Fx00B5/BV Components Meeting Read Timing Requirements (V_{CC} = 3.3V ± 10%)(1,3)

	25 MHz				20 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVQV} (max)			E	ABDE			ABDE	ABCDE
t_{ELQV} (max)			E	ABDE			BE	ABCDE
t_{GLQV} (max)			ABCDE	ABCDE		BE	ABCDE	ABCDE
t_{OH} (min)	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE
t_{GHQZ} (max)(2)					CDE	CDE	CDE	CDE

NOTES:

1. Read timing parameter requirements are specified for an interface which uses RD# to control OE#. Timing parameters given assume zero propagation delay due to any interface logic.
2. Specification for t_{GHQZ} is required only if no data bus buffer is used. If buffer is used, requirement is met by all 28Fx00B5/BV components for all frequencies shown.
3. A = 28F800BV-120, B = 28F800B5/BV-70, C = 28F400BV-120, D = 28F400B5/BV-80, and E = 28F400B5/BV-60.

5.0 WRITE TIMING ANALYSIS

This section of the application note discusses the key timing issues involved with the Intel386 EX embedded processor write cycle. A complete timing diagram which includes all essential microprocessor and flash device signals is shown in Figure 6, in Section 9.0.

5.1 Data Write Setup Time

The CPU places valid data on the data bus a maximum of one CLK2 period plus Write Data Valid Delay (t_{12}) from the beginning of a write cycle. This data must be valid for a time of t_{DVWH} prior to the negation of WE#, which may occur as early as 4 ns after the rising edge of CLK2 at the end of the write cycle. Adding a wait-state increases the setup time by two CLK2 periods.

5.2 Write Address Hold Time Verification

The address placed on the address bus by the CPU must remain valid until the WE# signal of the flash device is negated. The CPU meets this requirement at any frequency within its specified operating limits.

5.3 WE# Pulse Width

The WE# signal must be held low for a minimum duration of t_{WLWH}. The amount of time that the CPU asserts WR# is three CLK2 cycles minus the difference between the maximum WR# valid delay (t_{10a} max) and minimum RD# valid delay (t_{10a} min). Adding a wait-state increases the WR# pulse width by two CLK2 periods.

5.4 Memory Requirements

Tables 9 and 10 indicate the write timing parameters required of a flash device to operate with a specific number of wait-states for the given maximum processor frequencies. Tables 11 and 12 indicate which 28Fx00B5/BV components meet the parameter requirements.

Table 9. Write Timing Parameter Requirements (ns) for V_{CC} = 5V ± 5%(1)

	33 MHz				25 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVWH} (min)	35	65	96	126	50	90	130	170
t_{DVWH} (min)	26	56	87	117	41	81	121	161
t_{WLWH} (min)	26	56	87	117	41	81	121	161
t_{WHAX} (min)	0	0	0	0	0	0	0	0

Table 10. Write Timing Parameter Requirements (ns) for V_{CC} = 3.3V ± 10%(1)

	25 MHz				20 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVWH} (min)	50	90	130	170	65	115	165	215
t_{DVWH} (min)	33	73	113	153	45	95	145	195
t_{WLWH} (min)	33	73	113	153	45	95	145	195
t_{WHAX} (min)	0	0	0	0	0	0	0	0

Table 11. 28Fx00B5/BV Components Meeting Write Timing Requirements (V_{CC} = 5V ± 5%)(1,2)

	33 MHz				25 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVWH} (min)		ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE
t_{DVWH} (min)		ABCDE	ABCDE	ABCDE		ABCDE	ABCDE	ABCDE
t_{WLWH} (min)		ABCDE	ABCDE	ABCDE		ABCDE	ABCDE	ABCDE
t_{WHAX} (min)	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE

Table 12. 28Fx00B5/BV Components Meeting Write Timing Requirements (V_{CC} = 3.3V ± 10%)(1,2)

	25 MHz				20 MHz			
Timing Parameter	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**	**0 Waits**	**1 Wait**	**2 Waits**	**3 Waits**
t_{AVWH} (min)		BE	BDE	ABCDE		BE	ABCDE	ABCDE
t_{DVWH} (min)			BE	ABCDE		BE	BDE	ABCDE
t_{WLWH} (min)			BE	ABCDE		BE	BDE	ABCDE
t_{WHAX} (min)	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE

NOTES:

1. Write timing parameter requirements are specified for an interface which uses WR# to control WE#. Timing parameters given assume zero propagation delay due to any interface logic.
2. A = 28F800BV-120, B = 28F800B5/BV-70, C = 28F400BV-120, D = 28F400B5/BV-80, and E = 28F400B5/BV-60.

6.0 MAXIMUM OPERABLE FREQUENCY

The maximum operable frequency of a system is determined from the analysis of read and write timing parameters discussed in Sections 4.0 and 5.0. Tables 13 and 14 indicate the maximum operable frequency of a system using a given V_{CC}, Intel386 EX microprocessor, 28Fx00B5/BV component, and number of wait-states. Table 13 corresponds to an interface without a data bus buffer while Table 14 corresponds to an interface that employs a bi-directional data bus buffer as discussed in Section 4.3.

Table 13. Maximum Operable Frequency (MHz) Without Data Bus Buffer(1)

V_{CC} = 5V ± 5%						
i386™ CPU	**Wait-states(2)**	**28F800BV-120**	**28F800BV-70 28F800B5-70**	**28F400BV-120**	**28F400BV-80 28F400B5-80**	**28F400BV-60 28F400B5-60**
EXTC-33	0	13.25	19.80	13.25	18.02	21.74
EXTC-33	1	19.87	25.00	19.87	25.00	25.00
EXTC-25	0	12.74	18.69	12.74	17.09	20.62
EXTC-25	1	19.11	25.00	19.11	25.00	25.00
V_{CC} = 3.3V ± 10%						
i386™ CPU	**Wait-states(2)**	**28F800BV-120**	**28F800BV-70 28F800B5-70**	**28F400BV-120**	**28F400BV-80 28F400B5-80**	**28F400BV-60 28F400B5-60**
EXTB-25	0	8.47	11.11	8.47	10.20	12.82
EXTB-25	1	9.09	11.11	13.51	15.63	19.74
EXTB-20	0	8.33	11.11	8.33	9.85	12.27
EXTB-20	1	9.09	11.11	12.88	14.78	18.40

NOTES:

1. All specifications are for the interface shown in Figure 2 in Section 7.4. Operating frequencies given assume zero propagation delay due to any interface logic.
2. Adding waits states beyond what is shown in the table does not increase maximum operable frequency because of the t_{GHQZ} violation discussed in Section 4.3.

Table 14. Maximum Operable Frequency (MHz) with Data Bus Buffer

V_{CC} = 5V ± 5%						
i386™ CPU	**Wait-states**	**28F800BV-120**	**28F800BV-70 28F800B5-70**	**28F400BV-120**	**28F400BV-80 28F400B5-80**	**28F400BV-60 28F400B5-60**
EXTC-33	0	13.25	19.80	13.25	18.02	21.74
EXTC-33	1	19.87	29.70	19.87	27.03	32.97
EXTC-33	2	26.49	33.00	26.49	33.00	33.00
EXTC-33	3	33.00	33.00	33.00	33.00	33.00
EXTC-25	0	12.74	18.69	12.74	17.09	20.62
EXTC-25	1	19.11	25.00	19.11	25.00	25.00
EXTC-25	2	25.00	25.00	25.00	25.00	25.00
EXTC-25	3	25.00	25.00	25.00	25.00	25.00
V_{CC} = 3.3V ± 10%						
i386™ CPU	**Wait-states**	**28F800BV-120**	**28F800BV-70 28F800B5-70**	**28F400BV-120**	**28F400BV-80 28F400B5-80**	**28F400BV-60 28F400B5-60**
EXTB-25	0	8.47	12.35	8.47	10.20	12.82
EXTB-25	1	14.12	18.52	13.51	15.63	19.74
EXTB-25	2	19.77	24.69	18.02	20.83	25.00
EXTB-25	3	25.00	25.00	22.52	25.00	25.00
EXTB-20	0	8.33	11.56	8.33	9.85	12.27
EXTB-20	1	13.89	17.34	12.88	14.78	18.40
EXTB-20	2	19.44	20.00	17.17	19.70	20.00
EXTB-20	3	20.00	20.00	20.00	20.00	20.00

NOTE:
All specifications are for the interface shown in Figure 3 in Section 7.4. Operating frequencies given assume zero propagation delay due to interface logic.

7.0 INTERFACE LOGIC

This section of the application note describes the logic required to interface an Intel386 EX embedded microprocessor to 28Fx00B5/BV flash memory components. The physical means of implementing any required logic is not specified. This decision is left to the system designer.

7.1 Enable Signals

The timing of the CPU control signal outputs does not generate any setup or hold time violations relative to the flash device enable signals. Therefore, the CS#, RD#, and WR# signals from the CPU may be directly connected to the CE#, OE#, and WE# signals of the flash device without producing any timing conflicts.

7.2 Data Buffer

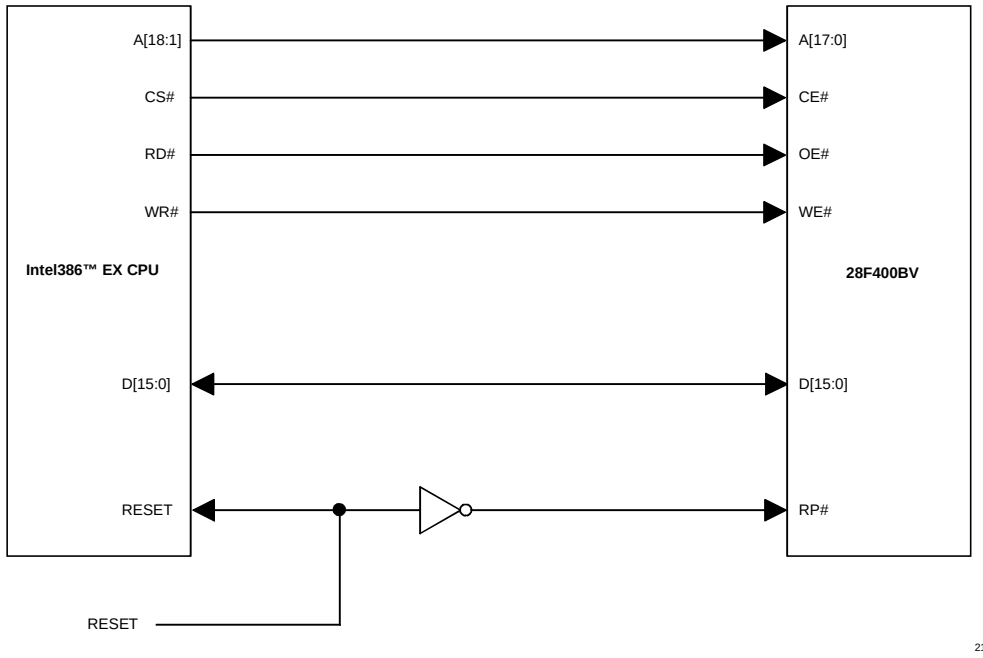
As discussed in Section 4.3, there is a data bus contention problem during the transition from a read to a write cycle with some 28Fx00B5/BV components under certain operating conditions. This problem may be eliminated by using a bi-directional buffer to control the ability of the flash device to drive the data bus. This buffer is enabled when either RD# or WR# is asserted, and the direction of the buffer is controlled by RD#. Refer to Tables 13 and 14 to determine if the buffer is necessary for a particular application.

7.3 Reset Interface

As discussed in Section 4.4, the reset time of the CPU is much longer than the reset time of a 28Fx00B5/BV component. However, the CPU reset signal is active high, and the flash device reset signal is active low. This requires that the external reset signal be inverted to interface to RP#.

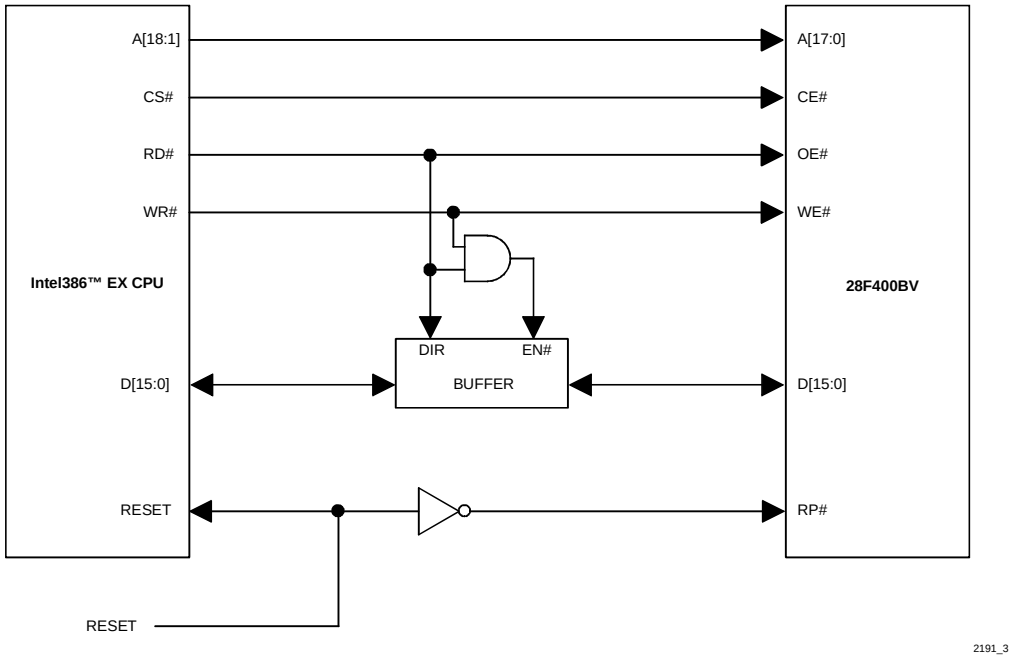
7.4 Interface Diagrams

Figure 2 shows the interface between the Intel386 EX embedded microprocessor and a 28F400BV flash device. This diagram shows the interface that does not include the use of a data bus buffer. Figure 3 shows the same interface with the addition of a bi-directional buffer to control the access of the data bus. Refer to Tables 13 and 14 to determine if the buffer is required for a particular application.



NOTE:
The physical means of implementing external logic is not specified. This decision is left to the system designer.

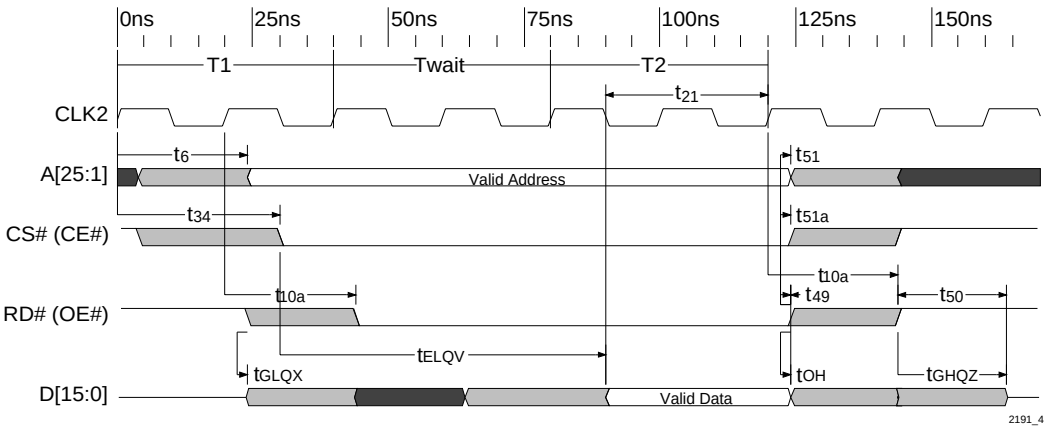
Figure 2. Interface Diagram (Without Data Bus Buffer)



NOTE:
The physical means of implementing external logic is not specified. This decision is left to the system designer.

Figure 3. Interface Diagram (with Data Bus Buffer)

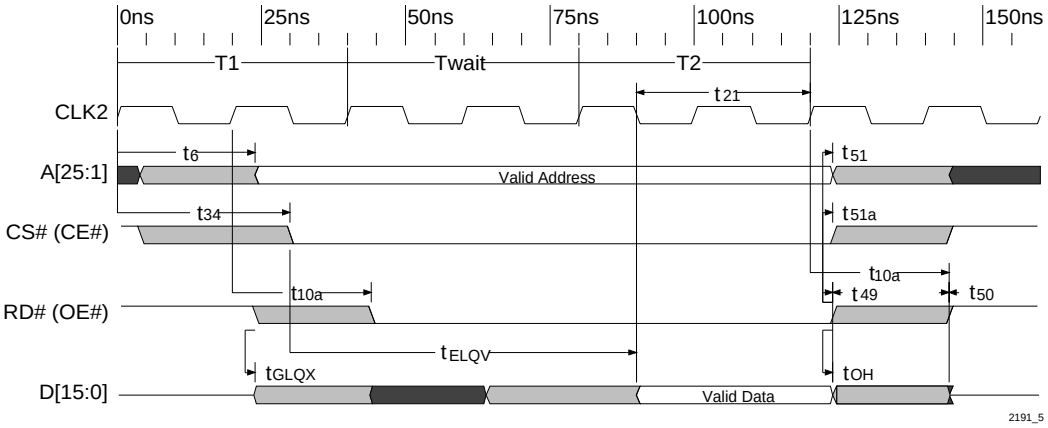
8.0 READ TIMING DIAGRAMS



NOTE:

This diagram shows a typical read cycle using one wait-state. Timing parameters used are from an Intel386™ EXTC-25 CPU and a 28F400BV-60 with a 50 MHz CLK2 and V_{CC} = 5V ± 5%. This diagram corresponds to the interface shown in Figure 2.

Figure 4. Read Cycle (Without Data Bus Buffer)

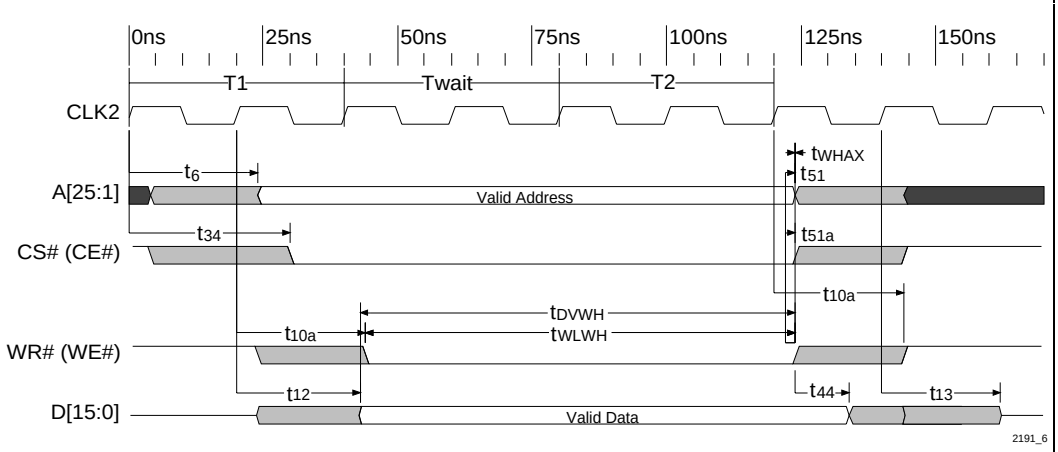


NOTE:

This diagram shows a typical read cycle using one wait-state. Timing parameters used are from an Intel386™ EXTC-25 CPU and a 28F400BV-60 with a 50 MHz CLK2 and V_{CC} = 5V ± 5%. This diagram corresponds to the interface shown in Figure 3.

Figure 5. Read Cycle (with Data Bus Buffer)

9.0 WRITE TIMING DIAGRAM



NOTES:

This diagram shows a typical write cycle using one wait-state. Timing parameters used are from an Intel386™ EXTC-25 CPU and a 28F400BV-60 with a 50 MHz CLK2 and V_{CC} = 5V ± 5%.

Figure 6. Write Cycle

10.0 CONCLUSION

This application note describes the logic necessary to interface the Intel386 EX embedded microprocessor to Intel Flash Memory. Any logic that is required is due to a mismatched timing between the two devices. Since most design optimizations reduce glue logic, future interface designs may not require the interface described.

If you are designing with Intel Flash Memory, or the Intel386 EX embedded microprocessor, contact your local Intel representative for the latest information.

APPENDIX A
UNDERSTANDING THE Intel386™ EX t_{50} EMBEDDED MICROPROCESSOR TIMING PARAMETER

Introduction

The worst case timing relations for interfacing the Intel386 EX embedded processor with flash indicates bus contention on a memory read followed by a memory write. However, the Intel386 EX embedded processor specification t_{50} resolves this issue and guarantees no bus contention if the flash data float time $t_{GHQZ} \leq t_{50}$.

Read Followed by Write Timing Analysis

To avoid bus contention, the flash memory must stop driving "read data" on the bus before the Intel386 EX embedded processor starts driving "write data" on the bus. Figure A1 shows the timing for a read followed by a write, along with the signals that must be evaluated to determine if bus contention exists. A full timing diagram with an incorrect analysis is presented in Figure A2.

Controller AC timings are specified with a minimum and a maximum value. For example, at 5V V_{CC}, 25 MHz the RD# valid delay, t_{10a}, is specified as 4 ns minimum and 22 ns maximum. These minimum and maximum specifications are provided to account for variations in temperature, voltage, and processing.

The contention occurs when RD# is deasserted late, e.g., 22 ns maximum delay, and D[15:0] is driven early, e.g., 4 ns minimum delay. However, spec t_{50} guarantees by design that there is no contention. This is guaranteed because if RD# goes high late, D[15:0] will also be driven late since the logic delays of t_{10a} and t_{12} track each other across temperature, voltage, and processing variations. Since t_{10a} and t_{12} track each other they are separated by a CLK2 (each are referenced to a rising clock edge, separated by a CLK2—20 ns at 25 MHz). Therefore the data float time of the flash memory must be less than CLK2 as spec'd by t_{50}.

The following example shows the typical calculation for data float without using t_{50} which yields an unrealistic float time requirement for flash memory. The example then shows the correct calculation using the Intel386 EX embedded processor specification of t_{50}.

At 5V V_{CC}:

t_{10a} = [4,22] = RD#, WR# Valid Delay [min, max]

t_{12} = [4,23] = D[15:0] Write Data Valid Delay

t_{GHQZ} = 20 ns (Flash Data Float Time, 28F400BV)

Incorrect Analysis without t_{50}:

t_{GHQZ} = CLK2 - t_{10a} (max) + t_{12} (min)

t_{GHQZ} = 20 ns - 22 + 4 = 2 ns.

Since the flash memory data float time, t_{GHQZ}, is longer than 2 ns, this would indicate a transceiver is necessary to avoid bus contention. However, as previously described, this analysis is incorrect because t_{50} guarantees no contention if $t_{GHQZ} \leq t_{50}$.

Correct Analysis using t_{50}:

$t_{GHQZ} \leq t_{50}$ = CLK2 = 20 ns @ 25 MHz.

Conclusion

The Intel386 EX embedded processor specification t_{50} resolves this issue and guarantees no bus contention if $t_{GHQZ} \leq t_{50}$. If this is the case, the Intel386 EX embedded processor and Intel Flash can be interfaced with no glue logic.

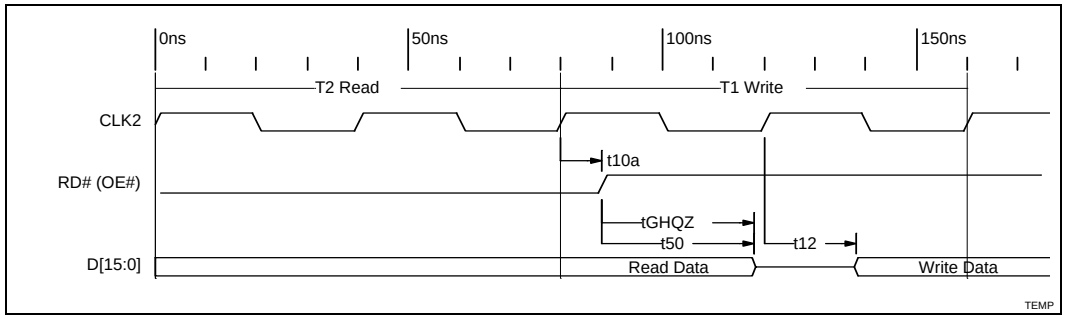


Figure A1. Correct Timing Analysis for Intel386™ EX Embedded Microprocessor Interface to Intel Flash

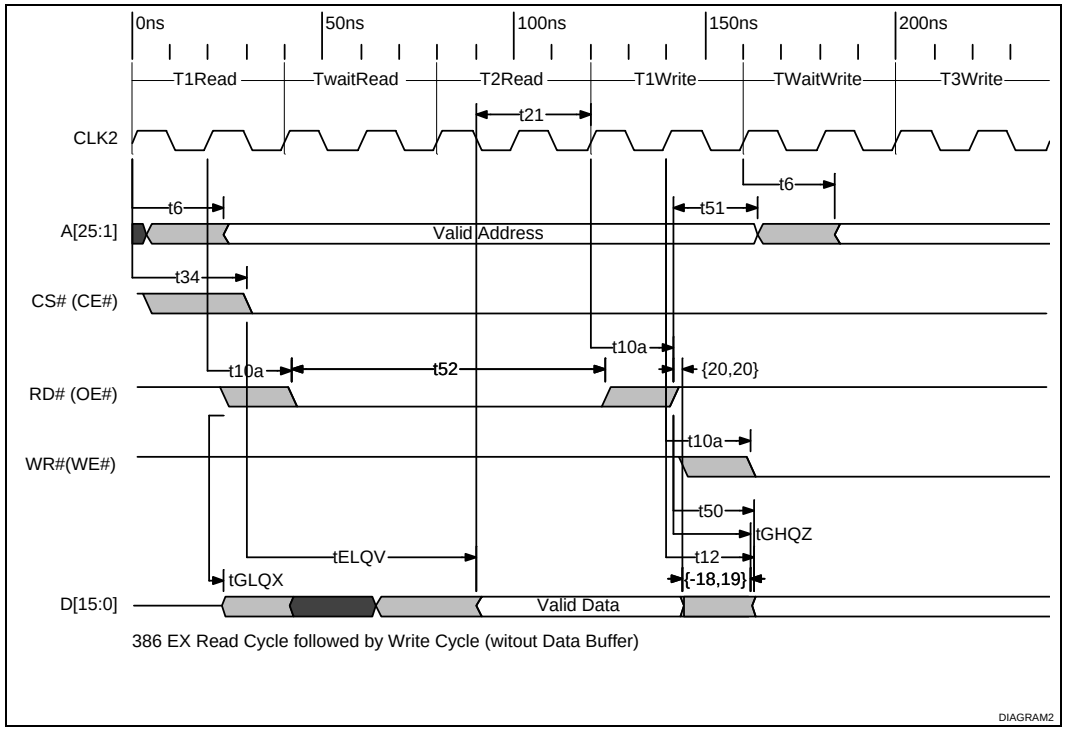


Figure A2. Incorrect Timing Analysis for Intel386™ EX Embedded Microprocessor Interface to Intel Flash

APPENDIX B
ADDITIONAL INFORMATION(1,2)

Order Number	Title
272420	*Intel386™ EX Embedded Microprocessor Datasheet*
290599	*Smart 5 Boot Block Flash Memory Family 2, 4, 8 Mbit*
290539	*8-Mbit SmartVoltage Boot Block Flash Memory Family*
290530	*4-Mbit SmartVoltage Boot Block Flash Memory Family*
290531	*2-Mbit SmartVoltage Boot Block Flash Memory Family*

NOTE:

1. Please call the Intel Literature Center at (800) 548-4725 to request Intel documentation. International customers should contact their local Intel or distribution sales office.
2. Visit Intel's World Wide Web home page at http://www.Intel.com for technical documentation and tools.