The Intel386™ EX Microprocessor's Enhanced DMA and DOS Compatibility

The new Intel386™ EX microprocessor is aimed at three major market segments:

- those who require DOS compatibility for development in native PC environment,

- those who require PC architecture compatibility to be able not only to develop and debug in the PC environment but also to run off-the-shelf software in conjunction with their own application, and

- the customer base looking for an upgrade path for 80C186 designs.

In addition to a fully static, modular Intel386 SX CPU core, one of the many peripheral functions of the Intel386 EX microprocessor is an enhanced Direct Memory Access (DMA) module. In view of the above market segments, the architectural task force had to strike a delicate balance between providing embedded control customers the performance they require and ensuring that the DMA module and its interface to the CPU remain DOS and PC compatible. This article explains "why we chose what we chose" regarding the DMA block.

The 8237A DMA and Its Limitations

A PC-AT compatible architecture (defined by ISA) incorporates two 8237A DMA controllers connected in a cascaded fashion. One of these DMA controllers allows 8-bit transfers, while the other allows 16-bit transfers through address shifting. In all, seven DMA channels are available. However, there are two major restrictions of this DMA unit:

- The 8237A has only 16-bit addressing capability. Hence, for an embedded processor like the Intel386 EX microprocessor, with a 26-bit (64 Mbyte) addressing scheme, a page register has to be used to allow address extension. This is cumbersome and may affect performance.

- The 8237A does not have a two-cycle data transfer mode to allow memory-to-memory transfers per channel. To accomplish it, two predefined DMA channels have to be used in a very specific manner. For embedded systems, it is common to have memory-mapped I/O devices and transferring data between these devices and memory would not be easy.

The QFD[1] process to determine customer requirements for the Intel386 EX microprocessor revealed the above limitations to be totally unacceptable to embedded control customers. Also, seven channels were deemed redundant.

The Intel386™ EX Microprocessor's DMA

The Intel386 EX microprocessor employs a DMA block that eliminates the above restrictions by providing two independent channels, each capable of transferring data between any combination of memory and I/O devices. The address registers and byte count registers are expanded to support a larger memory space. Other added features include buffer chaining, programmable priority levels, and byte assembly. Details about all the enhanced features of the Intel386 EX microprocessor's DMA will become available at a future date, but this discussion will be limited to DOS compatibility issues only. While providing the above enhancements, the developers have made careful efforts to avoid incompatibility with the PC/DOS standard as follows:

- An extensive study of enhanced DMA units available in various chip sets (some of them fully backward compatible and some not) was performed and features that could be incompatible were removed or modified. For example, even though the Intel386 EX microprocessor's DMA is capable of generating 26-bit addresses directly, an option to prevent the overflow from the lower 16 bits to the higher bits will be provided to emulate the page register on the 8237A.

- All of the 8237A's operating modes are supported (one exception is explained later).

- Also, similar to the 8237A, the Intel386 EX microprocessor's DMA controller is programmed through 8-bit registers.

[1] The Quality Function Deployment (QFD) process is a systematic approach to determine how to deploy available resources to provide product characteristics that customers value the most.

Even with such a conscious effort to maintain 8237A compatibility, it must be understood that the Intel386 EX microprocessor's DMA is **not** exactly an 8237A. The following section describes situations that may require full compatibility with 8237A and explain how those could be resolved in an Intel386 EX microprocessor-based system.

Compatibility for Ease of Development

It would be easier to talk about the ISA (PC-AT) compatibility from the viewpoint of the first two types of the market segments discussed in the beginning. Take the case of the user whose main requirement is to be able to develop and debug a significant portion of an application on the PC platform. Of course, when it is transferred to the embedded target system, timings will be different, but the functionality of the application could be checked out. First of all, DOS does **not** use DMA directly for any of its function calls. For basic PC-AT functionality, DMA is used only for floppy disk accesses through BIOS. A customized BIOS (as planned for the Intel386 EX microprocessor) would provide a transparent solution for similar usage. This, of course, assumes development of a "well-behaved" application that does not access DMA directly for floppy disk transfers.

Second, consider the development of an application that accesses DMA for other application-specific use through the ISA bus. For the two DMA channels offered on the Intel386 EX microprocessor, this should not be a problem, as 8237A registers, register addresses, and the programming and operating modes are maintained as a subset of the Intel386 EX microprocessor's DMA. It follows that the enhanced DMA features of the Intel386 EX microprocessor could not be tested on the PC development system.

Compatibility to Run Off-the-Shelf Applications

The most stringent requirements for compatibility will be demanded by an application design employing an off-the-shelf DOS application and utilizing DMA channels (e.g., a sound card) as one of its subsystems.

As mentioned above, the Intel386 EX microprocessor's DMA maintains compatibility with the 8237A in all respects but one: the absence of a contrived two-cycle transfer mode. Although the Intel386 EX microprocessor implementation is more elegant and desirable, it could be argued that some isolated applications may use that particular feature on the 8237A and may fail while running on an Intel386 EX microprocessor-based system. The following explains why this does not pose a problem.

In an 8237A, DMA channels 0 and 1 must be used to perform memory-to-memory transfers in two cycles. In the earlier PC-XT, DMA channel 0 was used for memory refresh. The subsequent PC-AT does not use DMA for refresh, but for backward compatibility, channel 0 is not re-assigned or made available to perform other functions. Also, channel 0 of the second 8237A (added for 16-bit transfers in PC-AT) is used for cascading. Thus, none of the 8237A could actually be used for two-cycle transfers in a PC-AT architecture. This renders the incompatibility of the Intel386 EX microprocessor's DMA with this feature a non-issue.

Finally, despite the above arguments, when an 8237A is required in an application design, the Intel386 EX microprocessor offers an elegant solution. The Intel386 EX microprocessor employs a special I/O address decoding scheme that "hides" registers of non-DOS peripherals (e.g., chip selects, watchdog timer, etc.) from the 1K I/O address range and decoding scheme that PC-AT architecture uses. The on-chip DMA can be remapped out of the DOS I/O space and hidden in this expanded I/O space. An external 8237A could now occupy the ISA-defined I/O addresses and provide the desired PC-AT compatible functions.

Conclusion

The 8237A DMA controller does not offer the features and performance demanded by embedded control applications. The DMA controller module on the Intel386 EX microprocessor meets these performance requirements and at the same time satisfies the need to be PC-AT architecture compatible in order to serve target market segments requiring DOS compatibility for ease of development and PC-AT architecture compatibility to be able to run off-the-shelf DOS software.