

COMPUTER SYSTEM HAVING DIRECT MEMORY ACCESSDescription

The present invention relates to a computer system with a DMA (Direct Memory Access) controller with which peripheral devices of number greater than the number of physical DMA channels can perform DMA data transfers.

Many present-day computer systems employ DMA channels to allow peripheral devices to transfer data, primarily to and from the main memory, without going through the CPU (Central Processing Unit). Bypassing the CPU of course has the advantage of increased data transfer rates and improved overall system efficiency because the CPU is free to perform other tasks during the data transfer.

The most common practice to date has been to provide one physical DMA channel for every peripheral unit that is allowed DMA access. An example of a computer system which employs such an arrangement is the IBM Personal Computer.

However, the number of peripheral devices which a user may attach to a computer system has been increasing and is expected to continue to do so. For instance, lately there have become available optical disk readers, additional communications devices, hard files, tape backup devices, high speed printers, etc., all of which can make advantageous use of DMA access.

On the other hand, adding further physical DMA channels is costly of system hardware, board space, and bus space. It has been considered to share DMA channels among peripheral devices with the restriction of mutually exclusive operation. Sharing of DMA channels is employed in, for instance, the IBM XT and AT computer systems. These schemes require extensive modifications to the operating system as additional subroutines are required to prevent the peripheral devices associated

with each DMA channel from operating simultaneously. A significant performance burden is also created when file controllers are not sufficiently "intelligent" to allow overlapped seek operations. The problem is compounded in certain situations, for example when a LAN (Local Area Network) feature and a file controller share the same DMA level and a LAN file server routine must trade back and forth between the two devices.

US-A-3,925,766 to Bardotti et al. discloses a computer system wherein peripheral devices are assigned priority levels, and requests for access to communicate with a memory are selected according to the priority levels. The priority level assignments can be changed according to the causes for which the request signals are generated or the particular load conditions of the central processor. No DMA paths are provided, however.

In US-A-4,400,771 to Suzuki et al. there is shown a multi-processor system in which each of the multiple processors can access a single memory. A programmable register circuit stores priority information designating a memory access grade priority for each of the processors. The priority information can be changed manually, by an external circuit, or by at least one of the processors. The Suzuki et al. patent does not employ DMA access to a memory from plural peripheral devices.

US-A-4,257,095 to Nadir is of interest in its teachings regarding bus arbitration techniques.

The following United States patents are mentioned for general background teachings related to computer systems employing DMA controllers: 4,371,932 to Dinwiddie, Jr., et al., 4,419,728 to Larson, 4,538,224 to Peterson, 4,556,962 to Brewer et al., and 4,584,703 to Hallberg.

It is an object of the present invention to provide a computer system employing a DMA controller wherein peripheral devices in a number

greater than the number of physical DMA channels provided in the system can share the DMA channels without encountering the problems of the prior art approaches discussed above.

According to the invention there is provided a computer system comprising a multi-channel direct memory access (DMA) device, and means for coupling a plurality of peripheral devices to the DMA device, said means for coupling being capable of coupling a number of peripheral devices, greater than the number of DMA channels, to the DMA device, the system being characterised in that each peripheral device has a respective channel assignment value associated therewith and at least one DMA channel is shared between peripheral devices through an arbitration control circuit, the arbitration control circuit comprising means for comparing the channel assignment value from a peripheral device requesting access to a DMA channel with channel assignment data programmed into the arbitration control circuit to determine current access to a shared channel.

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

Fig. 1 is a block diagram showing a computer system employing the invention;

Fig. 2 is a conceptual diagram used to explain the operation of the present invention;

Fig. 3 is a logic diagram depicting in detail an arbitration circuit provided in peripheral devices of the computer system of Fig. 1;

Fig. 4 is a diagram showing details of one of the buses in the computer system of Fig. 1;

Fig. 5 is a detailed logic diagram of a central arbitration control circuit employed in the computer system of the invention; and

Fig. 6 is a detailed logic diagram of a DMA control circuit used in the invention.

Referring first to FIG. 1 of the accompanying drawings, there is shown a block diagram of a computer system in which the present invention is used to advantage.

A CPU communicates with a main memory 15, bus controller 16 and math co-processor 14 via a system bus 26. Communication between the CPU and its associated peripheral devices is through a bus controller 16, the latter being coupled to the peripheral devices through a family bus 25. In the present example, the peripheral devices include an auxiliary memory 17, two communications devices 18 and 19, a hard file 20, an optical disk 21, and two floppy disks 23. Other peripherals can of course be used as well as system needs dictate. The peripheral devices are represented generically by DMA slave 24.

A DMA controller 12 is provided to allow at least selected ones of the peripheral devices direct memory access. For this purpose, as will be explained in more detail below, the family bus, or at least a portion thereof, is branched to the DMA controller 12. Each peripheral allowed DMA access is provided with an arbitration circuit 24, and each peripheral having an arbitration circuit is assigned an arbitration (priority) level. Again as will be explained in more detail later, a central arbitration control circuit 11 is associated with the DMA controller to arbitrate among peripheral devices concurrently requesting DMA access and to inform the DMA controller of which peripheral is to have access.

In the computer system embodying the invention, the number of peripheral devices to be allowed DMA access is greater than the number of physical DMA channels provided in the system. In accordance with the present invention, some of these devices are allotted their own dedicated DMA channel, while others share the remaining DMA channels. On the shared (programmable) channels, access is in order of preassigned priority.

In the present example, it is assumed that there are eight physical DMA channels, designated 0 through 7. It is further assumed that channels 0 and 4 are shared, and that remaining channels 1-3 and 5-7 are dedicated to individual peripherals.

Referring now to FIG. 2, which is a conceptual drawing used to explain the principles of the present invention, a comparator is provided with two sets of inputs. One set is composed of four lines from a bus ARBUS (ARbitration Bus). The value on the ARBUS is indicative of the peripheral currently requesting a DMA channel having the highest arbitration (priority) level. If that peripheral is one of those having one of the dedicated channels 1-3 or 5-7, access is granted directly to that channel. On the other hand, if the peripheral is one of those having to share a programmable DMA channel (channels 0 and 4), access is granted only if its priority level corresponds to one of the values preset in registers 6 and 7. This operation will be explained in more detail below with reference to Figs. 3-6.

FIG. 3 is a logic diagram of one of the arbitration circuits 28 used in each of the peripherals allowed DMA access. The arbitration level assigned to the peripheral is set in a register 70, hereinafter referred to as the channel priority assignment register. This may be done using any of a number of well-known techniques. Preferably, the CPU addresses the peripheral through a preassigned port so that the arbitration level can be set through software. This may be done through the BIOS (Basic Input-Output System), POST (Power-On Self Test) at reset, the operating system, or the application program as desired. Otherwise, it is possible to input the channel priority assignment value with hardware switches.

The actual arbitration circuitry is implemented with of an arbiter 72. This circuit, including the various signals indicated in FIG. 3, is disclosed and discussed in detail in American National Standard/IEEE Standard No. ANSI/IEEE Std 696-1983, published by the Institute of Electrical and Electronics Engineers, Inc., June 13, 1983. In general,

each device incorporated in the arbitration scheme is provided with such a circuit, and the circuits are connected together via an ARBUS. The ARBUS in the example shown has four data lines TMA0 - TMA3, allowing for 16 different arbitration levels. Any desired number though can be used. During an arbitration time period designated by the control signals pHLDA and HOLD, all devices desiring to gain control over the bus (those having the IWANT signal set to the high ("1") state) transmit their arbitration (priority) levels onto the ARBUS. This occurs in the example of FIG. 3 when the signal /APRIO ("/" indicating a "low-truth" signal) goes to the high ("1") state. The value then held in the register 70 is gated through the AND gates 71 to the arbiter 72 and thence onto the ARBUS lines TMA0 - TMA3. At the end of the arbitration period, the value on the ARBUS lines TMA0 - TMA3 will be the value of the arbitration level of the device having the highest level. The waveforms of the various signals mentioned here are shown in detail in the referenced ANSI/IEEE standard.

In the preferred embodiment under discussion, the ARBUS forms a part of the family bus 25 coupling the peripherals to the bus controller 16 and thence to the CPU 10. The relationship of the ARBUS to the overall family bus 25 is depicted in FIG. 4.

The details of the central arbitration control circuit 11 and the DMA controller 12 are shown in Figs. 5 and 6, respectively, to which reference is now made.

As indicated in FIG. 5, the signals HOLD and pHLDA from the ARBUS are used to generate a signal /ARBTIME. /ARBTIME is in the "0" state during the time when arbitration is to take place among the peripheral devices competing for a DMA channel. It remains in the "0" state long enough for the signals on the ARBUS to have reached steady-state conditions, that is, sufficiently long for the arbitration to be completed. To generate /ARBTIME, pHLDA is inverted by an inverter 61, then applied together with HOLD to respective inputs of an EXCLUSIVE-OR gate 62. A one-shot multivibrator 63, having an output pulse period

longer than the arbitration time on the ARBUS, is triggered by the leading edge of the output pulse from the EXCLUSIVE-OR gate 62. The output of the one-shot multivibrator is ORed with the output of the EXCLUSIVE-OR gate 62 by an OR gate 64 to thus produce /ARBTIME. Other arrangements are of course possible for generating /ARBTIME. In one approach, DMA request signals from each device are ORed together and the ORed output applied to generate a pulse signal of appropriate length. The basic requirement in any case is to set /ARBTIME to the "0" state to define the time periods when arbitration is taking place on the ARBUS.

One compare logic 40 is provided for each of the programmable physical DMA channels in the system, and one compare logic 49 is provided for each of the fixed channels. Each compare logic 40 for the programmable DMA channels (channels 0 and 4 in the present example) includes a register 41, referred to as a DMA channel assignment register, which is loaded by the CPU with a DMA channel assignment. The compare logics 49, namely, those for the fixed channels, are identical to the compare logics 40, except that the register 40 is replaced by a set of switches with which the channel assignment is manually set. Only one each of the compare logics 40 and 49 are shown in detail since the others ones of the compare logics 40 and 49 are identical and their input signals are the same.

The outputs from the registers 41 for the programmable channels (compare logics 40) and the outputs of the switches for the fixed channels (compare logics 49) are compared with the signals TMA0 - TMA3 by a set of exclusive-OR gates 42, the outputs of which are applied to inputs of a NOR gate 43. If a match is present between the channel assignment and the value represented by TMA0 - TMA3 at the end of the arbitration period, that is, if the corresponding signals are then in the identical states, the output from the NOR gate 43 (COMPARE 0 - COMPARE 7) will be in the "1" state. Of course, only one NOR gate 43 at a time can have an active output.

With reference now to FIG. 6, at the end of the arbitration time when /ARBTIME goes back to the "1" level, a "1" from the one of the signals COMPARE 0 - COMPARE 7 in the "1" state will be set in one of the two latches 51 of the DMA controller 12. The other ones of COMPARE 0 - COMPARE 7 will be in the "0" state, and hence a "0" will be set in the corresponding positions in the latches 51.

The outputs of the latches 51 are applied to corresponding DMA request inputs (DREQ0 - DREQ3) of two cascade-connected DMA controller IC's 52. In the embodiment under discussion, the IC's 52 are each a type 8237 programmable DMA controller manufactured by Intel Corporation. The DMA controller IC's 52 are cascade connected using NOR gates 53. The CPU may be any of the Intel iAPX 86 series of microprocessors, such as the 8088, 8086 or 80286. For details of the connections between the two DMA controller IC's and the CPU, reference may be made to the 1985 Intel Microsystem Components Handbook, pages 2-57 to 2-71.

Accordingly, for a peripheral having a dedicated DMA channel assignment, once the peripheral wins on the ARBUS, it is guaranteed immediate use of a DMA channel. For a peripheral sharing one of the DMA channels, when it wins on the ARBUS, it will gain the immediate use of a DMA channel only if its channel priority assignment value, as held in its channel priority assignment register 70, matches one of the DMA channel assignment values set in one of the two registers 41. Of course, the BIOS, operating system, or applications program can continually reprogram the channel assignment values held in the DMA channel assignment registers 41 of the two programmable channel compare logics 40 to assure that all peripherals needing access eventually are given the use of a DMA channel. Many different programming schemes can be implemented to control the values stored in the DMA channel assignment registers 41 and the channel priority assignment registers 70 of the arbitration circuits depending upon the application at hand. In a simple example, the values set in the priority assignment registers 70 are fixed and the values set in the DMA channel assignment registers 41 are rotated among at least some of the values in the priority assignment

registers 70 to thus give each peripheral associated with one of the programmable DMA channels a chance to gain the use of a DMA channel. If "intelligent" peripherals are employed having their own controller, more complex schemes may be implemented. For example, when a peripheral assigned to a programmable DMA channel wishes access, the peripheral can query the operating system or BIOS as to the availability of one of the programmable channels. If a channel is available, its number can then be set in the registers 70 and 41 to guarantee access. However, since the selection of the programming scheme is beyond the scope of the present invention and in the province of the user, further examples will not be discussed here.

Other applications of the invention are also contemplated. For example, programmable DMA channels provided by the use of the invention can be used as redundant back-up channels for ones of the dedicated channels, thereby providing improved system reliability.

CLAIMS

1. A computer system comprising a multi-channel direct memory access (DMA) device (12), and means for coupling a plurality of peripheral devices (17-24) to the DMA device, said means for coupling being capable of coupling a number of peripheral devices, greater than the number of DMA channels, to the DMA device, the system being characterised in that each peripheral device has a respective channel assignment value associated therewith and at least one DMA channel is shared between peripheral devices through an arbitration control circuit (11), the arbitration control circuit (11) comprising means for comparing (42, 43) the channel assignment value from a peripheral device requesting access to a DMA channel with channel assignment data programmed into the arbitration control circuit to determine current access to a shared channel.
2. A computer system according to Claim 1, in which each peripheral device (17 to 24) includes an arbitration register (70) for holding said channel assignment value, said register being coupled for loading from the system CPU (10).
3. A computer system according to Claim 1 or Claim 2, in which at least one DMA channel is a dedicated channel coupled to a single peripheral device, and for the, or each, dedicated channel, the arbitration control circuit (11) compares the channel assignment value from the coupled peripheral device with preset channel assignment data to determine access to the dedicated channel.
4. A computer system according to Claim 3, in which said preset channel assignment data is determined by manual switch means in the arbitration control circuit.
5. A computer system according to any of the previous claims, in which, for the, or each, shared DMA channel, the arbitration control circuit includes register means (41) for receiving channel assignment



data from the system CPU and compare circuit means (42, 43) for comparing data from the register means with said channel assignment value from a peripheral device requesting access to a DMA channel to provide a compare output when the compared data is equal.

6. A computer system according to Claim 5, in which said compare output provides a DMA request input to the DMA device.
- 

