2202658

COMPUTER SYSTEM WITH MODE CONVERSION OF COMPUTER COMMANDS

Description

This invention relates to computer systems with interrupt controllers and more particularly to more efficient systems that are conditioned to convert and respond to interrupt-related software commands that are normally incompatible with the system.

Systems of this nature have been described heretofore, such as the IBM Personal Computers XT and AT*, (*Personal Computer XT and Personal Computer AT are trademarks of International Business Machines Corporation) which have one or more programmable interrupt controllers interconnected in the system which handle the interrupts in the system and thereby relieve the Central Processor Unit (CPU) from software and processing overhead. A representative interrupt controller of this nature is the Intel* 8259A Programmable Interrupt Controller (PIC) which handles up to eight vectored priority interrupts for the CPU. (*Intel is a registered trademark of Intel Corporation). The 8259A features are described for example, in the Intel "Microsystem Components Handbook", 1984, Pp 2-120 to 2-137. The 8259A PIC can be programmed by software commands to respond to edge or level triggered interrupt modes. It is desirable with certain computer systems to operate primarily with software that uses only interrupt-related software commands in one mode but to be responsive to interrupt-related software commands from other system configurations or programs that are based on a different mode.

In accordance with the present invention, there is provided a computer system designed to operate normally in response to software commands in a first command mode including a control device programmable under control of a command to operate either in said first mode or a second mode, characterised by detecting means operable prior to transmission of a received command to said control device to detect the mode of the received command and, if it designates said second mode, to modify the command to designate said first mode for transmission to said control device, whereby said control device is programmed to operate in said first mode notwithstanding the designation of the received command.

In the practice of the present invention a significant advantage is gained since software, including applications programs, that would otherwise be incompatible are rendered compatible and a large number of such programs can be used on the system.

In a preferred embodiment to be described in detail later, a personal computer system utilises software that normally supplies only level sensitive interrupt commands at all times. It is desirable to be able to use otherwise incompatible software that supplies edge sensitive interrupt commands. Circuit logic is added to the level sensitive system which intercepts all incoming edge sensitive interrupt commands and treats them in the same manner as level sensitive commands.

The principles of response and conversion can be applied to the converse situation i.e., edge vs. level commands as well as to treatment of command types other than interrupt commands.

The preferred embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 illustrates a computer system including a CPU, a programmable interrupt controller and decode and buffer control logic for processing interrupt command signals in accordance with the present invention;

Figures 2A and 2B illustrate respectively, the format of a representative Initialisation Command Word (ICW1) and an Operation Command Word (OCW1) used in the system of Figure 1;

Figure 3 shows details of the buffer control logic of Figure 1 including a write gate and read gate;

Figure 4 is a timing chart showing signal waveforms occurring during initialization of the programmable interrupt controller, especially showing use of the ICW1 command word;

Figures 5A, 5B, and 5C illustrate logic and conditions pertaining to operation of the write gate in Figure 3; and

Figures 6A, 6B, and 6C illustrate logic and conditions pertaining to operation of the read gate in Figure 3.

The computer system in Fig. 1 includes a Central Processing Unit (CPU) 1 interconnected with a Programmable Interrupt Controller (PIC) 2 and by way of bus 3 (comprising address bus 14 and data bus 17) with other elements including Input/Output (I/O) devices 6-8, a Random Access Memory (RAM) 10 and Read Only Memory (ROM) 11.

Address Signals are directed by bus 14 to Address Decode and Instruction Decode block 15 and Data Signals by data bus 17 to block 15 and Buffer Control Logic 20. Various data, status, and control signals are supplied such as Chip Select on line 25, Address "0" on line 26 and Decode Result on line 27, which represents a -Write Gate (line 27a, Fig. 3) or -Read Gate signal (line 27b, Fig. 3). Buffer Control provides data by way of bus 17a to Interrupt Controller 2.

Programmable Interrupt Controller 2 handles up to eight vectored priority interrupts for CPU 1 by way of the eight line IR0, IR1, IR2, etc. to IR7 via bus 30. Only three I/O devices 6-8 are shown connected in the system of Fig. 1 to lines IR0, IR1 and IR2.

Interrupt Controller 2 functions as an overall manager. It accepts requests from the I/O devices, determines which of the incoming requests is of the highest importance (priority), ascertains whether an incoming request has a higher priority value than the level currently being serviced, and issues an interrupt via (INT) line 31 to the CPU based on this determination.

Each I/O device usually has a special program or "routine" that is associated with its specific functional or operational requirements which is referred to as a "service routine". Interrupt Controller 2, after issuing an interrupt to CPU 1, supplies information to CPU 1 that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and is typically referred to as vectoring data.

A selection of priority modes is available to the programmer so that the manner in which the requests are processed by Interrupt Controller 2 can be configured to match system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. Thus, the interrupt structure can be defined as required.

The interrupt sequence is fully described in the above referenced Microsystem Components Handbook.

Programming the 8259A

Interrupt Controller 2 accepts two types of command words generated by CPU 1.

(1) Initialisation Command Words (ICWs): The format of ICW1 is shown in Fig. 2A. Before normal operation can begin, Interrupt Controller 2 must be brought to a starting point by a sequence of 2 to 4 bytes timed by WR pulses.

(2) Operation Command Words (OCWs): The format of OCW1 is shown in Fig. 2B. These are the command words which command the 8259A to operate in various interrupt modes. These modes are described in the referenced handbook and are:

a) Fully nested mode
b) Rotating priority mode
c) Special mask mode

d) Polled mode

The OCWs can be written into the 8259A anytime after initialisation.

Whenever a command is issued with A0=0 and D4=1, this is interpreted as Initialisation Command Word 1 (ICW1), Fig. 24, ICW1 starts the initialisation sequence during which the following automatically occur.

a) The edge sense circuit is reset, which means that following initialisation, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
b) The Interrupt Mask Register is cleared.
c) IR7 input is assigned priority 7.
d) The slave mode address is set to 7.
e) Special Mask Mode is cleared and Status Read is set to IRR.
f) If IC4=0, then all functions selected on ICW4 are set to zero.

The edge and level triggered modes are programmed using bit 3 in ICW1. Refer to Fig. 2A.

If LTIM='0', an interrupt request will be recognised by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

If LTIM='1', an interrupt request will be recognised by a 'high' level on an IR input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The computer system in Fig. 1 is configured to normally operate with software, such as application programs, which supply interrupt

related commands (ICW1) at initialisation time that primarily establish a level sensitive mode for Interrupt Controller 2. It is desirable to maintain Interrupt Controller 2 in such level sensitive mode to enhance performance and "reduce noise problems" which result in false detection of an interrupt. When this is done, however, software that supplies edge sensitive command signals such as an edge mode ICW1 at initialisation time becomes incompatible. If the Interrupt Controller 2 is programmed to edge triggered mode, by software written for other computer systems, the system will not complete the interrupt sequence. Logic is added to prevent initialising the controller to edge triggered mode. This feature enables program compatibility with other systems. A system without this feature is likely to be less compatible to pre-existing Personal Computer Software applications.

In Fig. 1, whenever an edge trigger command is decoded by block 15, the Buffer Control logic block 20 traps the commands and the controller is conditioned to respond in the same manner as if a level sensitive command was received.

The edge triggered mode occurs during the first initialisation command word (ICW1). The occurrence of this condition is detected and the LTIM data bit for mode set on bus 17a is forced to be at a level sensitive mode. The logic required for this function in block 20 includes tri-state buffers to gate the bi-directional data, with a small amount of support logic to do the decode. Details are shown in Fig. 3.

In Fig. 3, a data buffer 35 is shown that supplies data via bus 17a to the Interrupt Controller (PIC) 2. Another PIC 2a can be provided, if desired, with appropriate selection of either one during operation. Buffer Control Logic 20 comprises blocks 40 and 47 and a pull up resistor 52. As noted the Decode Result 27 represents either a -Write Gate Signal on line 27a or a -Read Gate Signal on line 27b from a port in 15a of the Address Decode and Instruction Block 15. A -Write Gate Signal to block 40 controls write operations involving Interrupt Controllers 2 or 2a. Other Write logic includes blocks 43-45. A -Read

Gate Signal to block 47 controls read operations involving Interrupt Controllers 2 and 2a. Other Read logic includes blocks 48 and 49.

In Fig. 3, when writing a command word to one of the Interrupt Controllers 2 or 2a, the -Write Gate on line 27a is active except during a write for the command ICW1. By not activating -Write Gate during a write for ICW1, Data Bit 3 in the command is maintained at a high value by the pull-up resistor 52. When Data Bit 3=1 during a write to ICW1, the selected Interrupt Controller is set to or remains at level-sensitive mode. The status and conditions of the various control and data lines for Read and Write operations, but especially Write ICW1 are shown in Fig. 4.

The logic development for control of Write operations is illustrated in Figs. 5A - 5C as based on the following parameters:

Write Gate

Purpose

Do not allow data from the XDATA bus to be directed to the 8259 when the first initialisation command word (ICW1) is being written.

ICW1 = I/O Write when (A0=0 and D4=1)
(Trap)

The logic development for control of Read operations is illustrated in Figs. 6A-6C as based on the following parameters:

Read Gate

Purpose

Allow data from the 8259 to be directed to the XDATA bus during a read cycle to the chip or during an INTA cycle to the chip.

Therefore, whether the controllers were intended to be programmed to level or edge mode, the circuit always modifies the command to level mode.

Further, it is evident that the inventive principles set forth herein can be applied to level to edge command conversion or to conversion of command types other than interrupt commands and under other circumstances where it is desirable to establish compatibility among several otherwise incompatible systems, programs, or software.

Additionally or alternately, the Write gate signal may be used to signal the processing unit that an edge triggered command has been detected. This information may be used by the processor to flag other defects (time dependant code, incomplete I/O decodes, attempts to transfer data with unattached equipment) which may be associated with edge triggered commands. The processing unit may then elect to discontinue the operation, modify the operation or continue with the defects.

Claims

1. A computer system designed to operate normally in response to software commands in a first command mode including a control device (2) programmable under control of a command to operate either in said first mode or a second mode, characterised by detecting means (15) operable prior to transmission of a received command to said control device to detect the mode of the received command and, if it designates said second mode, to modify the command to designate said first mode for transmission to said control device, whereby said control device is programmed to operate in said first mode notwithstanding the designation of the received command.

2. A system according to claim 1, in which said software commands are interrupt commands and ~~saves~~ said control device is a programmable interrupt controller.

3. A system according to claim 1 or claim 2, in which said detecting means includes write and read control logic (40, 47) for controlling transfer of data to and from said control device during system operations and operable to decode each software command designating said second mode and to force the command to designate said first mode.

4. A system according to any of claims 1 to 3, in which said detecting means is operable to generate a mode conversion signal in response to detection of a command designating said second mode, said mode conversion signal serving as a flag to indicate whether software which generated the command is compatible with the system.

5. A system according to any of the previous claims, in which each said command comprises a predetermined number of data and command bits, and said detecting means logically inspects predetermined bit position in the command to determine whether the command designates the first or the second mode.

6. A system according to any of the previous claims, in which said modes comprise edge triggered mode and level sensitive mode.

7. A system according to claim 6, in which said first mode is level sensitive mode and said second mode is edge triggered mode.