mapped to the third highest priority of the vector address registers **200**, while the vector address A' for INT. A is placed in the third highest priority vector address register; the bit pattern 010 is placed in the interrupt configuration register corresponding to the interrupt INT. D, indicating that INT. D is to be mapped to the fourth highest priority of the vector address registers **200**, while the vector address D' for INT. D is placed in the fourth highest priority vector address register; the bit pattern 001 is placed in the interrupt configuration register corresponding to the interrupt INT. B, indicating that INT. B is to be mapped to the lowest priority of the vector address registers **200**, while the vector address B' for INT. B is placed in the lowest priority vector address register.

According to the example illustrated in FIG. **2**A, if an interrupt occurs, the corresponding vector address is supplied to the microprocessor for loading into its program counter. For example, if the interrupt INT. A occurs, the corresponding vector address A' is supplied to the microprocessor. If multiple interrupts are pending, such as the interrupts INT. A and INT. E, then only the vector address of the highest priority pending interrupt is supplied to the microprocessor. In this example, because the interrupt INT. E has a higher priority than the interrupt INT. A, the vector address E' will be supplied to the microprocessor first. Once the interrupt service routine for the interrupt INT. E has been executed, the vector address A' will be supplied to the microprocessor, assuming the interrupt INT. A is still pending.

Assume that the bit pattern 000 corresponds to a masked interrupt. It will be apparent, however, that any bit pattern can be selected to correspond to a masked interrupt. If it is desired to mask one or more of the interrupts INT. A through INT. E, the bit pattern in the configuration register corresponding to an interrupt signal to be masked is replaced with the bit pattern: 000. FIG. **2**B illustrates such an example. The example illustrated in FIG. **2**B is similar to that illustrated in FIG. **2**A except that the interrupt INT. E is masked. Note that because the interrupt INT. E is masked, there is no need to provide a corresponding vector address E'. Therefore, even if the interrupt E is active, no corresponding vector address will be provided to the microprocessor. If one or more of the remaining, non-masked, interrupts INT. A through INT. D is active, however, program flow will be interrupted and the appropriate vector address provided to the microprocessor. Any one, or multiple ones, of the interrupt signals A through E can be masked in this manner.

The present invention provides significant flexibility in configuring the interrupts and the relative priorities thereof. The vector address, masking and relative priority for each interrupt can be changed simply by altering the contents of the corresponding ones of either or both of the configuration registers **100** and vector address registers **200**.

As an example of this flexibility, referring again to FIG. **2**A, assume that upon servicing a highest priority interrupt, it is desired to change the relative priorities such that the highest priority interrupt becomes the lowest priority and the remaining interrupts are incremented in priority. This is accomplished according to the present invention by changing the entries in the interrupt configuration registers **100** and the vector address registers **200** as illustrated in FIG. **2**C. Note that in FIG. **2**A, the interrupt INT. C is mapped to the highest priority location in the vector address registers **200**, whereas, in FIG. **2**C, the interrupt INT. C is mapped to the lowest priority location in the vector address registers **200**. Additionally, in FIG. **2**C, each of the interrupts INT. A, INT. B, INT. D, and INT. E has been incremented in priority relative to their positions in FIG. **2**A.

The above examples assume that each interrupt signal is mapped to a unique interrupt service routine, however, it will be apparent that multiple interrupts can be mapped to a single interrupt service routine (or to a single interrupt handler) according to the present invention. FIG. **2**D illustrates two interrupt signals mapped to a single vector address. In particular, the interrupt signal INT. A and the interrupt signal INT. E are both mapped to the vector address A'.

FIG. **3** illustrates a block schematic diagram of a circuit **300** for implementing the present invention. For ease of explanation, the circuit **300** illustrated in FIG. **3** is for a system having five interrupts, INT. A–E, though it will be apparent that the principles disclosed herein can be applied to form a circuit in accordance with the present invention for a system having more or fewer interrupts. The interrupt signal INT. A is coupled to the input of a one-to-eight de-multiplexer **302**. A configuration register **102** for the interrupt signal INT. A holds three bits. The three bits of the register **102** are respectively coupled to three select lines of the de-multiplexer **302**. The interrupt signal INT. B is coupled to the input of a one-to-eight de-multiplexer **304**. A configuration register **104** for the interrupt signal INT. B holds three bits. The three bits of the register **104** are respectively coupled to three select lines of the one-to-eight de-multiplexer **304**.

The interrupt signal INT. C is coupled to the input of a one-to-eight de-multiplexer **306**. A configuration register **106** for the interrupt signal INT. C holds three bits. The three bits of the register **106** are respectively coupled to three select lines of the de-multiplexer **306**. The interrupt signal INT. D is coupled to the input of a one-to-eight multiplexer **308**. A configuration register **108** for the interrupt signal INT. D holds three bits. The three bits of the register **108** are respectively coupled to three select lines of the de-multiplexer **308**. The interrupt signal INT. E is coupled to the input of a one-to-eight multiplexer **310**. A configuration register **110** for the interrupt signal INT. E holds three bits. The three bits of the register **110** are respectively coupled to three select lines of the de-multiplexer **310**.

A "1" output of the de-multiplexer **302** is coupled to a first input of a logical "OR" gate **312**. A "2" output of the de-multiplexer **302** is coupled to a first input of a logical "OR" gate **314**. A "3" output of the de-multiplexer **302** is coupled to a first input of a logical "OR" gate **316**. A "4" output of the de-multiplexer **302** is coupled to a first input of a logical "OR" gate **318**. A "5" output of the de-multiplexer **302** is coupled to a first input of a logical "OR" gate **320**. The "0", "6" and "7" outputs of the de-multiplexer **302** have no connection.

A "1" output of the de-multiplexer **304** is coupled to a second input of the logical "OR" gate **312**. A "2" output of the de-multiplexer **304** is coupled to a second input of the logical "OR" gate **314**. A "3" output of the de-multiplexer **304** is coupled to a second input of the logical "OR" gate **316**. A "4" output of the de-multiplexer **304** is coupled to a second input of the logical "OR" gate **318**. A "5" output of the de-multiplexer **304** is coupled to a second input of the logical "OR" gate **320**. The "0", "6" and "7" outputs of the de-multiplexer **304** have no connection.

A "1" output of the de-multiplexer **306** is coupled to a third input of the logical "OR" gate **312**. A "2" output of the de-multiplexer **306** is coupled to a third input of the logical "OR" gate **314**. A "3" output of the de-multiplexer **306** is coupled to a third input of the logical "OR" gate **316**. A "4" output of the de-multiplexer **306** is coupled to a third input