FACILITATING TRANSPORT MODE INPUT/OUTPUT OPERATIONS BETWEEN A CHANNEL SUBSYSTEM AND INPUT/OUTPUT DEVICES

BACKGROUND

The present disclosure relates generally to input/output (I/O) processing, and in particular, to providing features to facilitate transport mode I/O operations.

Input/output (I/O) operations are used to transfer data between memory and I/O devices of an I/O processing system. Specifically, data is written from memory to one or more I/O devices, and data is read from one or more I/O devices to memory by executing I/O operations.

To facilitate processing of I/O operations, an I/O subsystem of the I/O processing system is employed. The I/O subsystem is coupled to main memory and the I/O devices of the I/O processing system and directs the flow of information between memory and the I/O devices. One example of an I/O subsystem is a channel subsystem. The channel subsystem uses channel paths as communications media. Each channel path includes a channel coupled to a control unit, the control unit being further coupled to one or more I/O devices.

The channel subsystem and I/O device may operate in a transport mode that supports the transfer of one or more command control blocks to transfer data between the I/O devices and memory. A transport control word (TCW) specifies one or more I/O commands to be executed. For commands initiating certain I/O operations, the TCW designates memory areas associated with the operation, the action to be taken whenever a transfer to or from the area is completed, and other options.

Such I/O operation systems are typically concerned with processing individual operations as either input data operations (e.g., read operations) or output data operations (e.g., write operations). These systems typically do not support bi-directional operations, and particularly do not include capability for generating or transferring indications of bi-directional support.

SUMMARY

An embodiment includes a computer program product for performing an input/output (I/O) operation initiated by an I/O operation instruction at a host computer system configured for communication with a control unit. The computer program product includes a tangible storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method including: sending, by a channel subsystem in the host computer system, a process login (PRLI) request message to the control unit to initialize a link between the channel subsystem and the control unit, the PRLI request message including a field having a value that indicates whether the channel subsystem supports bi-directional data transfer; receiving a PRLI response message from the control unit, the PRLI response message including a field having a value that indicates whether the control unit supports bi-directional data transfer; providing an indication to a host operating system that bi-directional data transfer is supported; and responsive to executing the I/O operation instruction received from the host computer system, performing a method. The method includes: gathering a plurality of commands associated with the I/O operation instruction received from the host computer system, at least one of the plurality of commands specifying an input data transfer and at least one of the plurality of commands specifying an output data transfer; transferring the plurality of commands to the control unit; sending at least one output data message to the control unit including output data to be transferred to the control unit, the output data message associated with the at least one of the plurality of commands specifying an output data transfer; and receiving at least one input message from the control unit including input data to be stored in a main storage of the host computer system, the input data message associated with the at least one of the plurality of commands specifying an input data transfer.

Another embodiment includes an apparatus for performing an input/output (I/O) operation initiated by an I/O operation instruction at a host computer system configured for communication with a control unit. The host computer system is configured to perform: sending, by a channel subsystem in the host computer system, a process login (PRLI) request message to the control unit to initialize a link between the channel subsystem and the control unit, the PRLI request message including a field having a value that indicates whether the channel subsystem supports bi-directional data transfer; receiving a PRLI response message

from the control unit, the PRLI response message including a field having a value that indicates whether the control unit supports bi-directional data transfer; providing an indication to a host operating system that bi-directional data transfer is supported; and responsive to executing the I/O operation instruction received from the host computer system, performing a method including: gathering a plurality of commands associated with the I/O operation instruction received from the host computer system, at least one of the plurality of commands specifying an input data transfer and at least one of the plurality of commands specifying an output data transfer; transferring the plurality of commands to the control unit; sending at least one output data message to the control unit including output data to be transferred to the control unit, the output data message associated with the at least one of the plurality of commands specifying an output data transfer; and receiving at least one input message from the control unit including input data to be stored in a main storage of the host computer system, the input data message associated with the at least one of the plurality of commands specifying an input data transfer.

Another embodiment includes a method of performing an input/output (I/O) operation initiated by an I/O operation instruction at a host computer system configured for communication with a control unit. The method includes: sending, by a channel subsystem in the host computer system, a process login (PRLI) request message to the control unit to initialize a link between the channel subsystem and the control unit, the PRLI request message including a field having a value that indicates whether the channel subsystem supports bi-directional data transfer; receiving a PRLI response message from the control unit, the PRLI response message including a field having a value that indicates whether the control unit supports bi-directional data transfer; providing an indication to a host operating system that bi-directional data transfer is supported; and responsive to executing the I/O operation instruction received from the host computer system, performing a method including: gathering a plurality of commands associated with the I/O operation instruction received from the host computer system, at least one of the plurality of commands specifying an input data transfer and at least one of the plurality of commands specifying an output data transfer; transferring the plurality of commands to the control unit; sending at least one output data message to the control unit including output data to be transferred to the control unit, the output data message associated with the at least one of the plurality of commands specifying an output data transfer; and receiving at least one input message from the control

unit including input data to be stored in a main storage of the host computer system, the input data message associated with the at least one of the plurality of commands specifying an input data transfer.

Additional features and advantages are realized through the techniques of the present embodiment. Other embodiments and aspects are described herein and are considered a part of the claimed invention. For a better understanding of the invention with the advantages and features, refer to the description and to the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

FIG. 1 depicts one embodiment of an I/O processing system incorporating and using one or more aspects of the present invention;

FIG. 2 depicts one embodiment of a transport control word (TCW);

FIG. 3 depicts one embodiment of a transport command control block (TCCB);

FIG. 4 depicts one embodiment of a transport command information unit (IU) that includes the TCCB of FIG. 3;

FIG. 5 depicts one embodiment of a transport command header of the transport command IU of FIG. 4;

FIG. 6 depicts one embodiment of a transport command area header (TCAH) of the TCCB of FIG. 4;

FIG. 7 depicts one embodiment of a transport command area (TCA) of the TCCB of FIG. 4;

FIG. 8 is a table describing exemplary determinations of various data counts for uni-directional data transfer operations;

FIG. 9 is a table describing exemplary determinations of various data counts for bidirectional data transfer operations;

FIG. 10 depicts one embodiment of a device command word (DCW);

FIG. 11 depicts one embodiment of a control flags field of the DCW of FIG. 10;

FIG. 12 depicts one embodiment of a transport command area extension (TCAX);

FIG. 13 depicts one embodiment of a CBC-offset block;

FIG. 14 is a flow chart illustrating an embodiment of a method of performing a transport mode I/O operation;

FIG. 15 depicts embodiments of transport-data IUs used to transport input and output data between a channel and a control unit and/or I/O device;

FIG. 16 depicts one embodiment of a transport response IU;

FIG. 17 depicts one embodiment of a status area of the transport response IU of FIG. 16

FIG. 18 depicts components of an exemplary process login (PRLI) request;

FIG. 19 depicts components of an exemplary PRLI response

FIG. 20 depicts an exemplary channel path description block provided to a host computer system OS from a channel subsystem; and

FIG. 21 depicts an exemplary channel description data of the description block of FIG. 20, including channel subsystem capability data.

DETAILED DESCRIPTION

Embodiments of the present invention facilitate input/output (I/O) processing in a computer system. In one embodiment, transport commands are provided to manage the I/O operation and the transfer of data via a channel path. Transport commands transfer command meta-information (TCMI) that is used to manage data transfer checking and the transfer of additional DCWs. TCMI is defined in one or more data transport information units (IUs) for

defining, specifying and verifying information regarding data to be transferred in an I/O operation. Transport commands may include an interrogate command, a transfer-CBC-Offset Block (TCOB) command, and transfer-TCA Extension (TTE) command.

In one embodiment, I/O processing is facilitated by enabling a host computer to increase the number of commands that can be sent to a device for an I/O operation. For example, TCMI in the form of a Transport Command Area Extension (TCAX) can be sent from the host to a device to increase the number of commands that can be associated with an I/O operation. In one embodiment, the TCAX is sent via a device command word (DCW), referred to as a Transfer TCA Extension (TTE), that includes commands to be performed in addition to commands sent in a Transport Command Area (TCA).

I/O processing can also be facilitated by providing a means by which an I/O device can continue an I/O operation when an incorrect length record is encountered. For example, a DCW-incorrect-length facility may be installed in the host system and a control unit or device. The facility provides for Incorrect Length (IL) checking that may be performed by a control unit. A suppress-incorrect-length or Suppress Length Indication (SLI) field may be added to a DCW sent to a device. When this bit is activated, DCW chaining is permitted to continue when an incorrect length condition is detected by the control unit.

I/O processing can also be facilitated by providing a means for sending both read and write commands to a device in a single I/O operation. Prior art TCW channel programs specify either all write commands or all read commands. A bi-directional operation can be specified by setting one or more flags in a transport command header (TCH) of a command IU, e.g., by setting both read and write flags to one and providing both a read and a write data count. In addition, a flag in an initiation or login message such as a Process Login (PRLI) request message can be set to indicate that the channel subsystem supports bidirectional data transfer. A response message such as a PRLI accept message can include a flag that can be set to indicate whether the device supports bi-directional data transfer. If bi-directional data transfer is supported by both the host and the control unit, the device will not report an error if both read and write flags are set in the command IU. This capability is an improvement over prior art Fibre Channel protocols, which do not support bi-directional data transfer operations. As described herein, "bi-directional data transfers" are transfers of both input

and output data that are performed within a single TCW I/O operation, e.g., sending and executing both read and write commands within a single TCW I/O operation.

FIG. 1 illustrates an exemplary embodiment of an I/O processing system 100 that includes a host computer system 102 that includes a data storage and/or processing system such as a zSeries® mainframe computer by International Business Machines Corporation (IBM®). IBM is a registered trademark of International Business Machines Corporation, Armonk, New York, USA. Other names used herein may be registered trademarks, trademarks or product names of International Business Machines Corporation or other companies. The host computer system 102 includes various processing, storage and communication elements. In one embodiment, the host computer system 102 includes one or more central processing units (CPUs) 104, memory components such as a main storage or memory 106, an expanded storage or memory 108, one or more operating systems (OSs) 110 that are executed by one or more of the CPUs 104. For example, one CPU 104 can execute a Linux® operating system 110 and/or a z/OS® operating system 110 as different virtual machine instances. CPU 104 is the controlling center of the I/O processing system 100. It contains sequencing and processing facilities for instruction execution, interruption action, timing functions, initial program loading, and other machine-related functions. CPU 104 is coupled to the main memory 106 and/or expanded memory 108 via a connection 113, such as a bidirectional or unidirectional bus.

The host computer system 102 also includes a channel subsystem 114 that provides a communication interface between the host system 101 and various I/O devices 116, which may controlled by one or more control units 118. I/O devices include equipment such as printers, magnetic-tape units, direct-access-storage devices, displays, keyboards, communications controllers, teleprocessing devices, and sensor-based equipment. In the description herein, the terms "control unit" and "device" may be used interchangeably, or a control unit may be considered to include one or more devices. The channel subsystem 114 directs the flow of information between the I/O devices 116 and the host computer system 102. It relieves the CPUs 104 of the task of communicating directly with the I/O devices 116 and permits data processing to proceed concurrently with I/O processing. The channel subsystem 114 is coupled to the CPUs 104, the main memory 106 and/or the expanded memory 108 via a connection 120, such as a bus.

In one embodiment, the channel subsystem 114 is connected to each I/O device 116 via a respective "channel path" 122 that connects the channel subsystem 114 to each control unit 118 via a connection 124 such as a serial or parallel link. Control units 118 may be attached to the channel subsystem 114 via more than one channel path 122, and an I/O device 116 may be attached to more than one control unit 118 and/or I/O device 116. In all, an individual I/O device 116 may be accessible by multiple channel paths. A channel path can use various types of connections, such as a parallel interface, a serial-I/O interface and a FICON I/O interface. For example, a serial channel path may include one or more optical fibers connected to a control unit 118 via, e.g., a dynamic switch 126 in a Fibre channel fabric, and a parallel interface may include a number of electrical or fiberoptic conductors.

In one embodiment, the channel subsystem 114 includes one or more individual channels 128 that are each connected to one or more control units 118 and/or I/O devices 116 via one or more channel paths 122. Each channel 128 includes processing electronics such as a local channel microprocessor 130 and a local channel memory 132 that is connected to and accessible by the local channel microprocessor 130. The local channel memory 132 may include information such as a channel-program designation, a channel-path identifier, a device number, a device count, status indications, as well as information on path availability and functions pending or being performed.

Also located within each channel 128 are one or more subchannels. Each subchannel is a data structure located within a channel memory 132 that provides information concerning an associated I/O device 116 and its attachment to the channel subsystem 114. The subchannel also provides information concerning I/O operations and other functions involving the associated I/O device 116. The subchannel is the means by which the channel subsystem 114 provides information about associated I/O devices 116 to the CPUs 104. In one embodiment, the number of subchannels provided by the channel subsystem is independent of the number of channel paths 122 to the associated I/O devices 116. For example, a device 116 accessible through alternate channel paths 122 still is represented by a single subchannel.

Each control unit 118 provides logic to operate and control one or more I/O devices 116 and adapts, through the use of common facilities, the characteristics of each I/O device 116 to the link interface provided by a channel 128. The common facilities provide for the execution

of I/O operations, indications concerning the status of I/O devices 116 and the control unit 118, control of the timing of data transfers over a channel path 122 and certain levels of I/O device control. A control unit 118 may be housed separately, or may be physically and logically integrated with an I/O device, the channel subsystem, or a CPU.

One or more of the above components of the I/O processing system 100 are further described in "IBM® z/Architecture Principles of Operation," Publication No. SA22-7832-08, 9th Edition, August 2010, which is hereby incorporated herein by reference in its entirety.

I/O operations are described as any operation that involves the transfer of data between the host computer system 102 and I/O devices 116. As described herein, an I/O operation includes the communications between the channel subsystem 114 and a device 116 (via, in one embodiment, a control unit 118) in which a single command (e.g., a channel command word or CCW), a single command message including multiple commands (e.g., a transport command information unit or transport command control block (TCCB)), or multiple chained commands (e.g., multiple CCWs) are sent from the channel subsystem 114 to a device. The I/O operation may also include one or more response messages generated by the device 116 or an associated control unit 118 in response to receiving and/or executing the command or chained commands.

In one embodiment, I/O operations are initiated with a device 116 by the execution of I/O instructions generated by an OS 110 that designate the subchannel associated with the device 116. Such instructions are executed in the host system by a CPU 104 by sending parameters to a channel 128 or subchannel to request that the channel subsystem 114 perform various functions in an I/O operation.

For example, the CPU 104 executes a "START SUBCHANNEL" instruction by passing parameters to the target subchannel requesting that the channel subsystem 114 perform a start function with the I/O device 116 associated with the subchannel. The channel subsystem 114 performs the start function by using information at the subchannel, including the information passed during the execution of the START SUBCHANNEL instruction, to find an accessible channel path to the device 116, and to execute the I/O operation once a channel path has been selected.

When an instruction such as a START SUBCHANNEL instruction is executed by the CPU 104, a channel 128 commences performing the I/O operation. In one embodiment, the channel subsystem 114 operates under a High Performance FICON (HPF) protocol for communication between the channel subsystem 114 and the devices 116 and/or control units 118. FICON and HPF are described further in "Fibre Channel: Single-Byte Command Code Sets Mapping Protocol – 4 (FC-SB-4)," T11 Project 2122-D, Revision 3.00, September 22, 2009, which is hereby incorporated herein by reference in its entirety.

In one embodiment, execution of the START SUBCHANNEL instruction passes the contents of an operation request block (ORB) to the channel subsystem 114. The ORB specifies a channel program that includes an address of one or more command words (e.g., a channel command word or a transport command word described further below). There are two modes of subchannel operation. In one embodiment, the host computer system 102 operates in a command mode and specifies command word(s) in the form of a channel command word (CCW). In another embodiment, the host system operates in a transport mode and specifies command word(s) in the form of a transport command word (TCW).

A subchannel may enter transport mode when an FCX (Fibre Channel Extensions) facility is installed and the start function is set at the subchannel as the result of the execution of a START SUBCHANNEL instruction that specifies a TCW channel program. The subchannel remains in transport mode until the start function is reset at the subchannel. At all other times, the subchannel is in command mode.

In command mode, the channel executes a CCW channel program that include a single channel-command word or a sequence of channel-command words executed sequentially that control a specific sequence of channel operations. A control unit executes a CCW I/O operation by decoding, accepting, and executing CCW commands by an I/O device. One or more CCWs arranged for sequential execution form a CCW channel program and are executed as one or more I/O operations, respectively.

The fibre-channel-extensions (FCX) facility is an optional facility that provides for the formation of a transport mode channel program that is composed of a transport control word (TCW) that designates a transport-command-control block (TCCB) and a transport-status block (TSB). The TCCB includes a transport-command area (TCA) which contains a list of

one or more (e.g., up to 30) I/O commands that are in the form of device-command words (DCWs). A TCW and its TCCB may specify either a read or a write operation. In one embodiment, a FCX-bidirectional-data-transfer facility may be installed in a system that supports transport mode operations, that allows the host computer system 102 to specify transfer of both input and output data in a single transport mode I/O operation if the connected device 116 and control unit 118 supports bidirectional-data transfer. When a control unit 118 recognizes bi-directional data transfers, a TCW and its TCCB may, depending on the device, specify both read and write data transfers.

In the transport mode, a single transport command word (TCW) specifies a location in memory of a TCCB (as well as a location in memory 106 or 108 of one or more data areas) that is sent in a single message instead of separate individual CCWs in the command mode. A control unit 118 executes a transport mode I/O operation by decoding, accepting, and executing a TCCB and the individual DCWs included therein. If the ORB specifies a TCW channel program, the channel subsystem 114 uses information in the designated TCW to transfer the TCCB to a control unit 118. The contents of the TCCB are ignored by the channel subsystem 114 after the TCCB is transferred to the control unit 118 and only have meaning to the control unit 118 and the associated I/O device 116.

In an exemplary embodiment, the control unit 118 generates a response message in response to executing the channel program. The control unit 118 may also generate a response message without executing the channel program under a limited number of communication scenarios, e.g., to inform the channel subsystem 114 that the channel program will not be executed. The control unit 118 may include a number of elements to support communication between the I/O communications adapter and I/O devices, as well as in support of channel program execution. For example, the control unit 118 can include control logic to parse and process messages, in addition to one or more queues, timers, and registers to facilitate communication and status monitoring.

FIG. 2 illustrates an embodiment of a transport-control word (TCW) 140, which is stored in the host system (e.g., main memory 106) and specifies at least one control block that is to be transferred to a control unit 118 from a channel 128. In one embodiment, the control block is a transport-command-control block (TCCB) whose contents are to be transported to the control unit 118 and I/O device 116 for processing. When the TCW 140 specifies a TCCB,

the TCCB includes a TCA that specifies one or more device-command words (DCWs) and associated options. For a DCW that specifies a command which initiates the transfer of data (with the exception of control data contained within the TCCB), the TCW 140 designates one or more storage areas where the data is located.

In one embodiment, the TCA includes a Transfer TCA Extension (TTE) command that is provided to transfer additional DCWs and/or control data in addition to those that can be accommodated in the TCCB. The TTE transfers a TCA extension (TCAX) to a control unit (which includes at least one DCW) and is considered a logical extension of the TCA to which it is associated, by, for example chaining the last DCW of the TCCB to the first DCW of the TTE.

Referring again to FIG. 2, and embodiment of the TCW 140 is a 64-byte control block that is designated on a 64-byte boundary. The TCW includes various fields described below.

For example, a number of flag fields 142 indicate whether direct or indirect data addressing is being used to locate input data, output data or the TCCB. An Output transport-indirect-data addressing (TIDA) field indicates whether output data is directly or indirectly addressed. For example, when word 0, bit 7 of the TCW 140 is zero and output data is specified, an output-data-address field 144 designates an output data location in absolute storage. When bit 7 is one, the output-data-address field 144 designates an absolute address of a TIDA word (TIDAW) or the first TIDAW in a list of TIDAWS (a TIDAL) that designates the output storage location or locations, respectively. An Input TIDA field indicates whether input data is directly or indirectly addressed. For example, when word 0, bit 5 is zero and input data is specified, an input-data-address field 146 designates the absolute address of the input storage location (i.e., where input data is to be stored. When bit 5 is one, the input-data-address field 146 designates the absolute address of a TIDAW or the first TIDAW in a TIDAL that designates the input storage location or locations. A Transport-command-control-block TIDA (TCCB-TIDA) field indicates whether the TCCB is directly or indirectly addressed. For example, when word 0, bit 6 is zero, a TCCB-address field 148 designates the absolute address of the TCCB for the TCW, and when bit 6 is one, the TCCB-address field 148 designates the absolute address of a TIDAW or TIDAL that designate the location or locations of the TCCB.

The TCW 140 also includes a Transport-Command-Control-Block Length (TCCBL) field 150 that specifies the length in bytes of the TCCB. For example, the TCCBL field includes an unsigned integer whose value (e.g., when added to 20 for unidirectional data transfers or when added to 24 for bidirectional data transfers) specifies the length of the TCCB in bytes.

A Read Operations (R) field 152 is non-zero (e.g., bit 14 of word 1 is one) when indicating the number of bytes to be transferred into main memory 106. A Write Operations (W) field 154 is non-zero (e.g., bit 15 of word 1 is one) when indicating the number of bytes to be transferred from main storage. If the R-field 152 and the W-field 154 are both one, a program-check condition is recognized, unless the target device 116 and control unit 118 support bi-directional data transfer. If the W-bit is one and the TCW is an interrogate TCW, a program-check condition is recognized.

The Output-Data Address field 144 indicates the location in storage of any output data to be sent to a device. For example, when bit 15 of word 1 (the W-field bit 154) is one and bit 7 of the flags field 142 (the output-TIDA flag) is zero, words 2-3 designate the 64-bit output location in absolute storage. When the W-bit is one and the output-TIDA flag is one, words 2-3 designate the 64-bit location in absolute storage of a TIDAW or list of TIDAWs that designate the output storage location or locations. The Input-Data Address field 146 indicates the location to which any input data is to be stored when received from a device. For example, when the bit 14 of word 1 (the R-field bit 152) is one and bit 5 of the flags field 142 (the input-TIDA flag) is zero, words 4-5 designate the 64-bit input location in absolute storage. When the R-bit is one and the input-TIDA flag is one, words 4-5 designate the location in absolute storage of a TIDAW or list of TIDAWs that designate the input storage location(s).

A Transport-Status-Block Address 156 specifies a location in storage of a transport-status block for the TCW. For example, words 6-7 designate the 64-bit location in absolute storage of the transport-status block for the TCW.

A Transport-Command-Control-Block Address field 148 specifies one or more (direct or indirect) addresses of the TCCB. For example, if the TCCB-TIDA bit (bit 6 of the flags field) is zero, words 8-9 designate 64-bit location in absolute storage of the TCCB. When the TCCB-TIDA bit is zero, the TCCB is specified to reside in a contiguous area of storage.

If the transport-command-control-block-TIDA bit is one, words 8-9 designate the 64-bit location in absolute storage of a TIDAW or list of TIDAWs that designate the location in absolute storage of the TCCB. When the TCCB-TIDA bit is one, the TCCB may be specified to reside in noncontiguous areas of storage

The Output Count field 158 specifies the number of output bytes for the TCW. For example, when bit 15 of word 1 (the W-bit) is one, word 10 contains the unsigned integer total count of output bytes for the TCW. The Input-Count field 160 specifies the number of input bytes for the TCW. For example, when bit 14 of word 1 (the R-bit) is one, word 11 contains the unsigned integer total count of input bytes for the TCW.

If the TCW specifies an interrogation operation, an Interrogate-TCW Address field 162 indicates a location in storage of an Interrogate TCW. For example, when a START SUBCHANNEL instruction designates a TCW, word 15 of the TCW is not checked. However, when a CANCEL SUBCHANNEL instruction designates a subchannel that is start pending for a TCW channel program and is not status pending, bits 1-31 of word 15 of the TCW designated by START SUBCHANNEL specify the 31-bit location in absolute storage of the interrogate-TCW that is used to initiate an interrogate operation for the subchannel. If word 15 contains zeros when CANCEL SUBCHANNEL is issued, an interrogate operation is not initiated. When CANCEL SUBCHANNEL is issued, bit 0 of word 15 must be zero, otherwise a program-check condition is recognized with interrogate-failed status indicated. When CANCEL SUBCHANNEL is issued and bits 1-31 of word 15 do not contain zeros, bits 1- 31 of word 15 must designate a storage location on a 64-byte boundary, otherwise a program-check condition is recognized with interrogate-failed status indicated. Word 14 may be reserved for extending the interrogate-TCW-address field to 64 bits.

Referring to FIGS. 3 and 4, the transport-command-control block (TCCB) 170 includes one or more individual commands as part of a TCW I/O operation, and is sent to a control unit 118 and/or device 116 by a channel 128 via a channel path. The TCCB 170 relieves the channel of having to send multiple messages or information units, and also transfers the responsibility of executing the operation to the control unit and removes the need for the control unit 118 to send responses for each command. Instead, the control unit 118 can execute all of the commands and send a response upon completion of the operation.

The TCCB 170 is variable in length, may contain header and trailer information, and one or more (e.g., from 1 to 30) commands as device-command words (DCWs) that are logically linked (e.g., chained) such that they are executed by the control unit 118 in a sequential manner. The TCCB 170 may reside as a single block of contiguous storage or may reside as multiple blocks of noncontiguous storage. For example, the TCCB-TIDA flag in the TCW 140 described above is used to specify whether the TCCB resides in contiguous storage.

Examples of a TCCB are shown in FIGS 3 and 4. As shown in FIG. 4, the TCCB 170 may be sent from a channel 128 to a control unit 118 as part of a transport command information unit (IU) 172 that is sent to the control unit 118 to initiate an I/O operation. In one embodiment, the transport command IU 172 is made up of an 8-byte SB-4 header 174, followed by a 4-byte transport command header (TCH) 176, and the TCCB 170. The TCCB 170 includes a 16-byte transport-command-area header (TCAH) 178, a variable length transport-command area (TCA) 180, and a transport command area trailer (TCAT) 182. The TCAT 182 may further include a 4-byte LRC field 184, a 4-byte data-transfer length (DL) field 186, and for bidirectional operations, a 4-byte bidirectional read data-transfer length (BRDL) field 188. These fields are described further below.

Referring to FIG. 4, the SB-4 header 174 provides FC-4 addressing information to identify the logical path and the device 116 for the data transfer. The SB-4 header 174 provides information including a channel image ID and a control unit ID for a logical path between a channel 128 and a control unit 118, as well as a device ID.

Referring to FIG. 5, the TCH 176 includes information about the TCCB 170 and the associated device operations. In one embodiment, the TCH 176 contains 4 bytes and immediately follows the SB-4 header 174 in the transport command IU 172. The TCH 176 includes fields such as a length field ("L1") 190, a read field ("R") 192 and a write field ("W") 194. The L1 field (located at e.g., bits 24-29), specifies the length, in words, of the TCA 180 plus the 1-word LRC field 184 that directly follows the TCA 180. For example, for control units 118 that do not support bidirectional operations, the total amount of data transferred in the transport command IU 172 is equal to the L1 field plus 8 (that is, the L1 field plus the 2-word SB-4 header, the 1-word TCH, the 4-word TCAH and the 1-word DL); otherwise, the control unit recognizes a transport-command IU integrity error due to a data count error. For control units that support bidirectional operations, the total amount of data

transferred in the transport-command IU 172 is equal to the L1 field plus 8 (that is, the L1 field plus the 2-word SB-4 header, the 1-word TCH, the 4-word TCAH and the 1-word DL) or the L1 field plus 9 (that is, the L1 field plus the 2-word SB-4 header, the 1-word TCH, the 4-word TCAH, the 1-word DL, and the 1-word BRDL for a bidirectional operation); otherwise, the control unit shall recognize a transport-command IU integrity error due to a data count error. If a transport-command IU integrity error is not recognized, the location of the LRC in the TCCB is located based on the L1 field and the LRC check can be performed.

The Read (R) field 192 specifies that the TCA 180 includes one or more read or input commands. In the embodiment of FIG. 5, the R field 192 is an R bit, bit 30, that indicates, when set to one, that the TCA 180 contains one or more commands that transfer read data. When the R bit is set to zero, the TCA 180 does not contain any commands that transfer read data. When the R bit is one and the W bit (e.g., bit 31) is zero, the operation is referred to as a read operation. When both the R and W bits are set to zero, the TCA 180 does not contain any commands that transfer read or write data, indicating that the I/O operation does not perform data transfer. When both the R and W bits are set to one, the TCA 180 may contain commands that transfer both read and write data, and the operation is referred to as a bidirectional operation. The device 116 or control unit 118 recognizes a TCH content error due to a read-write conflict if a bidirectional operation is specified and the control unit 118 does not support bidirectional operations, or the control unit 118 supports bidirectional operations but the transport-command IU 172 does not contain the BRDL field 188. When the R bit is zero and the control unit 118 or device 116 encounters a command in the TCA 180 that attempts to transfer read data, a TCCB content error is recognized due to an invalid read attempt.

The Write (W) field 194 specifies that the TCA 180 includes one or more write or output commands. In the embodiment of FIG. 5, the W field is a W bit, bit 31, that, when set to one, indicates the TCA 180 includes one or more commands that transfer write data. When the bit is set to zero, the TCA does not include any commands that transfer write data. When the W bit is one and the R bit (bit 30) is zero, the operation is referred to as a write operation. When the W bit is zero and the control unit 118 or device 116 encounters a command in the TCA 180 that specifies a write command, a TCCB content error is recognized due to an invalid write condition.

Referring to FIG. 6, an embodiment of the TCA Header (TCAH) 178 includes information about the TCA 180 and the operations described therein, such as the TCA length and device indications. In this embodiment, Byte 3 of word 1 is an "L2" length_field 196 that contains an unsigned binary integer that specifies the length, in bytes, of the TCA 180 plus words 2 and 3 of the TCA header, plus the 4-byte LRC field 184 that directly follows the TCA 180. A Service-Action Code field 198 (e.g., bytes 0-1 of word 2) includes an unsigned integer value that specifies the type of operation specified by the TCCB. For example, a Hex value of '1FFE' indicates a SB-4 Device I/O operation, and a Hex value of '1FFF' indicates SB-4 Device support functions. The Priority Number field 200 (e.g., byte 3 of word 2) includes the control unit I/O priority number.

Referring to FIG. 7, the transport-command area (TCA) 180 is a variable length area that contains one or more (e.g., from 1 to 30) commands as device-command words (DCWs) 202. The length of the TCA 180, in one embodiment, is an in integral number of 4-byte words. For DCWs 202 that specify device control data, the TCA 180 also contains the control data associated with each DCW 202. Each DCW 202 that specifies control data reduces the maximum-DCW capacity by one or more DCWs, depending on the size of the command-associated data. For DCWs 202 that specify input or output data, the TCW 140 designates the associated storage area or areas and the DCW designates the count of bytes to transfer. In one embodiment, the maximum size of the TCA is 240 bytes. As shown in the embodiment of FIG. 7, when sufficient space exists in the TCA 180 for the entire DCW 202, the DCW 202 begins on the word boundary that follows the previous DCW 202 or the control data associated with the previous DCW 202. When the last DCW 202 in the TCA 180 specifies control data that is not an integral number of words, the subsequent LRC field 184 begins on the word boundary that follows the control data.

For some devices, the list of DCWs 202 may extend beyond what will fit in the TCA 180. In such instances, a TCA extension (TCAX) field may be specified that includes additional DCWs and transferred as if the TCAX were output data (e.g., transferred as part of an output transport-data IU). However, the TCAX is treated as a logical continuation of the TCA 180 instead of as transfer data. The TCAX is specified in the TCA 180 by a transfer-TCA-extension (TTE) DCW. The TCAX and TTE DCW are described further below, and embodiments of the TCAX are shown in FIGS. 12 and 13.

As shown in FIGS. 3 and 4, the TCCB may include a transport-command-area trailer (TCAT) 182 that provides additional information about the TCCB, including various count information. In one embodiment, when uni-directional data transfer is specified (either the R-bit or the W-bit is set to one) or no data transfer is specified (both the R-bit and W-bit in the TCW are set to zero), the TCAT 182 is two words in length. When bidirectional data transfer is specified (both the R-bit and the W-bit in the TCW are set to one), the TCAT 182 is three words in length.

For example, when uni-directional data transfer is specified, the transport count or data length (DL) field 186 specifies the 32-bit unsigned integer count of total data to be transferred (the "transport count value"). When a read operation is specified (the TCW R-bit is one), the value in the transport-count field in the TCAT may be determined by summing the count field values in the DCWs 202 that each specify a read-type command. The sum is rounded upwards to the nearest multiple of 4, and the rounded sum is increased by 4 to arrive at the transport-count value. When a read operation is specified, the transport count value should be equal to the value in the TCW input-count field 160, rounded upwards to the next multiple of 4, plus 4.

When a write operation is specified (the W-bit in the TCW is one), the count field values in the DCWs 202 that specify a write command are summed. In addition, the count field values in all transport-command DCWs (described further below) that specify the transfer of transport command-meta information (TCMI), plus the size of the reserved fields in the TCMI are added to the sum. (Note that the size of a TCMI reserved field may be zero). The total of the counts of any TIDAW-specified control block check (CBC) or cyclical redundancy check (CRC) bytes and padding bytes is added to the sum. The sum is rounded upwards to the nearest multiple of 4, and the rounded sum is increased by 4 giving the transport-count value. When a write operation is specified, the transport count value should be equal to the value in TCW output-count field 158, rounded upwards to the next multiple of 4, plus 4. When neither a read nor a write operation is specified (both the W-bit and R-bit in the TCW are zero), the transport count value should be zero.

When bidirectional data transfer is specified, the DL field 186 is the write count and specifies the 32-bit unsigned integer count of total output data to be transferred. The transport count value in the DL field (write-count field) in the TCAT 182 is determined as

described above for a write operation. The bi-directional read data length (BRDL) field 188 specifies the 32-bit unsigned integer count of total input data to be transferred. The transport count value for the BDRL field 188 is determined as described above for read operations.

In one embodiment, for a write operation or bidirectional operation, the DL field shall contain a 4-byte unsigned binary integer that specifies the number of bytes to be transferred by the channel to the control unit for the operation and includes all intermediate and final pad and CRC bytes, and if a COB is included in the first transport-data IU of the write operation, it also includes the COB, any COB pad bytes and the COB CRC bytes. If a TTE DCW is present in the TCA, the DL field also includes the TCAX, any TCAX pad bytes and TCAX CRC bytes.

FIGS. 8 and 9 are tables summarizing the determination of the TCW input-count, output-count, and the TCAT transport count values for unidirectional and bi-directional transfers, respectively.

For a bidirectional operation, a bidirectional-read-data-length (BRDL) field includes a 4-byte unsigned binary integer that specifies the number of bytes to be transferred by the control unit to the channel for the operation and includes any pad bytes required to round to a word boundary if not already on a word boundary plus 4 bytes for CRC. For operations that are not bidirectional operations, the device shall recognize a TCH content error due to a read-write conflict if the transport-command IU contains the BRDL field.

Referring to FIG. 10, a device-command word (DCW) 202 specifies a command to be executed. For commands initiating certain I/O operations, it designates the count of bytes on which the operation is performed, the action to be taken whenever transfer to or from storage is completed, and other options. The storage area or areas associated with a DCW data-transfer operation are designated, depending on the operation specified by the command, by the input-data-address field 146 or the output-data-address field 144 of the TCW 140 that designates the TCCB 170 that includes the DCW 202. Whether the input-data-address field 146 or the output data-address field 144 designates the storage directly or indirectly is specified by the input-TIDA and output-TIDA flags in the TCW 140.

In one embodiment, the DCW 202 is an 8-byte control block that is designated in the TCW 140 on a word boundary. The DCW 202 includes fields such as a command code field 204, a control flags field 206, a control data count 208 and a DCW data count 210.

The Command Code field 204 (e.g., bits 0-7 of word 0) specifies the operation to be performed. Whether a command is valid is device dependent and dependent on the value of the service-action code field 198 in the TCAH 178. Basic commands include read, write, control, sense and transport. The channel subsystem 114 distinguishes among the following operations: control, output forward (write), input forward (read, sense, sense ID), input backward (read backward), branching (transfer in channel) and transport. Some commands, when executed, do not result in the transfer of data but cause the device to chain and start execution of the next command when all of the conditions for command chaining are satisfied. Each of the basic operations is described below.

A read command initiates execution of a device operation that performs device-to-channel data transfer. A write command initiates execution of a device operation that performs channel-to-device data transfer. A control command initiates execution of a device operation that makes use of control data provided in the DCW. The sense command is similar to a read command, except that the data is obtained from sense indicators rather than from a record source. Control commands and associated control data are provided for management of the specific I/O device and control of the device during execution of an I/O command. A transport command is provided to manage the I/O operation and the transfer of data via a channel path, and is not device specific. Such commands manage the transport of control block checking data such as cyclical redundancy check (CRC) data and the transport of additional DCWs in transport-data IUs.

The Control-Data (CD) Count field 208 specifies the number of control data bytes associated with a DCW 202. For example, byte 3 of word 0 specifies the length of the control data, in bytes. If the command code specifies a command that requires control data and byte 3 of word 0 specifies a control-data count that is less than required for the command, a unit-check condition or TCCB content error is recognized. If the command code specifies a command that requires control data and byte 3 of word 0 contains zero or contains a value that specifies data past the end of the TCA (or past the end of the specified TCAX), a device-detected-program check condition or TCCB content error is recognized.

The DCW Data Count field 210 specifies the byte count of the data (e.g., data to be transferred for the read or write operation, i.e., customer data) designated by the TCW for the present DCW 202. For example, the DCW data count is included in word one of the DCW 202 of FIG. 10, and specifies the 32-bit unsigned integer count of bytes in the storage area designated by the TCW for this DCW. The DCW data count field 210 indicates the number of bytes to be transferred between the channel and control unit during execution of the DCW, not including any pad or CRC bytes.

Referring to FIG. 11, in one embodiment, the DCW 202 includes various control flags 206. A chain-command (CC) flag 212 specifies an intent to perform chaining of commands. Upon normal completion of the current device operation and after recognition of device end at the device, chaining of commands causes the next DCW 202 in the TCA 180 to be executed. The CC flag 212 is set to zero in the last DCW in the TCA. In the embodiment of FIG. 10, bit 9 of word 0 (the CC bit), when one, specifies the chaining of commands. When one, the bit causes the operation specified by the next DCW to be initiated upon normal completion of the current DCW. When the control-data count is zero, the next DCW immediately follows the DCW in the TCA or in a TCAX. When the control-data count is not zero, the next DCW immediately follows the control-data, rounded to a word boundary, specified for the DCW. When the chain-command flag is one, in one embodiment, the next DCW location in the TCA or in a TCAX is determined by adding eight and the value in the CD count field to the location of the current DCW and rounding upwards to the nearest word boundary. If the chain-command flag is one in a DCW in a TCA, and the next DCW location is past the end of the TCA, and a TCAX is specified, the next DCW is at the beginning of the specified TCAX.

The control flags field, in one embodiment, include a Suppress Length Indication (SLI) field 214. When the DCW-incorrect-length facility is supported by a channel 128 and a control unit 118, a bit (e.g., bit 2) is provided in the SLI flag field 214 that controls whether command chaining is to occur on an incorrect-length condition and whether incorrect length is to be indicated in a transport-response IU by the control unit when an incorrect-length condition is detected by the control unit 118. An incorrect-length condition is detected by the control unit if the DCW data count does not match the amount of data required by the device for a write DCW or if the DCW data count does not match the amount of data

available at the device for a read DCW. When the DCW-incorrect-length facility is not supported by the channel and control unit, bit 2 is reserved and set to zero by the channel and ignored by the control unit. When the SLI flag 214 is one and an incorrect-length condition exists for the current DCW, command chaining, if indicated, shall be permitted and the control unit continues execution of the next following DCW. If the SLI flag is zero and an incorrect-length condition exists for a DCW, command chaining, if indicated, is not permitted and the control unit 118 aborts the I/O operation. When an incorrect-length condition exists for a DCW 202 and the SLI flag 214 in the DCW is set one, data transfer is performed as described below, otherwise the operation is terminated and an abnormal termination condition shall be reported for the operation.

For a read DCW 202, if the DCW data count is greater than the amount of data available at the device 116 for the command, then the following applies. If the CC flag 212 is one (chaining is indicated), the data available at the device 116 is transferred to the channel 128 and pad bytes (which are set to zero) are transferred so that the amount of data transferred for the DCW 202 is equal to the DCW data count. All data transferred, including pad bytes, are included in any computation of CRC required for the operation. If the CC flag 212 is zero (chaining is not indicated), the data available at the device shall be transferred to the channel and either no additional data shall be transferred or pad bytes (set to zero) are transferred so that the amount of data transferred for the DCW is equal to the DCW data count. The data transferred plus any pad bytes shall be included in the computation of CRC required for the operation. It is model dependent whether pad bytes are transferred for this case. If the DCW data count is less than the amount of data available at the device for the command, only an amount of data equal to the DCW data count is transferred to the channel 128 for the DCW 202. Only the data transferred shall be included in the computation of CRC required for the operation.

For a write DCW 202, if the DCW data count is greater than the amount of data required by the device 116 for the command, then the following applies. If the CC flag 212 is one (chaining is indicated), the amount of data specified by the DCW data count shall be transferred to the control unit 118. The data transferred shall be used in the computation of any CRC required for the operation, and data not required by the device 116 is discarded. If the CC flag is zero (chaining is not indicated), the amount of data required by the device 116

is transferred to the control unit 118. Additional data is transferred up to the next intermediate CRC word or until the data count for the DCW is exhausted. All data transferred is used in the computation of any CRC required for the operation. If the DCW data count is less than the amount of data required by the device 116 for the command, the amount of data specified by the DCW data count is transferred to the control unit 118. The data transferred is used in the computation of any CRC required for the operation.

A transport-command DCW is a type of DCW that does not specify a specific data transfer command (i.e., customer data that is requested to be transferred by a host computer instruction), but rather specifies a transport command that performs a support function associated with a transport-mode I/O operation. A transport-command DCW may specify control data and may also specify the transfer of transport-command-meta information (TCMI). TCMI includes data that is utilized to control transfer functions in the I/O operation, such as the status of an I/O operation at a device, information regarding additional DCWs that do not fit in or can be accommodated by a current TCCB, and control block checking (CBC) information such as cyclical redundancy check (CRC) data.

When a transport-command DCW specifies the transfer of TCMI to a device, the TCMI is transferred as output data (e.g., in a transport-data IU). The size of the TCMI is command dependent and is, for example, a multiple of 4. The TCMI may be extended by 4 reserved bytes when the size of the TCMI is an even multiple of 4, TIDAWs are used to specify the output storage areas, the insert-CBC flag is one in the last TIDAW used to specify the storage containing the TCMI, the chain-command bit is one in the transport command DCW that specifies the TCMI, and a subsequent DCW specifies the transfer of a TCMI or output data. Note that when a transport command specifies the transfer of a TCMI to a device and the TCMI is extended by 4 reserved bytes because all of the preceding conditions are met, the 4 reserved bytes are not included in the data-count value in the transport command DCW but are included in the following: the count value in the last TIDAW used to specify the storage containing the TCMI, the output-count value in the TCW and the transport-count value in the associated TCCB (for unidirectional data transfers or the write-count value in the associated TCCB for bidirectional transfer).

In one embodiment, TCMI includes interrogate information (via an Interrogate DCW), a CBC-Offset Block (via a Transfer-CBC-Offset Block DCW) and a TCA extension (via a Transfer-TCA Extension DCW).

FIG. 12 illustrates an embodiment of data specified by one type of transport-command DCW, referred to as a Transfer TCA Extension (TTE) command, which is included in a TCA 180 of a TCCB 170 and specifies a TCA extension (TCAX) 220 that is to be transferred to a control unit 118. The TCAX is considered a logical extension of the TCA 180. For some devices 116, the list of DCWs 202 associated with an I/O operation exceeds the number of DCWs that can be accommodated in the TCA 180. In such an instance, a TTE DCW may be included at the end of the TCW 140, which specifies a TCAX 220 that is transferred as output data (e.g., in a data-transport IU). The content of the TTE DCW may include a command code that includes the transfer-TCA extension command (e.g., value of 50 hex)

For example, a TCA 180 is variable in length with a maximum size of 240, and a DCW is 8 bytes. Thus, in this example, a maximum of 30 DCWs may be transmitted in a TCCB 170. However, for DCWs 202 that specify control commands that require control data, the control data immediately follows each DCW 202 in the TCCB. Thus, TCCBs 170 containing such commands are limited to less than 30 DCWs. The TTE DCW may be used to specify additional DCWs needed for the I/O operation that do not fit in the TCA 180.

As shown in FIG. 12, The TCA Extension (TCAX) is a variable-length area that is the logical continuation of the TCA 180 in another transport-command IU or TCCB, and includes a list of DCWs and associated control data. The TCAX contains one or more additional DCWs and associated control data for a TCW I/O operation (in addition to those transferred in an initial TCCB), and can be sent in another TCCB or in a transport-data IU to extend a list of DCWs required for an I/O operation. In one embodiment, the length of the TCAX is an integral multiple of 4.

As shown in FIG. 12, in one embodiment, the TCAX 220 includes a DCW 202 and/or control data for a previous DCW 202. When the last information in the TCAX 220 is a DCW 202, the end of that DCW 202 defines the end of the meaningful information in the TCAX 220. When the last information in the TCAX 220 is control data and the control data

ends on a 4-byte boundary, the end of the control data defines the end of the meaningful information in the TCAX 220. When the last information in the TCAX 220 is control data and the control data does not end on a 4-byte boundary, padding bytes are appended to the control data to reach a 4-byte boundary and the end of the padding bytes define the end of the meaningful information in the TCAX 220. When the size of the meaningful information in the TCAX 220 divided by four is an odd number, there are no reserved bytes; otherwise, the four bytes immediately following the meaningful information in the TCAX 220 are reserved and should contain zeros. When control data for the last DCW 202 in the TCA 180 is contained in the TCAX 220, that control data is the first data in the TCAX 220. Additional DCWs and control data in the TCAX 220 are processed as defined for the TCA 180.

In one embodiment, when a TCOB DCW (described further below) is not present in the TCA 180, the TTE DCW is the first DCW 202 in the TCA 180. When a TCOB DCW is present in the TCA 180, the TTE DCW is the second DCW 202 in the TCA 180. The TTE DCW data count specifies the length of the TCAX 220 and may be an integral multiple of four bytes. The TTE DCW control-data count is zero and the chain command flag in the TTE DCW is set to one. The TTE DCW data does not include the TCAX CRC or TCAX pad bytes.

A TTE command (TTE DCW) should be sent to a device that indicates support for the TTE command and is sent when the TCA length has a certain maximum (e.g., 60 words) and the control data for the last DCW in the TCA extends beyond the last byte in the TCA, or the TCA length is greater than, e.g., 58 words, and one or more additional DCWs are required for the I/O operation. When a COB is provided in the write transport-data IU, the TCAX shall follow the COB CRC. When a COB is not provided, the TCAX shall be sent in the first write transport-data IU of the I/O operation.

Another type of DCW is a Transfer CBC-Offset Block (TCOB) DCW, which is a command that transfers a CBC Offset Block (COB) to a control unit. CBC data refers to control block checking data that is used to verify that the correct amount of input or output data is being transferred in an I/O operation. In one embodiment, the CBC data is cyclical redundancy check (CRC) data, the COB is a CRC Offset Block, and the TCOB DCW is a Transfer CRC-Offset Block DCW. Generally, a cyclic redundancy check (CRC) is an error-detecting code

configured to detect accidental changes to raw data being sent between a channel 128 and a control unit 118. A CRC-enabled device (e.g., the channel 128) calculates a short, fixed-length binary sequence for each block of data to be sent or stored and appends it to the data, forming a codeword or "CRC word". For example, when CRC generation is provided, the channel 128 generates a CRC on the data sent and inserts the value calculated at the end of the last data byte. When a CRC word is received or read, the receiving device (e.g., the control unit 118) either compares its check value with one freshly calculated from the data block, or equivalently, performs a CRC on the whole codeword and compares the resulting check value with an expected residue constant. If the check values do not match, then the block contains an error in the transferred data. Although the CBC data is described in the embodiments herein as CRC data, it is not so limited.

As shown in FIG. 13, a COB 222 specifies the location of intermediate CRC words within the transport data for a write or read data transfer (e.g., in a transport-data IU). In one embodiment, a COB is provided when the first DCW 202 in a TCA 180 contains the TCOB command, i.e., is a TCOB DCW. Intermediate CRC offsets are CRC words located between data in a data transfer IU and located before the end of the IU. The COB 222 contains a list of 1-word values, referred to as CRC Offsets 224, each of which identify the byte offset of each intermediate CRC word in the transport data. For example, the COB is a variable-length control block that includes a list of 4-byte entries, each of which identifies the offset of a CRC specified by a TIDAW to be inserted in the output data.

In the embodiment shown in FIG. 13, the COB 222 includes 1 to N+1 words (CRC Offset words 224) of intermediate-CRC offsets. If the COB 222 is provided in a transport-data IU, then the COB may also includes 0 or 1 words of pad bytes and 1 word of CRC. Pad bytes may be provided so that the word containing the CRC will be on a word boundary that is not on a doubleword boundary. The last word of the COB shall contain a CRC that covers intermediate CRC offsets 0 through N and the pad word, if present. If the COB is provided as control data for the TCOB DCW, then there shall be no COB padding or CRC and the control-data count shall be equal to 4 times the number of intermediate-CRC-offset fields in the COB.

For write operations, the TCOB DCW specifies that a COB is transported to a device 116. The TCOB DCW command code includes the TCOB command which is a value of, e.g., 60

hex. The location of the COB is determined by the control-data count and may be provided in either the first write transport-data IU of an I/O operation or as control data following the TCOB DCW of an I/O operation. For example, when the CD count of the TCOB DCW is not zero, the COB 222 immediately follows the TCOB DCW in the TCA 180 and the CD count specifies the number of CRC offsets in the COB multiplied by 4. When the CD count is zero, the COB 222 is specified as TCMI that is transferred with the output data, and the location of the COB is specified by the output-data-address field 144 in the TCW 140.

Each intermediate-CRC offset 224 in the COB 222 indicates the relative offset, in bytes, from the first byte of data in a transport-data IU. If the COB 222 is provided in a transport-data IU, the first byte of data for the write transfer is either the first byte following the COB CRC if a TCAX 220 is not being provided or the first byte following the TCAX CRC if a TCAX is being provided. If the COB is provided in the TCA 180 as DCW control data, the first byte of data for the write transfer is either the first byte in the transport-data IU if a TCAX 220 is not being provided or the first byte following the TCAX CRC if a TCAX 220 is being provided. When provided in a transport-data IU, the COB is provided in the first write transport-data IU of an I/O operation.

Yet another type of transport-command DCW is an interrogate DCW that is used to determine the state of a TCW I/O operation at a device 116. The interrogate command does not initiate an operation at the device 116, does not affect the state of the device 116 and does not reset allegiances associated with the device 116 and an associated logical path. An exemplary interrogate DCW includes an interrogate command code (e.g., contains a value of 40 hex). With the exceptions of the command code, count and CD-count fields, all other fields in the DCW should contain zeros, otherwise a device-detected program-check condition. If the CD count of the interrogate DCW is greater than zero, interrogate data is specified.

An interrogate operation is performed on a logical path and device 116 that has a TCW I/O operation in progress in order to obtain information regarding the state of the operation at the device 116. In one embodiment, the TCCB 170 for an interrogate operation includes a single DCW with an interrogate device command code (e.g., X'40). The TCAH 178 for an interrogate operation may contain the R bit set to one to specify a read data transfer for the operation. An interrogate operation is recognized at the control unit 118 when the control

unit 118 accepts a transport-command IU that specifies the interrogate command. When an interrogate operation has been recognized at the control unit 118 and the control unit 118 is able to successfully perform the interrogate command, the control unit 118 may provide status in a transport-response IU that includes interrogate extended status describing the state of the logical path and device address specified by the interrogate transport-command IU.

Referring to FIG. 14, an embodiment of a method of performing a transport mode I/O operation 300 is shown. The method includes one or more stages 301-305. In one embodiment, the method includes the execution of all of the stages 301-305 in the order described. However, certain stages may be omitted, stages may be added, or the order of the stages changed.

In stage 301, the host computer (e.g., the OS 110) executes an instruction (e.g., START SUBCHANNEL) that causes an ORB to be transferred to a channel subsystem 114, that specifies a TCW 140. In one embodiment, program-check conditions related to the validity of TCW fields specific to capturing status and transporting a specified TCCB 170 are checked by the channel subsystem 114 to determine whether to transport the TCCB 170 to an I/O device 116. If such a program-check condition exists, the TCCB 170 is not transported and the program-check condition is recognized. For example, the TCCB-address field 148 is checked that it designates an available storage location. If the TCCB-TIDA flag is one, the TCCB-address field 148 is also checked that it designates an address that is on a quadword boundary. Additional program check conditions are further described below.

In stage 302, a TCW I/O operation is initiated with an I/O device (via, for example, a control unit 118) when a channel 128 transfers a transport-command IU 172 that includes a control block, such as a transport-command-control block (TCCB) 170 and associated control information for a TCW 140 to a selected device 116. In one embodiment, information associated with the execution of an I/O operation and the operation of a device (e.g., commands, input data and output data) is transferred between the channel 128 and the control unit 118 as Information Units (IUs). In one embodiment, the IUs are in the form of SB-4 Information Units (IUs).

In one embodiment, IUs or other messages are sent between the channel and the control unit via one or more exchanges. An exchange pair consisting of two unidirectional exchanges,

one used by a channel 128 to send IUs and one used by a control unit 118 to send IUs, are required for all SB-4 link-control functions and for all SB-4 device-level functions that are executed in command mode. A single bi-directional exchange, referred to as a transport exchange, is used for device-level functions executed in transport mode. IUs that a channel 128 sends during the execution of an SB-4 link-control function or the execution of an SB-4 device-level function in command mode are restricted to one exchange, and IUs which a channel receives during the operation are restricted to a different exchange. The exchange on which the channel 128 sends IUs is referred to as the outbound exchange, and the exchange on which the channel 128 receives IUs is referred to as an inbound exchange. When both an outbound exchange and an inbound exchange simultaneously exist between a channel 128 and a control unit 118 for the execution of the same link-level or device-level function, an exchange pair is said to exist, and the control unit 118 is said to be connected to the channel 128. A channel program which is executed in a single connection uses only one exchange pair. If the connection is removed by the closing of the exchanges during the channel program, a new exchange pair is generated to complete the channel program. A channel 128 can initiate an exchange pair by sending an IU which opens a new exchange (or, an initiation IU) as an unsolicited command or unsolicited control information category. A control unit 118 can initiate an exchange pair by sending an initiation IU as an unsolicited control or unsolicited data information category.

UIs, which a channel and control unit send during the execution of an I/O operation that is performed in transport mode are restricted to a single, bi-directional exchange referred to as a transport exchange. A channel 128 opens a transport exchange by sending a transport-command IU as an unsolicited command category (an Initiation IU). A channel 128 may open multiple transport exchanges, each for a different device 116 or for the same device 116 on different logical paths. A new transport exchange may be opened for a specific device on a logical path when a transport exchange already exists for that device and logical path to perform an interrogate operation; otherwise, the channel 128 waits for the existing transport exchange or exchange pair to be closed before initiating a new transport exchange for the device on the logical path.

In stage 303, the control unit 118 receives the TCCB 170 and transport-command IU 172, and may accept the TCCB 170 if certain conditions are met. When a transport-command IU

172 is accepted, the TCCB 170 is considered current at the control unit 118 until execution of the TCCB 170 at the control unit 118 is considered completed. If the transport-command IU 172 is not accepted because of an error condition, a transport-response IU is returned to the channel 128 with an initial status flag set to one to indicate that the error occurred prior to initiating execution of the TCCB 170 at the device 116. The control unit 118 may provide error information in status and I/O extended status fields in the transport-response IU (described below) to identify the transport-command IU error. In one embodiment, the exemplary conditions described below should be satisfied in order for the transport-command IU 172 to be accepted at the control unit 118:

1) The transport-command IU satisfies the following integrity checks:

a) For control units that do not support bidirectional operations, the L1 field in the TCH plus 8 shall specify an amount of data that is equal to the amount of data received by the control unit for the transport-command IU, otherwise, a transport-command IU integrity error due to a data count error shall be recognized;

b) For control units that support bidirectional operations, the L1 field in the TCH plus either 8 or 9 shall specify an amount of data that is equal to the amount of data received by the control unit for the transport-command IU, otherwise, a transport-command IU integrity error due to a data count error shall be recognized; and

c) The LRC field in the TCCB shall be valid, otherwise a transport-command IU integrity error due to invalid LRC error shall be recognized.

2) The specified logical path shall be established, otherwise a logical-path-not-established error shall be recognized;

3) For those commands that require the device to be installed and ready, the device address shall specify a device that is installed and ready, otherwise an address-exception condition shall be recognized;

4) The R and W bits shall not both be set to one in the TCH if bidirectional data transfer is not supported by the control unit or the transport-command IU does not contain the BRDL field; otherwise a TCH content error shall be recognized;

5) The TCCB in the transport-command IU shall satisfy all of the following conditions, otherwise, a TCCB content error shall be recognized:

a) The L2 field shall specify a length that is exactly 8 bytes greater than the L1 field and contain a value that is at least 20 bytes and not more than 252 bytes;

b) Byte 1 of word 0 of the TCAH shall be zero;

c) The format control field in the TCAH shall be equal to hex'7F';

d) The service-action code in the TCAH shall contain a valid value (see 8.13.4.4);

e) If the R and W bits are both set to zero, the data-length (DL) field shall be zero; and

6) If another TCW I/O operation is in progress for the logical path and device address specified in the transport-command IU, the service-action code shall specify hex'1FFF' and the command code in the first DCW of the TCA shall specify the interrogate command code, otherwise a non-interrogate-second-operation error shall be recognized.

In one embodiment, the channel 128 considers a connection to exist when the transport-command IU 172 is sent, and the control unit 118 considers a connection to exist when the transport-command IU 172 is accepted. The channel 128 is thus not aware of whether the control unit 118 has accepted the TCCB 170 or of the progress of the I/O operation at the device 116 until the I/O operation is terminated by the control unit 118 with a transport-response IU. In one embodiment, the channel 128 can set a time window for which to receive a response from the control unit, so that if the channel does not receive a transport-response IU within the window, the channel recognizes a transport command timeout.

In stage 304, when the TCCB 170 has been accepted, the control unit 118 processes the TCA 180 and executes each DCW 202. The DCW 202 being executed is recognized as the current DCW. A DCW 202 becomes current when it is the first DCW of a transport mode program (i.e., the first DCW in a TCCB) and has been selected by the I/O device 116 for execution or when, during command chaining, a subsequent DCW 202 takes over control of the I/O operation. The first DCW to be executed may be considered to be located at offset zero of the TCA 180 in the TCCB 170. Each additional DCW in the channel program is

located in the TCA 180 (or in a TCAX 220) and is used when the DCW is needed by the I/O device 116.

Command chaining is controlled by the DCW chain command (CC) flag in the DCW. This flag specifies the action to be taken upon the exhaustion of the current DCW. Chaining takes place between successive DCWs within the TCA. When a TCAX is specified, chaining also takes place between the last DCW in the TCA and the first DCW in the TCAX, and between successive DCWs within the TCAX. When the current DCW specifies command chaining and no unusual conditions have been detected during the operation, the completion of the current DCW causes the next DCW to become the current DCW and be executed by the device. If a TCAX is specified and the offset of the next DCW is past the end of the TCA, chaining continues with the first DCW of the TCAX whose TCAX offset is determined by subtracting the TCA length from the calculated offset. Thus, command chaining proceeds in ascending order of TCA offsets, and then TCAX offsets when a TCAX is specified. If a condition such as attention, unit check, unit exception, or incorrect length has occurred (unless a SLI field 214 in the DCW is activated), the sequence of operations is concluded, and the status associated with the current operation causes an interruption condition to be generated. The new DCW in this case is not executed.

Data that is requested to be transferred via the I/O operation are sent between the channel 128 and control unit 118 via one or more transport-data IUs during processing of the DCWs 202 in a TCCB 170. A read operation performs only a read data transfer, a write operation performs only a write data transfer and a bi-directional operation may perform both a read and write data transfer.

Referring to FIG. 15, for a write or bidirectional operation, the channel 128 sends one or more transport-data IUs 310 to the control unit 118 to transfer write data 312 for the operation. For a read or bidirectional operation, the control unit 118 sends at least one transport-data IU 314 to the channel 128 to transfer read data 316 specified for the operation. Pad bytes 318 may be included in the last word of a data area for which CRC is to be calculated and are used to pad the data area to the next word boundary when the data area to be covered is not an integral number of data words. This applies to data areas to be covered by COB, intermediate and final CRC. The value used for a pad byte is model dependent.

A final CRC field 320 is included in a transport-data IU 310, 314 when the transport-data IU is the last transport-data IU sent by the channel 128 or control unit 118 for a TCW I/O operation. In one embodiment, the final-CRC field 320 includes a word-aligned 32-bit redundancy-check code. For read or write operations, the DL field in the TCCB 170 specifies the amount of data to be transferred during an operation, which may includes all required pad and CRC bytes. For bidirectional operations, the DL field in the TCCB 170 specifies the amount of data to be transferred for the write data transfer portion of the operation and the BRDL field specifies the amount of data to be transferred for the read data transfer portion of the operation. These amounts may include all required pad and CRC bytes.

During a write or bi-directional data transfer, output data is transferred in one or more transport-data IUs 310 to the control unit 118 on the transport exchange associated with the TCW I/O operation. In one embodiment, when first-transfer-ready is disabled, except for the first write transport-data IU 310 of an operation, the channel 128 requires a transfer-ready IU from the control unit 118 prior to sending each transport-data IU 310. The control unit 118 may request additional data by sending additional transfer-ready IUs until it has requested all the data specified by the DL field 186 of the TCCB 170 for the write operation. For a write operation, the next IU sent by the control unit 118 following completion of the data transfer specified by the DL field in the TCCB is a transport-response IU. For a bidirectional operation, the next IU sent by the control unit 118 following completion of the data transfer specified by the DL field in the TCCB may be a transport data IU or a transport-response IU.

During a read data transfer, data is transferred in one or more transport-data IUs 314 to the channel 128 on the transport exchange associated with the TCW I/O operation. The amount of data sent in each transport-data IU is determined by the control unit 118 and may be set to any value as long that the total amount of data transferred in all transport-data IUs 314 for the operation does not exceed the value in the DL field 186 or, for bidirectional operations, the value in the BRDL field 188. For read operations, if the quantity of data sent by the control unit is less than the DL field in the TCCB, the DL residual count in the transport-response IU shall be the difference between the quantity of data sent by the CU and the DL field in the TCCB. The channel recognizes a device-level protocol error if the residual count

provided by the control unit in the transport-response IU does not match the difference between the DL and the number of bytes actually received by the channel.

For write and bi-directional operations, the DL residual count in the transport-response IU shall be the difference between the quantity of data sent by the channel and the DL field in the TCCB. The channel recognizes a device-level protocol error if the DL residual count provided by the control unit in the transport-response IU does not match the difference between the DL and the number of bytes actually sent to the channel. For bidirectional operations, if the quantity of data sent by the control unit is less than the BRDL field in the TCCB, the BRDL-residual count in the transport-response IU shall be the difference between the quantity of data sent by the CU and the BRDL field in the TCCB. The channel recognizes a device-level protocol error if the BRDL-residual count provided by the control unit in the transport-response IU does not match the difference between the BRDL and the number of bytes actually received by the channel.

For a write or bidirectional operation, when first-transfer-ready is disabled, the channel sends a transport-data IU immediately following the transport-command IU. For all transport-data IUs from the channel after the first transport-data IU and for all transport-data IUs from the channel when first-transfer-ready is not disabled, the channel shall send a write transport-data-IU only after receiving a transfer-ready IU. The channel shall send a transport-data IU after each transfer-ready IU is received until all write data specified by the TCCB is transferred or a transport-response IU is received.

For a read or bidirectional operation, the control unit sends at least one transport-data IU to the channel to transfer the read data specified for the operation.

For bidirectional operations, when first-transfer-ready disabled is not in effect, the control unit selects the first transport-data IU to be transferred. The control unit either sends a transfer-ready IU to the channel to request a transport-data IU or sends a transport-data IU to the channel. If first-transfer-ready disabled is in effect for the operation, the channel send a transport-data IU following the transport-command IU.

For bidirectional operations, both write and read data transfer may be performed. Write data transfer and read data transfer may be performed as described above. Except for the first transport-data IU when first-transfer-ready disabled is in effect, the transfer of data between

the channel and control unit is controlled by the control unit by either sending a read transport-data IU or by sending a transfer-ready IU to request a write transport-data IU.

The order in which data that has been received in a transport-data IU is stored to host storage in relation to the order in which data is fetched from host storage to send in a transport-data IU may be unpredictable for a bi-directional operation.

When the channel transfers a first transport-data IU when first-transfer-ready disabled is in effect or when the control unit requests a transport-data IU from the channel by sending a transfer-ready IU, the data sent by the channel in the transport-data IU may be for write DCWs that are executed subsequent to the execution of read DCWs in the TCA.

As shown in FIG. 15, one or more intermediate CRC words 320 may be present in a transport-data IU 310 sent by the channel 128. The intermediate CRC 320 words provide CRC checking of data areas prior to transmission of the entire data area specified by the TCA 180. A COB CRC word 322 may be present in a transport-data IU 310 when a COB 222 is being transferred in the transport-data IU.

Referring again to FIG. 14, in stage 305, the TCW I/O operation is ended by either the channel 128 or the control unit 118. The channel 128 may initiate the ending of a TCW I/O operation as the result of an abnormal condition or a program-initiated termination. The control unit 128 may initiate the ending of the operation as the result of the completion of the execution of the operation or as the result of an abnormal condition detected during the execution of the command. The control unit 128 may initiate the end of a TCW I/O operation by sending a transport-response IU or, in the case of certain errors, by aborting the exchange. In one embodiment, the control unit 118 may initiate the ending of a TCW I/O operation under any of the following circumstances: all of the DCWs 202 in the TCA 180 (and a TCAX if specified) having been executed; an incorrect-length condition having been detected for a DCW when the DCW-incorrect length facility is supported and the SLI flag 214 is set to zero; a unit check condition has been detected; an abnormal condition such as a transport error; or another error was recognized that required the exchange to be aborted.

FIGS. 16-18 illustrate an exemplary transport response IU 330 that may be sent by a control unit 118. The Transport Response IU 330 provides status for a TCW I/O operation, which may include a normal ending status or, when an abnormal condition has been detected,

termination status that indicates the cause for abnormal termination of the operation. The transport response IU 330 may also include an extended status field that provides further status for the operation. A transport-response IU may or may not close a transport exchange. In one embodiment, whether or not the exchange has been closed by a transport-response IU is indicated in the FC-FS-3 header. If the transport exchange has not been closed by the transport-response IU, the channel may send a transport-confirm IU that closes the exchange after receiving the transport-response IU.

In one embodiment, the transport-response IU 330 includes a SB-4 header 332 followed by a status field 334, a status LRC 344, and an optional extended-status field 346 containing from, e.g., 32 to 64 bytes. When extended status is provided, a 4-byte extended-status LRC field 340 may be provided as the last word of the transport-response IU 330. Pad bytes are added to the extended status to round to the next word boundary if the number of extended-status bytes is not on a word boundary. The SB-4 header 332 has a format similar to that of the transport command IU and is set equal to the SB-4 header in the transport command IU for this exchange.

Referring to FIG. 17, an embodiment of the status area 334 is, e.g., 20 bytes and contains information about the TCW I/O operation. A "status flags 1" field 336 includes one or more exception codes that are set by the control unit 118 to report an abnormal condition detected during a TCW I/O operation. Exemplary codes include:

0 - Device-level exception due to an address-exception condition;.

2 - Link-level reject due to a logical-path-not-established condition;

3 - Resetting event notification - a resetting event has occurred on the logical path and the device associated with the transport-command IU. When this code is set in the transport-response IU, the control unit requests status confirmation for the status. If status confirmation is received, the resetting event condition is reset at the device for the logical path; otherwise, the resetting-event condition remains pending;

4 - Device-detected program check/IFCC - the control unit has detected a condition that may result in a program check or IFCC to be reported. Errors that fall into this category include errors that indicate the transport-command IU arrived in a corrupted state (e.g., TCCB

integrity error), invalid CRC detected for write data, and receipt of a second I/O operation for a logical path and device address that is not an interrogate operation.

5 - Device-detected program check - the control unit has detected an error in the content of the TCH

In one embodiment, the status flags 1 field 336 includes an Incorrect Length (IL) flag 338. When the DCW-incorrect-length facility is supported by the channel and control unit, bit 0 is the IL flag and, when set to one, shall indicate that the TCW I/O operation was terminated due to an incorrect-length condition for the DCW indicated by the DCW offset. An incorrect length is detected by the control unit 118 if the DCW data count does not match the amount of data required by the device for a write DCW or if the DCW data count does not match the amount of data available at the device for a read DCW.

In one embodiment, if the DCW 202 includes an SLI flag 214, the IL flag 338 is set to one only when the SLI flag 214 is zero and when the device status includes channel-end status without unit-check status. When the IL flag 338 is set to one in a transport-response IU, the DCW offset identifies the DCW containing the incorrect-length condition and the DCW residual count indicates the amount of data transfer, if any, that was transferred for the incorrect-length DCW. Data transfer, if any, for DCWs that preceded the incorrect-length DCW in the TCA shall have completed and all CRC checking on transferred data shall have been performed as described below.

When the DCW containing the incorrect-length condition is a read DCW, the last transport-data IU sent to the channel INCLUDES CRC for all read data transferred to the channel during the TCW I/O operation. If write DCWs preceded the incorrect-length read DCW in the TCA, the data for those write DCWs shall have been received by the control unit and CRC checking shall have been performed on the data. Data for write DCWs in the TCA that follow the incorrect-length read DCW shall be transferred to the control unit as necessary to obtain the CRC for the write data and to perform the CRC checking.

The incorrect-length condition is not recognized by an interrogate command, a transfer-CRC-offset command or a TTE command regardless of whether the incorrect length facility is supported by the channel and the control unit.

The transport response IU 330 may also include a Data Length Residual Count (DLRC) field 340. For write and bidirectional operations, the DL-residual count is a 32-bit, unsigned binary integer that specifies the difference between the DL field in the transport-command IU and the number of bytes actually received from the channel. For read operations, the DL-residual count is a 32-bit, unsigned binary integer that specifies the difference between the DL field in the transport-command IU and the number of bytes actually sent to the channel.

For bidirectional operations, a BRDL residual count field 342 may also be included in the transport-response IU 330. For bidirectional operations, the BRDL-residual count is a 32-bit, unsigned binary integer that specifies the difference between the BRDL field in the transport-command IU and the number of bytes actually sent to the channel.

A Status Flags3 field 344 can be used by the control unit 118 to provide additional information about the associated transport-mode operation. This field may include an Extended Status Sent (ESS) bit to indicate that extended status, including possible sense data, has been sent in the transport-response IU. The extended status includes extended status (ES) Flags that include fields such as an ES Type Code field. Type codes include I/O status (the extended-status area contains valid ending status for the transport-mode I/O operation), I/O Exception (the extended-status area contains information regarding termination of the transport-mode I/O operation due to an exception condition), and Interrogate status (the extended-status area contains status for an interrogate operation).

When the ES-type code in the ES Flags is an I/O exception, the ES includes reason codes (RC). Exemplary reason codes include:

1 - TCCB integrity error: the control unit has determined that the TCCB arrived in a corrupted state (provided when exception code 4 is indicated in the status flags 1 field);

2 - Invalid CRC detected: invalid CRC was detected on received data (provided when exception code 4 is indicated in the status flags 1 field);

3 - Incorrect TCCB length specification (provided when exception code 5 is indicated in the status flags 1 field);

4 - TCAH specification error (provided when exception code 5 is indicated in the status flags 1 field);

5 - DCW specification error: there is an error with the DCW designated by the DCW-offset field in the extended status (provided when exception code 5 is indicated in the status flags 1 field);

6 - Transfer-direction specification error: the command specified by the DCW designated by the DCW-offset field in the extended status specifies a direction of data transfer that disagrees with the transfer direction specified in the TCH or both the R and W bits are set to one in the TCH and bidirectional data transfer is not supported by the control unit (provided when exception code 5 is indicated in the status flags 1 field);

7 - Transport-count specification error (provided when exception code 5 is indicated in the status flags 1 field);

8 - Two I/O operations active: While an I/O operation is active at the device a second non-interrogate TCCB has been transported to the device for execution. The RCQ field has no meaning. This reason code is provided when exception code 4 is indicated in the status flags 1 field; and

9 - One or more entries in the CRC-offset block indicate that an intermediate CRC is not at a location that is appropriate for the device and/or command being executed provided when exception code 4 is indicated in the status flags 1 field.

The Extended Status may also provide a Reason Code Qualifier (RCQ) that provides additional information about the reason(s) for the I/O exception. Different reason code qualifier types may be used for different types of errors, such as TCCB integrity errors, Output Data CRC Errors, Incorrect TCCB Length Specifications, TCAH Specification Errors and DCW Specification Errors. Exemplary RCQs for a TCCB integrity error include:

0 - No additional information;

1 - Data Count Error - the amount of data transferred for the transport-command IU is not equivalent to the amount of data specified by the L1 field plus 8 in the TCH for control units

that do not support bidirectional operations or is not equivalent to the amount specified by the L1 field plus either 8 or 9 for control units that do support bidirectional operations; and

2 LRC error - the LRC on the transport-command IU is invalid.

Exemplary RCQs for a DCW specification error include:

1 - Reserved-field specification error: A reserved field in the DCW that is required to contain zeros contains a non-zero value;

2 - Flags-field command-chaining specification error: The command-chaining bit is one and the offset of the next DCW is such that all or part of the next DCW extends past the end of the TCA, or the command-chaining bit is zero and more than 3 unused bytes remain in the TCA;

3 - Control-data count specification error: The CD count specifies control data past the end of the TCA;.

4 - TCOB DCW location error: The first TCOB DCW is not the first DCW in the TCA;

5 - TCOB DCW duplication error: More than one TCOB DCW is specified in the TCA;

6 - TCOB DCW multiple-count specification error: Both the CD count and the DCW data count are either zero or non-zero;

7 - TCOB DCW direction error: A TCOB DCW is specified in the TCA and the W bit in the TCH is zero;

8 - TCOB DCW chaining error: The chain-command bit in the TCOB DCW is zero;

9 - TCOB count-specification error: A TCOB DCW has a nonzero CD count or a data count that is not a multiple of 4;

10 - TTE DCW location error: A TCOB DCW was not specified and a TTE DCW was encountered that was not the first DCW in the TCA or a TCOB DCW was specified and the first TTE DCW encountered was not the second DCW in the TCA;

11 - TTE DCW duplication error: More than one TTE DCW was encountered in the TCA;

12 - TTE DCW CD-count specification error: The control-data count in the TTE DCW specifies a value that is not zero;

13 - TTE DCW data-count specification error: The data count in the TTE DCW specifies a value that is less than 8 or a value that is not a multiple of 4;

14 - TTE DCW direction error: A TTE DCW is specified and the W bit in the TCH is zero;

15 - TTE DCW chaining error: The chain-command bit in the TTE DCW is zero; and

16 - TCAX specification error: A TTE DCW is specified and either of the following is true:

The length of the TCA is 58 words or less, or

The length of the TCA is 59 words, the last DCW in the TCA specifies control data and the length of the control data extends beyond 59 words, or

The length of the TCA is the maximum length of 60 words and the last word in the TCA contains the first word of a DCW.

Exemplary RCQs for a Transfer-Direction Specification Error include:

1 - Read-direction specification error: The DCW specifies an input operation and the R bit in the TCH is zero;

2 - Write-direction specification error: The DCW specifies an output operation and the W bit in the TCH is zero. Note - A DCW specification is recognized when a TCOB or TTE DCW is specified and the W bit in the TCH is zero;

3 - Read-Write conflict: Both the R and W bits in the TCH are one and the control unit does not support bidirectional operations or the control unit supports bidirectional operations but the transport-command IU did not contain the BRDL field, or both the R and W bits are not one and the transport-command IU contains the BRDL field;

Exemplary RCQs for a Transport-Count Specification Error include:

1 - Read-count specification error: for read operations, the DL field in the TCCB specifies a value that is not equivalent to the total count of data bytes specified by the DCWs in the

TCA plus pad bytes and CRC; for bidirectional operations, the BRDL field in the TCCB specifies a value that is not equivalent to the total count of data bytes specified by the read DCWs in the TCA plus pad bytes and CRC; and

2 - Write-count specification error: The DL field in the TCCB specifies a value that is not equivalent to the total count of data bytes specified by the write DCWs in the TCA plus intermediate pad bytes, intermediate CRC, final pad bytes and final CRC bytes, and if a COB is included in the first transport-data IU of the write data transfer, the count includes the COB, any COB pad bytes and the COB CRC bytes. If a TTE DCW is present in the TCA, the count also includes the TCAX and TCAX CRC bytes.

A program check occurs when programming errors are detected by the channel subsystem. For a transport mode operation, programming errors may also be detected by the I/O device 116 and are reported as program checks. A program check condition can be due to any of the following reasons:

Invalid TCW Specification: When any of the following conditions is detected, an invalid TCW specification is recognized:

1. A reserved field that is checked for zeros in the TCW does not contain zeros.

2. A non-zero value is specified in the TCW format field.

3. The read and write bits in the TCW are both one, bit 10 of the TCW flags field is zero, and either the FCX-bidirectional-data-transfer facility is not installed or the specified device does not support bi-directional data transfers.

4. The bit 10 of the TCW flags field is zero and the TCCB-length field in a TCW specifies a length that is less than 12 or greater than 244.

5. Bit 10 of the TCW flags field is one and the write operations (W) bit of the TCW is zero, the read operations (R) bit of the TCW is zero, or both are zero.

6. Bit 10 of the TCW flags field is one and the designated subchannel is not associated with a control unit that has been configured to transport fibre-channel-services requests.

When a TTE DCW is used, a device-detected program check is recognized when any of the following conditions exist:

1. When a TCOB DCW is not specified, the TTE DCW is not the first DCW in the TCA. When a TCOB DCW is specified, the TTE DCW is not the second DCW in the TCA.

2. The TTE DCW is specified and a write operation is not specified (that is, the W-bit in the TCW is zero).

3. The chain-command flag in the TTE DCW is zero.

4. More than one TTE DCW is specified

5. The control-data-count field in the TTE DCW does not contain zero.

6. The count field contains less than 8 or a value that is not a multiple of 4.

7. Any of the following are true for the TCA:

The TCA does not contain at least one DCW that is not a transport-command DCW.

The TCA contains one or more DCWs that are not transport-command DCWs and the chain-command flag in the last DCW of the TCA is zero.

When a TTE DCW is used and additional TCMI and/or output data follows the TCAX, the following should be true; otherwise, a device-detected program check may be recognized:

The output-TIDA flag (flags bit 7) in the TCW must be one; or

When TIDAWs are used for the transfer of a TCAX and for the transfer of data, the insert-CBC control must be set to one in the last or only TIDAW that is used to transfer the TCAX. When TIDAWs are used for the transfer of only the TCAX, it is not necessary to set the insert-CBC control in the last or only TIDAW.

During the processing of a TCA, a TCCB content error may be recognized when any of the following conditions are detected:

1. The first TCOB DCW encountered is not the first DCW in the TCA;

2. a second TCOB DCW is encountered in the TCA;

3. a TCOB DCW is encountered in the TCA when the W bit in the TCAH is zero;

4. a TCOB DCW does not have the chaining flag set;

5. a TCOB DCW is encountered in which both the CD count and data count are zero;

6. a TCOB DCW is encountered in which both the CD count and data count are non-zero;

7. a TCOB DCW is encountered in which the CD count is zero and the data count is not a multiple of 4;

8. a TCOB DCW is encountered in which the data count is zero and the CD count is not a multiple of 4;

9. A TCOB DCW is not specified and the TTE DCW is not the first DCW in the TCA or a TCOB is specified and the TTE DCW is not the second DCW in the TCA;

10. A TTE DCW is encountered in the TCA when the W bit in the TCAH is zero;

11. A TTE DCW does not have the chaining flag set;

12. A second TTE DCW is encountered in the TCA;

13. A TTE DCW is encountered in which the CD count is non-zero;

14. A TTE DCW is encountered in which the data count is zero or is not a multiple of 4;

15. A TTE DCW is specified and any of the following is true: the length of the TCA is 58 words or less; the length of the TCA is 59 words, the last DCW in the TCA specifies control data and the length of the control data extends beyond 59 words; or the length of the TCA is the maximum length of 60 words and the last word in the TCA contains the first word of a DCW;

16. A DCW contains a command that requires control data and the CD count field is zero or contains a value that specifies data past the end of the TCA;

17. A DCW contains a CD count field that is non-zero and the command does not allow control data to be provided;

18. During a write operation, the device encountered a command in the TCA that attempted to perform a read data transfer;

19. During a read operation, the device encountered a command in the TCA that attempted to perform a write data transfer;

20. The CC bit is zero in a DCW and more than 3 unused bytes remain in the TCA;

21. The CC bit is one in a DCW and next DCW location is determined to be less than 8 bytes from the end of the TCA indicating and a TCAX has not been provided, or a TCAX has been provided and the next DCW location is determined to be less than 8 bytes from the end of the TCAX;

22. For a read or bidirectional operation, when a read DCW in the TCA becomes current, the sum of the data counts for all previous read DCWs and the current DCW plus all pad and CRC bytes exceeds the DL field in the TCCB or, for bidirectional operations, exceeds the BRDL field in the TCCB;

23. For a read or bidirectional operation, when the last DCW in the TCA becomes current, the sum of the data counts for all read DCWs plus all pad and CRC bytes is not equal to the DL field in the TCCB or, for bidirectional operations, is not equal to the BRDL field in the TCCB;

24. For a write or bidirectional operation, when a DCW in the TCA becomes current, the sum of the data counts for all write DCWs plus all pad and CRC bytes (including all intermediate pad and CRC bytes), and if a COB is included in the transport-data IU, the COB, COB pad and COB CRC bytes, and if a TCAX is present, the TCAX, TCAX pad, and TCAX CRC bytes, exceeds the DL field in the TCCB; or

25. For a write operation or bidirectional operation, when the last DCW in the TCA becomes current, the sum of the data counts for all write DCWs plus all pad and CRC bytes (including all intermediate pad and CRC bytes), and if a COB is included in the transport-data IU, the

COB, COB pad and COB CRC bytes, and if a TCAX is present, the TCAX, TCAX pad, and TCAX CRC bytes, is not equal to the DL field in the TCCB.

In one embodiment, prior to performing command mode or transport mode operations, links between the channel subsystem 114 and control units 118 and/or devices 116 are established. Each channel 128 in the I/O system that has a physical connection with one or more control units can be referred to as a "N_Port" that has a unique identifier (a "N_Port_ID"). Likewise, each control unit 118 can also be referred to an a N_Port having an associated N_Port_ID. Both a channel 128 and a control unit 118 may include multiple channel images or control unit images, respectively. Each N_Port includes an address identifier that is assigned during initialization and the performance of a login procedure. Explicit N_Port login may be performed by means of the N_Port login (PLOGI) extended link service during the initialization process. When an N_Port has performed N_Port login with another N_Port, that N_Port is said to be logged in with the other N_Port.

In one embodiment, link initialization between the channel subsystem and control units is performed using the process login (PRLI) extended link service (ELS) protocol. General aspects of the PRLI ELS, including the format of the PRLI ELS request and response, are given in "Fibre Channel: Link Services (FC-LS-2)," T11 Project 2103-D, Revision 2.00, June 26, 2008, which is hereby incorporated herein by reference in its entirety, with specific settings for FC-SB-4 defined in the sections below.

During a PRLI procedure, a channel 128 that supports PRLI sends a request to each control unit 118 in its configuration that also supports the process login ELS to determine whether the control unit 118 supports transport-mode operations. In one embodiment, the PRLI request is sent during channel initialization prior to establishing logical paths and may also be sent as the result of a process logout (PRLO). The PRLI ELS is used to exchange process login service parameters between a channel 128 and control unit 118. A PRLI ELS request may be sent by a channel to a control unit when logical paths are established with the control unit and applies to all the established logical paths. Parameters may be exchanged between a channel and control unit via a PRLI request and a PRLI response.

In one embodiment, if a control unit receives a PRLI ELS request that would modify the service parameters for a process login already in effect with the channel, the control unit

responds to the PRLI ELS request with a link service reject (LS_RJT). The LS_RJT response is sent with reason code hex'09', "unable to perform command request at this time". After the control unit sends the LS_RJT, the control unit shall send a process logout (PRLO) ELS request to the channel to allow a retry of the PRLI ELS request from the channel to be successfully performed.

Referring to FIG. 18, the PRLI request includes a service parameter page, which includes various fields used to establish service parameters. For example, the service parameter page 400 may include a Type Code field 402 that identifies a transport mode or FC-4 protocol (e.g., is set to hex'1B' to indicate the SB-4 protocol). A type code extension field 404 (e.g., byte 1 of word 0) is set to zero. A FC-LS-2 Flags field 406 (e.g., bits 16-19 of word 0) may be set by the channel as follows:

Bit 16 - Originator Process Associator Validity: this bit is set to zero by the channel and ignored by the control unit;

Bit 17 - Responder Process Associator Validity: this bit set to zero by the channel and ignored by the control unit;

Bit 18 - Establish Image Pair: this bit is set to zero by the channel and ignored by the control unit; and

Bit 19 - Reserved: this bit is set to zero by the channel and ignored by the control unit.

The service parameter page may include additional fields such as an Originator Process Associator field 408 (e.g., word 1) which is set to zero by the channel, and a Responder Process Associator field 410 (e.g., word 2). A Maximum Initiation Delay Time field 412 (e.g., byte 0 of word 3) contains a binary integer that specifies in units of seconds the maximum value that the control unit can set as the initiation-delay time in a process logout (PRLO) request. A PRLO request is used by the channel or the control unit to terminate a transport mode operation between the channel and the control unit. The Maximum Initiation Delay Time value is generally based on the amount of time that the channel can hold off initiating new I/O operations without disrupting the host system.

A FC-SB-4 Flags field 414 (e.g., Byte 3 of word 3 of the PRLI request page) includes one or more flags described below:

0 - Transport Mode Supported. When bit 0 is set to zero, the channel does not support transport-mode operations. When bit 0 is set to one, the channel supports transport-mode operations.

1 - Reserved.

2 - DCW Incorrect Length (IL) Facility Supported. For example, when an IL flag is active (e.g., when bit 2 is set to one), the DCW IL facility is supported by the channel 128. When bit 2 is set to zero, the DCW IL facility is not supported by the channel 128. The DCW IL facility includes support for the suppress-length-indicator (SLI) flag 214 and the incorrect-length flag 338. When the transport mode-supported bit (bit 0) is equal to zero, bit 2 shall be set to zero.

5 - Bidirectional Data Transfer Supported. When a bi-directional transfer indication is active (e.g., bit 5 is set to one) bidirectional data transfer is supported by the channel 128. When bit 5 is set to zero, bidirectional data transfer is not supported by the channel. This bit is only meaningful when the transport mode-supported bit is equal to one.

7 - First Transfer Ready Disabled Supported. When bit 7 is set to one, first-transfer-ready-disabled operation is supported by the channel. When bit 7 is set to zero, first-transfer-ready disabled operation is not supported by the channel. When the transport-mode-supported bit (bit 0) is equal to zero, bit 7 shall be set to zero.

When both the channel and control unit indicate support for first-transfer-ready disabled, then write data transfer in transport-mode is performed with first-transfer-ready disabled in effect, so that the channel can send a first transport-data IU to the control unit without having to wait for a transfer-ready IU from the control unit. If either the channel or control unit indicate they do not support first-transfer-ready disabled then first-transfer-ready disabled is not in effect for transport mode operations.

Referring to FIG. 19, in one embodiment, the PRLI response includes a PRLI Accept Service Parameter Page 420, which may include fields such as a Type Code field 402 that

identifies the transport mode or FC-4 protocol (e.g., is set to hex'1B' to indicate the SB-4 protocol), and a type code extension field 404 (e.g., byte 1 of word 0) that is set to zero. A FC-LS-2 Flags field 406 (e.g., bits 16-19 of word 0) may be set by the channel as follows:

Bit 16 - Originator Process Associator Validity: this bit is set to zero by the control unit and ignored by the channel;

Bit 17 - Responder Process Associator Validity: this bit set to zero by the control unit and ignored by the channel;

Bit 18 - Establish Image Pair: this bit is set to zero by the control unit and ignored by the channel; and

Bit 19 - Reserved: this bit is set to zero by the control unit and ignored by the channel.

A Response Code field 422 (e.g., bits 20-23 of word 0) contains a binary integer indicating the result of the PRLI Request. The meanings of the response codes values are defined by FC-LS-2. A First Burst Size field 424 (e.g., bytes 0-1 of word 3) contains a binary integer that specifies the maximum amount of data (e.g., in units of 4k bytes) that are allowed to be sent in the first transport-data IU for a write data transfer when first-transfer-ready disabled is supported by both the channel and control unit. A value of zero indicates that there is no first burst size limit. The first-burst size field 424 may be implemented by all control units that support first-transfer-ready disabled.

The PRLI Accept Service Parameter Page 420 may also include a FC-SB-4 Flags field 414 (e.g., byte 3 of word 3) that includes one or more flags defined described below:

0 - Transport Mode Supported. When bit 0 is set to zero, the control unit 118 does not support transport-mode operations. When bit 0 is set to one, the control unit 118 supports transport-mode operations.

2 - DCW Incorrect Length (IL) Facility Supported. When bit 2 is set to one, the DCW-incorrect-length (IL) facility is supported by the control unit 118. When bit 2 is set to zero, the DCW IL facility is not supported by the control unit 118. The DCW-incorrect-length

facility includes support for the SLI flag 214 and the incorrect-length flag 338. When the transport-mode-supported bit (bit 0) is equal to zero, bit 2 shall be set to zero.

5 - Bidirectional Data Transfer Supported. When a bi-directional data transfer indication is active (e.g., bit 5 is set to one), bidirectional data transfer is supported by the control unit 118. When bit 5 is set to zero, bidirectional data transfer is not supported by the control unit 118. This bit is only meaningful when the transport-mode-supported bit is equal to one.

7 - First Transfer Ready Disabled Supported. When bit 7 is set to one, first-transfer-ready-disabled is supported by the control unit. When bit 7 is set to zero, first-transfer-ready-disabled operation is not supported by the control unit. When the transport-mode-supported bit (bit 0) is equal to zero, bit 7 shall be set to zero.

Thus, the channel subsystem and the control unit may use login messages such as PRLI message to provide indications to one another regarding their respective support for bi-directional data transfers. The channel subsystem and/or the control unit may also provide indications to the host system (e.g., the OS 110). For example, an indication can be provided from the control unit to the OS that the control unit supports bi-directional data transfers. In one embodiment, the channel subsystem provides an indication of channel subsystem and/or control unit capabilities, including the capability to support bi-directional data transfers. For example, the channel subsystem may provide capability information in response to an instruction from the OS.

An example of capability indication is shown in FIGS. 20 and 21. As shown in FIG. 20, the channel subsystem may send a channel path description block 500 to the OS in response to an OS instruction (e.g., a STORE CHANNEL PATH DESCRIPTION instruction). The channel path description block 500 includes fields such as a channel path identifier (CHPID) field 502, a descriptor 504 and a channel description specific data (CDSD) field 506. Referring to FIG. 21, an exemplary CDSD field 506 is a word (e.g., word 5) of the channel path description block 500, and includes a maximum data length (e.g., maximum data length supported for a TCW/TCCB) field 508. For example, bits 0-15 of the CDSD field 506 set to (00 01h) means the maximum bytes per TCW/TCCB is 64k. An extension support field 510 (e.g., bit 31) indicates support for a protocol or protocol extension such as the FCX protocol. An additional field 512 (e.g., bits 16-30) may include information regarding channel

subsystem capabilities. For example, a BIDI bit (e.g., bit 28) of the field 512 indicates channel subsystem support for bi-directional data transfers.

Further description is provided in "Determining Extended Capability of a Channel Path," U.S. Patent Application No. 12/030,912, filed on February 14, 2008, published as U.S. Patent Publication No. 2009/0210557.

Technical effects and benefits of exemplary embodiments include the ability to transfer transport support data as well as additional commands to the control unit, in addition to I/O commands and device support data located in a TCCB. Technical effects also include the ability to continue processing of DCWs despite detection of an incorrect length condition, which allows a control unit to continue to process I/O commands without the need to terminate an I/O operation. Other technical effects include the ability to transport both input and output data between a channel and a control unit during a single I/O operation.

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one ore more other features, integers, steps, operations, element components, and/or groups thereof.

The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the

invention for various embodiments with various modifications as are suited to the particular use contemplated.

As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wire line, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

Aspects of the present invention are described below with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks.

The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

The flow diagrams depicted herein are just one example. There may be many variations to this diagram or the steps (or operations) described therein without departing from the scope of the invention. For instance, the steps may be performed in a differing order or steps may be added, deleted or modified. All of these variations are considered a part of the claimed invention.

While the preferred embodiment to the invention had been described, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first described.

CLAIMS

1. A non-transitory computer readable medium comprising executable instructions for performing an input/output (I/O) operation initiated by an I/O operation instruction at a host computer system configured for communication with a control unit, the computer readable medium readable by a processing circuit for performing a method comprising:

sending, by a channel subsystem in the host computer system, a process login (PRLI) request message to the control unit to initialize a link between the channel subsystem and the control unit, the PRLI request message including a field having a value that indicates whether the channel subsystem supports bi-directional data transfer;

receiving a PRLI response message from the control unit, the PRLI response message including a field having a value that indicates whether the control unit supports bi-directional data transfer;

providing an indication to a host operating system that bi-directional data transfer is supported; and

based on executing the I/O operation instruction received from the host computer system, performing a method comprising:

gathering a plurality of commands specified via the I/O operation instruction received from the host computer system, at least one of the plurality of commands specifying an input data transfer and at least one of the plurality of commands specifying an output data transfer;

transferring the plurality of commands to the control unit;

sending at least one output data message to the control unit including output data to be transferred to the control unit, the output data message sent based on the at least one of the plurality of commands specifying an output data transfer; and

receiving at least one input message from the control unit including input data to be stored in a main storage of the host computer system, the input data message sent based on the at least one of the plurality of commands specifying an input data transfer.

2. The computer readable medium of claim 1, wherein the plurality of commands are a plurality of device command words (DCW) that each specifies an I/O command, and gathering includes obtaining a transport command word (TCW) at the channel subsystem in the host computer system for the I/O operation, the TCW including a transport command control block (TCCB) address field including an address of a TCCB in the main storage of the host computer system,

the TCCB including the plurality of DCWs, at least one of the DCWs specifying an input data transfer and at least one of the DCWs specifying an output data transfer.

3. The computer readable medium of claim 2, wherein transferring the plurality of commands includes transferring the TCCB including the plurality of DCWs to the control unit.

4. The computer readable medium of claim 3, wherein the indication is configured to specify whether a first-transfer-ready disabled condition is in effect, and the method further comprises:

based on the first-transfer-ready disabled condition being in effect, sending the at least one output data message to the control unit following transfer of the TCCB and prior to receiving at least one message from the control unit; and

based on the first-transfer-ready disabled condition not being in effect, receiving the at least one message from the control unit prior to sending the at least one output data message.

5. The computer readable medium of claim 4, wherein the message from the control unit includes one of a transfer-ready message requesting that the channel subsystem send output data and the input data message.

6. The computer readable medium of claim 2, wherein the TCCB is included in a transport command information unit (IU) sent to the control unit, the transport command IU including a transport command header (TCH), the TCH including a field that indicates whether the channel subsystem is requesting bi-directional data transfer.

7. The computer readable medium of claim 6, wherein the TCCB includes a write data length (DL) field specifying an amount of output data to be transferred and a bi-directional read data length (BRDL) field specifying an amount of input data to be transferred.

8. The computer readable medium of claim 7, wherein the method further comprises, based on the control unit not supporting bi-directional data operations, receiving a transport response message from the control unit indicating that the I/O operation has ended, the transport response message including a TCH content error indication based on at least one of:

the control unit detecting that a read (R) bit and a write (W) bit in the TCH are both set to one and that the TCCB does not include the BRDL; and

the control unit detecting that the R bit and the W bit are both set to one and the control unit not supporting bi-directional data transfer.

9. The computer readable medium of claim 7, wherein the method further comprises receiving a transport response message from the control unit indicating that the I/O operation has ended, and the transport response message includes:

a Data Length Residual Count (DLRC) field that specifies a difference between a DL field value in the TCCB and an amount of output data received by the control unit; and

a BRDL-residual count that specifies a difference between a BRDL field value in the TCCB and an amount of input data sent by the control unit.

10. The computer program product of claim 9, wherein the channel subsystem detects an error based on at least one of:

a DLRC value not matching a difference between the DL field value in the TCCB and an amount of output data sent to the control unit; and

a BRDL-residual count value not matching a difference between the BRDL field value and an amount of input data received from the control unit.

11. The computer readable medium of claim 2, wherein the PRLI request message indicates a request to the control unit to indicate whether the control unit supports a transport mode protocol, the transport mode protocol supporting the use of the TCW and the TCCB, and the PRLI response message includes a field that indicates whether the control unit supports the transport mode protocol.

12. An apparatus comprising a channel subsystem for performing an input/output (I/O) operation initiated by an I/O operation instruction at a host computer system, the channel subsystem configured for communication with a control unit and including an interface for communication between the host computer system and the control unit, the channel subsystem configured to perform:

sending, by a channel subsystem in the host computer system, a process login (PRLI) request message to the control unit to initialize a link between the channel subsystem and the control unit, the PRLI request message including a field having a value that indicates whether the channel subsystem supports bi-directional data transfer;

receiving a PRLI response message from the control unit, the PRLI response message including a field having a value that indicates whether the control unit supports bi-directional data transfer;

providing an indication to a host operating system that bi-directional data transfer is supported; and

based on executing the I/O operation instruction received from the host computer system, performing a method comprising:

gathering a plurality of commands specified via the I/O operation instruction received from the host computer system, at least one of the plurality of commands specifying an input data transfer and at least one of the plurality of commands specifying an output data transfer;

transferring the plurality of commands to the control unit;

sending at least one output data message to the control unit including output data to be transferred to the control unit, the output data sent based on the at least one of the plurality of commands specifying an output data transfer; and

receiving at least one input message from the control unit including input data to be stored in a main storage of the host computer system, the input data message sent based on the at least one of the plurality of commands specifying an input data transfer.

13. The apparatus of claim 12, wherein the plurality of commands are a plurality of device command words (DCW) that each specifies an I/O command, and gathering includes obtaining a transport command word (TCW) at the channel subsystem in the host computer system for the I/O operation, the TCW including a transport command control block (TCCB) address field including an address of a TCCB in the main storage of the host computer system, the TCCB including the plurality of DCWs, at least one of the DCWs specifying an input data transfer and at least one of the DCWs specifying an output data transfer.

14. The apparatus of claim 13, wherein the indication is configured to specify whether a first-transfer-ready disabled condition is in effect, and the host computer system is configured to further perform:

based on the first-transfer-ready disabled condition being in effect, sending the at least one output data message to the control unit following transfer of the TCCB and prior to receiving at least one message from the control unit; and

based on the first-transfer-ready disabled condition not being in effect, receiving the at least one message from the control unit prior to sending the at least one output data message.

15. The apparatus of claim 13, wherein the TCCB is included in a transport command information unit (IU) sent to the control unit, the transport command IU including a transport command header (TCH), the TCH including a field that indicates whether the channel subsystem is requesting bi-directional data transfer.

16. The apparatus of claim 15, wherein the TCCB includes a write data length (DL) field specifying an amount of output data to be transferred and a bi-directional read data length (BRDL) field specifying an amount of input data to be transferred.

17. The apparatus of claim 16, wherein the channel subsystem is configured to further perform, based on the control unit not supporting bi-directional data operations, receiving a transport response message from the control unit indicating that the I/O operation has ended, the transport response message including a TCH content error indication based on at least one of:

the control unit detecting that a read (R) bit and a write (W) bit in the TCH are both set to one and that the TCCB does not include the BRDL; and

the control unit detecting that the R bit and the W bit are both set to one and the control unit not supporting bi-directional data transfer.

18. The apparatus of claim 16, wherein the method further comprises receiving a transport response message from the control unit indicating that the I/O operation has ended, the transport response message includes a Data Length Residual Count (DLRC) field that specifies a difference between a DL field value in the TCCB and an amount of output data received by the control unit, and a BRDL-residual count that specifies a difference between a BRDL field value in the TCCB and an amount of input data sent by the control unit, and the channel subsystem recognizes an based on at least one of

a DLRC value not matching a difference between the DL field value in the TCCB and an amount of output data sent to the control unit; and

a BRDL-residual count value not matching a difference between the BRDL field value and an amount of input data received from the control unit.

19. A method of performing an input/output (I/O) operation initiated by an I/O operation instruction at a host computer system configured for communication with a control unit, the method comprising:

sending, by a channel subsystem in the host computer system, a process login (PRLI) request message to the control unit to initialize a link between the channel subsystem and the control unit, the PRLI request message including a field having a value that indicates whether the channel subsystem supports bi-directional data transfer;

receiving a PRLI response message from the control unit, the PRLI response message including a field having a value that indicates whether the control unit supports bi-directional data transfer;

providing an indication to a host operating system that bi-directional data transfer is supported; and

based on executing the I/O operation instruction received from the host computer system, performing a method comprising:

gathering a plurality of commands specified via the I/O operation instruction received from the host computer system, at least one of the plurality of commands specifying an input data transfer and at least one of the plurality of commands specifying an output data transfer;

transferring the plurality of commands to the control unit;

sending at least one output data message to the control unit including output data to be transferred to the control unit, the output data message sent based on the at least one of the plurality of commands specifying an output data transfer; and

receiving at least one input message from the control unit including input data to be stored in a main storage of the host computer system, the input data message sent based on the at least one of the plurality of commands specifying an input data transfer.

20. The method of claim 19, wherein the plurality of commands are a plurality of device command words (DCW) that each specifies an I/O command, and gathering includes obtaining a transport command word (TCW) at the channel subsystem in the host computer system for the I/O operation, the TCW including a transport command control block (TCCB) address field including an address of a TCCB in the main storage of the host computer system, the TCCB including the plurality of DCWs, at least one of the DCWs specifying an input data transfer and at least one of the DCWs specifying an output data transfer.

21. The method of claim 20, wherein the indication is configured to specify whether a first-transfer-ready disabled condition is in effect, and the method further comprises:

based on the first-transfer-ready disabled condition being in effect, sending the at least one output data message to the control unit following transfer of the TCCB and prior to receiving at least one message from the control unit; and

based on the first-transfer-ready disabled condition not being in effect, receiving the at least one message from the control unit prior to sending the at least one output data message.

22. The method of claim 20, wherein the TCCB is included in a transport command information unit (IU) sent to the control unit, the transport command IU including a transport command header (TCH), the TCH including a field that indicates whether the channel subsystem is requesting bi-directional data transfer.

23. The method of claim 22, wherein the TCCB includes a write data length (DL) field specifying an amount of output data to be transferred and a bi-directional read data length (BRDL) field specifying an amount of input data to be transferred.

24. The method of claim 23, further comprising, based on the control unit not supporting bi-directional data operations, receiving a transport response message from the control unit indicating that the I/O operation has ended, the transport response message including a TCH content error indication based on at least one of:

the control unit detecting that a read (R) bit and a write (W) bit in the TCH are both set to one and that the TCCB does not include the BRDL; and

the control unit detecting that the R bit and the W bit are both set to one and the control unit not supporting bi-directional data transfer.

25. The method of claim 22, further comprising receiving a transport response message from the control unit indicating that the I/O operation has ended, and wherein the transport response message includes a Data Length Residual Count (DLRC) field that specifies a difference between a DL field value in the TCCB and an amount of output data received by the control unit, and a BRDL-residual count that specifies a difference between a BRDL field value in the TCCB and an amount of input data sent by the control unit, and the channel subsystem recognizes an error based on at least one of:

a DLRC value not matching a difference between the DL field value in the TCCB and an amount of output data sent to the control unit; and

a BRDL-residual count value not matching a difference between the BRDL field value and an amount of input data received from the control unit.