



AP-499

**APPLICATION
NOTE**

**Introducing Intel's
Family of Embedded
Intel386™
Microprocessors**

**PRANAV MEHTA
EPO APPLICATIONS
ENGINEERING**

February 1994

Order Number: 272425-002

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

MDS is an ordering code only and is not used as a product name or trademark of Intel Corporation.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

*Other brands and names are the property of their respective owners.

Additional copies of this document or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

CONTENTS

	PAGE		PAGE
1. INTRODUCTION	1	6. CONCLUSION	17
2. OVERVIEW	3	REFERENCES	17
2.1. The Intel386™ EX Microprocessor Core	3	FIGURES	
2.1.1. MODULAR BUS	4	1. Traditional Embedded-Application Design.....	1
2.1.2. EXTERNAL BUS INTERFACE	4	2. DOS-Based Embedded Application Using the Intel386™ EX Microprocessor.....	1
2.2. PC/DOS Compatible Features.....	5	3. Real-Time Embedded-Application Design Using the Intel386™ EX Microprocessor.....	2
2.2.1. TIMER/COUNTER UNIT.....	6	4. Integration of Value-Added Functions into a Small Form-Factor Package	3
2.2.2. INTERRUPT CONTROL UNIT	6	5. Intel386™ EX Microprocessor Bus Interface Unit	4
2.2.3. ASYNCHRONOUS SERIAL I/O (SIO) UNIT.....	6	6. Intel386™ EX Microprocessor Interface to 16-Bit Wide SRAM.....	5
2.2.4. DIRECT MEMORY ACCESS (DMA) CONTROLLER & BUS ARBITER.....	7	7. Intel386™ EX Microprocessor Interface to an 8-Bit Boot-Block EPROM.....	5
2.2.5. PORT 92H	7	8. Intel386™ EX Microprocessor Interrupt Structure	6
2.3. Embedded Control-Specific Features	7	9. Intel386™ EX Microprocessor Clock Generation Unit.....	8
2.3.1. CLOCK GENERATION & POWER MANAGEMENT UNIT	8	10. Interaction of SMM with Idle and Powerdown Modes	9
2.3.2. CHIP SELECT UNIT	9	11. A Simple Scheme for Interfacing the Intel386™ EX Microprocessor to DRAM	10
2.3.3. SYNCHRONOUS SERIAL I/O (SSIO) UNIT	9	12. ISA I/O Space.....	12
2.3.4. DRAM REFRESH CONTROL UNIT ...	10	13. EISA I/O Space.....	12
2.3.5. WATCHDOG TIMER UNIT	10	14. The Configuration Port for the Intel386™ EX Microprocessor, Address Remap Configuration Register (REMAPCFG)	13
2.3.6. PARALLEL I/O PORTS	11	15. Opening the Intel386™ EX Microprocessor's Expanded I/O Space.....	14
2.3.7. PACKAGING	11	16. DOS-Compatible Mode.....	15
2.4. JTAG Unit for Testability	11	17. Non-Intrusive DOS-Compatible Mode	15
3. I/O ADDRESS SPACE	11	18. Enhanced DOS Mode	16
3.1. ISA I/O Structure	11	19. Non-DOS Mode	16
3.2. Expanded I/O Space	12		
3.3. Configuration Port	13	TABLES	
3.3.1. OPENING AND CLOSING THE EXPANDED I/O SPACE	13	Table 1. Expanded I/O Address Map.....	14
4. CONFIGURING THE Intel386™ EX MICROPROCESSOR	14		
5. OPERATING MODES OF THE Intel386™ EX MICROPROCESSOR	15		
5.1. DOS-Compatible Mode	15		
5.2. Non-Intrusive DOS Mode.....	15		
5.3. Enhanced DOS Mode	16		
5.4. Non-DOS Mode.....	16		

1. INTRODUCTION

The Intel386™ microprocessor family has gained widespread acceptance in the world of embedded applications. To understand the reasons driving this phenomenon, it is worth looking at the market segments that are embracing this product family.

There are applications that seek to achieve quick time-to-market by using the personal computer-based development environment. Interestingly, the applications in this segment are very similar to traditional embedded markets (telecommunications, networking, factory automation, imaging). The product development cycle can be drastically reduced by embedding popular operating systems like DOS and Microsoft Windows* on the embedded target, obviating the need for software developers to develop proprietary operating systems. Figures 1 and 2 illustrate the difference between an embedded design using a proprietary operating system and an embedded design with DOS stored in ROM. The latter approach significantly reduces the required amount of application-specific development and provides a consistent and sophisticated user interface for embedded applications. Many home entertainment and office automation products provide a standard, PC-like user interface for ease of use. Applications like remote bar-code scanners, data loggers, and digital cameras not only require PC/DOS compatibility but also demand enhanced power management features.

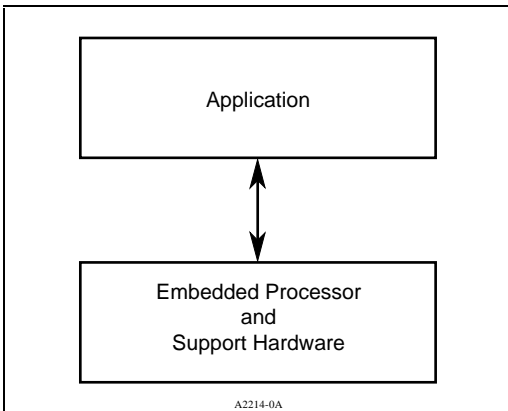


Figure 1. Traditional Embedded-Application Design

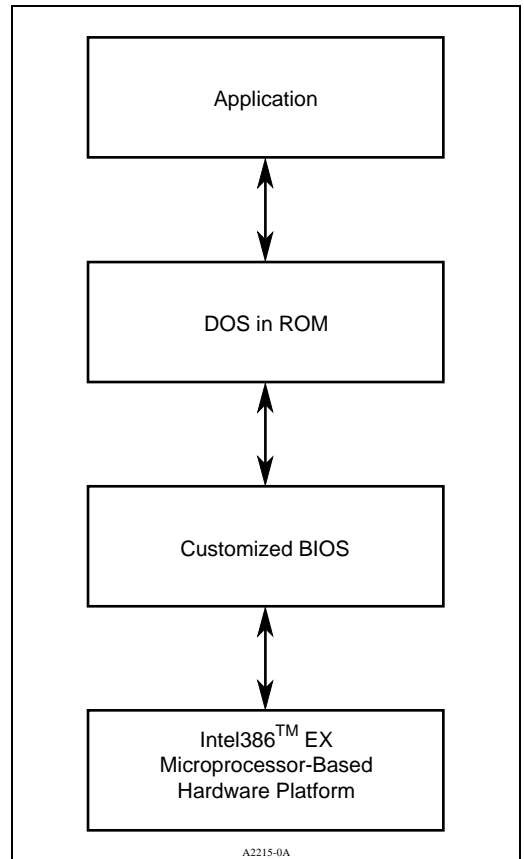


Figure 2. DOS-Based Embedded Application Using the Intel386™ EX Microprocessor

There is also a vast base of applications that use the 80C186 product family. When these applications require higher performance and address space, the Intel386 architecture provides a natural migration to protect their code investment in the Intel architecture. For such non-DOS applications, various real-time, multitasking operating systems and kernels are available, easing the transition to the 32-bit Intel386 architecture. Figure 3 shows an application that uses Intel's iRMX® EMB operating system. This is Intel's real-time multitasking kernel. The kernel isolates the designer from the details of the Intel386 microprocessor architecture.

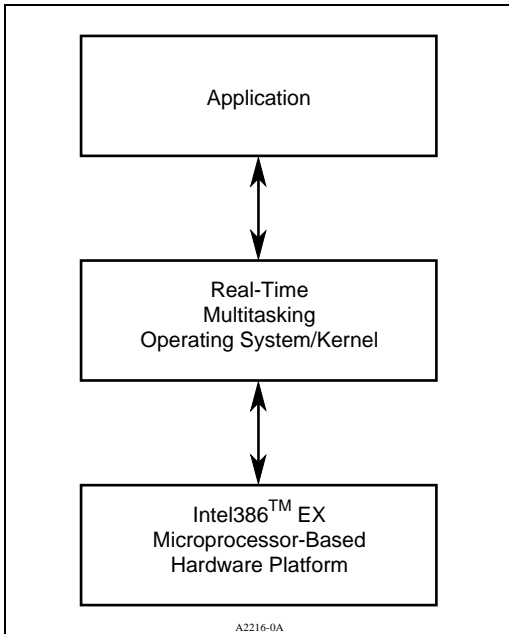


Figure 3. Real-Time Embedded-Application Design Using the Intel386™ EX Microprocessor

Against this background, Intel is introducing a new family of embedded Intel386 microprocessors to meet the requirements of various embedded market segments. The initial offerings of this family of fully static, 32-bit microprocessors are the following:

- the Intel386 EX microprocessor — a PC/DOS-compatible processor that has been optimized for embedded applications
- the Intel386 CX microprocessor — adds two additional address lines, low-power operation, and Intel’s System Management Mode (SMM) to the features of the Intel386 SX microprocessor
- the Static Intel386 SX microprocessor — pin-compatible with the dynamic Intel386 SX microprocessor

The features of the Intel386 SX microprocessor are well documented. This application note describes the Intel386 EX microprocessor. The Intel386 CX microprocessor is not covered in detail because its SMM and addressability are the same of those of the Intel386 EX microprocessor.

Figure 4 illustrates the basic functions offered on the Intel386 EX microprocessor. Although some of the peripherals are compatible with the PC architecture, the Intel386 EX microprocessor provides additional features,

such as the Power Management Unit, Chip Select Unit, DRAM Refresh Control Unit, Watchdog Timer Unit, Synchronous and Asynchronous Serial I/O Units, and JTAG Boundary Scan Unit.

The Intel386 EX microprocessor is **not** a PC-on-a-chip (integrated PC) solution. It differs from other integrated PC solutions in some significant areas:

- The Intel386 EX microprocessor is designed as a DOS-compatible platform optimized for embedded applications. This translates into integration of additional peripherals geared toward standard embedded functions (e.g., chip-select logic, watchdog timer, I/O ports, synchronous serial control unit). The I/O address space in a DOS configuration is limited to 1 Kbyte and is already assigned to standard PC peripherals. To obtain more register space for the additional peripherals, the Intel386 EX microprocessor incorporates a special address space extension mechanism.
- There is some departure from an exact PC configuration. Embedded applications require higher performance and superior functionality than those offered by the 8237A (the DMA module in PC architecture). The Intel386 EX microprocessor provides an enhanced DMA controller module that is a superset of the 8237A. Also, to be cost-effective, it reduces the number of DMA channels to two (from seven in the PC architecture).
- The Intel386 EX microprocessor does not supply the ISA bus signals directly. Most embedded applications are designed as single-board systems and do not require the ISA bus. However, the original Intel386 SX microprocessor’s bus is maintained, so the ISA bus can be easily recreated.
- Embedded applications do not adhere to any standard user interface. Hence, the Intel386 EX microprocessor does not have a video controller or a PC keyboard controller. However, the I/O addresses for these peripherals are reserved so that they can be added externally if a PC-compatible user interface is desired.

In short, only those features in PC architecture that are also useful in the embedded arena are retained in the Intel386 EX microprocessor. This makes the Intel386 EX microprocessor cost effective across a wide range of applications.

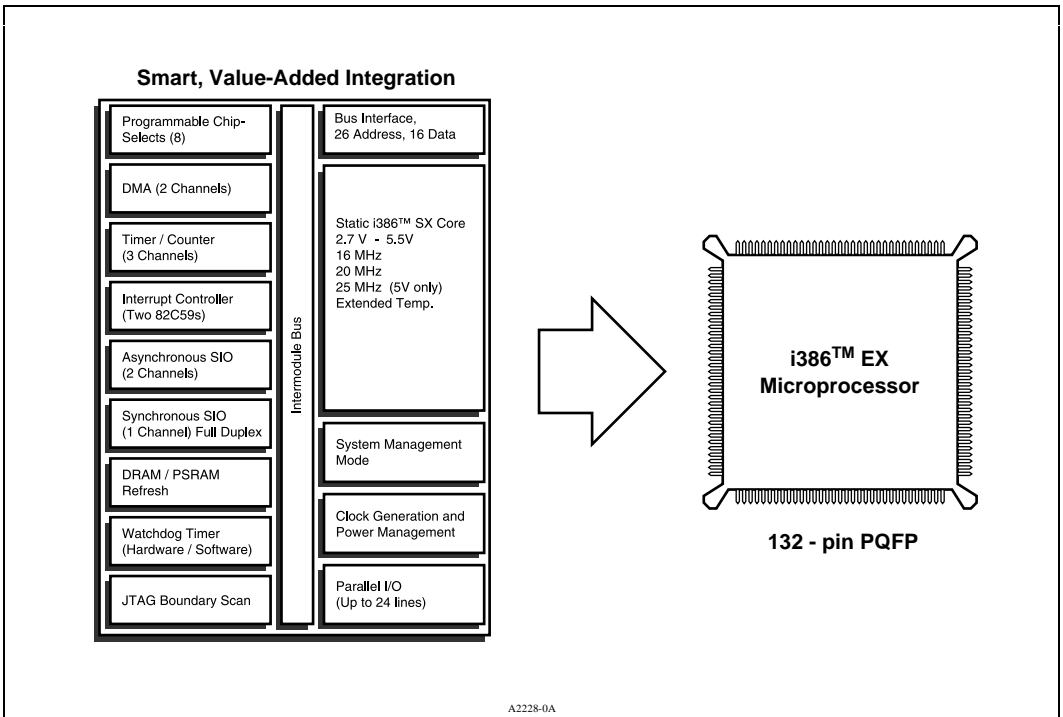


Figure 4. Integration of Value-Added Functions into a Small Form-Factor Package

2. OVERVIEW

As mentioned above, the Intel386 EX microprocessor brings PC/DOS compatibility to embedded applications, while offering many features useful in embedded systems. The following discussion describes the Intel386 EX microprocessor core, peripherals that provide DOS compatibility, and peripherals and features that are optimized for embedded applications.

2.1. The Intel386™ EX Microprocessor Core

The Intel386 EX microprocessor uses the same 32-bit CPU core as that of the standard Intel386 SX microprocessor, with some enhancements to make it suitable for the needs of the embedded market.

- The Intel386 EX microprocessor core is fully static, which means that the processor will retain its state even when the incoming clock signal is removed.

- The Intel386 EX microprocessor core design is modular, allowing easier implementation of future proliferations of the embedded Intel386 micro-processor product family.
- The Intel386 EX microprocessor is process-shifted from a 1.0 micron process to a 0.8 micron process, giving it more performance headroom. This is vital, as performance requirements typically grow over the long life span of an embedded processor.
- The Intel386 EX microprocessor is offered as a dual operating voltage part. This means that it operates at two supply voltage specifications, $V_{CC} = 5V \pm 10\%$ and $V_{CC} = 3V \pm 10\%$.
- Embedded applications are more likely to be exposed to wider temperature extremes than are standard desktop and portable PC applications. Components in an embedded design must be able to withstand this environment. The Intel386 EX microprocessor is offered with an extended temperature range of -40°C to $+85^{\circ}\text{C}$.

- Another feature added to the Intel386 EX microprocessor is the Intel System Management Mode (SMM). SMM is added as a fourth operating mode in the Intel architecture, along with real, protected, and virtual-86 modes. The SMM implementation of the Intel386 EX microprocessor is compatible with the SL-enhanced Intel486™ and Pentium™ processors. Although originally introduced for power management features in portable computers, SMM can be used in the embedded environment for other purposes, such as debugging, having an alternate operating system, or virtualizing I/O devices. More information on this topic will be available during the first quarter of 1994.

2.1.1. MODULAR BUS

The Intel386 EX microprocessor incorporates a modular design approach. Two internal buses are defined: an EM-bus and an EI-bus. As shown in Figure 5, the internal bus unit interfaces with the address, data, and control signals from the core and then splits them into the EM- and EI-bus. The EM-bus is a full 26-bit address bus connecting the Intel386 EX core to modules that need the full address range for operation (e.g., DMA). The 16-bit address EI-bus isolates the core from the majority of the I/O devices integrated on the chip. This prevents overloading the bus with signals coming from the core as peripherals are added to the EI-bus. This modular approach significantly reduces the development time for future proliferations of the embedded Intel386 microprocessor product family.

2.1.2. EXTERNAL BUS INTERFACE

The Intel386 EX microprocessor’s external bus is a superset of the Intel386 SX microprocessor’s bus. For example, it maintains the non-multiplexed 16-bit data bus, like that of the Intel386 SX microprocessor, for higher performance. When the data bus is multiplexed with the address bus, it must latch the address in the first clock cycle and then treat the bus as a data bus for the remainder of the bus cycle. A non-multiplexed bus obviates the need for address latching, requiring fewer clock cycles to complete a bus cycle.

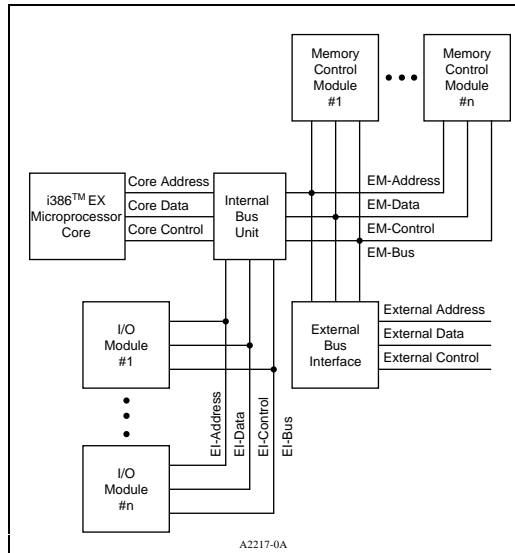


Figure 5. Intel386™ EX Microprocessor Bus Interface Unit

While the Intel386 EX microprocessor retains the original Intel386 SX microprocessor bus for compatibility with existing ASICs or chip sets, it adds the following enhancements to suit embedded applications.

- A standard Intel386 SX microprocessor has a 24-bit external address bus. This allows the operating system and application programs to address 16 Mbytes of physical memory directly. Conventional embedded applications require much less memory. However, as mentioned before, the new trend in embedded applications requires more sophisticated, graphical user interfaces, which demands larger data and code storage space. Keeping in mind the characteristically long life cycles of embedded applications, the Intel386 EX microprocessor provides 26-bit addressing that will support 64 Mbytes of memory space. Not so coincidentally, this range fits very well with the emerging PCMCIA standard. This will become important as PCMCIA cards find their way into embedded applications.

2.2.1. TIMER/COUNTER UNIT

The Timer/Counter Unit on the Intel386 EX microprocessor has the same functionality as the industry standard 82C54 counter/timer. It provides three independent 16-bit counters, each capable of handling clock inputs up to 8 MHz. The source of the timer clocks can be selected to be either the internal processor clock or an external clocking circuit. The timer/counter module can be programmed to operate in six different modes. Various modes allow the timer unit to be used as an event counter, elapsed time indicator, programmable one-shot, etc.

2.2.2. INTERRUPT CONTROL UNIT

The Intel386 EX microprocessor incorporates two 8259A Programmable Interrupt Control (PIC) units, the same peripherals used in the ISA architecture. In addition to using the same functional modules, these two PICs are connected in the same master/slave configuration as in the ISA system. For example, the output of the timer/counter channel 0 is connected to the IRQ0 input of the master 8259A. Figure 8 illustrates various connections.

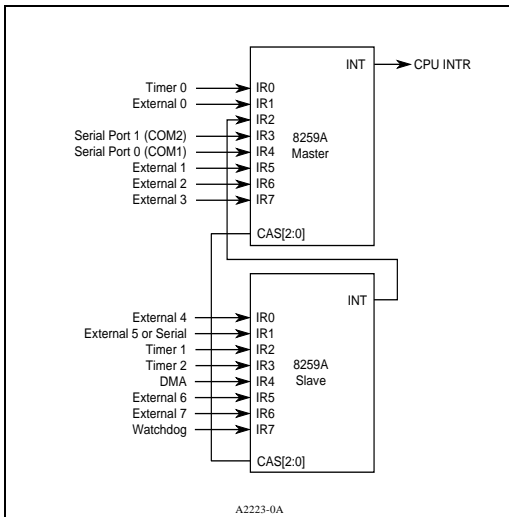


Figure 8. Intel386™ EX Microprocessor Interrupt Structure

The Intel386 EX microprocessor offers a total of eight external interrupt lines. Four of these, which come from the master 8259A, can be cascaded externally with additional 8259As. Thus the total of number of external interrupt lines can be expanded to 36. A set of configura-

tion registers associated with the interrupt control unit determines the interrupt sources, as described later.

2.2.3. ASYNCHRONOUS SERIAL I/O (SIO) UNIT

The Intel386 EX microprocessor's asynchronous serial port is a Universal Asynchronous Receiver/Transmitter (UART). Functionally, it is equivalent to the National Semiconductor NS16450 and INS8250A. The Intel386 EX microprocessor has two such serial ports.

The UART performs serial-to-parallel conversion on characters received from an external device and parallel-to-serial conversion on characters received from the CPU before transmitting them to the external world. The CPU can read the status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART and any error conditions (parity, overrun, framing, or break interrupt). Some of the other features of the SIO unit are the following:

- The UART includes a programmable baud-rate generator capable of dividing the baud-clock input by divisors of 1 to $(2^{16} - 1)$ and multiplying the clock by 16 to drive the internal transmitter and receiver logic.
- The baud-clock input can be selected to be either a derivative of the internal processor clock or an external clocking circuit.
- The "sticky" bit feature of the UART allows a nine-bit serial transfer mode used for multidrop configurations in which a master communicates with several slave processors.
- For DOS-compatible applications, serial transfer (baud) rates typically do not exceed 19.2 Kbaud. However, many non-DOS applications demand higher baud rates. The maximum baud rate achievable by the Intel386 EX microprocessor's SIO unit is 530K. To achieve approximately 500K baud rate with the CPU processing the serial interrupts requires almost 100% of the processor's bandwidth. This is due to the typical overhead involved in saving current processor information before jumping to the interrupt service routine. Thus at such high transfer rates the CPU cannot do other useful work.

To circumvent this problem, the serial interrupts in the Intel386 EX microprocessor can be selectively connected to the DMA module. In this case, instead of generating an interrupt whenever the transmit buffer is empty or receive buffer is full, the appropriate DMA request line (DREQ) is activated. The Bus Arbiter unit (explained in the next section) in turn

generates an internal HOLD request to the core. Unless the processor is executing LOCKed bus cycles, it relinquishes the external bus rather quickly. Once the Bus Arbiter gives control of the bus to the appropriate DMA channel, the data can be transmitted to (from) the memory from (to) the appropriate UART buffer within two bus cycles. Circumventing the interrupt latency of the processor in this fashion results in only 5% use of CPU bandwidth at the same 500K baud rate!

- The UART can be programmed to work in either a polled or an interrupt-driven environment.

2.2.4. DIRECT MEMORY ACCESS (DMA) CONTROLLER & BUS ARBITER

The Intel386 EX microprocessor's DMA Controller is a 2-channel DMA with a feature set that has enhancements beyond the 8237A DMA controller used in the PC (ISA) architecture. It incorporates full 26-bit addressing, a 2-cycle mode for memory-to-memory transfers using only one channel, buffer chaining mode, etc. However, the DMA module can be configured in an 8237A-like mode. The DMA circuitry is designed so that additional channels can be added on future embedded Intel386 microprocessor products.

The Intel386 EX microprocessor's DMA controller is capable of transferring data between any combination of memory and I/O, with any combination (8 or 16 bits) of data path widths. Bus bandwidth is optimized through the use of an internal temporary register that can disassemble or assemble data to or from either an aligned or a non-aligned destination or source.

The Intel386 EX microprocessor's DMA has two channels, each of which operates independently. Each channel has many different operating modes for data transfer (single, block or demand transfer with either "fly-by" or two-cycle mode). Many of the operating modes can be combined to provide a very versatile DMA controller.

The design of the DMA unit is backward-compatible with the 8237A, as follows:

- Even though the Intel386 EX microprocessor's DMA is capable of generating 26-bit addresses directly, to emulate the page register feature in PC architecture, an option is provided to prevent the overflow from the lower 16 bits to the higher bits.

- All 8237A operating modes are supported except the special two-cycle mode. Instead, a true two-cycle mode is provided that does not require two channels for memory-to-memory transfers.
- Similar to an 8237A, the DMA controller is programmed through 8-bit registers.

The Bus Arbiter works much like the priority resolving circuitry of a DMA. It receives requests from the two DMA channels, the external bus master, and the DRAM Refresh Controller. The Bus Arbiter requests bus ownership by asserting the internal HOLD signal to the processor and resolves priority issues among all active requests when bus ownership is granted.

2.2.5. PORT 92H

To overcome limitations of the 80286 processor, the early PC-AT systems incorporated two special features: A20GATE and FastCPUReset, as described below. These features were implemented in a somewhat random fashion in the beginning, but over time almost all OEMs settled on providing these two features via two register bits at I/O location 92H. The Intel386 architecture does **not** have similar limitations, but these features are implemented on the Intel386 EX microprocessor because of the need to be backward-compatible with the enormous amount of software written for 80286-based systems.

2.2.5.1. A20GATE

The A20GATE allows systems to emulate the wrap-around characteristic of the 8086 processor across the 1 Mbyte address boundary. When disabled, however, it allows programs to access 64 Kbytes of extra memory above the 1 Mbyte boundary.

2.2.5.2. FastCPUReset

Whenever this bit in Port 92H is set, it resets only the CPU of the Intel386 EX microprocessor. This provides backward compatibility with programs written for the 80286 that reset the processor to switch from the protected mode to the real mode.

2.3. Embedded Control-Specific Features

The following are the complementary features of the Intel386 EX microprocessor that make it a DOS-compatible engine optimized for embedded applications.

2.3.1. CLOCK GENERATION & POWER MANAGEMENT UNIT

The clock generation unit provides uniform, non-overlapping clock signals to the CPU and associated peripherals. Figure 9 shows the clock signals generated within the clock unit. The power management features control clock signals to provide different power management functions. The clock generation circuit includes a divide-by-two counter, a programmable divider for generating a prescaled clock (PSCLK), a divide-by-two counter for generating baud clock inputs to the serial control units (SERCLK), and a reset circuit.

The Intel386 EX microprocessor does not incorporate an on-chip oscillator circuit. Given the wide operating voltage specification of the Intel386 EX microprocessor, it is difficult to characterize an oscillator for a given frequency that provides an accurate time reference for the entire voltage range. Thus, in the absence of a CLKOUT signal from the on-chip oscillator, the CLK2 input (twice the operating frequency of the processor) provides the fundamental timing reference for the Intel386 EX microprocessor.

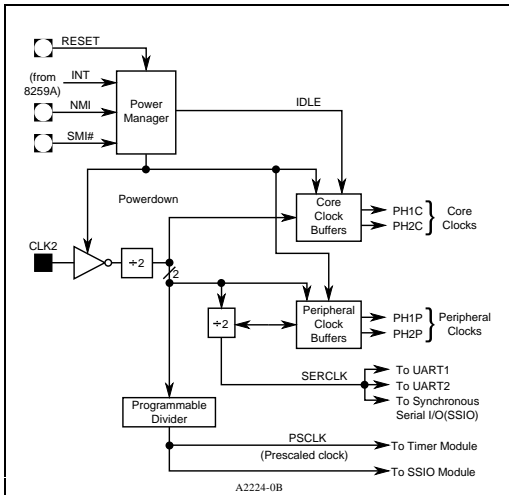


Figure 9. Intel386™ EX Microprocessor Clock Generation Unit

One of the key application requirements for the Intel386 EX microprocessor is low power consumption. Two power management modes are provided on the Intel386 EX microprocessor: Idle and Powerdown.

2.3.1.1. Idle Mode

The CPU of the Intel386 EX microprocessor consumes about half of the total device power. When the processor is waiting for an external event to occur, the Idle mode can be used to stop the CPU clocks during the waiting period. This reduces power consumption significantly. When an external event occurs, the processor exits Idle mode and resumes operation. While the CPU is idle, however, all the peripheral clocks are still active (for example, if a serial channel is set up to transfer data via the DMA, that process continues without any interruption).

Idle mode is entered by manipulating bits in the Power Control register and then executing a HALT instruction. Upon receiving an unmasked interrupt, NMI, or SMI, the processor exits Idle mode.

2.3.1.2. Powerdown Mode

There are instances in low-power applications when processor operation is not required until a certain external event occurs. The Powerdown mode allows clocks going to the CPU as well as all peripherals to be disabled. Processor current consumption is reduced to leakage current (microamps).

Powerdown mode is entered by manipulating bits in the Power Control register and then executing a HALT instruction. Upon receiving an unmasked interrupt, NMI, or SMI, the processor exits Powerdown mode. A PWRDOWN output signal is provided to control other external devices, if desired.

2.3.1.3. Interaction of SMM with Idle and Powerdown Modes

Upon receiving an SMI interrupt, the Intel386 EX microprocessor exits Idle or Powerdown mode and then enters the Intel System Management Mode (SMM). Once the appropriate actions have been taken in SMM, before exiting the SMM, it is possible to check if the SMM was entered while the processor was in a Halt state. At this point, if desired, the SM-code can set a flag that returns the processor to the HALT state upon exiting the SMM. Assuming that the Power Control Register bits were not changed in SMM, the processor then re-enters the Idle or Powerdown mode. Figure 10 describes these state transitions.

2.3.2. CHIP SELECT UNIT

The Chip Select Unit (CSU) decodes bus cycle address and status information and activates chip-select signals that are enabled. The individual chip-selects become valid in the same bus state as the address lines and follow the same timings as address lines.

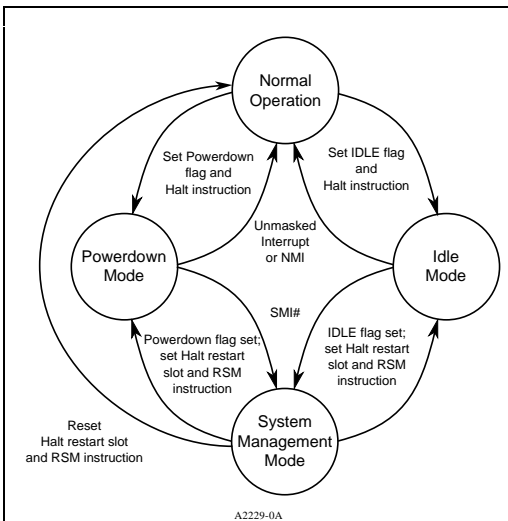


Figure 10. Interaction of SMM with Idle and Powerdown Modes

The CSU is divided into eight separate chip-select regions; each can enable one of the eight chip-select signals. Each chip-select region can be mapped into memory or I/O space. A memory-mapped chip-select region can start on any $2^{(n+1)}$ Kbyte address location (where $n = 0-15$, depending upon the mask register). An I/O-mapped chip-select region can start on any $2^{(n+1)}$ word

address location (where $n = 0-15$, depending upon the mask register). The size of the region is dependent upon the mask used.

A chip-select region is active when it meets all of the following criteria:

- The chip-select enable bit in the lower mask register is enabled.
- The bus status matches the programmed type (memory or I/O).
- The bus cycle address is equal to the nonmasked portion of the chip-select address.
- A memory address applies to memory read, memory write, and instruction fetch bus cycles. An I/O address applies to I/O read and I/O write bus cycles. Interrupt acknowledge and HALT bus cycles will not activate the chip-select regions.
- After power-on or reset, only the UCS chip-select region is active, selecting the UCS# chip-select pin.

The following summarizes the features of the Chip Select Unit:

- Eight chip-selects with 2 Kbyte granularity for memory accesses and word granularity for I/O accesses.
- Support of SMM memory addressing.
- Programmable start address with mask register to indicate chip-select region, allowing overlap.
- Memory or I/O bus cycle decoder.
- Programmable wait state generator (up to 31 wait states).
- On-chip 8-bit bus size generator.
- Provision to disable a chip-select.
- Provision to override bus ready.

2.3.3. SYNCHRONOUS SERIAL I/O (SSIO) UNIT

The Synchronous Serial I/O (SSIO) unit of the Intel386 EX microprocessor allows simultaneous bidirectional communications. It consists of a transmit channel, a receive channel, and a dedicated baud rate generator. The transmit and receive channels may be operated independently (using different clocks) to provide non-lock-step, full-duplex communications. Each channel is capable of originating the clocking signal (Master Mode) or receiving an externally generated clocking signal (Slave Mode).

The following summarizes the features of the SSIO unit:

- 6.25 Mbaud maximum shift clock speed (at 25 MHz)
- Double-buffered 16-bit data register for each channel
- Shared baud rate generator
- Gated interrupt outputs
- Independently enabled transmit and receive functions
- Programmable connection of SSIO interrupts to the on-chip DMA, similar to that of the UARTs

2.3.4. DRAM REFRESH CONTROL UNIT

Several vendors offer fully integrated DRAM controllers, either as single-chip solution or as part of a chip-set. These devices are useful, but there are several factors that may lead designers to design their own DRAM interface:

- Off-the-shelf DRAM controllers are expensive.
- They typically have high pin counts and power requirements.
- They offer many special features such as nibble, page, and static-column modes that are not commonly used in embedded applications. This represents non-value added functionality for the user.

A simple DRAM interface can be implemented by using a couple of multiplexers and an inexpensive Programmable Logic Device (PLD). Figure 11 shows basic conceptual elements of such a design. The resulting logic would be the most cost-effective solution for a specific application.

The Intel386 EX microprocessor incorporates simple DRAM refresh control logic that provides three basic functions of a complete DRAM interface:

- Programmable refresh interval
- Bus arbitration logic whenever a refresh cycle needs to be generated
- Row address generation to refresh DRAM cells

A REFRESH# pin is provided that can be used by external logic to generate appropriate “RAS only,” or “CAS before RAS,” or simple dummy read cycles to refresh the DRAM.

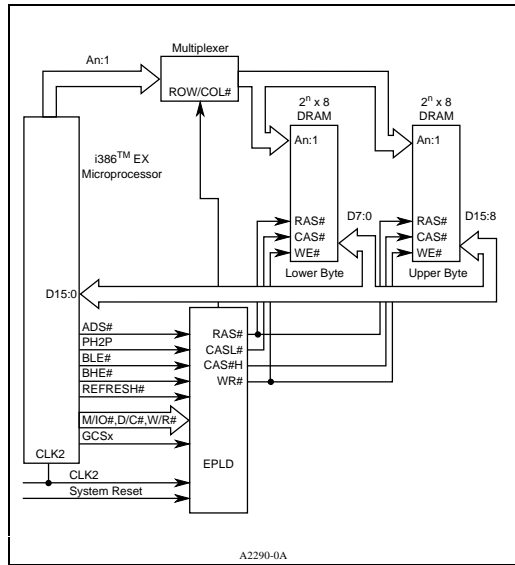


Figure 11. A Simple Scheme for Interfacing the Intel386™ EX Microprocessor to DRAM

2.3.5. WATCHDOG TIMER UNIT

A variety of sources can cause system failures; for example, runaway software can work its way into an endless loop waiting for an event that never occurs. Not all sources of system failures can be anticipated and guarded against. The Watchdog Timer (WDT) unit provides a method for graceful recovery from unexpected hardware and software upsets. It contains a 32-bit programmable counter that generates a pulse on the WDTOUT pin for 8 PHZP cycles (or 16 CLK2 cycles, under normal operation). This signal can be directly fed back to the RESET, NMI, or any other interrupt.

The WDT unit on the Intel386 EX microprocessor can be configured in one of three possible states.

2.3.5.1. Software Watchdog Mode

Runaway software may cause a system to hang. If the WDT is programmed to be a software watchdog, then the executing software needs to reload the timer before the WDT timer expires. A special instruction sequence, one that errant code would be very unlikely to produce, is used to reload the timer.

2.3.5.2. Bus Monitor Mode

In some cases, a hardware upset or errant code can cause a bus access to a peripheral in an I/O location that does not exist in hardware in a system. In a “normally not ready” system, this would cause a hang-up because the processor would be waiting for a Ready to terminate the bus cycle. The Bus Monitor mode protects against this scenario. When in Bus Monitor mode, the timer is loaded on every falling edge of ADS# (beginning of a bus cycle) and the counting is stopped when READY# goes low. If the READY# signal is not asserted within a programmed interval, the WDTOUT signal is activated.

2.3.5.3. Free-Running Counter Mode

If neither the Software Watchdog function nor the Bus Monitor function is required, the Watchdog Timer can be used as a general-purpose, 32-bit, free-running counter.

In all instances, the only action the WDT takes on expiration of timer is activating the WDTOUT signal. The system designer can route this signal to the Reset pin, NMI, or any other nonmaskable interrupts to satisfy the system requirements. The Intel386 EX microprocessor also allows the designer to route the WDTOUT signal internally to the internal slave 8259A interrupt controller.

2.3.6. PARALLEL I/O PORTS

Many applications do not require all of the on-chip peripheral functions available on the Intel386 EX microprocessor. For example, a relatively small system would require only a few of the eight chip-select lines provided. In other applications, not all the modem control signals are required. In the cost-sensitive embedded world, maximum utilization of the pin real estate is desired.

The Input/Output ports allow system designers the flexibility to replace the function of unused peripheral pins with general-purpose I/O ports. Many on-chip peripheral pin functions are multiplexed with an I/O port. All the Intel386 EX microprocessor's I/O ports are bit-programmable to be input-only, output-only, or open-drain bidirectional. Ports 1 and 2 share an input/output pad buffer cell that has 8 mA drive capability. Port 3 has 16 mA drive capability to interface directly with heavier-load devices such as LEDs. These drive capabilities are offered at $V_{CC} = 5$ volts.

2.3.7. PACKAGING

Initially, the Intel386 EX microprocessor will be offered in a 132-pin JEDEC Plastic Quad Flat Pack (PQFP)

package. This will be followed by a 144-pin EIAJ TQFP. Figure 4 (on page 3) shows the value-added integration of PC peripherals and other embedded features into a small form-factor package.

2.4. JTAG Unit for Testability

With shrinking package size (and pitch), increasing system complexity, and the advent of surface-mount technology, circuit board testing has become a major issue. One of the emerging standards in the industry to simplify board testing is the JTAG boundary scan technique. The Intel386 EX microprocessor employs the IEEE 1149.1 compliant JTAG boundary scan standard to facilitate manufacturability and testability of the end products.

3. I/O ADDRESS SPACE

It is appropriate at this juncture to discuss the Intel386 EX microprocessor's I/O address space. The PC architecture, as defined by the ISA standard, has a peculiar scheme of I/O address decoding. I/O addresses of most common peripherals found in the PC architecture are fixed. While the I/O addressing scheme for the Intel386 EX microprocessor needs to be compatible with the PC I/O scheme, it needs to go beyond that to accommodate the extra peripherals designed for embedded applications.

3.1. ISA I/O Structure

The Intel386 EX microprocessor's I/O address space is 64 Kbytes. On PC, XT, AT, and ISA platforms, DOS assumes that only 1K of the total 64K I/O address space is used. The first 256 bytes (addresses 0000H to 00FFH) of this 1K I/O space are reserved for I/O platform (motherboard) resources, such as the interrupt and DMA controllers. The remaining 768 bytes (addresses 0100H to 03FFH) are available for “general” I/O peripheral card resources. Since only 1K of the address space is supported, ISA add-in I/O peripheral cards decode only the lower 10 address lines; consequently, the upper address lines (A10–A15) are ignored. Because the upper address lines are not decoded, the 256 platform address locations and the 768 bus address locations are repeated 64 times (on 1K boundaries), covering the entire 64K address space. (See Figure 12.)

Add-in I/O peripheral cards do not use the I/O addresses reserved for the platform resources. Software running on the platform can use any of the 64 copies of the 256 I/O address locations reserved for platform resources.

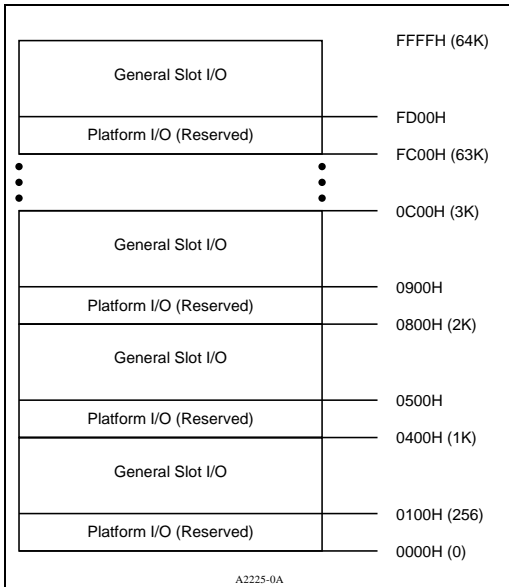


Figure 12. ISA I/O Space

3.2. Expanded I/O Space

The Intel386 EX microprocessor's I/O address scheme (similar to that of EISA(32) systems) exploits the above facts by assigning 63 of the 64 repetitions of the first 256 address locations to specific *slots*¹. The partitioning is such that 4 groups of 256 address locations are assigned to each slot, for a total of 1024 specific address locations per slot. (See Figure 13.) Since add-in I/O cards decode only the lower 10 address lines, they respond to the "general" 768 bytes (repeated 64 times). Thus, each slot has 1K addresses (in four 256-byte segments) that can potentially contain extended peripheral registers. [1]

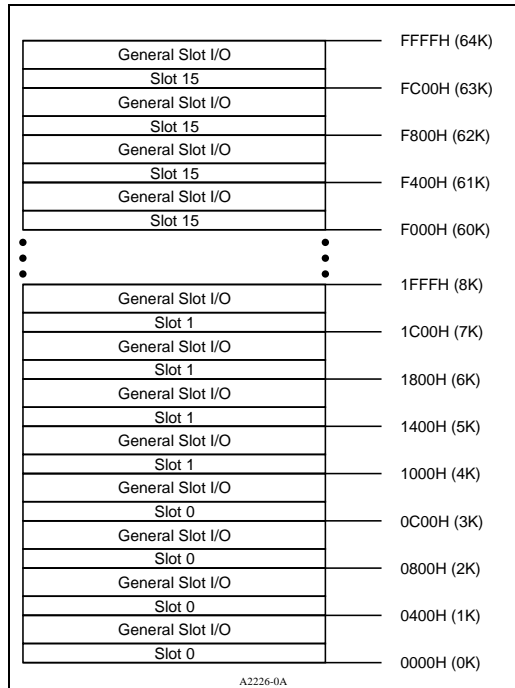


Figure 13. EISA I/O Space

Slot 0 refers to the platform. (Again, many of the peripherals found on a standard PC platform (motherboard) are integrated on the Intel386 EX microprocessor). Thus, a total of 1K unique I/O addresses are assigned to the platform (in addition to the 768 bytes which are repeated). The first 256 address locations are the same platform resources as defined across all platforms. The remaining three groups of 256 address locations can be used for a specific platform such as EISA.

For the Intel386 EX microprocessor, slot 15 is used for the extra registers needed for integrated non-DOS peripherals. This is a safe method because, although the EISA method allows 16 slots to be defined, traditionally only 8 to 10 slots are available on a platform. In other words, slot 15 is typically not used².

¹ In a PC, a *slot* is a socket used to connect add-on boards to standard XT, AT or ISA buses for expanding the functionality of the system. For embedded processors like the Intel386 EX microprocessor, the term *slot* is more of a concept than physical reality. A *slot* could be viewed as part of the total I/O address space.

² In addition to slot 15, Intel reserves slot 14 for future expansion of the embedded Intel386 microprocessor product family.

3.3. Configuration Port

I/O address locations 22H and 23H in slot 0 (within the DOS I/O area) offer a special case. These address locations are not used for any peripheral registers on the main platform. The Intel386 SL microprocessor and other integrated PC solutions use them to enable extra address space required for configuration registers specific to these products. These same address locations are used for enabling the extra address space for the Intel386 EX microprocessor.

Also, the 16-bit register at I/O location 22H can be used to control mapping of various on-chip peripherals in I/O address space. This register, designated the REMAPCFG

register (often referred to as the Configuration Port), is defined in Figure 14.

The Remap bits of this register control whether the on-chip peripherals are mapped into the regular DOS I/O address space (i.e., the first 256 bytes of slot 0) or mapped out to the expanded I/O address space (i.e., slot 15). If a bit associated with a particular peripheral is 0, then it is mapped into the DOS I/O space. One can individually set bits for various peripherals to map them out. It is important to note, however, that the contents of this register can be modified only after the expanded I/O address space is enabled as specified below. At Reset, this register is cleared, which maps on-chip peripherals into DOS I/O space.

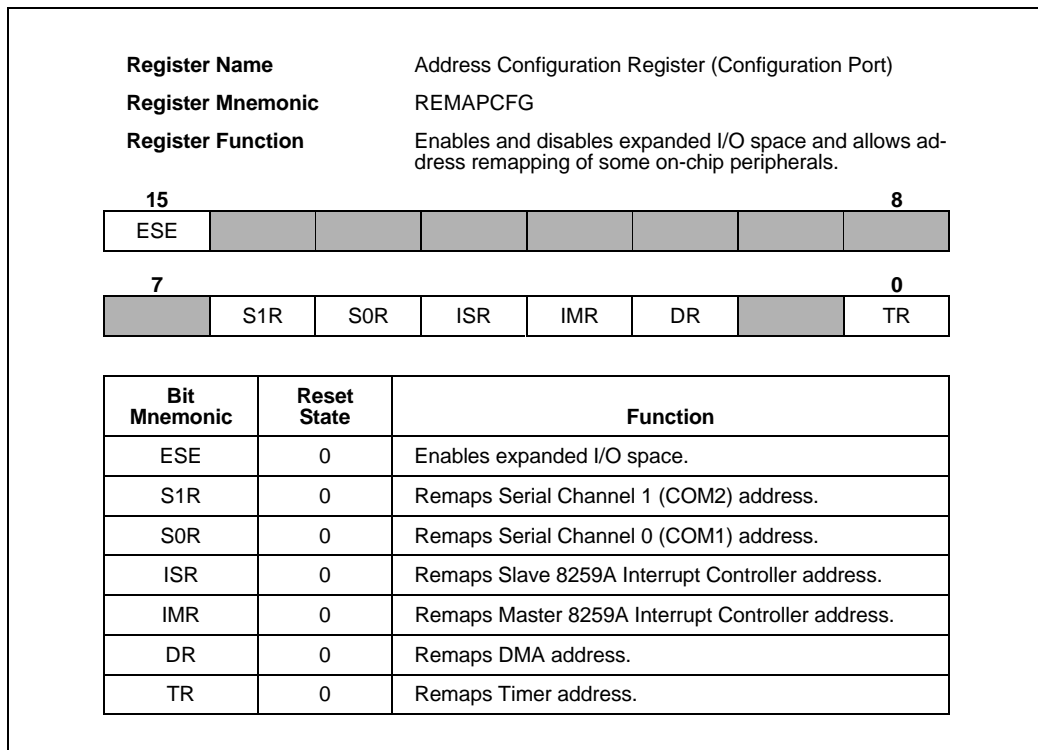


Figure 14. The Configuration Port for the Intel386™ EX Microprocessor, Address Remap Configuration Register (REMAPCFG)

3.3.1. OPENING AND CLOSING THE EXPANDED I/O SPACE

The Intel386 EX microprocessor's expanded I/O space is enabled by a specific write sequence to I/O addresses 22H and 23H. Once the expanded I/O space is enabled,

on-chip peripherals like the timers, DMA, interrupt controllers and serial communication channels can be mapped out of regular DOS I/O space and into expanded I/O space (using the REMAPCFG register). Registers associated with other on-chip peripherals (Chip Select

Unit, Power Management module, Watchdog Timer, etc.) can be accessed. The I/O address map for this expanded mode is illustrated in Table 1.

Table 1. Expanded I/O Address Map

Register Description	I/O Address Range
DMA Controller 1	F000H – F01FH
Master Interrupt Controller	F020H – F03FH
Programmable Interval Timer	F040H – F05FH
DMA Page Registers	F080H – F09FH
Slave Interrupt Controller	F0A0H – F0BFH
Math Coprocessor	F0F0H – F0FFH
Chip Select Unit	F400H – F47FH
Synchronous Serial I/O Unit	F480H – F49FH
DRAM Refresh Control Unit	F4A0H – F4BFH
Watchdog Timer Unit	F4C0H – F4CFH
Asynchronous Serial I/O Channel 0 (COM1)	F4F8H – F4FFH
Clock Generation and Power Management Unit	F800H – F80FH
External/Internal Bus Interface Unit	F810H – F81FH
Chip Configuration Registers	F820H – F83FH
Parallel I/O Ports	F860H – F87FH
Asynchronous Serial I/O Channel 1 (COM2)	F8F8H – F8FFH

The REMAPCFG register is write-protected until the expanded I/O space is unlocked (see Figure 15). When the unlocking write sequence is executed, it sets the Expanded I/O Space Enabled (ESE) bit, which is bit 15 of the REMAPCFG register (Figure 14). A program can check this bit to see whether it has access to the expanded I/O space registers. Clearing the Expanded I/O Space Enabled (ESE) bit disables the expanded I/O space. This again locks the REMAPCFG register and makes it read-only.

As indicated by the I/O address map, all the registers associated with the Intel386 EX microprocessor's peripherals and controls (except REMAPCFG) are physically located in the unique address space for Slot 15 (i.e., address ranges F000H–F0FFH, F400H–F4FFH, and F800H–F8FFH). However, some of these registers need to be mapped into the lower I/O address range (slot 0) for

DOS compatibility. A special I/O address-decoding technique is required to accomplish this. The I/O address-decoding scheme is based on the values of the Expanded I/O Space Enabled (ESE) bit and individual Address Remap bits in the REMAPCFG register. Various combinations of these bit values define four basic operating modes for the Intel386 EX microprocessor, discussed in Section 5.

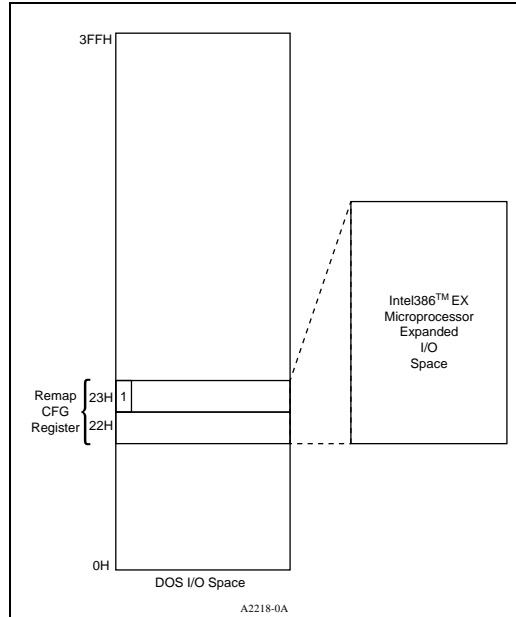


Figure 15. Opening the Intel386™ EX Microprocessor's Expanded I/O Space

4. CONFIGURING THE Intel386™ EX MICROPROCESSOR

During chip initialization (and at other times as needed), the Intel386 EX microprocessor can be configured in different ways. Configuration choices are available to interconnect various on-chip peripheral modules in certain ways. Functions of many device pins can also be programmed. The Intel386 EX microprocessor incorporates module configuration registers and pin configuration registers (located between F820H and F83FH in the Expanded I/O space) for these purposes. For example, the output of timer channel 1 can be internally connected to the IRQ2 line of the slave 8259A and then, at the pin level, the shared pin can be programmed to be PORT3.1 instead of the timer output TMROUT1. The idea is to allow as much flexibility for the end-user as possible,

given the wide array of features available on the Intel386 EX microprocessor.

5. OPERATING MODES OF THE Intel386™ EX MICROPROCESSOR

As discussed earlier, the Intel386 EX microprocessor can be programmed to operate in four different operating modes depending on the application.

5.1. DOS-Compatible Mode

This mode is characterized by the following bit values:

ESE = 0 and all peripherals' Remap bits = 0.

All DOS peripherals are mapped into the DOS I/O space. Only address lines A9–A0 are decoded for on-chip peripherals. In other words, accesses to DOS peripherals are valid, whereas non-DOS peripherals are inaccessible (see Figure 16).

This mode is useful for accessing the on-chip timer, interrupt controller, UARTs, or DMA controller in a DOS-compatible environment.

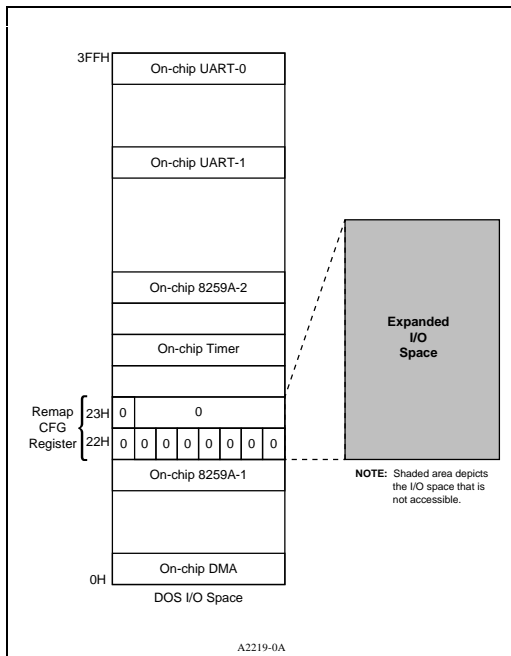


Figure 16. DOS-Compatible Mode

5.2. Non-Intrusive DOS Mode

This mode is characterized by the following bit values:

ESE = 0 and individual peripheral's Remap bit = 1.

Peripherals whose corresponding Remap bits are set will be mapped out to expanded I/O space. Still, only A9–A0 address lines are decoded internally. Mapped out DOS peripherals and non-DOS peripherals are inaccessible (see Figure 17).

This mode is useful when the on-chip DOS-compatible peripheral is not to be used for a DOS-compatible function but, instead, a DOS stand-alone peripheral is connected externally. For example, a designer might connect an external 8237A DMA for 100% DOS compatibility rather than using the Intel386 EX microprocessor's enhanced DMA module. In this case, the Remap bit for the integrated DMA block will be set (1). The external 8237A can be accessed in the slot 0 I/O space, while the integrated DMA can be accessed only after the expanded I/O space is enabled.

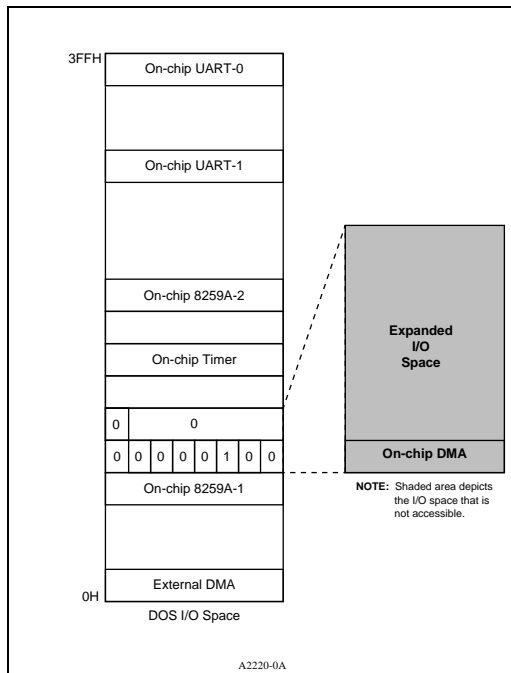


Figure 17. Non-Intrusive DOS-Compatible Mode

5.3. Enhanced DOS Mode

This mode is characterized by the following bit values:

ESE = 1 and all peripherals' Remap bits = 0.

Internally, address lines A15–A0 are decoded. At the same time, the on-chip DOS peripherals are mapped into the slot 0 I/O space (00H–3FFH). Any access to a DOS peripheral accesses the physical register in slot 15 I/O space, but at the same time is reflected in slot 0 I/O space (see Figure 18).

If the application frequently requires the additional peripherals available on the Intel386 EX microprocessor but at the same time wants to maintain some level of DOS compatibility for ease of development, this is the most useful mode. External I/O decoding logic must decode address lines A15–A0 for this mode.

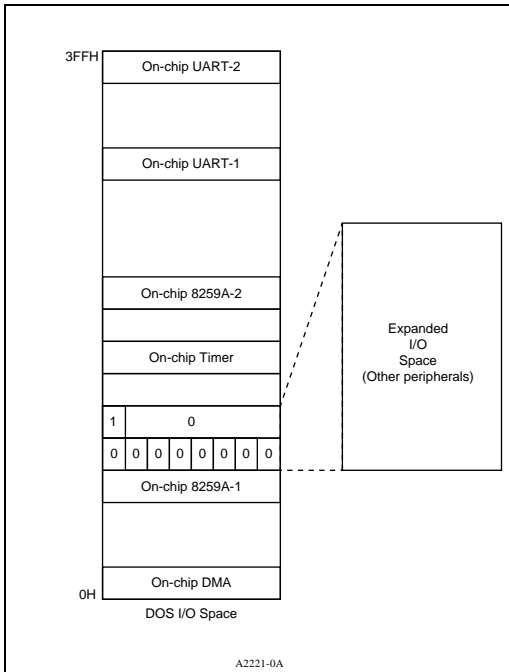


Figure 18. Enhanced DOS Mode

5.4. Non-DOS Mode

This mode is characterized by the following bit values:

ESE = 1 and individual peripheral's Remap bit = 1

Internally, address lines A15–A0 are decoded. Corresponding on-chip DOS peripherals can be accessed only in the slot 15 I/O space. The designer can place other peripherals in slot 0 with no conflict (see Figure 19). Again, external logic must decode A15–A0.

This mode is for those systems that don't require DOS compatibility and have other custom peripherals in slot 0 I/O space. A complete non-DOS mode is accomplished when all remap bits are set (1).

For all DOS peripherals, the lower 10 bits in the DOS I/O space and in the expanded I/O space are identical (except the UARTs, whose lower 8 bits are identical). This makes remapping easier. Also, the UARTs have fixed I/O addresses. This differs from the standard PC configurations, in which these address ranges are programmable. However, if some other device is located at the same I/O location, then the customized BIOS for the Intel386 EX microprocessor can detect them, remap the corresponding SIO channel out of the DOS I/O space, and then write the SIO address (F4F8H–F4FFH, for example, for SIO channel 0) into the BIOS data table describing the I/O map. Of course, in this case, one would enter Enhanced DOS mode to access SIO channel 0.

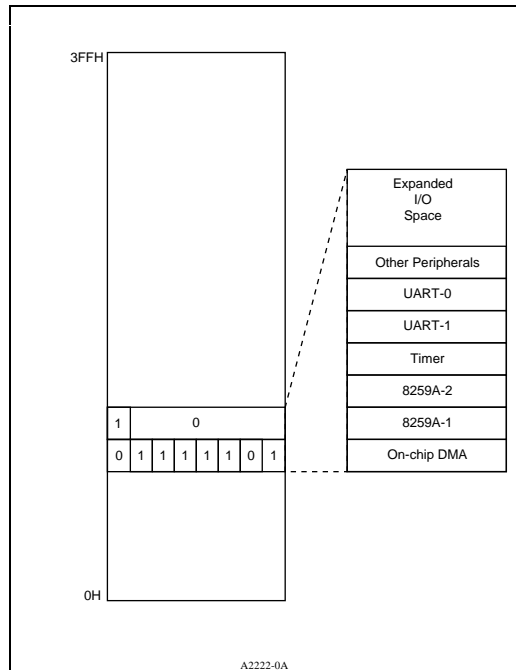


Figure 19. Non-DOS Mode

6. CONCLUSION

There is a growing need in many embedded market segments for processor solutions that enable a fast development cycle and low-power, high-performance applications requiring minimum board space. Intel plans to meet the needs of these market segments with the new embedded Intel386 microprocessor product family. Of the three initial product offerings, the Intel386 SX and Intel386 CX microprocessors offer high-performance computing power without integration. The third member of the family, the integrated Intel386 EX microprocessor, was reviewed. The Intel386 EX microprocessor is an Intel386 architecture-based, PC/DOS-compatible processor that has been optimized for embedded applications.

REFERENCES

[1] Solari, Edward, *AT Bus Design*, Annabooks, San Diego, CA, 1991.