Description

The present invention relates to a data processing system including a main processor unit and a co-processor unit and tackles the problem of the efficient handling of interrupts to the main processor in response to the generation of error signals by the co-processor.

Main multiprocessor systems have been shown in the prior art. These vary from systems in which each processor operates substantially independently within the system to those in which one processor is slaved to the other. In one group of such systems, each processor has full access to other units in the system with full access to I/O and memory devices with bus contention devices determining access of the processors to address and data busses. In a further group, one processor is defined as a main processor and a further processor as a co-processor. In systems of this group, the co-processor is used to enhance the performance of the system for certain classes of operations, such as numeric processing. Both processors have access to both the system address and data busses, but the main processor exerts control over the co-processor for operand transfers. One example of such a system is the iAPX 88/20 system produced by Intel Corporation and comprising an 8088 CPU with an 8087 numeric processor extension.

In another group of systems, the co-processor is slaved to the main processor. In such systems, the co-processor has access to the system data bus, but not to the address bus. Thus, operand transfer to or from the co-processor is fully controlled by the main processor. In effect, the main processor looks upon the co-processor as an I/O device. One example of such a system is the iAPX 286/20 system produced by Intel Corporation and comprising an 80286 main processor with an 80287 numeric processor extension. The architecture of this system is described briefly in I.E.E.E. Electro. Vol. 8 Suppl., 1983, pages 7/2 1-5. It is to systems of this type that the present invention is directed.

Referring back to the iAPX 88/20 system, unmasked exceptions during operation of the 8087 co-processor are indicated by means of an interrupt (INT) output. This signal is normally applied directly to a selected one of the interrupt request inputs of an interrupt controller. These inputs each have a different priority, so the designer can designate a particular interrupt priority for co-processor exceptions.

In the iAPX 286/20 system, the 80287 co-processor has an error output pin. In the recommended circuit for this system, as illustrated in the article referred to, this pin is directly coupled to an error input of the 80286 main processor. A signal on this input interrupts the main processor with an internally defined vector value, and no interrupt acknowledge cycles are performed, as is the case with other interrupts which are handled via a priority sensitive interrupt control unit external to either processor. This creates a problem for a designer if he wishes to build a machine using an iAPX 286/20 system which is compatible with previous machines using an iAPX 88/20 system. This is because he has no choice in determining the main processor interrupt used in response to co-processor errors, and he can not use acknowledge cycles for such interrupts.

The approach of the present invention is to provide a logic system responsive to error signals from a co-processor to generate control signals for handshaking interrupt sequences between a priority interrupt controller and a main processor.

In both the iAPX 286/20 and iAPX 88/20 systems, the co-processor includes a busy output pin. An active signal on this pin indicates the operating condition of the co-processor, and this signal, on occurrence of an error, remains active until the error condition is cleared. It has been found that in some cases spurious signals on the error output can appear when the co-processor is not busy. This, of course, means that the main processor may be interrupted unnecessarily.

The present logic system is responsive only to both error and busy signals from a co-processor to initiate interrupt sequences, using the interrupt control unit, for a main processor.

Collecting these points together, according to the invention there is provided a data processing system comprising a main processor and a co-processor having a busy output which, when active, indicates that the co-processor is executing a command and is not available for external data transfer, and an error output which, when active, indicates a co-processor malfunction, a priority sensitive interrupt controller, external to either processor being provided to handle interrupts, characterized in that there is included a logic circuit having inputs coupled respectively to the error and busy outputs of the co-processor and an output connected to the main processor via the interrupt controller, and including first logic means responsive only to coincident active error and busy outputs from the co-processor to generate an interrupt request signal directed at the main processor to initiate an error interrupt routine.

The present invention will be described further by way of example with reference to an embodiment thereof as illustrated in the accompanying drawings in which:

Figure 1 is a block digram showing major components of the embodiment; and

Figure 2 is a block diagram of an interrupt control logic circuit employed in the system of Figure 1.

Figure 1 is a simplified block diagram of the system, showing its main processor 1, co-processor 2 and coupling logic. Only those components and connections relevant to the present invention are shown in order to set out the features of the invention clearly. It is further pointed out that the type of logic used is sometimes referred to as negative logic with the result that many of the lines carry signals which indicate the absence of the corresponding state of signifi-

cance and the presence of a small circle at the point of joining of a line to a logic element indicates that that element response to or emits signals of reversed polarity.

The main microprocessor 1 and mathematical co-processor 2 are respectively type 80286 and type 80287 processors manufactured by Intel Corporation which together form an iAPX 286/20 computing system. Clocking signals (not shown) for the processors are provided by an 82284 clock generator and ready interface unit 3, also manufactured by Intel Corporation. Unit 4 is a type 82288 bus controller and unit 5 is an 8259A programmable interrupt controller, both of which are also produced by Intel Corporation. One interrupt request input 7 to interrupt controller 5 is provided by a logic circuit 6. Details of this circuit, which may be implemented in discrete form or in the form of a programmable array logic (PAL) chip, are shown in Figure 2. The final element in Figure 1 is a decoder circuit, which is responsive to selected address lines from microprocessor 1 to prdouce chip select signals for co-processor 2.

As mentioned above, the Figure 1 is a simplified diagram of the system, and with reference to the 80286 microprocessor 1, a number of the 63 pin connections have been omitted. Those shown include the status outputs $\overline{S0}$ and $\overline{S1}$ which indicate initialisation of a bus cycle and, in conjunction with other outputs, including the M10 (memory, not I/O) output, indicate the type of bus cycle to be performed. The outputs of interest in this description are interrupt acknowledge (M10, $\overline{S1}$, $\overline{S0}$=0), and I/O read (M10, $\overline{S1}$=0, $\overline{S1}$=1). D0 through D15 are the 16 bit data bus input/output pins. The INTR pin is an interrupt request input which, when high, requests the processor to suspend its current program to service an external interrupt, the $\overline{BUSY}$ pin is used to indicate the operating condition of co-processor 2, it is active, that is low, when the co-processor is carrying out an operation. Normally this pin is coupled directly to a $\overline{BUSY}$ pin on the co-processor, but in accordance with the present invention, it will be seen that it is coupled from logic circuit 6 through a line 10. As mentioned above, outputs from a M10 in operate in conjunction with the $\overline{S0}$ and $\overline{S1}$ outputs, the M10 output, when high, indicates that a memory cycle is in progress, when low it indicates an I/O cycle or an interrupt acknowledge cycle. Lastly, A0 through A23 provide a 24 bit address bus output.

Turning now to the 80287 co-processor 2, the RESET input, when high causes the co-processor to terminate is present activity and to enter a non-operative state. The $\overline{NPSI}$ (numeric processor select 1) input on line 18 is used to select the co-processor, data transfers to or from the co-processor can only take place when this pin is low. The $\overline{S0}$ and $\overline{S1}$ inputs are used to monitor ESCAPE instructions in micro-processor 1, which instructions control data transfer between the processors. The $\overline{ERROR}$ output, when low, indicates that an unmasked error condition has arisen in the co-processor. This output is normally coupled to an $\overline{ERROR}$ input in the 80286 main processor 1, but in accordance with this invention, it provides an input to logic circuit 6 over line 11. Lastly pins D0 through D15 are coupled, as inputs or outputs, to a data bus 13.

Note that the 80287 has no address outputs. Management of data transfer to and from the co-processor is controlled by main processor 1. In effect, to the main processor, the co-processor looks like an I/O device selected by address outputs which are decoded by a decoder 8 to provide a co-processor chip select signal on line 12 to logic circuit 6 and from this circuit to co-processor 2 over line 18.

The 82284 clock generator and ready interface unit 3 provides a system clock output (not shown) for processors 1 and 2 and a bus controller 4. In response to the $\overline{S0}$ and $\overline{S1}$ inputs, together with control inputs (not shown), interface unit 3 develops a $\overline{READY}$ output which, when low, indicates the end of a bus cycle. A RESET output is used to enter the system into an initial reset state on "power on".

The 82288 bus controller 4 is responsive to a READY signal from clock generator 3 to prepare for the next bus cycle. The function of this bus cycle is determined by the values of the signals $\overline{S0}$, $\overline{S1}$ and M10 from processor 1 as set out in the above description of the outputs of that processor. It has nine outputs to effect the control of data flow through the system, such as I/O read, I/O write, memory read and memory write. The two outputs shown are $\overline{INTA}$, the interrupt acknowledge output which, when low, tells an interrupting device that its interrupt request has been acknowledged, and $\overline{IOWC}$, the I/O write command which, when low, instructs an I/O device to read data from the data bus.

Unit 5 is the 8259A programmable interrupt controller. This includes a number of interrupt request inputs IRQ0 through IRQ7. These are coupled to receive interrupt requests from I/O devices and have different priorities with IRQ0 the highest and IRQ7 the lowest. An interrupt request input is an active high signal. The 8259A evaluates requests, and in response to that one with the highest priority, sends an interrupt (INTR) active high signal over line 14 to the INTR input of main processor 1. When the system replies with an interrupt acknowledge signal ($\overline{INTA}$, active low) on line 15, register bits within the 8259A indicating the highest priority interrupt input and the level actually being serviced are used to generate a corresponding 8 bit subroutine address on lines D0—D7 of the data bus 13 identifying the interrupt subroutine to be taken. Upon receipt of a further $\overline{INTA}$ signal, a lower 8 bit subroutine address is released to the data bus 13 for receipt by the main processor 1. Each of these addresses address points to the main processor routine used to service the interrupt.

Decoder 8 is employed to produce a number of control outputs, of which only one, the co-processor chip select output is shown. This output is

active low when address inputs A5—A7 from main processor are high, address inputs A8 and A9 are low and the processor is not emitting a hold acknowledge (HLDA) signal.

In operation, logic circuit 6 looks for an active error indication, $\overline{ERROR}$ signal down, when the $\overline{BUSY}$ signal from co-processor is also down, indicating that the co-processor is active. When these signals coincide, the $\overline{BUSY}$ 286 output on line 10, which normally corresponds to the $\overline{BUSY}$ signal from the co-processor is latched and the INTR output on line 7 goes high. This provides an interrupt request to the interrupt controller. When this request is the highest priority request, the interrupt controller issues a corresponding interrupt signal, over line 14, to the main processor INTR input. Upon receipt of this input, main processor 1 suspends its current program and initiates two interrupt bus cycles. On each of these, the M10, $\overline{S0}$ and $\overline{S1}$ outputs are low, so bus controller 4 issues a $\overline{INTA}$ signal. On the first of these cycles, interrupt controller 5 is set up to define a main processor routine which is to service the interrupt by defining a pointer to this routine. Interrupt controller 5 does not drive the data bus at this time. During the second interrupt bus cycle, that is, when the main processor issues the second $\overline{INTA}$ signal, interrupt controller 5 releases an 8 bit pointer, defining the interrupt routine, on to the data bus from which it is read by the main processor. In the main processor, the pointer initiates an error handler routine to clear the co-processor error condition. During this routine, the main processor produces particular outputs on the A0 through A9 address lines to define particular functions. The functions of interest with respect to the present invention are as follows:

Co-processor chip select:

A8, A9=0, A3, A5, A6 and A7=1. Note that this particular function is not limited to the error handling routine. It is used to select the co-processor for data transfer on any I/O write command to the co-processor.

Clear busy:

A0, A3, A8, A9=0, A5, A6 and A7=1.

Co-processor reset:

A3, A8, A9=0, A0, A5, A6, A7=1. Note that this particular function must be used to exit the co-processor from its protected mode. In these groups, bits A8 and A9, when zero, define the main processing unit and not attached I/O devices, bits A5, A6 and A7, when one, define the co-processor, and bits A0 and A3 define the functions as shown.

At the end of the second $\overline{INTA}$ signal, interrupt controller 5 is automatically reset to service the next interrupt request. At the end of the error handler routine, main processor 1 and co-processor 2 continue with their defined processing functions.

Figure 2 shows details of logic circuit 6. This circuit includes three latch circuits which generate internal signals. These latches comprise AND gate 36 and OR gate 25 for a signal $\overline{Q1}$, AND gate 37 and OR gate 39 for a signal $\overline{Q2}$ and AND gate 33 and OR gate 26 for a signal $\overline{Q3}$. The function of these latches is to provide a latched high output $\overline{Q3}$ on line 27 when the $\overline{INTR}$ signal on line 24 goes low, and to clear this latched condition when a clear busy (CLRBSY) signal from OR gate 28 goes high.

An OR gate 20 monitors the $\overline{ERROR}$ input and, through AND gate 21, the $\overline{BUSY}$ input. The output ($\overline{INTR}$) of OR gate gate 20 is normally high, but goes low when both the $\overline{BUSY}$ and $\overline{ERROR}$ inputs go low, indicating a co-processor error signal when that processor is in the busy condition. The low $\overline{INTR}$ signal is latched through AND gate 21 to provide a latched high INTR signal for the interrupt controller at the output of inverter 23. The purpose of this circuit is to ensure that a latched INTR signal is generated only if a co-processor error signal is generated when the co-processor is busy. When the co-processor is not busy, transient signals on the $\overline{ERROR}$ input cannot generate a latched INTR output. The $\overline{INTR}$ signal on line 24, as explained above, now latches the $\overline{BUSY}$ 286 output of OR gate 22 into its active (low) state to ensure that, at this time, no data transfer can take place between the main processor and co-processor even if the co-processor is no longer issuing a $\overline{BUSY}$ signal. This ensures that main processor 1 will honour the interrupt before executing another co-processor instruction.

As indicated in the Figure 1 description, the INTR input to the interrupt controller, when serviced in its priority order, generates an INTR signal for the main processor. The processor responds by setting its M10, $\overline{S1}$ and $\overline{S0}$ outputs low to cause bus controller 4 to issue an active $\overline{INTA}$ signal (low). At this stage, the inputs to the Figure 2 logic circuit are such that its outputs remain unchanged.

On the second $\overline{INTA}$ signal, the pointer is passed to main processor 1 from interrupt controller 5 to initiate the error handling routine. A clear busy command is issued by the error handling routine. As indicated above, address bits A0 and A3 are both "0" for this command. In addition, at this time, M10 from the main processor is "0", the $\overline{CS287}$ signal from decoder 8 is "0", the $\overline{INTA}$ signal is now "1". The main processor drives bus controller 4 to generate an 10WC (I/O write command), active low, providing the $\overline{10W}$ input to the logic circuit. AND gate 32 is therefore enabled to apply a high output through OR gate 28 along line 34 to AND gate 33. This disables the high latched $\overline{Q3}$ output from OR gate 26 on line 27, to clear the $\overline{BUSY}$ 286 output to the main processor. Next, the main processor selects the co-processor by providing a "1" output on the A3 address line. With the other inputs to AND gate 35 remaining unchanged, this gate is enabled to lower the $\overline{NPCS}$ output of the logic circuit to select the co-processor. The error

handling routine may clear the co-processor error without the need to reset it. If, however, reset is required, this command, in which address bit A0 is "1", is issued by the main processor. At this time, M10, $\overline{10W}$ and $\overline{CS287}$ are still low and $\overline{INTA}$ and A0 are high. Accordingly, the output of OR gate 31 goes low. As the RESET input from clock generator 3 is always low except during a power on sequence, the output of AND gate 30 goes from high to low. This output is inverted by inverter 38 to provide a positive reset signal, RESET 287, to co-processor 2.

The RESET input to AND gate 30 is used to ensure that the co-processor is reset during a power up sequence of the system. When the system is initially powered up, this input is high, thereby forcing a low output from AND gate 30 to reset the co-processor. In addition, it provides an input to OR gate 28 to ensure that output $\overline{Q3}$ on line 27 is not initially latched positive to generate a $\overline{BUSY}$ 286 signal.

As mentioned above, the circuit of Figure 2 may be implemented on a programmable array logic chip. This may be embodied on a Monolithic Memories Inc. 20 pin PAL16L8 chip programmed as follows:

Pin list: RESET, ERROR, BUSY, A0, A3, $\overline{10W}$, M10, $\overline{CS287}$, $\overline{INTA}$, GND, TEST, NPCS, Q1, Q2, Q3, CLRBSY, RESET (RST) 287, INTR, BUSY (BSY) 286, VCC, where GND is ground, TEST is a test pin, grounded in operation of the circuit, VCC is the voltage supply pin and the remaining pins are as shown in Figure 2. The chip is programmed to perform the following boolean expressions:

$$\overline{NPCS}=\overline{M10}\times A3\times CS287\times\overline{INTA}$$
$$\overline{Q1}=\overline{Q1}\times Q2+\overline{CLRBSY}+\overline{INTR}$$
$$\overline{Q2}=\overline{Q2}\times CLRBSY+\overline{INTR}+Q1$$
$$\overline{Q3}=\overline{Q3}\times\overline{Q2}\times CLRBSY+Q1$$
$$\overline{CLRBSY}=\overline{A3}\times\overline{A0}\times 10W\times\overline{M10}\times CS287\times\overline{INTA}+RST287$$
$$\overline{RST287}=\overline{RESET}\times M10+\overline{RESET}\times A3+\overline{RESET}\times\overline{A0}+\overline{RESET}\times\overline{10W}+\overline{RESET}\times\overline{CS287}+\overline{RESET}\times INTA$$
$$\overline{INTR}=\overline{ERROR}+\overline{BUSY}\times\overline{INTR}$$
$$\overline{BSY286}=\overline{Q3}+BUSY.$$

In summary, there has been shown a logic circuit which is coupled to monitor the busy and error outputs of a co-processor. On detection of an error signal in the presence of a busy signal, the logic circuit supplies an interrupt request to an interrupt controller whilst maintaining a co-processor busy input to a main processor. After the interrupt controller services the request to point to a main processor routine, the logic circuit is responsive to main processor command signals to remove the busy signal, then select the co-processor for data transfer and lastly, to reset the co-processor if it is in its protected mode, which is the only way in which this processor can exit from this mode.

Claims

1. A data processing system comprising a main processor (1) and a co-processor (2) having a busy output which, when active, indicates that the co-processor is executing a command and is not available for external data transfer, and an error output which, when active, indicates a co-processor malfunction, a priority sensitive interrupt controller (5), external to either processor being provided to handle interrupts, characterised in that there is included a logic circuit having inputs coupled respectively to the error and busy outputs of the co-processor and an output connected to the main processor via the interrupt controller, and including first logic means (20, 21, 23) responsive only to coincident active error and busy outputs from the co-processor to generate an interupt request signal directed at the main processor to initiate an error interrupt routine.

2. A data processing system as claimed in claim 1 in which the first logic means includes latching means operable to provide a latched interrupt request signal subsequent to the reception of the coincident busy and error signals irrespective of the condition of subsequent busy signals.

3. A data processing system as claimed in either preceding claim in which the logic circuit includes second logic means (22) responsive to an active busy signal from the co-processor to provide an active busy signal to the main processor and latching means (26, 33, 39) responsive to the interrupt request signal to latch the active busy signal to the main processor to prevent the main processor from executing a co-processor instruction until the interrupt is handled by the routine.

4. A data processing system as claimed in claim 3 in which the logic circuit includes third logic means (28) responsive to a clear busy command from the main processor to clear the active busy signal to an inactive state.

5. A data processing system as claimed in any preceding claim in which the logic circuit includes fourth logic means (35) responsive to signals representing an I/O write command from the main processor defining the co-processor to generate an active chip select signal to the co-processor to enable data transfer from the co-processor to the main processor.

6. A data processing system as claimed in claim 5 in which the logic circuit includes fifth logic means (30) responsive to signals representing a co-processor reset command from the main processor to generate a reset signal to reset the co-processor to a dormant state.

Patentansprüche

1. Datenverarbeitungssystem mit einem Hauptprozessor (1) und einem Coprozessor (2) mit einem Tätigausgang, welcher, wenn er aktiv ist, anzeigt, daß der Coprozessor gerade einen Befehl ausführt und für einen externen Datentransfer nicht zur Verfügung steht, und einem Fehlausgang, welcher, wenn er aktiv ist, eine Coprozes-

handling routine may clear the co-processor error without the need to reset it. If, however, reset is required, this command, in which address bit A0 is "1", is issued by the main processor. At this time, M10, $\overline{10W}$ and $\overline{CS287}$ are still low and $\overline{INTA}$ and A0 are high. Accordingly, the output of OR gate 31 goes low. As the RESET input from clock generator 3 is always low except during a power on sequence, the output of AND gate 30 goes from high to low. This output is inverted by inverter 38 to provide a positive reset signal, RESET 287, to co-processor 2.

The RESET input to AND gate 30 is used to ensure that the co-processor is reset during a power up sequence of the system. When the system is initially powered up, this input is high, thereby forcing a low output from AND gate 30 to reset the co-processor. In addition, it provides an input to OR gate 28 to ensure that output $\overline{Q3}$ on line 27 is not initially latched positive to generate a $\overline{BUSY}$ 286 signal.

As mentioned above, the circuit of Figure 2 may be implemented on a programmable array logic chip. This may be embodied on a Monolithic Memories Inc. 20 pin PAL16L8 chip programmed as follows:

Pin list: RESET, ERROR, BUSY, A0, A3, $\overline{10W}$, M10, $\overline{CS287}$, $\overline{INTA}$, GND, TEST, NPCS, Q1, Q2, Q3, CLRBSY, RESET (RST) 287, INTR, BUSY (BSY) 286, VCC, where GND is ground, TEST is a test pin, grounded in operation of the circuit, VCC is the voltage supply pin and the remaining pins are as shown in Figure 2. The chip is programmed to perform the following boolean expressions:

$$\overline{NPCS}=\overline{M10}\times A3\times CS287\times\overline{INTA}$$
$$\overline{Q1}=\overline{Q1}\times Q2+\overline{CLRBSY}+\overline{INTR}$$
$$\overline{Q2}=\overline{Q2}\times CLRBSY+\overline{INTR}+Q1$$
$$\overline{Q3}=\overline{Q3}\times\overline{Q2}\times CLRBSY+Q1$$
$$\overline{CLRBSY}=\overline{A3}\times\overline{A0}\times 10W\times\overline{M10}\times CS287\times\overline{INTA}+RST287$$
$$\overline{RST287}=\overline{RESET}\times M10+\overline{RESET}\times A3+\overline{RESET}\times\overline{A0}+\overline{RESET}\times\overline{10W}+\overline{RESET}\times\overline{CS287}+\overline{RESET}\times INTA$$
$$\overline{INTR}=\overline{ERROR}+\overline{BUSY}\times\overline{INTR}$$
$$\overline{BSY286}=\overline{Q3}+BUSY.$$

In summary, there has been shown a logic circuit which is coupled to monitor the busy and error outputs of a co-processor. On detection of an error signal in the presence of a busy signal, the logic circuit supplies an interrupt request to an interrupt controller whilst maintaining a co-processor busy input to a main processor. After the interrupt controller services the request to point to a main processor routine, the logic circuit is responsive to main processor command signals to remove the busy signal, then select the co-processor for data transfer and lastly, to reset the co-processor if it is in its protected mode, which is the only way in which this processor can exit from this mode.

Claims

1. A data processing system comprising a main processor (1) and a co-processor (2) having a busy output which, when active, indicates that the co-processor is executing a command and is not available for external data transfer, and an error output which, when active, indicates a co-processor malfunction, a priority sensitive interrupt controller (5), external to either processor being provided to handle interrupts, characterised in that there is included a logic circuit having inputs coupled respectively to the error and busy outputs of the co-processor and an output connected to the main processor via the interrupt controller, and including first logic means (20, 21, 23) responsive only to coincident active error and busy outputs from the co-processor to generate an interupt request signal directed at the main processor to initiate an error interrupt routine.

2. A data processing system as claimed in claim 1 in which the first logic means includes latching means operable to provide a latched interrupt request signal subsequent to the reception of the coincident busy and error signals irrespective of the condition of subsequent busy signals.

3. A data processing system as claimed in either preceding claim in which the logic circuit includes second logic means (22) responsive to an active busy signal from the co-processor to provide an active busy signal to the main processor and latching means (26, 33, 39) responsive to the interrupt request signal to latch the active busy signal to the main processor to prevent the main processor from executing a co-processor instruction until the interrupt is handled by the routine.

4. A data processing system as claimed in claim 3 in which the logic circuit includes third logic means (28) responsive to a clear busy command from the main processor to clear the active busy signal to an inactive state.

5. A data processing system as claimed in any preceding claim in which the logic circuit includes fourth logic means (35) responsive to signals representing an I/O write command from the main processor defining the co-processor to generate an active chip select signal to the co-processor to enable data transfer from the co-processor to the main processor.

6. A data processing system as claimed in claim 5 in which the logic circuit includes fifth logic means (30) responsive to signals representing a co-processor reset command from the main processor to generate a reset signal to reset the co-processor to a dormant state.

Patentansprüche

1. Datenverarbeitungssystem mit einem Hauptprozessor (1) und einem Coprozessor (2) mit einem Tätigausgang, welcher, wenn er aktiv ist, anzeigt, daß der Coprozessor gerade einen Befehl ausführt und für einen externen Datentransfer nicht zur Verfügung steht, und einem Fehlausgang, welcher, wenn er aktiv ist, eine Coprozes-

* C L M S *

sor-Fehlfunktion anzeigt, wobei eine prioritätsempfindliche Unterbrechungssteuerung (5) außerhalb beider Prozessoren zur Handhabung von Unterbrechungen vorgesehen ist, dadurch gekennzeichnet, daß eine Logikschaltung enthalten ist, welche beziehentlich mit dem Fehler- und dem Tätigausgang des coprozessors gekoppelte Eingänge und einen mit dem Hauptprozessor über die Unterbrechungssteuerung verbundenen Ausgang aufweist, und wobei erste Logikmittel (20, 21, 23) enthalten sind, welche nur auf gleichzeitig aktive Fehler- und Tätigausgänge des Coprozessors damit ansprechen, eine auf den Hauptprozessor gerichtetes Unterbrechungsanforderungssignal zu erzeugen, um eine Fehlerunterbrechungsroutine einzuleiten.

2. Datenverarbeitungssystem nach Anspruch 1, bei welchem die ersten Logikmittel Verriegelungsmittel enthalten, die so betreibbar sind, daß sie ein verriegeltes Unterbrechungsanforderungssignal nachfolgend auf den Erhalt der zusammenfallenden Tätig- und Fehlersignale unabhängig vom Zustand nachfolgender Tätigsignals liefern.

3. Datenverarbeitungsystem nach irgendeinem vorstehenden Anspruch, bei welchem die Logikschaltung zweite Logikmittel (22), die auf ein aktives Tätigsignal vom Coprozessor damit ansprechen, ein aktives Tätigsignal zum Hauptprozessor zu liefern, und Verriegelungsmittel (26, 33, 39) umfassen, die auf ein Unterbrechungsanforderungssignal damit ansprechen, daß aktive Tätigsignal zum Hauptprozessor zu verriegeln, um zu verhindern, daß der Hauptprozessor einen Coprozessorbefehl ausführt, bis die Unterbrechung durch die Routine gehandhabt wird.

4. Datenverarbeitungssystem nach Anspruch 3, bei welchem die Logikschaltung dritte Logikmittel (28) enthält, die auf einen Löschtätigbefehl des Hauptprozessors damit ansprechen, das aktive Tätigsignal in einen inaktiven Zustand zu löschen.

5. Datenverarbeitungssystem nach irgendeinem vorstehenden Anspruch, bei welchem die Logikschaltung vierte Logikmittel (35) enthält, die auf Signale, die einen den Coprozessor definierenden E/A-Schreibbefehl von Hauptprozessor darstellen, damit ansprechen, ein Aktivchipauswahlsignal an den Coprozessor zu erzeugen, um einen Datentransfer vom Coprozessor zum Hauptprozessor zu ermöglichen.

6. Datenverarbeitungssystem nach Anspruch 5, bei welchem die Logikschaltung fünfte Logikmittel (30) enthält, welche auf Signale, die einen Coprozessorrücksetzbefehl vom Hauptprozessor darstellen, damit ansprechen, ein Rücksetzsignal zum Rücksetzen des Coprozessors in einen Ruhezustand zu erzeugen.

Revendications

1. Système de traitement de données comprenant un processeur principal (1) et un coprocesseur (2) comportant une sortie d'occupation qui, lorsqu'elle est active, indique que le coprocesseur est en train d'exécuter une instruction et n'est pas disponible pour un transfert extérieur de données, et une sortie d'erreur, qui, lorsqu'elle est active, indique une défaillance du coprocesseur, et un contrôleur d'interruptions (5) répondant aux priorités, qui est situé à l'extérieur des processeurs de manière à traiter des interruptions, caractérisé en ce que ce système inclut un circuit logique comportant des entrées reliées respectivement aux sorties d'erreur et d'occupation du coprocesseur, et une sortie raccordée au processeur principal par l'intermédiaire du contrôleur d'interruptions, et incluant des premiers moyens logiques (20, 21, 23) sensibles uniquement à des signaux de sortie actifs coïncidents d'erreur et d'occupation, délivrés par le coprocesseur, de manière à produire un signal de demande d'interruption appliqué au processeur principal pour déclencher un sous-programme d'interruption d'erreur.

2. Système de traitement de données selon la revendication 1, dans lequel les premiers moyens logiques incluent des moyens de verrouillage commandables pour délivrer un signal verrouillé de demande d'interruption, à la suite de la réception des signaux coïncidents d'occupation et d'erreur, indépendamment de l'état de signaux ultérieurs d'occupation.

3. Système de traitement de données selon l'une ou l'autre des revendications précédentes, dans lequel le circuit logique inclut des seconds moyens logiques (22) sensibles à un signal actif d'occupation, délivré par le coprocesseur, pour appliquer un signal actif d'occupation au processeur principal, et des moyens de verrouillage (26, 33, 39) sensibles au signal de demande d'interruption pour verrouiller le signal actif d'occupation à destination du processeur principal de manière à empêcher ce dernier d'exécuter une instruction du coprocesseur, avant que l'interruption n'ait été traitée par le sous-programme.

4. Système de traitement de données selon la revendication 3, dans lequel le circuit logique inclut des troisièmes moyens logiques (28) sensibles à une instruction de suppression d'occupation délivré par le processeur principal pour supprimer le signal actif d'occupation de manière à l'amener à un état inactif.

5. Système de traitement de données selon l'une quelconque des revendications précédentes, dans lequel le circuit logique inclut des quatrièmes moyens logiques (35) sensibles à des signaux représentant une instruction d'écriture d'entrée/sortie délivré par le processeur principal et désignant le coprocesseur, afin d'engendrer un signal actif de sélection de circuit vers le coprocesseur pour autoriser le transfert de données depuis le coprocesseur au processeur principal.

6. Système de traitement de données selon la revendication 5, dans lequel le circuit logique inclut des cinquièmes moyens logiques (30) sensibles à des signaux représentant une instruction de rétablissement du coprocesseur, délivrée par le processeur principal, pour produire un signal de rétablissement servant à rétablir le coprocesseur dans un état de sommeil.

* C L M S *