PATENT SPECIFICATION (11) 1 589 180

1 589 180

(21) Application No. 51428/77 (22) Filed 9 Dec. 1977

(31) Convention Application No 757403 (32) Filed 6 Jan. 1977 in

(33) United States of America (US)

(44) Complete Specification Published 7 May 1981

(51) INT. CL.[3] G06F 13/00

(52) Index at Acceptancce
G4A 10P 12N 12P 13E 16D 16J FG

(72) Inventors: MAX ABBOTT BOUKNECHT
LOUIS PETER VERGARI

(19)

(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 provides data processing apparatus having a central processing unit, an I/O (input/output) channel, a first I/O bus directly connected with the central processing unit and I/O attachments attaching or to attach I/O devices connected to be powered by a first power supply and a second I/O bus and I/O attachments attaching or to attach further I/O devices connected to be powered by a second power supply, each I/O bus including data, address and control signal lines, there being an I/O interface connector circuit to control the interconnection of data, address, and control signal lines between the first I/O bus and the second I/O bus, the I/O interface connector circuit appearing to the first I/O bus as an attached I/O device and including bidirectional driver means interconnecting data and address lines of the first I/O bus and corresponding data and address lines of the second I/O bus, unidirectional driver means interconnecting control signal lines of the first and second I/O buses, and logic means, connected and responsive to predetermined signals being transferred between any attached I/O device and the I/O channel, and connected to said bidirectional driver means, to control the direction of signal transfer on the data and address lines.

By I/O attachments is meant connective circuitry for attachment of devices to the bus whether or not such devices are actually so attached since most such devices can be replaceable units.

The control of transfer of data between the storage of the central processing unit and peripheral I/O devices over an input/output bus, or interface, can take many forms. Among the forms of input/ouput control for the purpose of data transfer are, direct program instruction control for each transfer of data, the initiation of data transfers by a central processor after which subsequent data transfer is accomplished under control of the peripheral device without use of the central processor, and logic for handling interrupt requests from peripheral devices to inform the central processor of peripheral device status. Included in various of these concepts are control mechanisms for permitting peripheral devices to initiate interrupt handling in the central processor by directly informing the processor of the device and status requiring handling, or the interrupt request may require the central processor to initiate a polling signal to all attached devices, in series, to subsequently permit information to be transferred to the central processor identifying the device and status causing the interrupt.

In systems utilizing direct program control for each data transfer between a peripheral device and main storage, interfaces are usually provided which require, in response to the program instruction, the sequential transfer of device addesses, commands, and/or data to the peripheral device.

In data processing systems which not only provide for direct program control data transfers, but also cycle steal data transfers, different forms of program instructions are normally required. Even if different forms of initiating instructions are not required, different forms of peripheral device control information are required, which must be recognized by the peripheral device control unit and handled differently. Therefore, each

peripheral device control unit must have specialized logic. Further, if the input/output control system is also required to handle asynchronous requests for interrupt servicing by the processor, additional circuitry in the peripheral device control unit must be provided.

During cycle steal data transfer operations when a peripheral device control unit has been provided with sufficient information to initiate and control further use of the interface bus for controlling the storage unit independent of the processor, certain exceptional conditions may occur prior to completion of the data transfer requiring special handling by the central processor before re-initiating the data transfer.

Normally, input/output control systems which are adapted to handle direct program control, cycle steal, or interrupt request transfers on a common interface, must handle each of these situations exclusively on the interface bus, preventing any other form of request from being processed.

In prior systems, polling logic has been provided for responding to an unknown interrupt request which signals the priority of the interrupt request. I/O control logic responds with a serial poll signal combined with identification from the central processor of the priority of the interrupt request being polled, to cause selection by a proper peripheral device control unit for subsequent use of the interface bus. The priority interrupt request made by a peripheral device control unit can be modified by a central processor. However, in these prior systems, the modification of the priority level of a peripheral device control unit could only be accomplished when the device associated with the peripheral device control unit is not busy with some previous command. Further, prior art systems which combine cycle steal data transfers with interrupt request handling have been required to provide separate logic within a peripheral device control unit and input/output control logic of the central processor to poll for the two forms of communication required.

Prior systems which incorporate a serial poll signal for the purpose of selecting one of a plurality of peripheral device control units, all of which are requesting service, require use of logic within each peripheral device control unit to propagate the serial poll signal to succeeding devices. In these prior systems it is readily evident that if a particular peripheral device control unit, or device, were physically removed from the input/output bus, proper functioning of the poll propagation would not be possible.

Further complications in a conventional data processing system result when more peripheral devices are needed to be attached to the I/O interface than can be handled due to insufficient driving power from a channel. If an extension is made to an interface bus that includes bidirectional signal lines through use of a special attachment, the drivers on the special attachment must normally be provided with additional control information to indicate the direction of signals on the bus. If an extension to the interface bus includes its own power supply, fluctuations may cause erroneous signals on the original bus. Further, a special attachment would normally be required if it were desired to repower signals from the base system to a single remote peripheral device.

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

Figure 1 is a block diagram showing the major components of a data processing system which forms one embodiment of the present invention.

Figure 2 is a representation of the physical configuration of the data processing system.

Figure 3 identifies the lines of an input/output (I/O) interface (I/F) bus interconnecting I/O control (Channel) logic in the central processing unit (CPU) and a peripheral device control unit;

Figure 4 is a block diagram showing the major components of the I/O control logic of the data processing system.

Figure 5 shows certain registers and data busses of the central processing unit.

Figure 6 shows certain registers and busses of a central processing unit for handling address information.

Figure 7 is a representation of a data processing system program instruction and immediate device control block (IDCB) for initiating I/O operations.

Figure 8 is a representation of information in an immediate device control block transferred to a peripheral device control unit, and the timing involved.

Figure 9 represents the interaction and contents of, an operate I/O instruction (I/O), immediate data control block (IDCB), data control block (DCB), and data transferred.

Figure 10 is a representation of the contents of a data control block and control word within a data control block stored in main storage of a data processing system utilized for controlling I/O operations.

Figure 11 is a representation of the I/O interface bus lines and timing involved in transfer of data on a cycle stealing (C/S) basis between a data processing system storage unit and peripheral device control unit.

Figure 12 is a representation of the I/O interface bus lines and timing for polling of

peripheral device control units to initiate further communication on the interface bus.

Figure 13 is a general representation of the concept of a serial poll signal being propagated serially from peripheral device control unit to peripheral device control unit selecting a device to use the interface bus.

Figure 14 shows major components for receiving poll signals from a preceeding peripheral device control unit, capture of the interface, and signalling thereof back to the I/O control logic of the data processing system.

Figure 15 is a representation of the major components of a peripheral device control unit attached to an I/O interface bus.

Figure 16 is a block diagram of the major components of channel interface logic connecting the interface bus to the peripheral device control unit.

Figure 17 shows the major components of a micro processor utilized as part of a peripheral device control unit.

Figure 18 is a block diagram representation of the interconnection and I/O interface within the device control logic of a peripheral device control unit.

Figure 19 is a general block diagram of the connector circuitry of a base I/O interface attachment card.

Figure 20 is a detailed logic diagram of gate control logic of Figure 19 which responds to normal control signals of an input/output interface bus for controlling proper energization of a pair of drivers to provide proper signal transfer in either of two directions on a bidirectional bus.

Figure 1 illustrates in no great detail whatsoever a basis for discussing a data processing system that includes a central processing unit (CPU) 30, a main storage unit 31 for storage of data, machine instructions, and input/output (I/O) control information, and I/O control logic (Channel) (32). The area of interest of the invention lies in the control of the transfer of data and control information to I/O devices 33 through peripheral device control units or I/O attachments 34 utilizing an I/O interface (I/F) bus 35 which connects the various units in parallel for the transfer of data, address information and control information. Also shown is a poll signal on a line 36 which interconnects the peripheral device control units 34 in series for the purpose of selecting a particular I/O device 33 for attachment to the I/O interface 35 during a particular transfer cycle.

A physical representation of the data processing system utilizing the present invention is shown in Figure 2. The physical arrangement includes a power supply 37, a rack, or card file 38, and a plurality of pluggable cards 39 containing the circuits making up the various units of the data processing system.

Three cards 40, 41, and 42 include circuits which comprise the processor 30. Various parts of the I/O control logic 32 are distributed on the processor cards. A number of storage cards 43, depending on the amount of storage desired, are plugged into the card file 38.

The I/O attachment 34 as shown in Figure 1 is represented by each of a selected plurality of cards 44. If it is desired to attach additional I/O devices to the system, a repower and isolation card 45 may be included. The repower card 45 has the function of repowering the I/O interface lines 35 to an additional rack (which has its own power supply) and isolating the components shown in Figure 2 if any additional racks should lose power and thereby normally render the I/O interface 35 ineffective.

Card 42 is a read ony store (ROS) card containing a microprogram control mechanism for the data processing system. The address (ADDR) card 41 contains all the program accessible hardware such as data and status registers, and forms addresses used to access the storage unit 31 and I/O devices 33. The DATA card 40 performs all arithmetic and logical operations and provides the gating for data to and from the I/O interface 35 and the storage unit 31.

Interface Lines

In Figure 3, there is depicted the I/O control logic for channel 32 distributed between the address card 41, data card 40, and read only store card 42. Also depicted is an I/O attachment card 44 of Figure 2 for a peripheral device 33. The interface bus 35 in accordance with the present invention may accommodate any number of different devices 33. However, in accordance with a preferred embodiment of the present invention, each I/O attachment card 44, which represents a peripheral device control unit, will have common circuitry divided between channel logic 46 and a microprocessor 47. Added to the common circuitry is device logic 48, which is dependent upon the particular device 33 to be controlled.

Subsequent discussion of the operation of a peripheral device control unit 34, in accordance with a preferred embodiment of the present invention, will discuss operation of a micro processor 47. However, the common circuitry 46 and 47 could be comprised of only combinatorial and sequential logic.

There are three basic forms of communication between an I/O device 33 and the I/O control logic 32 which, depending on the type of device 33 may require use of up to 81 lines on the I/O interface 35. Two forms of communication are initiated by a program instruction identified as Operate I/O (OIO). These two forms of communication are primarily for exchange of data, and are identified as a direct program control (DCP) transfer or cycle steal (CS) transfer. In the DPC form of transfer, each OIO instruction effects the transfer of one item of information between the storage unit 31 and I/O device 33 in either direction. The cycle steal form of transfer is initiated by the processor 30, and involves the transfer of I/O command information to the peripheral device control unit 34 for subsequent use by the peripheral device control unit 34 to control the transfer of a plurality of items of data between the storage unit 31 and peripheral device 33. This transfer is independent of, and concurrent with, other processor 30 operations. The third form of communication required between the processor 30 and device 33, is the initiation of program interrupt sequences in the processor 30 in response to requests for processor service by a peripheral device 33. The interaction of the I/O control logic 32, interface bus 35, and peripheral device control unit 34 to accomplish these forms of communication will now be discussed in detail.

Each of the 81 lines of the I/O interface 35 will now be briefly defined utilizing the representation in Figure 3. There are two bidirectional busses essential to the operation, and these include a 17-bit, bidirectional, address bus 49, and a 16-bit plus two parity bit I/O data bus 50.

The control communication on the I/O bus 35 in response to the decode of an OIO instruction, for the purpose of transferring data or I/O control information on the data bus 50, requires use of the address bus 49. Other interface lines required to control the transfer are address gate 51, address gate return 52, condition code in bus 53, and data strobe line 54, which will be energized in proper sequence to control the communication.

During cycle steal (CS) communications, transfer of data on the data bus 50 and storage unit 31 address information on the address bus 49 is effected from the device control unit 34. Additional lines on the interface bus 35 required for this type of transfer include a service gate signal 55, service gate return 56, input/output indicator 57, word/byte indicator 58, and a four-bit status bus 59 distributed between the read only store card 42 and address card 41. If the central processor 30 and storage unit 31 have a storage protect mechanism installed, the condition code bus 53 is utilized during CS operations to transfer a storage protect key from the device control unit 34 to the storage protect mechanism.

Normally, transfer of cycle steal information between the I/O control logic 32 and device control unit 34 involves a single transfer followed by selection of another device for further operations. An additional type of transfer may be implemented, and would be identified by a signal on a line labeled burst return 60. The burst return signal 60 energizes controls in both the device control unit 34 and I/O control logic 32 to permit, in response to one selection of the device 33, a plurality of transfers of cycle steal information on the I/O bus 35 before selecting another device.

A third basic form of communication involves the requirement to signal the I/O control logic 32 that a particular device 33 desires to interrupt the central processor 30. The interface 35 lines primarily involved are a request in bus 61 and a poll identifier bus 62. To explain more fully, a device control unit 34 will have been assigned a particular priority interrupt level by means of a prepare command. The interrupt level assigned may be one of four different levels, although the concept of the invention in this regard can accommodate up to 16 different levels. When a device 33 requires interrupt service, a portion of the channel interface logic 46 of the peripheral device control unit 34 will energize a particular one of the four lines on bus 61 indicating an interrupt request. The particular one of the lines on bus 61 energized is associated with the priority interrupt level assigned. One additional line in the bus 61 is identified as bit 16, and is energized to inform the I/O control logic 32 of a need for transfer by a device 33 utilizing the cycle steal form of transfer.

When a particular device 33 has signalled either a priority interrupt request, or cycle steal request, on bus 61, interrupt control logic in the I/O control logic 32 and processor 30 determines which one of a plurality of priority levels or cycle steal requests can be acknowledged to establish connection between the I/O control logic 32 and device 33. The poll identifier bus 62 is coded with binary information to indicate which interrupt priority level is being acknowledged, or will signal a particular binary code on the poll identifier bus 62 indicating that any cycle steal request is being acknowledged.

As part of the selection of a device 33 to be permitted connection to the I/O bus 35 in response to an interrupt request or cycle steal request, the I/O control logic 32 generates a poll signal 63 and a poll prime signal 64. The poll signals 63 and 64 are propagated serially through all device control units 34 attached to the interface bus 35. As part of the device selection for use of the bus 35, the poll and poll prime signals 63 and 64 interact with the coded information on the poll identifier bus 62 to cause a particular device control unit 34 to

be selected. When a peripheral device control unit 34 recognizes a poll identification on bus 62 corresponding to its present priority interrupt level, or that it requires cycle steal transfer and recognizes the particular identification code, and also receives the poll and poll prime signals 63 and 64, the selection is made, and this fact is returned to the I/O control logic 32 on a signal line poll return 65. Receipt by a device control unit 34 of the poll and poll prime signals 63 and 64, and in the absence of recognizing a proper code on the poll identification bus 62, will cause the device control unit 34 to propagate the poll and poll prime signals 63 and 64 to succeeding peripheral device control units 34.

Additional signal lines on the interface bus 35, not previously discussed and which do not form part of the present invention, are included in the preferred embodiment of the I/O bus 35. These lines include a halt or machine check signal line 65 for halting a device previously started, two lines 66 utilized for control and transfer during an initial program load from a device 33 to storage unit 31, power on reset line 67 to cause all logic in the device control units 34 to be reset to a known state, and system reset line 68 for establishing known conditions in response to processor controls.

Throughout the remainder of the description and on the remaning drawings, signal lines and busses will be identified as shown in Figure 3. Any reference to a particular binary bit on a larger bus will be identified by the bus number, hyphen, and bit number. For example, the line labeled 16 on bus 61 will be identified as 61-16.

CPU - I/O Control Logic General Description

Major functional components of the I/O control logic 32 shown in Figure 1 are shown in Figure 4. A preferred embodiment of the present invention may be used with a central processor 30 which has has a logical mechanism for indicating the level of importance of a particular program being executed in the processor 30. Any requests for execution of a program of more or less importance than the current level will determine the response of the processor 30 to such request. As part of the I/O control logic 32, there is interrupt logic 69 for comparing the importance of an interrupt request from I/O devices as signalled on bus 61 with the level of importance of the present processor 30 program indicated in a current level register 70. As in many other data processing systems, the ability of any particular interrupt to be effective can be modified by the use of an interrupt mask contained in a register 71. The contents of the current level register 70 and interrupt mask 71 can be modified by data on the processor data bus 72 in accordance with programmed instructions. In accordance with the settings of the current level register 70, mask 71, and level of the interrupt request in on bus 61, the read only store control of the processor 30 can be notified on a line 73 of the requirement to control the processor 30 to discontinue operation at the current level and initiate an interrupt.

After necessary housekeeping functions in the processor 30, the read only store control mechanism will return a signal on line 74 indicating that an interrupt request or cycle steal request indicated on bus 61-16 can be acknowledged.

At this point in time, the processor 30 and therefore programs stored in the storage unit 31 do not know the identity of the particular device that made the request that is being acknowledged. Therefore, the I/O control logic 32 further includes a poll sequence control mechanism 75 which initiates a poll signal on line 63 along with coded information on the poll identifier bus 62 indicating whether a cycle steal request is being honored, or identifies a particular priority interrupt level being acknowledged. In response to a signal on poll return 65 indicating that a device 33 has captured the poll signal 63, the poll sequence control 75 initiates the necessary exchange of signals between the I/O control logic 32 and peripheral device control unit 34.

The control of the transfer of signals, and response thereto, in the I/O control logic 32 is accomplished in logic called interface gate control 76. If, as previously discussed, a poll sequence was initiated for interrupt or cycle steal reasons, the primary signal lines energized and responded to in the interface gate control 76 are the service gate 55, service gate return 56, and data strobe 54. If the selection has been made for cycle steal transfers, various cycle steal status information on bus 59 is transferred to the peripheral device control unit 34 indicating various conditions of the cycle steal operation.

If the interface gate control 76 is to initiate and control information transfer a signal on line 77 will be received from the instruction register of the processor 30 indicating the decode of an Operate I/O instruction. Response to the signal on line 77 requires energization of and response to the signal lines address gate 51, address gate return 52, and data strobe 54. Further, response to each OIO instruction by the addressed peripheral device control unit 34 is signalled by information on the condition code in bus 53 which is entered into latches 78 for presentation to level status registers in the processor 30 on the three lines 79. If cycle steal transfers are taking place, the storage protect key will be sent to the storage protect mechanism on lines 80.

Interface check control logic 81 responds to and generates various signals indicating the correctness of the operation of the I/O control logic sequence on a line 82, other I/O and device related errors on I/O check line 83, and responds to a signal on a line 84 indicating that a parity error was detected during a cycle steal data transfer. The designation PSW refers to the processor status word in the processor 30. The PSW can be sensed by program control to monitor and indicate various errors and exceptions within the data processing system.

Control of the timing between the I/O control logic 32 and the storage unit 31 is accomplished generally on lines 85. Completion of an I/O sequence is signalled to the processor 30 on a line 86 and control of gates within the processor labeled A, B, and C, required to accomplish data transfer is signalled on three lines 87. The decode of a halt I/O instruction by the processor 30 is signalled to the interface gate control 76 on a line 88 and any requirement to reset the I/O control mechanism is signalled on a line 89 from the processor 30. During cycle steal operations, any parity error detected on the interface in the transfer of data into storage unit 31 is signalled on a line 90. Various other lines to and from the processor 30 have been named in Figure 4 and are essentially self explanatory and not required for an understanding of the operation of the present invention.

In Figures 5 and 6 various registers and busses contained in a processor 30 for accomplishing I/O operations are shown. All the busses and registers shown are comprised of binary bits. The processor bus 72 has a number of other units attached such as the arithmetic and logic unit, local storage, and additional registers primarily concerned with data processing functions.

Data from storage unit 31 is received on a bus 91, and entered into storage unit 31 on a bus 92. Data received from the storage unit 31, when to be used primarily within the processor 30 is received in a CPU storage data register (CPU SDR) 93, and when data is being transferred between peripheral devices 33 and the storage unit 31 during cycle steal operations, data will be entered into a cycle steal storage data register (CS SDR) 94.

Also shown in Figure 5, is an operation register 95 which receives program instructions from storage unit 31 and bus 91 and CPU SDR 93 to be docoded for control of operations of the system. Of particular concern to the present invention is the decoding of an instruction called Operate I/O (OIO).

When the OIO instruction is to effect a direct program control transfer of data from the storage unit 31 to a peripheral device 33, the data will be received from the storage unit 31 on bus 91, entered into the CPU SDR 93, transferred to the processor bus 72 on a further bus 96, entered into one of the CPU registers 97, and presented to the I/O data bus 50 on a bus 98 in response to energization of IF gate A 99 in response to controls from the I/O control logic 32. Direct program control of data transfer from an I/O device 33 to the storage unit 31 would be accomplished by presenting data on the I/O data bus 50 to the processor bus 72 by energization of gates represented at 100, entering the data into the CPU SDR 93 from a bus 101, and transferring the data to the storage unit 31 on bus 92.

Transfer of data during cycle steal operations from the I/O device 33 to the storage unit 31 will involve transfer of data from the I/O data bus 50 into the CS SDR 94 on a bus 102, by energization of IF gate B 103, followed by transfer of the data from the CS SDR 94 to the storage unit 31 on bus 92.

Cycle steal output transfers would involve the transfer of data from the storage unit 31 on the bus 91 into the CS SDR 94, followed by energization of IF gate C 104 to present data on a bus 105 to the I/O data bus 50.

The generation of parity bits 106 to be included with data from the I/O data bus 50, or the signalling of parity errors on line 84, is accomplished in the interface parity check/generator 107 during I/O operations.

Figure 6 shows the busses and processor 30 registers required for the transfer of address information between I/O devices 33 and the storage unit 31. Addresses are presented to the storage unit 31 on a bus 108 from either a CPU storage address register (CPU SAR) 109 or, during cycle steal transfers from a cycle steal storage address register (CSSAR) 110. As part of the present invention, selection of a particular I/O device 33, and transmission of commands to the device 33, is accomplished utilizing the I/O address bus 49. This information is presented to the I/O address bus 49 from a further CPU register 111 which receives the information from the processor bus 72.

OIO-IDCB-DCB Formats and Timing

Figure 7 shows the two word (32-bit) Operate I/O (OIO) instruction decoded in the operation register 95 of Figure 5 which initiates all I/O operations from the processor 30. It is a privileged instruction and may only be fetched in supervisor state. If this instruction is fetched in problem state, a privilege violate program check is set, and a class interrupt is taken.

The effective address, generated by this instruction, points to, and addresses, an

Immediate Device Control Block (IDCB) in storage unit 31. The IDCB contains a command field (bits 0-7), device address field (bits 8-15), and the immediate data field (bits 16-31).

In the command field, the first hex digit (bits 0-3) identifies the type of command, and the second hex digit (bits 4-7) is a modifier. The types of command are Read, Read ID, Read Status, Write, Prepare, Control, Device Reset, Start, Start Cycle Steal Status, and Halt I/O.

The device address field contains the device 33 address. Device 33 addresses are selectable by switches or jumpers on each I/O attachment card 34.

For direct program control (DPC) operations, the immediate filed of the IDCB in storage unit 31 contains the word to be transferred from the storage unit 31 to the I/O device 33, or the word from the device 33 to be stored in storage unit 31. For cycle steal operations, the immediate field contains the address in storage unit 31 of a device control block (DCB).

The Read command transfers a word or byte from the addressed device 33 to immediate field word of the IDCB. If a single byte is transferred, it is placed in bits 24-31 of the data word.

The Read ID command transfers an identification word from the device 33 to the immediate field of the IDCB. The device identification word contains physical information about the device and is used by diagnostic programming to tabulate a system configuration. This word is not related to the interrupt ID word associated with interruption processing.

The Read Status command transfers a device status word from the device 33 to the immediate field of the IDCB. Contents of the status word are device dependent.

The Write command transfers a word or byte of data to the addressed device 33 from the immediate field of the IDCB. If a single byte is to be transferred, it is placed in bits 24-31 of the data word and bits 16-23 are ignored.

The Prepare command transfers a word to the addressed device 33 that controls its interruption level. The word is transferred from the second word of the IDCB in which bits 16-26 are zeros, bits 27-30 are a level field, and bit 31 is an I-bit. A priority interruption level is assigned to the device 33 by the level field. The I-bit (device mask) controls the device interruption capability. If the I-bit equals 1, the device is allowed to interrupt.

The Control command initiates a control action in the addressed device 33. A word, or byte, transfer from the immediate field of the IDCB to the addressed device may or may not occur, depending on device requirements.

The Device Reset command resets the addressed device 33. A pending interruption from this device is cleared. The device mask (I-bit) is not changed.

The start command initiates a cycle steal operation for the addressed device 33. The second word, or immediate field, of the IDCB is transferred to the peripheral device control unit 34. It contains a 16-bit storage unit 31 address of a device control block (DCB) to be used by the peripheral device control unit 34 to control further operations.

The Start Cycle Steal Status command initiates a cycle steal operation for the addressed device 33.

Its purpose is collecting status information relative to the previous cycle steal operation. The immediate field of the IDCB is transferred to the peripheral device control unit 34, and contains a 16-bit address of a DCB.

The Halt I/O command is an I/O control logic 32 directed command that causes a halt of all I/O activity on the I/O interface 35. No data is associated with this command. All pending device interruptions are cleared. Device priority interruption level assignments and device masks (I-bits) are unchanged.

Figure 8 depicts the contents of register 97 of Figure 5 and register 111 of Figure 6 and the timing of signals on various lines of the interface 35.

This represents the initial action when an OIO program instruction is decoded, whether for DPC Read or Write, transfer of the DCB address for cycle steal operations, or transfer of interrupt level codes for a Prepare command. The Data Bus 50 will be energized with the data being transferred between the device 33 and the immediate field of the IDCB in storage unit 31 that was addressed by the Operate I/O instruction.

Address bus 49, bits 0-15 contain the first word of the IDCB. The Address bus 49 is active prior to the rise of Address Gate 51 and until the fall of Address Gate Return 52. Equality between the wired device address and bits 8-15 of the Address bus 49, with bit 16 binary 1, constitutes initial selection of a peripheral device control unit 34. Bit 16 is added to the Address bus 49 by the Interface Gate control 76 of Figure 4 from a decoder 112 to distinguish use of the address bus 49 for I/O operations as opposed to other operations using the address bus 49.

Address Gate 51 is the outbound tag used to signal the device 33 to respond to initial selection and begin the operation specified by the command (bits 0-7 Address Bus).

Address Gate Return 52 is the tag raised by the peripheral device control unit 34 to signal

the I/O control logic 32 that is has received Address Gate 51, has recognized its address, and has activated status information on the Condition Code In Bus 53. This tag must rise within a certain time limit of the rise of Address Gate 51 as seen at the output of the channel. If not, condition code 0 is returned to the I/O control logic 32 and the sequence is terminated. Address Gate 51 falls and the Address Bus 49 is cleared.

The Condition Code In bus 53 is a three bit field that is binary encoded. The I/O device 33 passes status to the channel on this bus during the Address Gate Return tag time. The condition code bits are placed into the current level status register (LSR) of the CPU 30. The condition code values and meaning are shown in Table I.

TABLE I

CC Value	Meaning
0	Device not attached
1	Busy
2	Busy after reset
3	Command reject
4	Intervention required
5	Interface data check
6	Controller busy
7	Satisfactory

Data Strobe 54 is an outbound signal generated by the I/O control logic 32, and may be used by the device to register data being sent to the device. Data Strobe 54 falls with the fall of Address Gate 51.

Figures 9, 10 and 11 will be utilized to describe further details of cycle steal input/output operations. In Figure 9, the decoding of an OIO instruction with storage unit address 200 will cause the processor 30 to address and access from location 200 in the storage unit 31, the two words of the IDCB 113. The IDCB will be transferred to the peripheral device control unit 34 selected by the device address portion of the IDCB in accordance with the sequence shown in Figure 8. The immediate field of the IDCB identifies and provides the address of the location of a device control block (DCB) 114 in the storage unit 31. The command Start Cycle Steal or Start Cycle Steal Status will be decoded in the peripheral device control unit 34 and initiate a first cycle steal operation using the address information 500 to the storage unit 31 for the purpose of transferring the DCB 114 to the peripheral device control unit 34.

The DCB contents will identify the address in storage unit 31 involved with the data transfer, and as depicted in Figure 9, it is shown to be located at address 800 in the storage unit 31, thereby defining a data area 115. The amount of data to be transferred is specified by a byte count field. At the completion of the transfer controlled by DCB 114, an additional DCB, identified as a chained DCB 116, may be transferred to the peripheral device control unit 34 to provide further control for the previously selected peripheral device 33. As shown in Figure 9, DCB 114 contains control information providing the address in storage unit 31 of chained DCB 116 which is located in the storage unit 31 beginning at address 600.

During Cycle Steal operations, each of the eight words which comprise a DCB is transferred to the previously selected peripheral device control unit 34 on a cycle steal request basis. Figure 10 depicts the contents of a DCB contained either in the storage unit 31 or as received by a peripheral device control unit 34 in response to utilization of the IDCB information, which in turn was transferred in response to the OIO instruction. The DCB is an eight word control block residing in the supervisor area of storage unit 31. It describes the specific parameters of the cycle stealing operation. The peripheral device control unit 34 fetches the DCB using storage protect key zero. The following describes the contents of the Control Word of each DCB.

If bit 0 is equal to one, a DCB chaining operation is indicated. After satisfactorily completing the current DCB operation, the device does not interrupt (excluding PCI interruptions). Instead, the device fetches the next DCB in the chain.

If bit 1 is equal to one, the device presents a programmed controlled interruption (PCI) at the completion of the DCB fetch. A pending PCI does not inhibit data transfers associated with the DCB. If the PCI is pending when the device encounters the next interrupt causing condition, the PCI condition is discarded by the device and replaced with the new interrupt condition.

The setting of bit 2 tells the device the direction of a data transfer; 0 = Output (main storage to device) and 1 = Input (device to main storage). For bidirectional data transfers

under one DCB operation, this bit must be set to one. For control operations involving no data transfer, this bit must be set to zero.

If bit 3 is equal to one, the transfer of data takes place in burst mode. This mode dedicates the channel and I/O interface to the device until the last data transfer associated with this DCB is completed.

If bit 4 is equal to one, an incorrect length record is not reported. The device continues the operation. The classes of incorrect length record are: (1) a record that is longer than the specified count. and (2) a record that is shorter than the specified count. Reporting of incorrect length record may be suppressed for one or both classes depending on the individual device.

Bits 5-7 are the cycle steal address key. This key is presented by the device during data transfers. It is used to ascertain storage access authorization.

Bits 8-15 may be used to describe functions unique to a particular device.

Parameter words 1-3 are device-dependent control words and are implemented as required. If suppress incorrect length (SIL) is used by a device, parameter word 4 specifies a 16-bit storage unit address called the status address. This address points to a residual status block that is stored when the following two conditions are met: (1) The SIL bit (bit 4 of the DCB control word) is set to one, and (2) All data transfers for the current DCB have been completed without error.

The size of the residual status block varies from one to three words depending on the individual device. The first word contains the residual byte count. Additional words (maximum of two) contain device dependent status information. If suppress incorrect length is not used by a device, meaning of the device parameter word 4 is device dependent and has the same meaning as parameter words 1-3.

If the DCB chaining bit (bit 0 of the control word) is equal to one, Parameter Word 5 specifies a 16-bit main storage address of the next DCB in the chain. If chaining is not indicated, this parameter word is device dependent.

The count word contains a 16-bit unsigned integer representing the number of data bytes to be transferred for the current DCB. Count is specified in bytes with a range of 0 through 65,535. It must also be even for the start cycle steal status operation.

The Data Address word contains the starting main storage address for the data transfer.

The cycle steal mechanism allows data service to or from an I/O device 33 while the CPU 30 is doing other processing. This overlapped operation allows multiple data transfers to be initiated by one operate I/O instruction. The CPU executes the Operate I/O instruction; then continues processing the instruction stream while the I/O device steals main storage 31 data cycles when needed. The operation always terminates with a priority interruption from the device. A poll tag 63 is generated by the channel to resolve contention between multiple devices requesting cycle steal transfers. The poll tag also resolves contention for priority interruptions on the same level.

All cycle steal operations include certain capabilities that are provided on a device feature basis:

1. Burst mode
2. Command chaining
3. Data chaining
4. Programmed controlled interruption (PCI)
5. Storage addresses and data transfers byte or word

All cycle steal operations terminate with a priority interruption.

The purpose of the Cycle Steal Start command is for data transfer. The purpose of the Start Cycle Steal Status comman is to obtain residual parameters from the device if the previous cycle steal operation terminates due to an error or exception condition. The DCB format is the same as that for a normal cycle steal operation, with words 1-5 set to zeros.

During start cycle steal status operations, data is transferred to main storage 31 starting at the data address specified in the DCB. This data consists of residual parameters and device dependent status information. The first word transferred contains the main storage address of the last attempted cycle steal transfer associated with a Start command. If an error occurs during a start cycle steal status operation, this address is not altered. The residual address may be a data address, a DCB address, or a residual-status-block address and is cleared only by a power-on reset. It is updated to the current cycle-steal storage address upon execution of cycle steal transfers. For word transfers, the residual address points to the high-order byte of the word. Device reset, Halt I/O machine check, and system reset have no effect on the residual address in the device.

The second status word transferred contains the residual byte count of a device. The residual byte count is initialized by the count field of a DCB associated with a Start comand, and is updated as each byte of data is successfully transferred via a cycle steal operation. It is not updated by cycle-steal transfers into the residual status block. The residual byte count

is not altered if an error occurs during a start cycle steal status operation. It is reset by (1) power-on reset, (2) system reset, (3) device reset, (4) Halt I/O, and (5) machine check condition. The contents of the device cycle-steal-status word 1 are device dependent if the device does not: (1) implement suppress incorrect length (SIL), or (2) store a residual byte count as part of its cycle-steal status.

Other device dependent status words can be transferred depending on the device type. Two conditions can cause bits to be set in the device dependent status words.

1. Execution of an I/O command that causes an exception interruption.
2. Asychronous conditions in the device that indicate an error or exception.

The bits are reset as follows:

1. For the first condition listed above, the bits are reset by the acceptance of the next I/O command (except Start Cycle Steal Status) following the exception interruption. These bits are also reset by a power-on reset, system reset, or execution of a Halt I/O command.
2. For the second condition, the bits are reset on a device dependent basis.

Figure 11 depicts the Interface 35 lines used and timing during cycle steal operations. Prior to this operation, the device had sent a cycle steal request (Bit 16 on the Request in bus 61), to the I/O control log in 32 which responded with the Polling sequence, and this device captured that poll.

Service Gate 55 is raised by the I/O control logic 32 to indicate to the device 33 that captured the poll 64, and signalled Poll return 65, that data transfers may begin.

When the device detects Service Gate 55 it sends Service Gate Return 56 to the channel 32 to indicate that it has placed the necessary data and control information on the I/O interface 35. Any data provided by the device for the transfer, is activated no later than the raise of this tag line. This tag line may fall no sooner than the fall of Service Gate 55 and Data Strobe 54 as seen at the output of the I/O device.

The Address Bus 49 contains the storage unit 31 address that is used for the data word to be transferred. The contents of the Address bus are gated to the Cycle Steal SAR 110 in Address card 41. A storage cycle takes place and the word is placed into the Cycle Steal SDR 94. The Data Bus 50 contains the word being transferred.

The Condition Code In bus 53 contains the Address key to be used during the storage unit 31 access. Condition Code In bits 0, 1, 2 correspond to bits 0, 1, 2 of the address key. This bus is activated with the rise of Service Gate Return 56 and is maintained till the fall of Service Gate 55.

Data Strobe 54 is an outbound tag and may be used by the device to register data being sent to the device. Data Strobe 54 falls with the fall of Service Gate 55.

The Status Bus 59 is used by the I/O control logic 32 to signal the peripheral device control unit 34 in the event an error is detected during cycle steal operations. The bits of this bus have the following meaning:

Bit 0 Storage Data Check
Bit 1 Invalid Storage Address
Bit 2 Protect Check
Bit 3 Interface Data Check

If this bus is activated, the device retains the information for presentation in an Interrupt Status Byte at interruption time. The cycle steal operation is terminated and the device presents an end interruption.

If the device had already raised Cycle Steal Request for the next transfer, or is in Burst Transfer mode, it must complete one more servicing over the interface. This servicing is a dummy cycle where no device held parameters are updated or any status bits accumulated.

The Input/Output Indicator 57 tag = 0 indicates to the I/O control logic 32 that the operation is an output from storage 31, and = 1 indicates an input to storage 31.

The Word/Byte Indicator 58 tag = 0 indicates to the I/O control logic 32 that a word transfer is to take place, and = 1 indicates a byte transfer.

Polling

Figures 12 through 14 generally depict the polling concept. The polling logic is common to the selection of peripheral device control units 34 in response to either interrupt requests or cycle steal requests. In accordance with the sequence shown of Figure 12, the Request In Bus 61 is energized by any peripheral device 33 on the Interface Bus 35 which requires interrupt handling or use of the bus 35 for cycle steal data transfers. The line labeled "Bit 16" of the Request In Bus 61 is energized whenever any device requires cycle steal data transfers. The remaining lines of the request in bus 61 are each associated with a particular interrupt level. The energization of lines on the Request In Bus 61 will remain at a steady state value as long as any device requires servicing for interrupt or cycle steal transfers.

At the time the I/O control logic 32 determines that any of the requests on the Request In

Bus 61 should be acknowledged, the signal lines of the poll identifier bus 62 will be energized in a coded fashion to indicate to all devices that a poll and selection process is being intitiated either for cycle steal transfers or interrupt handling at a particular interrupt level identified by the poll identifier bus 62. After the poll identifier bus 62 is energized, a poll signal 63 is serially generated to all peripheral device control units 34 on the bus 35 to resolve contention between peripheral device control units 34 requesting interrupts on the same priority level and cycle stealing requests. Each peripheral device control unit 34 receives the poll tag 63 and redrives, or propagates, it to the next peripheral device control unit 34 by sending poll propagate if the device does not capture the poll. If a particular device control unit 34 has requested service of the type identified by the poll identifier bus 62, it responds with the poll return signal on line 65, and the poll signal 63 is not propagated to any further peripheral device control unit 34.

In Figure 13, there is shown a representation of three peripheral device control units 34 all having interrupts pending. The first device is indicating a request to interrupt at level 2 while the remaining devices are requesting interrupts at level 1. The poll identifier bus 62 will be coded to specify a poll for any device making a request at level 1. Since the poll identifier bus 62 is not equal to the request at level 2 by the first device, the poll signal 63 will be propagated to the next following device. The first device signifying a request at level 1 captures the poll, and degates the poll propagate signal to the next following devices. At the same time, the first level 1 device will generate the poll return signal 65 for informing the I/O control logic 32 that the poll has been captured. The I/O control logic 32 will respond with service gate 55, the device responds to the service gate 55 with service gate return 56 and starts utilization of the interface bus 35.

As shown in Figure 13, the polling signal which is serially propagated from device to device is in fact two separate signals labeled poll 63 and poll prime 64. The internal logic for each of the peripheral device control units 34 generates an internal poll signal in response to receipt of a signal on both the poll input line 63 and poll prime input line 64. This feature permits the proper functioning of the polling mechanism even though a particular peripheral device control unit 34 is physically removed from the interface bus 35. In the extreme all alternate peripheral device control units 34 may be physically removed.

Figure 14 shows additional detail of the internal logic of a peripheral device control unit 34 utilized in receiving poll signals 63 and 64, and generating the poll return signal 65. An AND circuit 117 receives both the poll signal 63 and poll prime signal 64, received on first and second inputs respectively. The output of AND circuit 117 on line 118 is the internal poll signal. An AND circuit 119 and a compare circuit 120 determined from the present device interrupt level or cycle steal request indication on a line 121, and the coded information on the poll identifiers bus 62 whether or not the particular device shown should capture the poll and generate a poll return signal 65 from AND circuit 122.

The output of either Compare Circuit 120 or AND circuit 119 will be effective at the AND circuits 122 along with an internal poll signal 118 to generate the poll return signal 65 and inhibit operation of the poll propagate logic to a next following device.

AND circuit 117 is shown to have, on each of the first and second inputs, a resistor 123 connected to a positive voltage. The poll line 63 and poll prime line 64 will normally be held at negative, ineffective levels, in the absence of the generation of the respective signals. If the next preceeding peripheral device control unit 34 were removed from the interface bus 35, the resistor 123 to the positive voltage supply will clamp the first input of AND circuit 117 to a positive level indicating a normal poll signal on line 63. At this time, receipt of the poll prime signal on line 64 from a peripheral device control unit 34 next preceeding the peripheral device control unit 34 removed from the inerface bus, will combine with the clamped first input of AND circuit 117, and be effective to generate the internal poll on signal line 118. If the peripheral device control unit 34 generating the poll prime signal 64 were removed from the bus 35, the second input to AND circuit 117 would be clamped, and AND circuit 117 would respond to the poll signal 63 from the next preceeding device control unit 34.

Peripheral Device Control Unit

Figure 15 depicts in slightly more detail the arrangement of the major parts of a peripheral device control unit 34 shown in Figure 3. The channel interface logic 46 is connected in parallel with other channel interface logic to the interface 35, and also receives the serially transmitted poll signal 63. In certain situations, the channel interface logic 46 could contain all of the combinatorial and sequential logic required to directly control a device 33. However, in a preferred embodiment of the present invention, basic control for the peripheral device control unit 34 is effected by a microprocessor 47 which includes its own storage 124 for programs, data, and peripheral device control information. The transfer of data, control, and sense information is effected by the microprocessor 47 data

bus out 125, data bus in 126, and address bus 127. The microprocessor 47 instruction set includes OP codes and address information wherein the address information on bus 127 identifies particular registers, triggers, latches, and gates within the peripheral device control unit 34 to be effected or sensed.

Figure 16 shows the major components of the channel interface logic 46 as connected to the interface bus 35 and microprocessor 47 busses. The major units include data register byte 0 with parity check and parity generate, and byte 1 of the data register with parity check and parity generate. Interrupt and cycle steal sequencing is controlled in logic which includes checking of priority level and poll identification. Further logic includes byte 0 of an address register, which as previously discussed carries the command for a device and therefore also includes a command decode mechanism. Other logic receives byte 1 of the address information, which as previously discussed, addresses a particular device which is compared with a wired address on address jumpers. There is logic that includes the cycle steal status register, condition code generation and other reset and interface controls. There is a decoder for the address information from the micro-processor which is effective to control and sense various latches in the peripheral device control unit 34.

Figure 17 shows a block diagram of the major components of a microprocessor 47 suitable for use in the peripheral device control unit 34. The previously mentioned storage 124, output and input data busses 125 and 126, and address bus 127 are shown. The microprocessor is controlled by the entry of 16-bit instructions into an OP register 128, the OP code portion of which is utilized by cycle control 129 and a timing clock 130 to generate necessary control signals within the microprocessor. The storage 124 is accessed by address information from a storage address register (SAR) 131 which receives address information from several sources. These sources include address information in instructions contained in the OP register 128, an instruction address register 132, a link register 133, and from instruction addressable data address register (DAR) stack 134. In combination with the instruction address register 132 and link register 133, a backup register 135 and incrementer 136 provide the necessary controls for controlling sequence of programmed instruction execution, including branch, branch and return, branch and link types of sequence controls.

A further addressable register stack 137 and data from storage 124, presented through an assembler or multiplexor 138, can be stored in an A register 139 and/or B register 140. Registers 139 and 140 provide input to arithmetic and logic unit 141, and are the registers utilized for transfer of data utilizing the Data Bus Out 125 or Data Bus In 126.

Figure 18 shows additional detail of the channel interface logic 46 discussed briefly in connection with Figure 16. As the result of a decode in the processor 30 of an OIO instruction, the I/O control logic or channel 32 must communicate with the peripheral device control units 34 to utilize the interface bus 35 to transfer the immediate data control block (IDCB). The channel interface logic 46, whether controlled by a microprocessor 47 in accordance with a preferred embodiment, or by combinatorial and sequential logic must include a number of basic elements, and these include a 16-bit data register 142, address register 143, command register 144, and attachment or device selection address compare circuit 145.

As indicated earlier, the interface address bus 49 carries the first word of the IDCB which includes the device command in bits 0 through 7 and the device address in bits 8 through 15. An initial selection of all peripheral device control units 34 is made by bit 16 of the address bus 49 to distinguish use of the bus for I/O operations as opposed to other operations. The initial device or attachment selection is made by comparing the device address in bits 8 through 15 on the address bus 49 with the wired device address 133 in the address compare circuit 145 to provide an initial device or attachment selection signal on line 146. The recognition of the device address will cause bits 1 through 7 on the address bus 49 to be gated into the command register 144 for presentation to a command decode circuit 147. The microprocessor address bus 127 is decoded in a decode circuit 148. The decoder 148 output is combined with the decoder 147 output in attachment logic 149. For data transfer operations, either a first or second output signal on lines 150 and 151 will indicate a direct program control transfer or a cycle steal transfer respectively.

The initial device or attachment select signal 146 provides one enabling signal to an AND circuit 152 which responds to the I/O control logic generated address gate 51 to generate an address gate return signal 52. In response to the address gate return signal 52, the I/O control logic 32 is informed of the device selection.

The 16-bit data register 142 is interconnected to the interface data bus 50 by busses 153 and 154. The 16-bit data register 142 communicates with the 8-bit microprocessor data bus out 125 or the data bus in 126, in two separate cycles in response to control from the microprocessor. In the case of a direct program control read operation, the data register 142 will have received the data from the microprocessor data bus out 125 for presentation to the interface data bus 50 over bus 154. If the DPC operation is to write data, the contents of the

interface data bus 50 would have been placed in the data register 142 over the bus 153 for subsequent presentation, in two separate cycles, to the microprocessor data bus in 126.

If the immediate data control block command portion has called for a Start Cycle Steal operation, the contents of the data register 142 received by bus 153 contains address information which will be transferred over the microprocessor data bus in 126, to the storage 124 of the microprocessor shown in Figure 17. Further, in response to the decoding of a Start Cycle Steal Operation, the contents of the command register 124 will be transferred by a bus 155 to the microprocessor data bus in 126 for storage in the storage 124 of the microprocessor. Therefore, the storage 124 of the microprocessor will be utilized as command storage and storage unit 31 address storage for the pupose of controlling subsequent cycle steal data transfer operations.

During subsequent cycle steal transfer operations, the address register 143 will receive on a bus 156 in two consecutive cycles, from the microprocessor storage 124, the previously stored storage unit 31 address information. This address information on subsequent cycle steal data transfer operations will be transferred over a bus 157 to the interface address bus 49 for presentation to the address mechanism of the storage unit 31 of the central processing unit. The data register 142 will contain the data of a cycle steal transfer whether for a read or write operation.

A third type of command received in the immediate data control block is a prepare command signalled on a line 158. In response to a prepare command signalled on line 158, bits 11 through 14 on the interface data bus 50 will be stored in a priority level register 159, and the previously mentioned I bit 15 will be stored in a trigger 160. The contents of the priority level register 159 signify the priority level of the device when interrupt requests must be made. An interrupt request signal on line 161, initiated by the attached microprocessor 47, will be effective at an AND circuit 162 if the I bit stored in 160 is a binary 1. This signifies that the device may interrupt at any level. If the device can interrupt, and an interrupt request has been made on line 161, a level decoder 163 will be rendered effective to energize one of the signal lines on the interrupt request in bus 61. The line energized will be associated with the priority interrupt level registered in the priority level register 159.

As previously discussed in connection with Figure 14, the I/O control logic 32 responds to any signal on the request in bus 61, whether for an interrupt request or a cycle steal request signalled on a line 61-16, by initiating a polling sequence. The polling sequence includes the transmission of the poll identification on bus 62 to all attached peripheral device control units 34 along with the serially transmitted poll signal 63. If the poll identification on bus 62 indicates a poll for any device requesting a cycle steal transfer, a signal on line 164 will be generated. This enables AND gate 119, to produce an output if the device shown in Figure 18 has requested a cycle steal transfer as indicated on line 121.

If the poll identification bus 62 is coded with a priority interrupt level which compares with the present level assigned as indicated in register 159, and the device shown in Figure 18 has requested an interrupt, as indicated by an output from AND circuit 162, a signal from an AND circuit 165 will be generated. OR circuit 166 will generate an output signal in response to an output of AND circuit 119, or the output of AND circuit 165 to degate propagation of the poll to succeeding peripheral device control units 34, which signal is shown at 167. Other logic of the peripheral device control unit 34 of Figure 18 will be signalled of the poll capture on a line 168. Further, an AND circuit 169 will be enabled to thereby cause the peripheral device control unit 34 to respond to the service gate signal 55 on the interface 35 by generating the service gate return signal 56 for the purpose of controlling further transfer on the interface bus 35.

Figures 19 and 20 show details of the connector circuitry included on an attachment card such as 45 shown in Figure 2. The processor cards 40, 41, and 42 are capable of handling a predetermined number of peripheral devices attached to I/O cards 44 which are included in the base data processing system depicted Figure 2. In certain cases, it may be required that additional peripheral devices must be attached to the base and these will normally be included in a cabinet such as that shown in Figure 2 which includes a power supply 37, a rack, or card file 38, and additional I/O device attachment cards 44. The connector circuitry on the repowering card 45 will be effective to interconnect the signal lines of the interface bus 35 of the base unit of Figure 2 to the further interface bus 35 of the expansion unit.

Figure 19 is a block diagram of the repower attachment card 45 of Figure 2. The interface connector circuitry includes bidirectional drivers 170 and 171 which interconnect the address bus 49 and data bus 50, respectively, of the base data processing system of Figure 2 to the further input/output interface which also includes a bidirectional address bus 172 and data bus 173.

Unidirectional drivers 174 respond to various outbound busses and tags from the channel

of the base data processing system on a bus depicted at 175 and repower these signals to corresponding signal lines on a further interface on a bus shown at 176. Unidirectional drivers 1175 respond to signal lines of the further interface bus 176 to repower and drive various inbound bus and tag signal lines on a bus 177 to the channel of the base data processing system.

Gate logic 178, to be more fully described in connection with Figure 20, responds to certain of the normal signal lines involving transfer of address, data and other control signals between the channel of the base data processing system and connected peripheral devices, to control the energization of a proper driver of the bidirectional drivers 170 and 171 to ensure transfer of information on these bidirectional busses is in the proper direction. As a result of this, additional information over and above that normally transmitted on the interface, is not required to determine which direction drivers 170 and 171 are to be effective.

Figure 19 shows a block 179 labeled poll logic. The attachment card directly connected to the interface of the base system for accomplishing repowering to an extension unit of I/O devices can be utilized to interconnect the base system interface to a peripheral device which may be physically many feet from the base system interface, such that repowering of the interface signal lines is required to provide proper control to the physically remote peripheral device. As part of the interface operation, as described in connection with Figures 13, 14, and 18, a serial poll, poll prime, and poll return signal must be generated to determine the particular peripheral device to utilize the interface in response to interrupt or cycle steal requests. If the poll signal, poll propagate, or poll return signal is required to traverse the path from the interface to the remote peripheral device, time will be lost in operation of the interface. Therefore the attachment card which provides repowering is also provided with the poll logic 179 which, in combination with a further set of inbound busses and tags on a bus 180, and terminals, noted generally at 181, to which the remote peripheral device may be attached, the poll capture, propagate, or poll return signalling can be achieved at the attachment card itself. This saves the propagation time of signals to and from the remote peripheral device for the purpose of poll capture as previously described.

Signals such as interrupt request 161, or cycle steal request 121, shown in Figure 18, would be received from the remote peripheral device into the poll logic 179 for the purpose of energizing a proper one of the interrupt request in bus signal lines 61. A bus 182 connected to the bus 177 provides this information. The poll logic 179 would also include the logic which would respond to an interface poll ID on lines 62 in combination with the priority level assigned to the peripheral device in register 169 to achieve the poll capture, poll propagate, or poll return signalling on the interface of the base system.

Figure 20 shows more detail of the logic of the connector circuitry on the repowering attachment card 45 of Figure 2 which responds to certain signals on the interface bus of the base system, energized during normal interface utilization, for controlling the direction of signal transfer on the bidirectional address bus 49 or data bus 50.

Only a single bidirectional signal line of the address bus 49 is shown at the bottom of Figure 20. All other signal lines of the address bus 49 and all of the signal lines of the data bus 50 would be depicted identically. Associated with each bidirectional signal line, is a first driver 183 enabled by a first AND gate 184 for enabling the driver 183 to generate signals on the further interface bus 172, in accordance with information on the address bus signal line 149. A second driver 185 and a second AND gate 186 will be effective to generate signals in the opposite direction, that is from information on the further I/O bus 172 to the channel of the base system on line 49.

The logic for controlling the direction of signal transfer on the bidirectional address lines includes a first inverter circuit 187 and a second inverter circuit 188. In earlier descriptions, it was indicated that the only time address information is being generated from the channel to all peripheral devices attached to the interface bus is during initial selection of a peripheral device in response to an OIO instruction in the central processing unit. When this occurs, address bit 16 on the address bus 49 is caused to be a binary 1. When bit 16 of the address bus 49 is a binary 1, it is inverted to a binary 0 by inverter 187 and reinverted to a binary 1 by inverter 188 to enable AND gate 184 to respond to the information on the address bus signal line 49. This enables the driver 183 to repower the information to the further interface bus signal line 172. In all other cases, bit 16 of address bus 49 will be a binary zero, which when inverted by inverter 187, will enable AND gate 186 to enable driver 185 to repower signals from the further interface bus 172 onto the address bus 49 to the channel. This operation is effective during cycle steal data transfers in which the peripheral device transmits address information to the central processing unit.

For each bidirectional signal line of the data bus 50, a pair of drivers 183, 185 and enabling AND gates 184 and 186 would be included. The selective and alternate generation of either of the drivers will be controlled by the output of an OR circuit 189 and inverter

190. The logic for determining the direction of signal transfer on the data bus is depicted in the remainder of Figure 20.

A first situation in which data is to be gated from the further interface bus to the channel, as signalled on a line 191, is indicated by an AND circuit 192 which receives information from address bus 49 bit positions 1 and 16. A binary 1 on address bus signal line 49-16, as previously indicated, indicates the initial selection of a peripheral device in response to an OIO instruction. During the initial selection, if the selection is for the purpose of effecting a direct program control (DPC) read operation, bit position 1 of the address bus 49 is part of the command code indicating a read command and will be a binary 0 which is inverted by an inverter 193 to enable AND circuit 192 to generate a signal through OR circuit 189 enabling the peripheral device to transfer data to the channel in response to the read command.

An interrupt service gate trigger 194 and an AND circuit 195 indicate another situation in which the drivers of the bidirectional data bus are to transfer information to the channel as indicated by a signal on line 191. During a cycle steal data transfer by a peripheral device, a binary 1 will be indicated on line 57 as being an input cycle steal operation, and the servicing of the cycle steal data transfer will be indicated by the normal service gate signal on line 55. The only other determination that must be made for this particular situation is that the service gate signal 55 is not being generated for the purpose of handling interrupts, this being signalled by the fact that trigger 194 will be in the off condition providing the third enabling input to AND circuit 195.

The third input to OR circuit 189 is the binary 1 signal indicated by the fact that trigger 194 is in the on, or set state, which causes the information on the bidirectional data bus to be gated to the channel because an interrupt service is in process. Information such as interrupt ID must be gated to the CPU to be placed in internal registers. An interrupt poll sample trigger 196 remembers that a poll for an interrupt service has been received on line 63 when the poll ID bit 0 on line 62-0 is binary 0, inverted through an inverter 197.

Therefore, the trigger 196 sets only when a poll for an interrupt has occurred, as opposed to a poll for a cycle steal operation. When the trigger 196 has been set and is active, the next service gate signal received on line 55 will be effective to set trigger 194.

When the interrupt service gate trigger 194 is turned on, the interrupt poll sampled trigger 196 is reset through an inverter 198 and OR circuit 199. An AND circuit 200 through an OR circuit 201 is effective to reset the interrupt service gate trigger 194. This occurs when the interrupt poll sample trigger 196 has been reset and the service gate on line 55 has fallen as indicated through an inverter 202, indicating completion of the interrupt service cycle.

A further OR circuit 203 responds to various signals on the interface from the channel and is effective to reset all of the logic shown in Figure 20.

One additional feature is effected by an additional input to the AND gate 186 associated with all of the bidirectional signal lines. When an additional card file and power supply are to be added to the base system, the power supply system will include a signal line called power on reset which is comparable to the power on reset signal line 57 of the base data processing system. The normal condition of a power on reset signal on the line 204, associated with the further interface bus, is deactivated, for a binary 0 condition. Through the operation of inverter 205, AND gate 186 will remain enabled. Even if the power supply of the further interface bus is turned off, the interface of the base system can continue the function with all the devices directly attached to it.

Any attempt to select a peripheral device on the further interface bus would merely cause a condition code of 0 to be returned to the channel indicating that it is inoperative. Since the driver 185 represents only a single load on the interface signal line 49, no matter how many peripheral devices on the further interface bus are removed, there will be no adverse effect realized with signal levels on the interface of the base system.

The function of the input 204 from the power supply of the further interface is that when power on reset is effected, line 204 becomes positive which disables AND gate 186. During the process of power on reset, in which various components on the further interface bus are switching states and signal conditions, these extraneous signals will not be propagated through AND gate 186 to the interface bus of the base system.

There has thus been shown, in accordance with the present invention, standard connector circuitry included on an attachment card for connection to an interface bus of a base data processing system, for the purpose of repowering both bidirectional and unidirectional signal lines. The simple logic responds to normal signals on the interface thus to control the direction of signal transfer on the bidirectional busses. Additional logic is not required in either the channel or on the attachment card to recognize address information indicating the need to energize the drivers for the purpose of sending signals to a further input/output interface bus. Also, propagation delays for critical polling signals is reduced by including, on the connector circuitry attachment card, certain logic which would normally be

associated with a peripheral device physically remote from the interface bus. Further, adverse effects of the power fluctuations or spurious signals associated with the further interface bus are eliminated, or isolated, through the operation of simple logic on the repowering card.

WHAT WE CLAIM IS:-

1. Data processing apparatus having a central processing unit, an I/O (input/output) channel, a first I/O bus directly connected with the central processing unit and I/O attachments attaching or to attach I/O devices connected to be powered by a first power supply and a second I/O bus and I/O attachments attaching or to attach further I/O devices connected to be powered by a second power supply, each I/O bus including data address and control signal lines, there being an I/O interface connector circuit to control the interconnection of data, address, and control signal lines between the first I/O bus and the second I/O bus, the I/O interface connector circuit appearing to the first I/O bus as an attached I/O device and including bidirectional driver means interconnecting data and address lines of the first I/O bus and corresponding data and address lines of the second I/O bus, unidirectional driver means interconnecting control signal lines of the first and second I/O buses, and logic means, connected and responsive to predetermined signals being transferred between any attached I/O device and the I/O channel, and connected to said bidirectional driver means, to control the direction of signal transfer on the data and address lines.

2. Apparatus as claimed in claim 1 wherein each said bidirectional driver means includes: a first driver for generating signals on the second I/O bus, a second driver for generating signals on the first I/O bus, a first gate connected for enabling said first driver, and connected and responsive to signals on the first I/O bus and said logic means, a second gate connected for enabling said second driver, and connected and responsive to signals on the second I/O bus and said logic means, and said logic means includes means for selectively rendering either said first or said second gate effective.

3. Apparatus as claimed in accordance with claim 2 wherein each said second gate is connected and responsive to a predetermined signal on a predetermined one of the unidirectional control signal lines of the further bus for rendering said second gate inoperative.

4. Apparatus as claimed in claim 1 including peripheral devices directly attached to the first I/O bus in a multi-drop configuration and including interface capture logic responding to a poll signal from the I/O channel to either serially propagate the poll signal to a succeeding peripheral device or degate the poll signal to capture use of the I/O bus, and at least one peripheral device is directly attached to the second I/O bus, wherein the connector circuit also includes: interface capture logic including means connected and responsive to the presence or absence of a request signal from the peripheral device's on the second I/O bus to render said interface capture logic effective to degate or propagate, respectively, the poll signal; and means directly connecting said interface capture logic to certain of the control signal lines of the first I/O bus.

5. Data processing apparatus 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 Croydon Printing Company Limited, Croydon, Surrey, 1981.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

associated with a peripheral device physically remote from the interface bus. Further, adverse effects of the power fluctuations or spurious signals associated with the further interface bus are eliminated, or isolated, through the operation of simple logic on the repowering card.

WHAT WE CLAIM IS:-

1. Data processing apparatus having a central processing unit, an I/O (input/output) channel, a first I/O bus directly connected with the central processing unit and I/O attachments attaching or to attach I/O devices connected to be powered by a first power supply and a second I/O bus and I/O attachments attaching or to attach further I/O devices connected to be powered by a second power supply, each I/O bus including data address and control signal lines, there being an I/O interface connector circuit to control the interconnection of data, address, and control signal lines between the first I/O bus and the second I/O bus, the I/O interface connector circuit appearing to the first I/O bus as an attached I/O device and including bidirectional driver means interconnecting data and address lines of the first I/O bus and corresponding data and address lines of the second I/O bus, unidirectional driver means interconnecting control signal lines of the first and second I/O buses, and logic means, connected and responsive to predetermined signals being transferred between any attached I/O device and the I/O channel, and connected to said bidirectional driver means, to control the direction of signal transfer on the data and address lines.

2. Apparatus as claimed in claim 1 wherein each said bidirectional driver means includes: a first driver for generating signals on the second I/O bus, a second driver for generating signals on the first I/O bus, a first gate connected for enabling said first driver, and connected and responsive to signals on the first I/O bus and said logic means, a second gate connected for enabling said second driver, and connected and responsive to signals on the second I/O bus and said logic means, and said logic means includes means for selectively rendering either said first or said second gate effective.

3. Apparatus as claimed in accordance with claim 2 wherein each said second gate is connected and responsive to a predetermined signal on a predetermined one of the unidirectional control signal lines of the further bus for rendering said second gate inoperative.

4. Apparatus as claimed in claim 1 including peripheral devices directly attached to the first I/O bus in a multi-drop configuration and including interface capture logic responding to a poll signal from the I/O channel to either serially propagate the poll signal to a succeeding peripheral device or degate the poll signal to capture use of the I/O bus, and at least one peripheral device is directly attached to the second I/O bus, wherein the connector circuit also includes: interface capture logic including means connected and responsive to the presence or absence of a request signal from the peripheral device's on the second I/O bus to render said interface capture logic effective to degate or propagate, respectively, the poll signal; and means directly connecting said interface capture logic to certain of the control signal lines of the first I/O bus.

5. Data processing apparatus 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 Croydon Printing Company Limited, Croydon, Surrey, 1981.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

