



Interrupt Latency in 80386EX Based System

This document will attempt to explain the Interrupt latency and Interrupt Response Time for a 386EX based systems in Real Mode of operation. Similar explanations can be used with some modifications for the task switch and privilege level verifications, for the Protected Mode of operation, but however, this will be covered at a later time. Intel 80386EX CPU core is same as the 386SX core and hence most of the Interrupt latency descriptions for the SX core would apply directly.

As a start, some of the terminology which will be used in the course of this document will be explained (may be slightly different from other documentation).

Interrupt Latency: The time that elapses before an interrupt request is serviced by the CPU (recognition of the interrupt by the CPU with an interrupt acknowledge cycle). Also called latency time.

Interrupt response time: Time that elapses between the occurrence of an interrupt and the execution of the first instruction of that Interrupt Service Routine (ISR) by the CPU.

In response to an interrupt, the following operations takes place in a microprocessor system while in the Real Mode of operation as indicated below. However, a very similar procedure would be followed in the protected mode except that the Task switch may be necessary and the privilege levels may cause the instruction executions to take longer to execute than in normal real mode operation [1].

- 1. Recognition of the INTR by the CPU.
- 2. The processing of the current instruction is completed.
- 3. Micro-code of the CPU Core executes the INTR (1 cycle for 386SX).
- 4. Get INTR vector from the 8259.
- 5. Branching to the ISR+ save the microprocessor state.
- 6. First instruction of the ISR executed.
- 7. Continue with interrupt service routine.
- 8. Restore the saved status of the microprocessor and return to the instruction that follows the interrupted instruction.

The following diagram illustrates the above sequence: