

# SOFTWARE CONFIGURABLE TECHNIQUE FOR PRIORITIZING INTERRUPTS IN A MICROPROCESSOR-BASED SYSTEM

## FIELD OF THE INVENTION

The invention relates to the field of microprocessor-based systems. More particularly, the invention relates to configuring and prioritizing interrupts in microprocessor-based systems.

## BACKGROUND OF THE INVENTION

Microprocessor-based systems generally operate by sequentially executing instructions of a software program stored in consecutive memory locations. A program counter contains a memory address of a next stored instruction in the sequence. The program counter is incremented for each instruction executed. The software program can include branches taken according to conditions specified in the program. For a conditional branch, if the condition is satisfied, a branch address is placed in the program counter and sequential execution resumes beginning at the branch address.

The microprocessor-based system, however, typically must respond to the occurrence of events that are not synchronized with respect to the stored software program. For example, data from a peripheral device, such as a modem or a keyboard, may need to be processed soon after the data is acquired. Accordingly, microprocessors are typically provided with an ability to receive one or more interrupt signals activated by one or more peripheral devices. In response to an activated interrupt signal, the microprocessor will interrupt the sequence of instructions that is currently being executed and will begin executing another set of instructions, commonly referred to as an interrupt service routine, appropriate to the activated interrupt.

To begin executing the appropriate interrupt service routine, the address in the program counter is saved and a vector address, also referred to as an interrupt vector, is placed in the program counter. There are generally two techniques for determining the appropriate vector address for executing the interrupt service routine: vectored and auto-vectored. When an interrupt is vectored, the peripheral device provides the starting address of the appropriate interrupt service routine. This starting address is placed in the program counter of the microprocessor. Alternately, the address placed in the program counter can be an address of a location in memory which contains the starting address of the interrupt service routine. When an interrupt is auto-vectored, the vector address points to a predetermined location in memory, regardless of which interrupt is active. This predetermined location is typically the starting address of a software routine commonly referred to as an interrupt handler. The interrupt handler then initiates an interrupt service routine that is appropriate for the active interrupt signal.

Once the microprocessor has executed the interrupt service routine, the program counter is restored such that the execution of the original sequence of instructions resumes.

When two or more interrupt signals are simultaneously pending, the microprocessor must respond to the interrupts in an appropriate priority. Therefore, microprocessor-based systems generally provide an ability to prioritize the interrupt signals. In addition, under certain circumstances, it is desired that the microprocessor ignore one or more of the interrupt signals. For example, the microprocessor may be

performing a critical operation that cannot be interrupted. For this reason microprocessors generally include a means for masking interrupts.

An example of a widely utilized microprocessor is the Motorola, Inc. 68000 series processors. The interrupts for a Motorola 68000 series processor are implemented by connecting a hard-wired interrupt signal line from each peripheral device to an interrupt priority encoder. The interrupt priority encoder is a hardware logic circuit which accepts each active interrupt signal and provides only the highest priority active interrupt signal to one of a plurality of interrupt inputs to the Motorola 68000 series processor. The processor then initiates the appropriate interrupt service routine according to the vectored or auto-vectored technique.

A drawback to this technique for implementing the interrupts in the Motorola 68000 series microprocessors is that to change the relative priorities of the interrupt signals, the connections of the hard-wired interrupt signal lines to the interrupt priority encoder have to be re-arranged. Thus, once such a system is configured, the relative priorities among the interrupts cannot practically be altered.

Another example of a commonly utilized microprocessor is the MIPS Technologies, Inc. reduced instruction set computers (RISC), such as the R2000, R3000, R4000, and R6000 processors. The MIPS RISC processor is notified of pending interrupts by setting appropriate bits in a specialized register, referred to as a cause register. The processor then initiates the appropriate interrupt service routine by first executing an interrupt handler. Priority among pending interrupts is determined by the interrupt handler.

A drawback to this technique for implementing the interrupts in the MIPS RISC processors is that to change the relative priorities of the interrupt signals, the interrupt handler has to be altered. Thus, once such a system is configured, the relative priorities among the interrupts cannot practically be altered. In addition, latency in servicing the interrupts is introduced by the time required to access and execute the interrupt handler. In such a system, the latency can reach undesirable levels.

Therefore, what is needed is more readily configurable technique for prioritizing interrupts in a microprocessor-based system.

## SUMMARY OF THE INVENTION

The invention is a software configurable technique for prioritizing and masking interrupts in a microprocessor-based system. A plurality of interrupt configuration registers are provided for mapping a corresponding one of a plurality of interrupt signals to an appropriate one of a plurality of interrupt vector address registers and for masking selected ones of the interrupt signals. Each of the plurality of interrupt configuration registers is software configurable to include a bit pattern that identifies one of the plurality of vector address registers or that indicates that the corresponding interrupt signal is to be masked. The plurality of vector address registers are arranged according to a predetermined priority and each of the plurality of vector address registers is software configurable to contain the starting address of an appropriate interrupt service routine for the corresponding interrupt.

Each of the interrupt signals is coupled to an input of a corresponding one of a plurality of de-multiplexers. Each of the plurality of interrupt configuration registers is coupled to the select lines of a corresponding one of the plurality of de-multiplexers. The outputs of each of the plurality of