Description

The present invention relates to data transfer on a computer bus.

General purpose computer systems are usually designed to transfer data between various subsystems over a shared data path known as a bus. The various subsystems, such as a central processor, cache controller, and input/output (I/O) subsystems such as keyboard, video, and mass storage controllers may all be attached to a single system bus. The system main memory may also be attached to the system bus. In some systems, main memory is connected directly to the central processor, with the system bus being reserved for I/O data transfer through a Direct Memory Access (DMA) device or other I/O controller.

Since a large amount of data is transferred between numerous subsystems on a single bus, the bus itself can often become a performance bottleneck for the overall system. Designing a system bus to be able to handle the heavy load placed upon it is a difficult task. The difficulty of this task is increased in many systems because the precise nature of the equipment to be included in the system may not be known in advance to the bus designer. Different subsystems, made by different manufacturers and having different performance specifications, may all be attached to a single system bus at the same time.

The system bus must be designed to perform correctly regardless of which subsystems are attached to it. In practice, this usually means that a bus is designed to a "lowest common denominator", in which certain aspects of the bus design are chosen to sacrifice performance in order to assure correct operation with all subsystems.

The nature of subsystems to be attached to the system bus can vary. Some subsystems, such as keyboard I/O, usually transfer only a single character or word at one time. Other subsystems, such as video device controllers or DMA controllers attached to mass storage subsystems, usually transfer data in blocks of words. In order to support efficient transfer of block data, many system buses include a block transfer mode, also called a "burst mode" on some systems, which is designed for the express purpose of transferring blocks of consecutive data words. When a block transfer is in effect, some of the standard bus control transactions are not used, which decreases bus overheads and increases the data transfer rate.

Although block transfer modes allow blocks of data to be more efficiently transferred on a bus, the overall system bus performance is limited by the fact that all of the subsystems will be attached to the bus at all times. Thus, bus performance limitations, which are included to allow some low performance subsystems to operate properly, prevent potentially higher performance subsystems from being able to transfer data at a higher rate. In traditional computer system bus designs it is not possible to have different subsystems transfer data over a single bus at different rates.

US patent 4530053 describes a microcomputer system wherein a direct memory access controller is used to transfer data directly between peripheral devices and main memory. Circuitry is provided to be used in association with a single mode transfer DMAC device to enable control of the number of bytes transferred during a DMA transfer cycle. US patent 4095267 describes a clock pulse control system for use in a microcomputer system which operates to synchronise the operating speed of an I/O unit with that of the CPU during transfer of information therebetween.

The prior art, therefore, does not provide a computer system bus to support a relatively high speed data transfer protocol which does not interfere with the proper operation of lower performance devices attached to the bus. Also the prior art does not provide a high speed data transfer protocol on such bus which can accommodate subsystems having different, predefined performance limitations, so that certain data transfers between subsystems can be made at a rate limited only by the subsystems involved in the transfer without regard to performance limitations of other devices attached to the bus.

Accordingly the invention provides a system for transferring data in a computer between a master subsystem and a slave subsystem comprising: a bus coupled to the master subsystem and the slave subsystem for communicating signals therebetween, said bus including a plurality of address lines, data lines and control lines; and a clock line in said bus for communicating a normal clock signal under control of the master subsystem; characterised in that said bus includes a high speed clock control line for communicating a separate high speed clock under control of the master subsystem; and said bus includes a transfer control line for communicating to the master subsystem a transfer mode signal indicative of one or more data transfer capabilities of the slave subsystem, said transfer mode signal comprising a high speed mode signal indicating whether the slave subsystem is capable of transferring data according to said high speed clock signal; wherein the master subsystem, in response to said high speed mode signal, selects whether data is transferred between the master subsystem and the slave subsystem according to said normal clock or according to said high speed clock.

The invention further provides a method for transferring data on a computer system bus between a master subsystem and a slave subsystem, said bus including a line for communicating a normal clock signal under control of the master subsystem, the method comprising the steps of: the master subsystem initiating a data

transfer with the slave subsystem; the slave subsystem issuing a transfer mode signal to said master subsystem over a control line of said bus, said transfer mode signal indicating whether the slave subsystem is capable of transferring data according to a high speed clock, separate from the normal clock; the master subsystem selecting, in response to said indication, whether to transfer data between the master subsystem and the slave subsystem according to said high speed clock or according to said normal clock; in the case of a transfer according to said high speed clock, the master subsystem driving said high speed clock signal on the bus; and in the case of a transfer according to said normal clock, transferring data between said master and slave subsystems according to said normal clock.

Therefore, according to the present invention, a computer system bus includes a mechanism for a high speed block data transfer between a master subsystem and a slave subsystem. Several bus control signals are dedicated specifically to this function. When a master subsystem and a slave subsystem are both capable of supporting a high speed block data transfer, such fact is communicated on the special control lines. Selected bus control signals are held in a preselected state so that the remaining devices attached to the bus are not affected by the high speed block data transfer.

A separate high speed clock signal is used for the high speed data transfers instead of the normal bus clock signal. The frequency of the high speed clock can be adjusted to be consistent with the performance specifications of both the master and the slave.

Preferably, in the case of a high speed block data transfer, the data transfer rate can be varied to accommodate the maximum transfer speeds of the master subsystem and slave subsystem.

In a preferred embodiment, the word width of the data transfer can be adjusted to accommodate the maximum word width transfer supported by both the master subsystem and the slave subsystem.

In order that the invention may be fully understood, a preferred embodiment thereof will now be described, by way of example only, with reference to the accompanying drawings in which:

FIGURE 1 is a block diagram showing a master subsystem and a slave subsystem connected to a computer system bus;

FIGURE 2 is a flowchart illustrating operation of a high speed data transfer mode;

FIGURES 3a, 3b and **3c**, considered together are a flowchart illustrating the operation of a bus master device;

FIGURES 4a and **4b**, considered together, are a flowchart illustrating operation of a bus slave device; and

FIGURES 5 is a timing diagram illustrating bus signal sequencing for a sample data transfer.

The preferred embodiment of the invention which will now be described is based upon the Micro Channel System2 bus which is incorporated in PS/2 microcomputer products available from IBM (Micro Channel, PS/2 and IBM are trademarks of the International Business Machines Corporation). Only the control signals necessary for illustrating this preferred embodiment of the present invention are included in this description; additional aspects of the bus upon which this description is based are available in published literature.

Figure 1 shows a computer system bus **10** with two attached subsystems **12, 14.** For convenience, the bus signals are broken down into addressing signals ADDR, data signals DATA, and control signals CTRL. The addressing signals ADDR define memory locations within the memory map of the system. For systems such as the PS/2 family which have separate address spaces for system memory and I/O devices, the address signals ADDR also include a signal (not shown) indicating whether the current bus address is a memory address or an I/O address.

In a high speed block data transfer according to the present invention, two devices are involved. One device **12** coupled to the bus will be considered the bus master device, with the other device referred to as the bus slave device **14.** Consistent with traditional usage, the bus master **12** is the subsystem which is in charge of the transfer. The slave **14** generates signals and may terminate the transfer, but its control logic is generally simpler than that of the master **12.**

In many systems, any given subsystem may be capable of operating either as a master or a slave. In particular, subsystems which are capable of operating as a bus master are of ten capable of operating as a slave instead. Many simpler devices, such as memory subsystems, are only capable of operating as slaves.

The high speed data transfer which will now be described is referred to herein as a streaming data transfer. This terminology is used to distinguish this new transfer mechanism from previous ("normal") block mode transfers which operate on different principles.

The discussion of **Figures 2, 3, 4,** and **5** makes the assumption, in all cases, that the bus master can perform the described high speed transfer process. If the master cannot perform the high speed transfer, either by design or because of something in the master's current operational state, data transfer is performed in the normal mode used by the Micro Channel bus.

Figure 2 is a high level block diagram illustrating the data transfer process according to the present invention. First, a normal bus data transfer cycle begins **20.** At this point, no decision has been made as to wheth-

er this cycle will be a normal data transfer or a streaming data transfer. After the cycle begins, the bus master determines whether the slave device supports streaming data transfers **22**. If not, a normal bus cycle is completed **24**. If the slave does support streaming data and, assuming that the master does also, a check is made **26** to see whether the transfer is complete. If it is not, data is transferred **28** and control refers to step **26**. During the time that control is looping between steps **26** and **28**, the remaining devices attached to the bus can see only that the original bus cycle begun in step **20** has not yet completed. Therefore, they ignore any changing of signals which occurs during the block data transfer.

Once the transfer is complete, as tested in step **26**, the normal bus cycle completes **24**. The remaining devices on the bus then operate in normal fashion again. It is possible that only a single data transfer will be made at the higher rate, in which case little or nothing is gained. However, the maximum number of data items which may be transferred can be quite large. In many systems, the maximum number will be determined by a bus timeout period. Since a data cycle is begun in step **20**, but has not yet completed, a typical bus timeout controller will think that the slave device has not responded at all.

Figures 3 and **4** are flowcharts illustrating operation of the bus master **12** and slave **14**, respectively. Although steps are shown sequentially, some are actually performed in parallel. For example, steps **34** and **38** in **Figure 3** are actually performed simultaneously. Steps **40, 44, 46, 50** and **52** are also performed together. Such implementation details are performed in the same manner as previously done on the Micro Channel bus.

Figure 3 is a detailed flowchart illustrating operation of a bus master **12** as used with the preferred embodiment. If the master **12** does not support streaming data transfers, it will simply operate in the normal mode. The flowchart of **Figure 3** illustrates the flow of control within a bus master **12** which does support streaming data transfers.

At the beginning of a bus cycle, the master asserts the address and select signals **30**. The select signal, SELECT, is a signal indicating whether the address placed on the bus is a memory address or an I/O port address. Only a single address must be asserted for a streaming data transfer. As will be described, the master **12** and the slave **14** both automatically increment an internal address each time a data item is transferred. This is somewhat analogous to prior art block transfers, in which a starting address and a count of the number of data items to be transferred are the only address related signals required by a DMA device. However, in prior art DMA transfers, the DMA controller typically asserts an address for each data item transferred. In the presently described system, both the master **12** and slave **14** increment addresses internally, so that the address bus is not used after the start of the streaming data transfer. As described below, in the presently described system no block length is defined in advance. Instead, either the master **12** or the slave **14** may terminate the transfer.

Shortly after the address is asserted, a READ/WRITE signal is asserted **32**. This indicates whether data will be transferred from the master **12** to the slave **14** (a write cycle), or from the slave device **14** to the master **12** (a read cycle).

Next, an address latch signal, ADL is asserted **34**. This signal indicates that the address currently on the address lines of the bus **10** is valid, and typically causes the slave device **14** to write the address into its internal buffers. Bus definitions typically specify a minimum delay period between the time that the addresses become valid and their latching with the ADL signal.

The bus master **12** determines the data width at which it is capable of communicating **36**. A master **12** may not be capable of transferring data at the maximum width allowed by the bus **10**. For example, on a 32 bit data bus, a given device acting as a bus master for this cycle may only be capable of transferring 8 or 16 bits at a time. In most bus masters, the maximum width of the master does not change, and will actually have been determined at some fixed time in the past. However, the results of that determination should be available at step**36**.

If the current data transfer is a write, thereby transferring data from the master to the slave, the master **12** then asserts (places on the data bus) data **38**. Such data may be asserted simultaneously with ADL. The master then begins the data cycle **40**, by transitioning the bus clock. In the preferred embodiment, as illustrated in **Figure 5**, the clock signal is transitioned from high to low to start the data cycle.

At the time the data cycle begins, the master **12** detects whether the slave **14** is ready to transfer data **42**. This is done by sampling a control line driven by the slave **14** which indicates whether the slave **14** is ready. If the slave **14** is not ready, control passes to step **80** (in **Figure 3c**). If the slave **14** is ready, the master **12** proceeds to determine the data width which can be handled by the slave **44**. Width is sensed on control lines reserved for this purpose, which are driven by the currently addressed slave during each normal bus clock cycle. If the master word width is greater than that supported by the slave **14**, the master **12** will need to abort or complete this cycle and resend the same data in smaller segments which are acceptable by the slave.

In a preferred embodiment, slave data width is indicated by a 2 bit device size signal DS(16.32). The possible values of DS(16.32), and their corresponding meanings, are shown in Table I. The master **12** will transfer

data at the lesser of the master's word width and the slave's word width.

Table I

DS(16.32)	Data Width
00	32 bits
01	16 bits
10	Reserved
11	8 bits

Up to and including step **44**, all of the actions performed by the master **12** are those which are performed in every bus cycle. That is, no special actions have yet been undertaken in support of a streaming data transfer.

In step **46**, an SDR signal is tested by the master **12** to see if it is valid. The SDR signal indicates whether the slave device **14** supports streaming data transfers. If it does not, control is transferred to step **47** (in figure **3b**). If the master was reading during this bus cycle it fetches the data from the data bus **47**, and completes the data cycle **48**. Completing the data cycle is accomplished by driving the clock signal from low to high. The READ/WRITE signal should also be driven high if it is still low. READ/WRITE may be driven high anytime after the SDR signal has been tested. At this point, the master **12** has completed a normal bus cycle.

In step **46**, if the slave **14** does in fact support streaming data transfers, control passes to step **50**. If the master **12** has only a single data item to transfer, it will typically complete a normal data cycle by forcing control of the transfer to step **47** regardless of the state of the slave's SDR signal.

When the streaming data transfer is initiated, the master first determines the maximum transfer rate which the slave device supports **50**. In a preferred embodiment, this rate is indicated by the same SDR signal which indicates whether the slave supports streaming data transfer.

Table II

SDR(0.1)	Operation & Speed
00	SD at 20 MHz (50ns)
10	SD at 16 MHz (62.5ns)
01	SD at 10 MHz (100ns)
11	Normal Transfer Cycle

In a preferred embodiment, the signal SDR is a 2 bit signal, with the four possible values indicated in Table II. The bit pairs 00, 10, and 01 indicate that the slave **14** can support streaming data transfers at 20 MHz, 16 MHz, and 10MHz respectively. These clock speeds correspond to high speed clock cycle times of 50ns, 62.5ns, and 100ns, respectively.

If the SDR signal set by the slave **14** is 11, then the slave does not support streaming data transfer at all, and the master **12** will cause only the normal transfer cycle to be used as described above. This is the value tested for by the master **12** in step **46**. If the value of SDR is 11, the SDR signal is considered to be not valid.

Once the transfer rate has been determined **50**, the master **12** drives a high speed clock (step **52**), referred to as SD STROBE. This is a free running clock which is a completely separate signal from the normal bus clock.

As described above, in the case of a normal bus cycle, the normal bus clock is driven low in step **40**, and driven high in step **48** to complete the bus cycle. In a streaming data transfer, the normal bus clock remains low during the entire transfer.

The next step for the master **12** is to strobe the data (step **54**, figure **3b**). This means that the master **12** waits for the next cycle of SD STROBE. In a preferred embodiment, the high to low transitions of SD STROBE actually cause the data strobing referred to in step **54**.

At the time that the data is strobed, the slave **14** may not yet be ready to transfer data. The slave indicates this on a signal line which is reserved for this purpose, and which is sensed by the master (step **56**) at the time SD STROBE transitions high-to-low. If the slave **14** is not ready, then the attempted data transfer must be made again. Therefore, if the master **12** is writing to the slave **14**, the same data is again asserted on the data lines **58**, and control returns to step **54**.

Once the slave **14** successfully transfers the data, indicated by the yes branch from step **56**, the master

12 tests, in step 60 to see whether the SDR signal driven by the slave 14 is still valid. SDR is not valid when SDR(0.1)=11, as shown in Table II. If it is not valid, this shows that the slave 14 is terminating the data transfer, and the master 12 transfers control to step 62, which causes the high speed clock (SD STROBE) to stop. The slave 14 is required to indicate its termination of the data transfer before the last data item is transferred. Thus, after the slave 14 terminates the transfer, one last data item is transferred when the normal data cycle completes in step 48.

If the SDR signal remains valid in step 60, control transfers to step 64. If the master 12 is reading data from the slave 14, if fetches the data from the bus 64 and latches it into its internal buffers. The master next increments its internal address 66. After incrementing its address, if the master is writing data to the slave, it asserts the next data item 68.

Next, in step 70, the master checks to see if this is the master's penultimate (next to last) cycle. If it is not, there are at least two more data items to be transferred, and control returns to step 54.

If this is the master's next to last cycle, the master asserts the last cycle indicator 72. In the preferred embodiment, indication of the last cycle is done by raising the READ/WRITE signal which is originally asserted in step 32. The data is then strobed 74, and a check is made to see if the slave is ready 76. If not, the same data is again asserted if the master is writing to the slave 78, and control returns to step 74. As shown in Figure 5, this determination of slave readiness is made by waiting for the SDR signals to go high.

When the slave is ready in step 76, the high speed clock (SD STROBE) is stopped 62, data is fetched 47 if the master is reading from the slave, and in step 48 the data cycle is completed. The last item of data to be transferred is always transferred on the normal cycle completion, regardless of whether the streaming data transfer was terminated by the master 12 or the slave 14.

Returning to step 42, if the slave 14 was not ready, control passed to step 80, figure 3c. In step 80 the master 12 waits for a specified minimum delay 80, and then determines the slave's 14 data width in step 82. During this delay the slave 14 must assert a valid data width signal, even if it is not otherwise ready. This delay allows a bus converter, used to attach a second bus to the bus 10 as known in the art, to function properly. The use of this delay was previously included in the Micro Channel bus design.

At the same time as the slave's 14 data width is determined 82, the SDR signal is again tested 84 for validity. SDR must also be asserted during step 80, even if the slave 14 is not otherwise ready. If SDR is now valid, the master 12 waits until the slave 14 is ready 86, and control returns to step 50 for a high speed streaming data transfer.

If the slave 14 does not support streaming data transfers, step 84 will result in an invalid SDR signal, and in step 87 the master 12 resets the READ/WRITE signal. When the slave 14 is ready to complete the normal bus transfer (step 88), control returns to step 47, figure 3b, and the normal transfer is completed.

Figure 4 illustrates the sequence of steps performed by the slave device 14 during a data transfer. Prior to a new data cycle, the master places the appropriate address on the address lines, and these are decoded 90 by the slave. The slave 14 then asserts its device size 92 on DS(16.32). The asserted value is the appropriate value from Table I.

If the slave 14 is not ready to transfer data at the time the address is latched 90, it signals not ready 96 on a line (CHRDY in Figure 5) reserved for this purpose.

The next action taken by the slave 14 depends on whether or not it is capable of supporting streaming data transfers, determined in step 98. If the slave is so capable, it asserts in step 99 the streaming data rate which it can handle. This rate is selected from those shown in Table 2. The slave then waits until it is ready 100, and, when ready, signals such fact in step 101. The data cycle starts in step 102 when the master 12 drives the bus clock. (step 40 in figure 3a) A test is made 104 to see if, during the forthcoming data cycle, the master 12 is preparing to transfer its last data item. If not, the slave 14 determines in step 106 whether or not it is ready to transfer the next data item, and, if not, signals not ready 108 to the master 12. If the slave 14 is ready, and if this is a read cycle, the slave asserts data onto the bus in step 110.

Next, the data is strobed 112, and, if this a write from the master 12 to the slave 14, the slave 14 fetches the data from the bus 114 and loads it into its internal buffers. The slave 14 then increments its internal address 116, and determines (step 118, figure4b) whether this is the slave's penultimate cycle. This is a determination made of the status of the slave device 14; therefore in some transfers, the slave device 14 will determine the size of the streaming data transfer. This situation can occur, for example, in a system in which the slave is a buffered I/O device with a limited buffer size. During a read (slave to master) cycle, the slave will know how much data it has to transmit, and can terminate the transfer when all the data has been transmitted. In a write (master to slave), the slave may need to terminate a transfer in order to prevent buffer overflow.

In any event, if the slave 14 is not prepared to terminate the transfer after the next data item, control returns to step 104, figure 4a. If however the slave is preparing to terminate the transfer, it checks to see whether it is ready 120, and, if not, signals not ready 122. When the slave is ready, and if the data transfer is a read, and

it asserts its last data item **124**.

The slave **14** then resets its SDR rate to 11 in step **126**. This is interpreted by the master **12** as a slave termination of transfer as described in connection with **Figure 3** (step **60**). The data cycle then completes **128**, and if this was a write transfer, the slave **14** fetches the data from the bus **130** and latches it into its buffers.

If the master **12** terminates the transfer, the yes branch will be taken from step **104**, transferring control to step **132**. As described in connection with **Figure 3**, the master termination event is indicated by driving the READ/WRITE signal high. If the slave **14** is not ready **132** it simply waits. When it is ready, the slave responds by resetting the SDR rate to 11 in step **134**. If this is read transfer, the slave places its data on the bus **136**, and waits for the data cycle to complete **128**. As in the usual case, if this is a write transfer, data is fetched from the bus **130** after the cycle completes.

If the slave **14** is not ready in step **132**, it is not necessary to signal this fact as is done in steps **96** and **122**. Instead, when the master **12** terminates the transfer and raises the READ/WRITE signal, the slave **14** then signals that it is ready by resetting the SDR rate (step **134**). This hand shaking protocol prevents the master **12** from completing the cycle until the slave **14** is ready to transfer the last data item.

Returning to step **98**, if in fact the slave **14** does not support streaming data transfers, control branches to step **144**. In step **144** the slave **14** waits until it is ready, and when ready signals such fact in step **142**. The normal data cycle starts **140**, and, if this is a read cycle, the slave **14** asserts data **136**. The data cycle then completes **128** as described above.

Figure 5 is a timing diagram illustrating most of the operations described in connection with **Figures 3** and **4** with a sample transfer. The sample transfer described is a four word streaming data transfer. Several options, such as read and write transfers, and master and slave terminated transfers, are illustrated.

The address signals of the bus **10** are shown as ADDR. For the timing diagram of **Figure 5**, ADDR includes the memory/I/O select line (SELECT). The signals S0 and S1 are control lines driven by the master **12** to indicate whether this is a read or write transfer. If S0 is low, the transfer is a read. If S1 is low, it is a write transfer. S0 and S1 may not both be low at the same time.

The signal ADL is the signal used to latch the address, and the CMD is the normal bus clock. SD STROBE is the streaming data, high speed clock used in streaming data transfers. The signals ADDR, S0, S1, ADL, CMD, and SD STROBE are all driven by the bus master **12**.

The signal DATA contains 8, 16, or 32 bits of parallel data, whichever data width is supported by both the master **12** and slave **14**. The DATA lines are driven by the master 12 during a write transfer, and by the slave **14** during a read transfer.

The signal CHRDY is used by the slave **14** to indicate whether or not it is ready to transfer data as described in connection with **Figure 4**. The 2 bit signal SDR(0.1) indicates whether or not the slave 14 supports streaming data transfers, and the speed of such transfers if they are supported, as defined in Table II. The 2 bit signal DS(16.32) indicates the slave word width as defined in Table I. The signals CHRDY, SDR(0.1), and DS(16.32) are all driven by the slave**14**.

Initially, the address lines are not guaranteed to contain valid values **200**. To initiate a transfer, the master **12** drives valid address signals **202** onto the lines ADDR. Next, the appropriate read or write signals S0, S1 are driven low **204**. The slave **14** drives its data width onto DS(16.32) **205** when it recognizes that it has been addressed. After the ADDR signals have been valid for at least a defined minimum time, ADL is driven low **206** to latch the address into the slave's buffers. When the slave **14** detects the transition of ADL, it sets the SDR signals **208** to show the transfer rate it supports. The SDR signals are typically driven to a high value **210** from a tri-state condition when the slave device **14** detects that it is being addressed. The tri-state to high transition is not controlled by the transition of ADL, as was the case with the DS signals.

If this is a write transfer, the master **12** next places information **212** on DATA, which was previously tri-stated. The master **12** then begins the data cycle by driving CMD low **214**. Preferably, SD STROBE is driven low **216** at the same time, and ADL is driven high **217**. Steps **40**, **44**, **46**, **50**, and **52** shown in **Figure 3** are thus occurring simultaneously, in that the master **12** is starting the SD STROBE, and determining the data width and transfer rate of the slave simultaneously, at the time CMD transitions low **214**.

After CMD is driven low **214**, the ADDR signals are allowed to become undefined after a defined minimum delay, and are no longer relevant to the transfer. As known in the art, an address for the next bus cycle can be asserted during this invalid period. The slave **14** can also allow DS(16.32) to become undefined **219**. If this is a read (slave to master) transfer, the slave drives the first data item (D0) onto the data lines **220** in response to the high-to-low transition **214** of CMD.

The SD STROBE signal is now free running. The "strobe data" steps of **Figures 3** and **4** occur on the high-to-low transitions **222**, **224**, **226**, and **228** of SD STROBE. After each such transition, the device which is driving data onto the bus places the next data item on DATA. For so long as the slave can keep up with the transfer, a new item will be placed in DATA for each cycle of SD STROBE.

In the example of **Figure 5**, the slave determines during the second high-to-low transition **222** of SD STROBE that it will be unable to transfer data item D1 at the next cycle. This is the determination made in step **106** of **Figure 4**. Since the slave **14** is not ready to transfer the next data item, it drives the signal CHRDY low **230**. When the signal SD STROBE next makes a high-to-low transition **224**, the master **12** detects that the slave **14** has indicated that it is not ready. This detection is made in step **56** of **Figure 3**.

The signal SD STROBE continues running, even though no data is being transferred. If this is a write cycle, the master **12** continues placing the same data onto the bus. If it is a read cycle, the master ignores whatever data happens to be on the bus, as this data is assumed to be invalid. **Figure 5** shows CHRDY being held low by the slave **14** for only a single clock cycle, but it may be held low longer if necessary. When the slave **14** is ready to transfer the next data item, control in **Figure 4** passes to step **110**, and CHRDY is allowed to go high **232**. At the next high-to-low transition**226** of SD STROBE, the master **12** detects that the data has properly transferred, and continues its normal function.

A master terminated transfer is indicated when the master **12** drives the appropriate signal, S0 or S1, high **234**. This is preferably done simultaneously with a high-to-low transition **228** of SD STROBE. This transition **234** is detected in step **104** of **Figure 4**, and the slave **14**, when ready, drives the signals SDR(0.1) high **236**. This is shown in step **134** of **Figure 4**. Once the slave **14** has indicated ready **236**, the master **12** drives CMD high **238** to complete the data cycle. This is shown in step **48** of **Figure 3** and step**128** of **Figure 4**. The last data item, D3 in **Figure 5**, is transferred on the positive-going transition **238** of CMD

After the slave **14** detects the low-to-high transition 238 of CMD, it tri-states SDR(0.1) **240**, and DATA if this was a read transfer **242**, in order to prevent bus collisions with other devices.

If the slave **14** terminates the transfer, it indicates such fact by a low-to-high transition **244** of SDR(0.1). The master **12** senses this transition asynchronously, so that the timing of the transition **244** need not be synchronized with SD STROBE. The master **12** senses this transition in step **60** of **Figure 3**. In response to the end of transfer indication **244** from the slave **14**, the master **12** drives S0 or S1 high **246**, and also drives CMD high **238**. As before, the final data item D3 is transferred on the low-to-high transition **238** of CMD. Also, DATA and SDR(0.1) are tri-stated after CMD goes high.

The remaining devices on the bus are completely unaffected by the transfer of **Figure 5** because CMD has remained low throughout the entire transfer. When ADL was driven low **206**, the remaining devices, which were not addressed, became idle. They are awakened from their idle state only when CMD transitions from low to high **238**. To the remaining devices, the streaming data transfer appears simply as a stretched-out normal transfer.

The events which occur during a normal transfer are those shown to the left of the first dashed line **250**, and those shown to the right of the second dashed line **252**. In a normal transfer, CMD goes low **214**, D0 is placed on the data bus by the master **12** or by the slave **14**, and that same D0 is transferred when CMD transitions low-to-high **238**. As described above, the remaining devices on the bus are unaffected by any events occurring between the dashed lines.

The details of the bus interfaces needed to implement the master and slave as described above are well within the skill of a person skilled in the art. The interfaces can utilize programmable logic arrays or other programmable devices to reduce parts count.

As will be appreciated by those skilled in the art, the streaming data transfer described above allows a very high transfer rate for block data. It allows both high performance and low performance devices to operate normally on the same bus. Those devices for which a high speed block transfer is suitable, such as mass storage controllers and video controllers, can take advantage of streaming data transfer, while other devices can be built for less cost.

Claims

1. A system for transferring data in a computer between a master subsystem (12) and a slave subsystem (14) comprising:

bus (10) coupled to the master subsystem and the slave subsystem for communicating signals therebetween, said bus including a plurality of address lines, data lines and control lines; and

a clock line in said bus for communicating a normal clock signal (CMD) under control of the master subsystem;

characterised in that said bus includes a high speed clock control line for communicating a separate high speed clock (SD STROBE) under control of the master subsystem; and

said bus includes a transfer control line for communicating from the slave subsystem to the master subsystem a transfer mode signal (SDR (0.1)) indicative of one or more data transfer capabilities of the

In the example of **Figure 5**, the slave determines during the second high-to-low transition **222** of SD STROBE that it will be unable to transfer data item D1 at the next cycle. This is the determination made in step **106** of **Figure 4**. Since the slave **14** is not ready to transfer the next data item, it drives the signal CHRDY low **230**. When the signal SD STROBE next makes a high-to-low transition **224**, the master **12** detects that the slave **14** has indicated that it is not ready. This detection is made in step **56** of **Figure 3**.

The signal SD STROBE continues running, even though no data is being transferred. If this is a write cycle, the master **12** continues placing the same data onto the bus. If it is a read cycle, the master ignores whatever data happens to be on the bus, as this data is assumed to be invalid. **Figure 5** shows CHRDY being held low by the slave **14** for only a single clock cycle, but it may be held low longer if necessary. When the slave **14** is ready to transfer the next data item, control in **Figure 4** passes to step **110**, and CHRDY is allowed to go high **232**. At the next high-to-low transition**226** of SD STROBE, the master **12** detects that the data has properly transferred, and continues its normal function.

A master terminated transfer is indicated when the master **12** drives the appropriate signal, S0 or S1, high **234**. This is preferably done simultaneously with a high-to-low transition **228** of SD STROBE. This transition **234** is detected in step **104** of **Figure 4**, and the slave **14**, when ready, drives the signals SDR(0.1) high **236**. This is shown in step **134** of **Figure 4**. Once the slave **14** has indicated ready **236**, the master **12** drives CMD high **238** to complete the data cycle. This is shown in step **48** of **Figure 3** and step**128** of **Figure 4**. The last data item, D3 in **Figure 5**, is transferred on the positive-going transition **238** of CMD

After the slave **14** detects the low-to-high transition 238 of CMD, it tri-states SDR(0.1) **240**, and DATA if this was a read transfer **242**, in order to prevent bus collisions with other devices.

If the slave **14** terminates the transfer, it indicates such fact by a low-to-high transition **244** of SDR(0.1). The master **12** senses this transition asynchronously, so that the timing of the transition **244** need not be synchronized with SD STROBE. The master **12** senses this transition in step **60** of **Figure 3**. In response to the end of transfer indication **244** from the slave **14**, the master **12** drives S0 or S1 high **246**, and also drives CMD high **238**. As before, the final data item D3 is transferred on the low-to-high transition **238** of CMD. Also, DATA and SDR(0.1) are tri-stated after CMD goes high.

The remaining devices on the bus are completely unaffected by the transfer of **Figure 5** because CMD has remained low throughout the entire transfer. When ADL was driven low **206**, the remaining devices, which were not addressed, became idle. They are awakened from their idle state only when CMD transitions from low to high **238**. To the remaining devices, the streaming data transfer appears simply as a stretched-out normal transfer.

The events which occur during a normal transfer are those shown to the left of the first dashed line **250**, and those shown to the right of the second dashed line **252**. In a normal transfer, CMD goes low **214**, D0 is placed on the data bus by the master **12** or by the slave **14**, and that same D0 is transferred when CMD transitions low-to-high **238**. As described above, the remaining devices on the bus are unaffected by any events occurring between the dashed lines.

The details of the bus interfaces needed to implement the master and slave as described above are well within the skill of a person skilled in the art. The interfaces can utilize programmable logic arrays or other programmable devices to reduce parts count.

As will be appreciated by those skilled in the art, the streaming data transfer described above allows a very high transfer rate for block data. It allows both high performance and low performance devices to operate normally on the same bus. Those devices for which a high speed block transfer is suitable, such as mass storage controllers and video controllers, can take advantage of streaming data transfer, while other devices can be built for less cost.

Claims

1. A system for transferring data in a computer between a master subsystem (12) and a slave subsystem (14) comprising:

bus (10) coupled to the master subsystem and the slave subsystem for communicating signals therebetween, said bus including a plurality of address lines, data lines and control lines; and

a clock line in said bus for communicating a normal clock signal (CMD) under control of the master subsystem;

characterised in that said bus includes a high speed clock control line for communicating a separate high speed clock (SD STROBE) under control of the master subsystem; and

said bus includes a transfer control line for communicating from the slave subsystem to the master subsystem a transfer mode signal (SDR (0.1)) indicative of one or more data transfer capabilities of the

* C L M S *

slave subsystem, said transfer mode signal comprising a high speed mode signal indicating whether the slave subsystem is capable of transferring data according to said high speed clock signal;

wherein the master subsystem, in response to said high speed mode signal, selects whether data is transferred between the master subsystem and the slave subsystem according to said normal clock or according to said high speed clock.

2. A system as claimed in claim 1, in which said transfer mode signal includes a transfer speed control signal;

said transfer speed control signal being used by the slave subsystem, in the case when the slave subsystem is capable of data transfers according to said high speed clock, to indicate to the master subsystem the maximum rate at which the slave subsystem is capable of making such data transfers.

3. A system as claimed in any preceding claim, in which said transfer mode signal includes a transfer width control signal (DS (16.32)) indicating the maximum data word width which can be transferred by the slave subsystem;

and in which the master subsystem, in response to said transfer width control signal, selects a data word width within the capabilities of the slave subsystem for transferring data between the master subsystem and the slave subsystem.

4. A system as claimed in any preceding claim in which said bus includes a ready control line used by the slave subsystem to communicate to the master subsystem, during a data transfer according to said high speed clock signal, whether the slave is ready to complete the transfer of an individual data item;

and in which said transfer is not completed by the master until the slave indicates that it is ready to complete said transfer.

5. A system as claimed in any preceding claim, in which the master communicates only one address on the bus during a data transfer according to said high speed clock signal.

6. A device (12) for transferring data on a computer system bus (10), comprising:

means for communicating a normal clock signal (CMD) on a clock line in said bus;

means for communicating, on said bus, a high speed clock signal (SD STROBE) separate from said normal clock signal;

means for addressing a second device (14) connected to said bus;

means for detecting a transfer mode signal (SDR (0.1)) on the bus from said second device, which signal indicates one or more data transfer capabilities of said second device, said transfer mode signal comprising a high speed mode signal indicating whether said second device is capable of transferring data according to said high speed clock signal;

means for transferring data to or from said second device according to said normal clock signal;

means for transferring data to or from said second device under the control of the high speed clock signal; and

means for selecting, in response to said high speed mode signal, whether to transfer data, between said device and said second device, according to the normal clock or according to the high speed clock.

7. A device (14) for transferring data on a computer system bus (10), comprising:

means for recognising that the device is being addressed on the bus by a second device (12) connected to the bus;

means for communicating a transfer mode signal (SDR (0.1)) the bus to said second device, which signal indicates one or more data transfer capabilities of the device, said transfer mode signal comprising a high speed mode signal indicating whether the device is capable of transferring data in a high speed data transfer mode;

means for transferring data to or from said second device under the control of a high speed clock signal, wherein the high speed clock signal is separate from a normal clock signal.

8. A method for transferring data on a computer system bus (10) between a master subsystem (12) and a slave subsystem (14), said bus including a line for communicating a normal clock signal (CMD) under control of the master subsystem, the method comprising the steps of:

the master subsystem initiating a data transfer with the slave subsystem;

the slave subsystem issuing a transfer mode signal (SDR (0.1)) to said master subsystem over a control line of said bus, said transfer mode signal indicating whether the slave subsystem is capable of

* C L M S *

transferring data according to a high speed clock (SD STROBE), separate from the normal clock;

the master subsystem selecting, in response to said indication, whether to transfer data between the master subsystem and the slave subsysten according to said high speed clock or according to said normal clock;

in the case of a transfer according to said high speed clock, the master subsystem driving said high speed clock signal on the bus; and

in the case of a transfer according to said normal clock, transferring data between said master and slave subsystems according to said normal clock.

9. A computer system comprising a master subsystem (12) including a device as claimed in claim 6 connected by means of a computer bus (10) to a slave subsystem (14) including a device as claimed in claim 7.

Patentansprüche

1. Ein System zur Datenübertragung in einem Rechner zwischen einem übergeordneten Subsystem (12) und einem nachgeordneten Subsystem (14), enthaltend:
Einen Bus (10), der an das übergeordnete Subsystem und an das nachgeordnete Subsystem angeschlossen ist zur Übermittlung von Signalen zwischen ihnen, wobei der Bus eine Vielzahl von Adressenleitungen, Datenleitungen und Steuerleitungen aufweist; und
eine Taktleitung im Bus zur Übermittlung eines normalen Taktsignals (CMD) unter Steuerung des übergeordneten Subsystems vorhanden ist,
dadurch gekennzeichnet, daß der Bus eine Hochgeschwindigkeitstakt-Steuerleitung zur Übertragung eines gesonderten Hochgeschwindigkeitstakts (SD STROBE) unter der Steuerung des übergeordneten Subsystems aufweist; und
der Bus eine Übertragungssteuerleitung beinhaltet, um vom nachgeordneten Subsystem an das übergeordnete Subsystem ein Übertragungsmodussignal (SDR(0.1)) zu übermitteln, das hinweisend ist für eine oder mehrere Datenübertragungsfähigkeiten des nachgeordneten Subsystems, wobei dieses Übertragungsmodussignal ein Hochgeschwindigkeitsmodussignal enthält, das anzeigt, ob das nachgeordnete Subsystem in der Lage ist, Daten gemäß diesem Hochgeschwindigkeitstaktsignal zu übertragen;
wobei das übergeordnete Subsystem als Reaktion auf dieses Hochgeschwindigkeitsmodussignal auswählt, ob die Daten zwischen dem übergeordneten Subsystem und dem nachgeordneten Subsystem gemäß dem normalen Takt oder gemäß dem Hochgeschwindigkeitstakt übertragen werden.

2. Ein System gemäß Anspruch 1, in dem das Übertragungsmodussignal ein Übertragungsgeschwindigkeitssteuersignal enthält;
wobei das Übertragungsgeschwindigkeitssteuersignal vom nachgeordneten Subsystem benutzt wird in den Fällen, in denen das nachgeordnete Subsystem in der Lage ist Datenübertragungen gemäß dem Hochgeschwindigkeitstakt durchzuführen, um dem übergeordneten Subsystem die maximale Rate anzuzeigen, zu der das nachgeordnete Subsystem eine solche Datenübertragung durchführen kann.

3. Ein System gemäß einem der vorstehenden Ansprüche, in dem das Übertragungsmodussignal ein Übertragungsbreite-Steuersignal (DS(16.32)) enthält, das die maximale Datenwortbreite anzeigt, die vom nachgeordneten Subsystem übertragen werden kann;
und in dem das übergeordnete Subsystem als Reaktion auf dieses Übertragungsbreitensteuersignal eine Datenwortbreite auswählt, die innerhalb der Fähigkeiten des nachgeordneten Subsystems zum Übertragen der Daten zwischen dem übergeordneten Subsystem und dem nachgeordneten Subsystem liegt.

4. Ein System gemäß einem der vorstehenden Ansprüche, in dem der Bus eine Bereit-Steuerleitung aufweist, die vom nachgeordneten Subsystem benutzt wird, um dem übergeordneten Subsystem während einer Datenübertragung gemäß dem Hochgeschwindigkeitstaktsignal mitzuteilen, ob das nachgeordnete Subsystem bereit zum Abschluß der Übertragung eines einzelnen Datenfelds ist;
und in dem die Übertragung vom übergeordneten Subsystem nicht abgeschlossen wird, bis das nachgeordnete Subsystem anzeigt, daß es zum Abschluß der Übertragung bereit ist.

5. Ein System gemäß einem der vorstehenden Ansprüche, in dem das übergeordnete Subsystem während einer Datenübertragung gemäß dem Hochgeschwindigkeitstaktsignal nur eine einzige Adresse auf den

* C L M S *

Bus legt.

6. Eine Vorrichtung (12) zum Übertragen von Daten auf einen Rechnersystembus (10), enthaltend:
Mittel zum Legen eines normalen Taktsignals (CMD) auf eine Taktleitung in dem Bus;
Mittel zum Legen eines Hochgeschwindigkeitstaktsignals (SD STROBE) auf den Bus, das getrennt vom normalem Taktsignal ist;
Mittel zum Ansprechen einer zweiten Vorrichtung (14), die an den Bus angeschlossen ist;
Mittel zum Erfassen eines Datenübertragungssignals (SDR(0.1)) von der zweiten Vorrichtung auf dem Bus, wobei dieses Signal eine oder mehrere Datenübertragungsfähigkeiten dieser zweiten Vorrichtung anzeigt, wobei dieses Übertragungsmodussignal ein Hochgeschwindigkeitsmodussignal beinhaltet, das angibt, ob diese zweite Vorrichtung in der Lage ist, Daten gemäß dem Hochgeschwindigkeitstaktsignal zu übertragen;
Mittel zum Übertragen von Daten zu oder von dieser zweiten Vorrichtung gemäß dem normalen Taktsignal;
Mittel zum Übertragen von Daten zu oder von dieser zweiten Vorrichtung unter der Steuerung des Hochgeschwindigkeitstaktsignals; und
Mittel zum Wählen in Reaktion auf das Hochgeschwindigkeitstaktsignal, ob die Daten zwischen der Vorrichtung und der zweiten Vorrichtung gemäß dem normalen Takt oder gemäß dem Hochgeschwindigkeitstakt übertragen werden.

7. Eine Vorrichtung (14) zum Übertragen von Daten auf einen Rechnersystembus (10), enthaltend:
Mittel zum Erkennen, daß die Vorrichtung auf dem Bus von einer zweiten Vorrichtung (12), die an den Bus angeschlossen ist, angesprochen wird;
Mittel zum Übertragen eines Datenübertragungssignals (SDR(0.1)) auf den Bus zu der zweiten Vorrichtung, wobei dieses Signal eine oder mehrere Datenübertragungsfähigkeiten der Vorrichtung anzeigt, wobei dieses Übertragungsmodussignal ein Hochgeschwindigkeitsmodussignal beinhaltet, das angibt, ob diese zweite Vorrichtung in der Lage ist, Daten gemäß dem Hochgeschwindigkeitstaktsignal zu übertragen;
Mittel zum Übertragen von Daten zu oder von der zweiten Vorrichtung unter der Steuerung eines Hochgeschwindigkeitstaktsignals, wobei das Hochgeschwindigkeitstaktsignal von einem normalen Taktsignal getrennt ist.

8. Ein Verfahren zur Übertragung von Daten über einen Rechnersystembus (10) zwischen einem übergeordneten Subsystem (12) und einem nachgeordneten Subsystem (14), wobei der Bus eine Leitung enthält zur Übermittlung eines normalen Taktsignals (CMD) unter der Steuerung des übergeordneten Subsystems, wobei dieses Verfahren die folgenden Schritte umfaßt:
Das übergeordnete Subsystem läßt eine Datenübertragung mit dem nachgeordneten Subsystem anlaufen;
das nachgeordnete Subsystem legt ein Übertragungsmodussignal (SDR(0.1)) an das übergeordnete Subsystem auf eine Steuerleitung des Busses, wobei dieses Übertragungsmodussignal anzeigt, ob das nachgeordnete Subsystem in der Lage ist, Daten gemäß einem Hochgeschwindigkeitstakt (SD STROBE) zu übertragen, der vom normalen Takt getrennt ist;
dabei wählt das übergeordnete Subsystem als Reaktion auf diese Angabe, ob die Daten zwischen dem übergeordneten Subsystem und dem nachgeordneten Subsystem gemäß dem Hochgeschwindigkeitstakt oder gemäß den Normaltakt übertragen werden;
im Falle einer Übertragung gemäß dem Hochgeschwindigkeitstakt legt das übergeordnete Subsystem das Hochgeschwindigkeitstaktsignal auf den Bus; und
im Falle einer Übertragung mit dem Normaltakt überträgt es die Daten zwischen dem übergeordneten Subsystem und dem nachgeordneten Subsystem gemäß dem Normaltakt.

9. Ein Rechnersystem, enthaltend ein übergeordnetes Subsystem (12) einschließlich einer Vorrichtung gemäß Anspruch 6, die über einen Rechnerbus (10) an ein nachgeordnetes Subsystem (14) angeschlossen ist, einschließlich einer Vorrichtung gemäß Anspruch 7.

Revendications

1. Système de transfert de données dans un ordinateur entre un sous-système maître (12) et un sous-

* C L M S *

système esclave (14) comprenant:

un bus (10) connecté au sous-système maître et au sous-système esclave pour communiquer des signaux entre eux, ledit bus comprenant une pluralité de lignes d'adresses, de lignes de données et de lignes de contrôle, et

une ligne d'horloge dans ledit bus pour communiquer un signal d'horloge normale (CMD) sous la commande du sous-système maître,

caractérisé en ce que ledit bus comprend une ligne de contrôle d'horloge à grande vitesse pour communiquer un signal d'horloge à grande vitesse séparé (SD STROBE) sous la commande du sous-système maître, et ledit bus comprend une ligne de commande de transfert pour communiquer du sous-système esclave au sous-système maître un signal de mode de transfert (SDR(0.1)) indiquant une ou plusieurs possibilités de transfert de données du sous-système esclave, ledit signal de mode de transfert comprenant un signal de mode à grande vitesse indiquant si le sous-système esclave est capable de transférer des données selon ledit signal d'horloge à grande vitesse,

dans lequel le sous-système maître, en réponse audit signal de mode à grande vitesse, sélectionne si les données sont transférées entre le sous-système maître et le sous-système esclave selon ladite horloge normale ou selon ladite horloge à grande vitesse.

2. Système tel que revendiqué dans la revendication 1, dans lequel ledit signal de mode de transfert comprend un signal de commande de vitesse de transfert,

ledit signal de commande de vitesse de transfert étant utilisé par le sous-système esclave, dans le cas où le sous-système esclave peut effectuer des transferts de données selon ladite horloge à grande vitesse, pour indiquer au sous-système maître la vitesse maximum à laquelle le sous-système esclave est capable d'effectuer de tels transferts de données.

3. Système tel que revendiqué dans l'une des revendications précédentes, dans lequel ledit signal de mode de transfert comprend un signal de commande de largeur de transfert (DS(16.32)) indiquant la largeur de mot de données maximum qui peut être transférée par le sous-système esclave,

et dans lequel le sous-système maître, en réponse audit signal de commande de largeur de transfert, sélectionne une largeur de mot de données dans les possibilités du sous-système esclave pour transférer des données entre le sous-système maître et le sous-système esclave.

4. Système tel que revendiqué dans l'une des revendications précédentes, dans lequel ledit bus comprend une ligne de contrôle prêt utilisée par le sous-système esclave pour communiquer au sous-système maître, durant un transfert de données selon ledit signal d'horloge à grande vitesse, que le sous-système esclave est prêt à compléter le transfert d'un élément de données individuel,

et dans lequel ledit transfert n'est pas complété par le sous-système maître jusqu'à ce que le sous-système esclave indique qu'il est prêt à compléter ledit transfert.

5. Système tel que revendiqué dans l'une des revendications précédentes, dans lequel le sous-système maître communique seulement une adresse sur le bus durant un transfert de données selon ledit signal d'horloge à grande vitesse.

6. Dispositif (12) de transfert de donnés sur un bus (10) de système ordinateur comprenant:

des moyens pour communiquer un signal d'horloge normale (CMD) sur une ligne d'horloge dudit bus,

des moyens pour communiquer, sur ledit bus, un signal d'horloge à grande vitesse (SD STROBE) séparé dudit signal d'horloge normale,

des moyens d'adressage d'un deuxième dispositif (14) connecté audit bus,

des moyens pour détecter un signal de transfert de mode (SDR/0.1) sur le bus à partir dudit deuxième dispositif, lequel signal indique une ou plusieurs possibilités de transfert de données dudit deuxième dispositif, ledit signal de transfert de mode comprenant un signal de mode à grande vitesse indiquant si ledit deuxième dispositif est capable de transférer des données selon ledit signal d'horloge de grande vitesse,

des moyens pour transférer des données vers ou à partir dudit deuxième dispositif selon ledit signal d'horloge normale,

des moyens pour transférer des données vers ou à partir dudit deuxième dispositif sous la commande du signal d'horloge à grande vitesse, et

des moyens pour sélectionner, en réponse audit signal de mode à grande vitesse, s'il faut transférer

* C L M S *

des données, entre ledit dispositif et ledit deuxième dispositif, selon l'horloge normale ou l'horloge à grande vitesse.

7. Dispositif (4) de transfert de données sur un bus de système d'ordinateur (10) comprenant:

des moyens pour reconnaître que le dispositif est adressé sur le bus par un deuxième dispositif (12) connecté au bus,

des moyens pour communiquer un signal de mode de transfert (SDR(0.1)) sur le bus audit deuxième dispositif, lequel signal indique une ou plusieurs possibilités de transfert du dispositif, ledit signal de mode de transfert comprenant un signal de mode à grand vitesse indiquant si le dispositif est capable de transférer des données dans un mode de transfert de données à grande vitesse,

des moyens pour transférer des données vers ou à partir dudit deuxième dispositif sous la commande d'un signal d'horloge à grande vitesse, dans lesquels le signal d'horloge à grande vitesse est séparé d'un signal d'horloge normale.

8. Méthode pour transférer des données sur un bus de système d'ordinateur (10) entre un sous-système maître (12) et un sous-système esclave (14), ledit bus comprenant une ligne pour communiquer un signal d'horloge normale (CDM) sous la commande du sous-système maître, la méthode comprenant les étapes suivantes:

le sous-système maître initialise un transfert de données avec le sous-système esclave,

le sous-système esclave délivre un signal de mode de transfert (SDR(0.1)) audit sous-système maître sur une ligne de contrôle dudit bus, ledit signal de mode de transfert indiquant si le sous-système esclave est capable de transférer des données selon un signal d'horloge à grande vitesse (SD STROBE) séparé du signal d'horloge normale,

le sous-système maître décide, en réponse à ladite indication, s'il faut transférer des données entre le sous-systèmes maître et le sous-système esclave selon ladite horloge à grande vitesse ou selon ladite horloge normale,

dans le cas d'un transfert selon ladite horloge à grande vitesse, le sous-système maître commande ledit signal d'horloge à grande vitesse sur le bus, et

dans le cas d'un transfert selon ladite horloge normale, le transfert de données s'effectue entre lesdits sous-système maître et esclave selon ladite horloge normale.

9. Système d'ordinateur comprenant un sous-système maître (12) comportant un dispositif tel que revendiqué dans la revendication 6, connecté au moyen d'un bus d'ordinateur (10) à un sous-système esclave (14) comprenant un dispositif tel que revendiqué dans la revendication 7.

* C L M S *