(12)

PATENT SPECIFICATION (11) 1 571 858

1 571 858

(21) Application No. 20894/77 (22) Filed 18 May 1977
(31) Convention Application No. 693 821 (32) Filed 7 June 1976 in
(33) United States of America (US)
(44) Complete Specification published 23 July 1980
(51) INT. CL.³ G06F 13/00
(52) Index at acceptance G4A SX



(72) Inventors WILLIAM CLYDE COCKERILL
LOUIS MICHAEL HORNUNG
DONAVON WILLIAM JOHNSON
RICHARD ALAN VRBA

(54) DATA PROCESSING APPARATUS

(71) We, INTERNATIONAL BUSINESS MACHINES CORPORATION, a Corporation organized and existing under the laws of the State of New York in the United States of America, of Armonk, New York 10504, United States of America do hereby declare the invention for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement: —

The present invention relates to data processing apparatus and is concerned with controlling the transmission of data signals between a processor and its associated memory and input/output devices.

Through the utilization of LSI MOS technology the size and cost of processors have decreased manyfold. Accordingly it has become possible to construct an entire processor on a single LSI MOS chip. However, these devices suffer from the disadvantage of having a high circuit to pin ration, meaning pins are at a premium. One method in the prior art sought to overcome this problem by using the same pins for input and output and using internal circuitry to switch the data between input and output buffers of the processor. With this arrangement the system is limited to performing input and output operations serially. That is, an input or output operation must be completed before the next input or output can take place. This degrades processor performance because the speed of operation of the processor is limited by the speed of the input and output device.

LSI MOS technology has also led to dramatic size and cost reductions in memory. In using this technology for memory application, there is a critical tradeoff between cost and speed of application. The faster the memory operates the larger the electronic devices with which the memory must be built — hence the more expensive the memory. One of the key parameters by which the speed of the memory is measured is called access time. Access time is the time from which the memory is initially commanded to read the data at a certain address to which that data is available from the memory. Typically in the prior art the address from which data is to be read, the reading of that data, and the transfer of the data into a register in the processor all takes place within a period of time called a machine cycle. Only after all of this has been accomplished can the address for the next read (or write) be specified to the memory and the next machine cycle commenced. The memory access time is, consequently, very critical to determining the speed with which the processor operates.

The present invention provides digital data processing apparatus including a processor (CPU), a memory, and plurality of (relatively low speed) I/O devices and buss control apparatus which comprises a first data buss connected to the output of the CPU and the input of the memory for transferring data signals from the CPU to the memory, a second data buss connected to the output of the memory, gating means having a first input connected to the first data buss and a second input connected to the second data buss, a third data buss connected to the output of the gating means and to the input of the CPU for transferring data from the gating means to the CPU, an I/O data buss connected to the output of the gating means and to the (relatively low speed) I/O devices for transferring data between the gating means and the I/O devices, and decoding means connected to the gating means for decoding signals from the CPU for controlling the gating means to select the proper buss onto which received data is to be passed.

In one form of apparatus according to the invention disclosed hereinafter by way

of example, there is provided control circuitry which selectively steers processor inputs and outputs between the memory data buss and the I/O data buss. The system includes unidirectional data busses for communication between the processor and the memory and a bidirectional data buss for communication with the plurality of I/O devices. The steering control circuit includes a decoder and gate. The gate is constructed of circuitry which provides a set of common electrical buffering for all the devices attached to the bidirectional I/O channel. Electrical buffering consists of a power up function to interface the low drive current of FET I/O devices to the higher drive currents of faster bipolar channels which interface between the processor and the memory. The decoder receives signals from the processor and controls the gate to steer the transmitted data onto the proper buss for transmission to the appropriate device. By selectively steering the data, processor inputs can be overlapped with output operations enhancing the operation of the processor in a pipelined mode.

Such a bussing structure allows the address for the next memory cycle, and the data if this next cycle is a write into memory, to be emitted from the processor to the memory before the data read from the memory has been transferred to the processor. Using this technique the speed of the processor is greatly enhanced without bearing the cost usually attended to a corresponding increase in the memory speed. Obviously data can not be utilized in the processor until after the data has been transferred to the processor. Consequently, this arrangement is usable only with processors which are designed to tolerate this delay between the specification of the memory address and the entering of the data at that address into the processor. Processors with this capability are described herein as "pipelined processors".

Performance degradation is minimized while at the same time saving pins in critical areas by using unidirectional busses in combination with bidirectional busses. In a typical system, most of the pins are consumed by the I/O adapters. Most of the data transfers, however, are between the processor and the memory. Using a bidirectional buss at the adapters and unidirectional busses at the processor and memory constitutes an excellent cost/performance design point.

The aforementioned form of apparatus according to the invention is now described in more detail with reference to the accompanying drawings in which: —

FIGURE 1 is a block diagram of the data processing apparatus;

FIGURE 2 is a functional block diagram of the data steering logic of the apparatus;

FIGURE 3 is an overall system diagram including a detailed implementation of the gate logic of Figure 2.

Figure 1 is a block diagram of a data processing system which includes the data steering logic of the present invention. The system includes a CPU 1 whose output connected to data bus 11 and whose input is connected to data buss 14. The CPU 1 contains a processor and a direct memory access controller (DMA) for transferring data between the memory and I/O device between processor cycles. Both data buss 11 and data buss 14 are preferably unidirectional data busses, meaning data will travel in only one direction on the buss. The CPU 1 also is connected to decode 4 of buss control logic 3 through control signal line 15. Control signal line 15 carries coded timing and control data to buss control logic 3 from the system clock and the CPU 1. Data buss 11 is connected to the input of memory 2 and carries address and data signals from the CPU to memory 2. Data buss 11 is also connected to the input of gate 5 and carries output data signals from the CPU 1 to gate 5.

The output of memory 2 is connected to unidirectional data buss 12 which carries data from memory 2 into gate 5. Gate 5 is connected to decode 4 through signal line 16 and receives control signals from decode 4 causing gate 5 to select buss 14 and buss 13 when required for low speed I/O data. Buss 13 is a bidirectional buss which communicates with low speed I/O devices 6-10 and buss 14 is a unidirectional bus connected to the input of CPU 1.

Referring now to Figure 2, a more detailed drawing of decode 4 and gate 5 is shown. Decode 4 is connected to CPU 1 through control lines 15 and receives coded control signals from the CPU 1 which determine the configuration gate 5 will place the data busses in. The signals received by decode 4 includes $\overline{\text{CWRT}}$ 451 which determines whether or not the operation is a processor write, $\overline{\text{POR}}$ 421 which determines whether or not the operation is a power on "reset", $\overline{\text{ADR12}}$ 419 which indicates the address cycle of the operation, $\overline{\text{MEMCY}}$ 423 which indicates whether or not the operation is a memory cycle operation initiated by the processor direct memory access

controller, $\overline{\text{DMAM}}$ 457 which samples the request line of the direct memory access controller and determines whether or not the operation is to be a direct memory access (or cycle steal) operation, and XCC 475 and $\overline{\text{CHNSW}}$ 405 which provide the timing signals for the operation. XCC defines the beginning of a cycle and corresponds with the processor's phase one. $\overline{\text{CHNSW}}$ defines the time that the data in buss (DIϕ) 14 should be switched during the processor's phase two. Two other inputs, $\overline{\text{INHDI}}$ 401 and $\overline{\text{INHIO}}$ 413 provide test signals which can be used to override the decode and gating functions during testing of the module. The Direct Memory Access Controller (DMA), shown in Figure 3, is dotted with the processor to perform direct transfers of data between memory and the I/O devices between cycles of operation of the processor. The DMA is connected to the processor input buss 14 and output buss 11 and operates in the overlapped mode. The combination of logical ones and zeros on control line 15 is decoded by the control decode 4 and the result outputted to gate 5 on line 16. If the operation requested by the CPU is an output then gate 5 will connect data buss 11 to data buss 13 and data buss 14 enabling data to be transferred from the CPU to the selected one of the I/O devices 6-10 or to an I/O device on buss 14. If the requested operation is an input to the CPU 1 from I/O devices 6-10, the gate 5 will connect data buss 13 to data buss 14 for transfer of data to the CPU 1. It additionally connects data buss 11 to data buss 14 for transfer of data to the CPU 1 from I/O devices on data buss 11. If the operation does not request one of the I/O devices 6-10, then the buss control logic 3 electrically isolates the relatively low speed I/O data buss 13 from the processor output data buss 11 and the processor input data buss 14. Processor data busses 11 and 14 are high speed data busses which allow the processor to operate in the pipelined mode by overlapping input and output accesses to memory 2. Referring to Figure 1, I/O devices which operate at a speed comparable to memory 2 may be dotted directly onto data busses 11 and 14 in parallel with memory 2 without degrading processor performance as shown by dashed box 23. Also other high speed memory devices may be connected to data busses 11 and 14 as shown by dashed box 24. Memory devices 24 must have their own gate logic analogous to gate logic 5. Dashed box 25 shows a second set of bus control logic called SLIO which is identical to bus control logic 3 and can be used to control data transfer to other low speed I/O devices physically located on other circuit cards.

The I/O devices 6-10 operate at comparably slower speeds than memory 2 and degrade system performance by preventing the overlapping of input and output. However, accesses to these devices require a relatively small amount of the processor time and overall system performance is greatly enhanced by isolating these slower devices from the system during high speed accesses.

High speed I/O devices which do not require a DMA type read and write function may be connected to either data buss 11 or data buss 14. In this configuration, the I/O devices are connected directly to the processor on buss 11 for output, and input to the processor from the devices takes place through gate 5 and data in buss 14. If the I/O device is connected directly to data in buss 14, then inputs to the processor from the device are performed directly, but outputs from the processor must pass through gate logic 5. More importantly, when connecting I/O devices directly to data in buss 14 provision must be made to prevent the I/O device from using the data buss 14 simultaneously with processor reads from memory 2. To prevent the I/O device from destroying data being transferred from memory 2 to the processor, the I/O device must be made such that they do not start putting their data onto data buss 14 until CHNSW. If these high speed I/O devices require both DMA read and write, they must connect to both data buss 11 and data buss 14.

Referring now to Figure 3 a more detailed drawing of the gate logic 5 is shown. The outputs from the control decodes 4 are received by logic gate 5 over control line 16. These signals are $\overline{\text{OUTPUT}}$ 417 which defines whether or not the operation is an output from CPU 1, $\overline{\text{DMARD}}$ 455 and $\overline{\text{DMAWT}}$ 463 which define whether or not the operation is a direct memory access read or write to memory INOUT 441 and $\overline{\text{INOUT}}$ 437 which define whether or not the operation is an I/O operation, INHDI 403 which defines whether the operation is a test, and $\overline{\text{IOCIN}}$ 429 which defines whether the operation is an input from I/O.

$\overline{\text{IOCIN}}$ 429 and $\overline{\text{DMAWT}}$ 463 are inputs to NAND gate 504. NAND gate 504 has its output dotted with the output of NAND gate 502 and controls whether data on the bidirectional data buss 13 coming from the

I/O units 6-10 is input to NOR gate 506. NOR gate 506 drives buss 11 which is connected to the input of memory 2 and also dotted at node 505 with inputs 11a and 11b to gates which control buss 14 into the processor and buss 13 for output to the I/O devices on that buss. The input of NAND gate 508 is connected to output from NOR gate 506 by buss 11a. The operation of NAND gate 508 is controlled by signal INOUT on input line 441. NAND gate 508 controls the passage of data from the I/O devices onto the buss 14 to the input of the processor. The output of NAND gate 508 is dotted with the outputs of NAND gates 510 and 512 at node 507. NAND gate 512 has its input connected to the output of memory 2 by data buss 12. NAND gate 512 is controlled by control signal $\overline{\text{INOUT}}$ on input line 437 to control the passage of data from the memory onto data buss 14 as input to the processor. The input to NAND gate 510 is control signal INHDI on line 403. INHDI can be used to inhibit the passage of data by either NAND gate 508 or NAND gate 512 during testing. The dotted outputs of NAND gates 508, 510 and 512 drives the input of NOR gate 514. NOR gate 514 is the buss driver for data buss 14 connected to the input of the CPU 1.

The dotted output of NOR gate 506 is also connected to input of NAND gate 516 by buss 11b. This provides a path from buss 11 for earlier data output from the processor to the I/O devices on buss 13. This drives the input of NAND gate 516. Data buss 11b provides an alternative path for data being output from the processor on data buss 11 to an I/O device attached to I/O buss 13. This enables the processor to overlap outputs to the I/O devices with processor reads from the memory 2 which take place through data busses 12 and 14. Thus, when a read from memory is initiated by the processor, the processor does not have to wait for the memory data transfer, but can continue operating and may overlap the output of data either to I/O devices connected to data buss 11 or data buss 13. The operation of NAND gate 516 is controlled by control signal $\overline{\text{DMARD}}$ on line 455. The output of NAND gate 516 drives NOR gate 522 and is dotted with the outputs of NAND gates 518 and 520. NAND gate 518 has its input dotted with the output of NOR gate 514 and controls the passage of data from the memory 2 to the I/O devices during the optional cycle steal operation. The operation of NAND gate 518 is controlled by control signal $\overline{\text{OUTPUT}}$ on line 417. NAND gate 520 is controlled by the control signals $\overline{\text{DMARD}}$ and $\overline{\text{OUTPUT}}$ and inhibits the passage of data to I/O buss driver 522 when the operation is an input of data to the processor from memory 2 or one of the I/O devices. NOR gate 522 is the buss driver from I/O buss 13.

The buss drivers 506, 514 and 522 are all open collector drivers. Therefore each data buss is provided with a pull-up resistor R connected to a power source V.

OPERATION

Referring to Figure 2 and 3, assume that the CPU 1 requests a transfer of data from the memory 2 to the CPU 1. The processor and system clock output the following signals to control decode 4 on control buss 15:

a logical zero for $\overline{\text{MEMCY}}$ on line 423 indicating that a memory cycle is requested,

a logical one for $\overline{\text{DMAM}}$ on line 457 indicating that the operation is not a direct memory access, a logical one for $\overline{\text{CWRT}}$ one line 451 indicating that the operation is a processor read, a logical one for $\overline{\text{POR}}$ indicating that the operation is not a power on sequence, and logical ones for $\overline{\text{INHDI}}$ and $\overline{\text{INHIO}}$ indicating the operation is not a test. The remaining signals, $\overline{\text{ADRI2}}$ on line 419, $\overline{\text{CHNSW}}$ on line 405 and XCC on line 475 provide address and cycle timing information.

These signals are decoded by control decode logic 4 to provide the following gate control signals to logic gates 5 on control lines 16: a logical one for $\overline{\text{IOCIN}}$ on line 429 indicating that the operation is not an I/O input, a logical one for $\overline{\text{DMAWT}}$ on line 463 indicating that the operation is not a direct memory access write, a logical zero for INOUT and a logical one for $\overline{\text{INOUT}}$ indicating that the operation is not an I/O operation, a logical zero for INHDI on line 403 indicating that the operation is not a test, a logical one for $\overline{\text{DMARD}}$ on line 455 indicating that the operation is not a direct memory access read, and a logical one for $\overline{\text{OUTPUT}}$ indicating that the operation is not an output from the processor.

Data is output from the memory onto

memory out data buss MOϕ 12 which drives NAND gate 512. The control signal $\overline{\text{INOUT}}$ to NAND gate 512 on line 437 is a logical one and controls NAND gate 512 to pass the data through output node 507 to the input of buss driver 514. Since INOUT on line 441 which controls NAND gate 508 and INHDI on line 403 which controls NAND gate 510 are both logical zeroes, no inhibit signal is provided at node 507 to prevent the transfer of data from gate 512 to buss driver 514. Buss driver 514 transfers the data onto data in buss DIϕ 14 which is connected to the input of processor 1. The data is also input to NAND gate 518 which is dotted with data buss 14. However, the combination of $\overline{\text{DMARD}}$ and $\overline{\text{OUTPUT}}$ inhibits the transfer of the data into buss driver 522 which controls the I/O data buss 13.

For a process write into memory operation the gate logic decodes remain the same and the processor writes directly into memory 2 on data buss MIϕ.

For an output from the processor to a selected one of the I/O devices, the processor outputs the following control signals to control decode 4 on control buss 15: a logical one for $\overline{\text{MEMCY}}$ on line 423 indicating that the operation is not a memory cycle, a logical one for $\overline{\text{DMAM}}$ on line 457 indicating that the operation is not a direct memory access, a logical one for $\overline{\text{POR}}$ on line 421 indicating that the operation is not a power on sequence, and logical zero for $\overline{\text{CWRT}}$ on line 451 indicating that the operation is a write operation. The remaining signals remain the same as for processor read from memory.

These signals are decoded by control decodes 4 to provide the following control signals to logic 5 on control lines 16: a logical one for $\overline{\text{IOCIN}}$ on line 429 indicating the operation is not an I/O input, a logical one for $\overline{\text{DMAWT}}$ on line 463 indicating that the operation is not a direct memory access write into memory, a logical one for INOUT on line 441 and a logical zero for $\overline{\text{INOUT}}$ on line 437 indicating that the operation is an I/O operation, a logical zero for INHDI on line 403 indicating that the operation is not a test, a logical one for $\overline{\text{DMARD}}$ on line 455 indicating that the operation is not a direct memory access read and a logical zero for $\overline{\text{OUTPUT}}$ on line 417 indicating that the operation is a processor output.

The data is output from the processor onto data buss 11 into the data steering gate logic 5. Data buss 11 is dotted at node 505 with the output of buss driver 506 and the inputs to NAND gates 508 and 516. The control signal for NAND gate 516 is $\overline{\text{DMARD}}$, a logical one, on line 455. $\overline{\text{DMARD}}$ controls NAND gate 516 to pass the data to buss driver 522. $\overline{\text{OUTPUT}}$ produces a logical zero at the outputs of both NAND gate 518 and 520 so as not to inhibit the passage of data from NAND gate 516 to buss driver 522. Buss driver 522 then places the data on I/O buss 13 to the selected I/O device.

For an input operation from I/O, the only change in the processor control outputs is $\overline{\text{CWRT}}$ on line 451 becomes a logical 1 indicating that the operation is not a processor write. This change is decoded by the control decode 4 to yield a change of state in $\overline{\text{OUTPUT}}$ on line 417 which becomes a logical one indicating that the operation is not an output and $\overline{\text{IOCIN}}$ on line 429 which becomes a logical zero indicating that the operation is an input from I/O. Input data is placed on the data buss 13 from the selected I/O device. $\overline{\text{IOCIN}}$ as a logical zero produces a logical one at the output of NAND gate 504 so as not to inhibit the passage of data through NAND gate 502 which is output dotted with NAND gate 504. The output of NAND gate 502 drives buss driver 506 to drive the data into NAND gates 508 and 516. For high speed I/O devices connected to data buss 11, the data is placed on data buss 11 and input to NAND gates 508 and 516 over lines 11a and 11b respectively. Control signal INOUT on line 441 is a logical one and causes gate 508 to pass the data to buss driver 514. NAND gates 510 and 512 which dotted with NAND gate 508 at 507 do not inhibit the output from NAND gate 508 because INHDI and $\overline{\text{INOUT}}$ are at logical zero. Thus buss driver 514 drives the data onto data in buss DIϕ 14 which is connected to the input of the processor.

Outputs from the processor to memory or I/O are overlapped with processor reads from memory 2. During the current cycle it sets up the memory address. The data is transferred during the second half of the

current cycle or the first half of the next cycle. The processor outputs the address and data to the memory 2 which latches up the address in the well known manner. The output of the memory onto data buss 14 may be delayed up to one full processor cycle time. During this time the processor is free to input or output data to the I/O devices or access the memory again. If the next instruction is an output then the data is transferred onto data buss 11 to common node 505. The data is inhibited from passing through NAND gate 508 by control signal INOUT. This prevents the data from interfering with the transfer of data out of memory which must pass through common node 507. However, data buss 11b provides a by pass for the data into NAND gate 516. The output of NAND gate 516 is controlled by $\overline{\text{OUTPUT}}$ to pass the data into buss driver 522 as was previously described for the output operation. The data from memory 2 passes through memory out buss 12, gate 512, buss driver 514 and data in buss 14 as was previously described.

The buss control logic also controls the transfer of data during two special operations called Direct Memory Access read (DMARD) and Direct Memory Access Write (DMAWT). The two operations are controlled by a direct memory access controller which is dotted with the processor and shares the same power of circuits. The direct memory access controller executes a cycle steal operation between cycles of the processor and transfers data directly between the memory 2 and the I/O devices 6-10 or vice versa.

During direct memory access read from the memory 2 to a selected one of the I/O devices, the processor output signals remain the same as for a processor read but the DMA controller changes the state of $\overline{\text{DMAM}}$ to a logical zero indicating that the operation is a direct memory access. The change in state of the $\overline{\text{DMAM}}$ on line 457 signal causes the control decode 4 to change the state of $\overline{\text{DMARD}}$ on line 455 from logical one to logical zero indicating that the operation is a DMA read. Data output from memory 2 on data buss 12 then takes the same path, as previously discussed for a processor read, through NAND gate 512 and buss driver 514 onto data buss in 14. However, the change of state of $\overline{\text{DMARD}}$ on control line 455 causes the data to also pass through NAND gate 518 and into buss driver 522. Buss driver 522 then places the data on data buss 13 to the selected I/O device 6-10. Since the data also appears on data in buss 14, the operation can be a read of data into the DMA itself as well as to I/O.

Similarly for a DMA write into memory 2 from an I/O device the processor signals remain the same as for a processor write except $\overline{\text{DMAM}}$ on line 457 becomes a logical zero. This in turn causes the control decode to convert $\overline{\text{DMAWT}}$ from a logical one to a logical zero indicating that the operation is a direct memory access write into memory. The selected I/O device places data onto data buss 13 which activates NAND gate 502. The change of $\overline{\text{DMAWT}}$ on control line 463 removes the inhibition at node 503 and allows the output of NAND gate 502 to drive buss driver 506. Buss driver 506 then places the data onto data buss 11 for input into the memory 2 on memory in buss MIϕ. Note that for the DMA write function to be incorporated into the system, data buss 11 must be a bidirectional data buss. The DMA controller also can move data from one memory address to another as well as initiate the input of data to memory from I/O.

While the preferred embodiment of the invention has been described with reference to NAND and NOR logic gates and a particular configuration of uni- and bi- directional busses, it will be appreciated by those of skill in the art that the invention may be implemented using AND and OR gates or any other suitable combination and other directional configurations of buss, as, for example, busses 12 and 14 unidirectional; 11 and 13 bidirectional or busses 11, 13 and 14 bidirectional and buss 12 unidirectional.

WHAT WE CLAIM IS:—

1. Digital data processing apparatus including a processor (CPU), a memory, and plurality of (relatively low speed) I/O devices and buss control apparatus which comprises a first data buss connected to the output of the CPU and the input of the memory for transferring data signals from the CPU to the memory, a second data buss connected to the output of the memory, gating means having a first input connected to the first data buss and a second input connected to the second data buss, a third data buss connected to the output of the gating means and to the input of the CPU for transferring data from the gating means to the CPU, an I/O data buss connected to the output of the gating means and to the (relatively low speed) I/O devices for transferring data between the gating means and the I/O devices, and decoding means connected to the gating means for decoding

current cycle or the first half of the next cycle. The processor outputs the address and data to the memory 2 which latches up the address in the well known manner. The output of the memory onto data buss 14 may be delayed up to one full processor cycle time. During this time the processor is free to input or output data to the I/O devices or access the memory again. If the next instruction is an output then the data is transferred onto data buss 11 to common node 505. The data is inhibited from passing through NAND gate 508 by control signal INOUT. This prevents the data from interfering with the transfer of data out of memory which must pass through common node 507. However, data buss 11b provides a by pass for the data into NAND gate 516. The output of NAND gate 516 is controlled by $\overline{\text{OUTPUT}}$ to pass the data into buss driver 522 as was previously described for the output operation. The data from memory 2 passes through memory out buss 12, gate 512, buss driver 514 and data in buss 14 as was previously described.

The buss control logic also controls the transfer of data during two special operations called Direct Memory Access read (DMARD) and Direct Memory Access Write (DMAWT). The two operations are controlled by a direct memory access controller which is dotted with the processor and shares the same power of circuits. The direct memory access controller executes a cycle steal operation between cycles of the processor and transfers data directly between the memory 2 and the I/O devices 6-10 or vice versa.

During direct memory access read from the memory 2 to a selected one of the I/O devices, the processor output signals remain the same as for a processor read but the DMA controller changes the state of $\overline{\text{DMAM}}$ to a logical zero indicating that the operation is a direct memory access. The change in state of the $\overline{\text{DMAM}}$ on line 457 signal causes the control decode 4 to change the state of $\overline{\text{DMARD}}$ on line 455 from logical one to logical zero indicating that the operation is a DMA read. Data output from memory 2 on data buss 12 then takes the same path, as previously discussed for a processor read, through NAND gate 512 and buss driver 514 onto data buss in 14. However, the change of state of $\overline{\text{DMARD}}$ on control line 455 causes the data to also pass through NAND gate 518 and into buss driver 522. Buss driver 522 then places the data on data buss 13 to the selected I/O device 6-10. Since the data also appears on data in buss 14, the operation can be a read of data into the DMA itself as well as to I/O.

Similarly for a DMA write into memory 2 from an I/O device the processor signals remain the same as for a processor write except $\overline{\text{DMAM}}$ on line 457 becomes a logical zero. This in turn causes the control decode to convert $\overline{\text{DMAWT}}$ from a logical one to a logical zero indicating that the operation is a direct memory access write into memory. The selected I/O device places data onto data buss 13 which activates NAND gate 502. The change of $\overline{\text{DMAWT}}$ on control line 463 removes the inhibition at node 503 and allows the output of NAND gate 502 to drive buss driver 506. Buss driver 506 then places the data onto data buss 11 for input into the memory 2 on memory in buss MIϕ. Note that for the DMA write function to be incorporated into the system, data buss 11 must be a bidirectional data buss. The DMA controller also can move data from one memory address to another as well as initiate the input of data to memory from I/O.

While the preferred embodiment of the invention has been described with reference to NAND and NOR logic gates and a particular configuration of uni- and bi- directional busses, it will be appreciated by those of skill in the art that the invention may be implemented using AND and OR gates or any other suitable combination and other directional configurations of buss, as, for example, busses 12 and 14 unidirectional; 11 and 13 bidirectional or busses 11, 13 and 14 bidirectional and buss 12 unidirectional.

WHAT WE CLAIM IS:—

1. Digital data processing apparatus including a processor (CPU), a memory, and plurality of (relatively low speed) I/O devices and buss control apparatus which comprises a first data buss connected to the output of the CPU and the input of the memory for transferring data signals from the CPU to the memory, a second data buss connected to the output of the memory, gating means having a first input connected to the first data buss and a second input connected to the second data buss, a third data buss connected to the output of the gating means and to the input of the CPU for transferring data from the gating means to the CPU, an I/O data buss connected to the output of the gating means and to the (relatively low speed) I/O devices for transferring data between the gating means and the I/O devices, and decoding means connected to the gating means for decoding

signals from the CPU for controlling the gating means to select the proper buss onto which received data is to be passed.

2. Apparatus as claimed in claim 1, wherein the first, second, and third data busses are unidirectional data busses and the I/O data buss is a bidirectional data buss.

3. Apparatus as claimed in claim 1, wherein the second and third data busses are unidirectional data busses and the first and I/O data busses are bidirectional data busses.

4. Apparatus as claimed in claim 1, wherein the first, third and I/O data busses are bidirectional data busses and the second data buss is a unidirectional data buss.

5. Apparatus as claimed in claim 4 also including a plurality of relatively high speed I/O devices, one of the high speed I/O devices being a cycle steal device sharing the third data buss.

6. Apparatus as claimed in claim 5, wherein the decoding means controls the gate means to provide a path between the memory and the cycle steal I/O device during cycle steal operations.

7. Apparatus as claimed in any preceding claim, wherein the buss control apparatus is arranged to electrically isolate the relatively low speed I/O devices from the remainder of the system during non-accesses to such low speed I/O devices.

8. Apparatus as claimed in any preceding claim, wherein the gating means include means for connecting the output of the processor to a selected one of the I/O devices while simultaneously connecting the input of the processor to the output of the memory.

9. Apparatus as claimed in claim 1, wherein the control apparatus is substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

I. M. GRANT
Chartered Patent Agent.
Agent for the Applicants.

Printed for Her Majesty's Stationery Office by The Tweeddale Press Ltd., Berwick-upon-Tweed, 1980.
Published at the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained