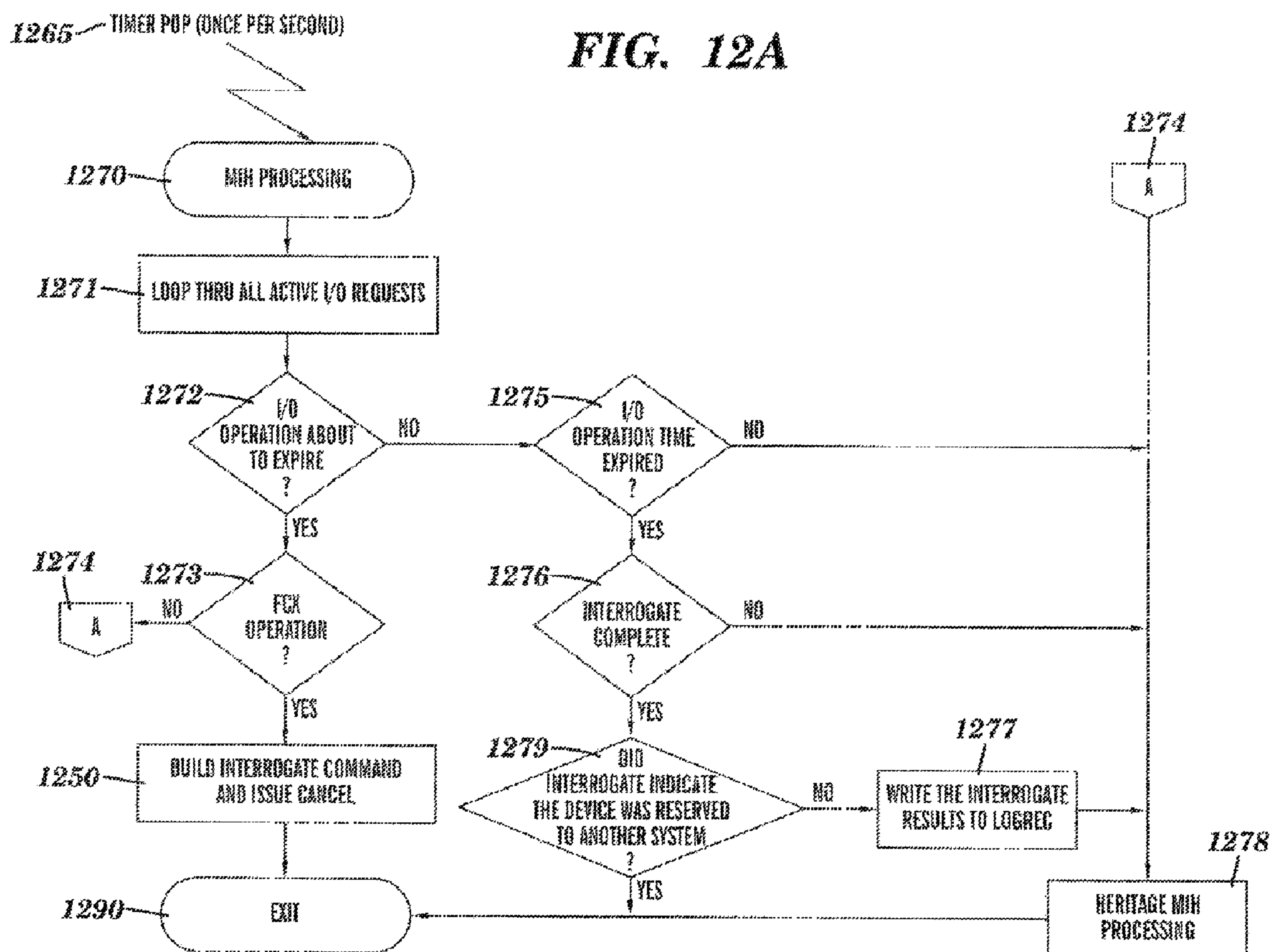




(86) Date de dépôt PCT/PCT Filing Date: 2009/02/09
(87) Date publication PCT/PCT Publication Date: 2009/08/20
(85) Entrée phase nationale/National Entry: 2010/04/28
(86) N° demande PCT/PCT Application No.: EP 2009/051461
(87) N° publication PCT/PCT Publication No.: 2009/101056
(30) Priorité/Priority: 2008/02/14 (US12/030,989)

(51) Cl.Int./Int.Cl. *G06F 13/12* (2006.01)
(71) Demandeur/Applicant:
INTERNATIONAL BUSINESS MACHINES
CORPORATION, US
(72) Inventeurs/Inventors:
CASPER, DANIEL, US;
CARLSON, SCOTT, US;
FLANAGAN, JOHN, US;
HATHORN, ROGER, US;
HUANG, CATHERINE, US;
KALOS, MATTHEW, US;
RICCI, LOUIS, US;
...
(74) Agent: WANG, PETER

(54) Titre : COMMANDE INFORMATIQUE ET REPONSE POUR DETERMINER L'ETAT D'UNE OPERATION D'E/S
(54) Title: COMPUTER COMMAND AND RESPONSE FOR DETERMINING THE STATE OF AN I/O OPERATION



(57) Abrégé/Abstract:

A state of an input/output (I/O) operation in an I/O processing system is determined. A request for performing the I/O operation is received from an I/O operating system at a channel subsystem and forwarded to a control unit controlling an I/O device for



(72) **Inventeurs(suite)/Inventors(continued)**: RIEDY, DALE, US; SITTMANN, GUSTAV, III, US; NJOKU, UGOCHUKWU, US; YUDENFRIEND, HARRY, US

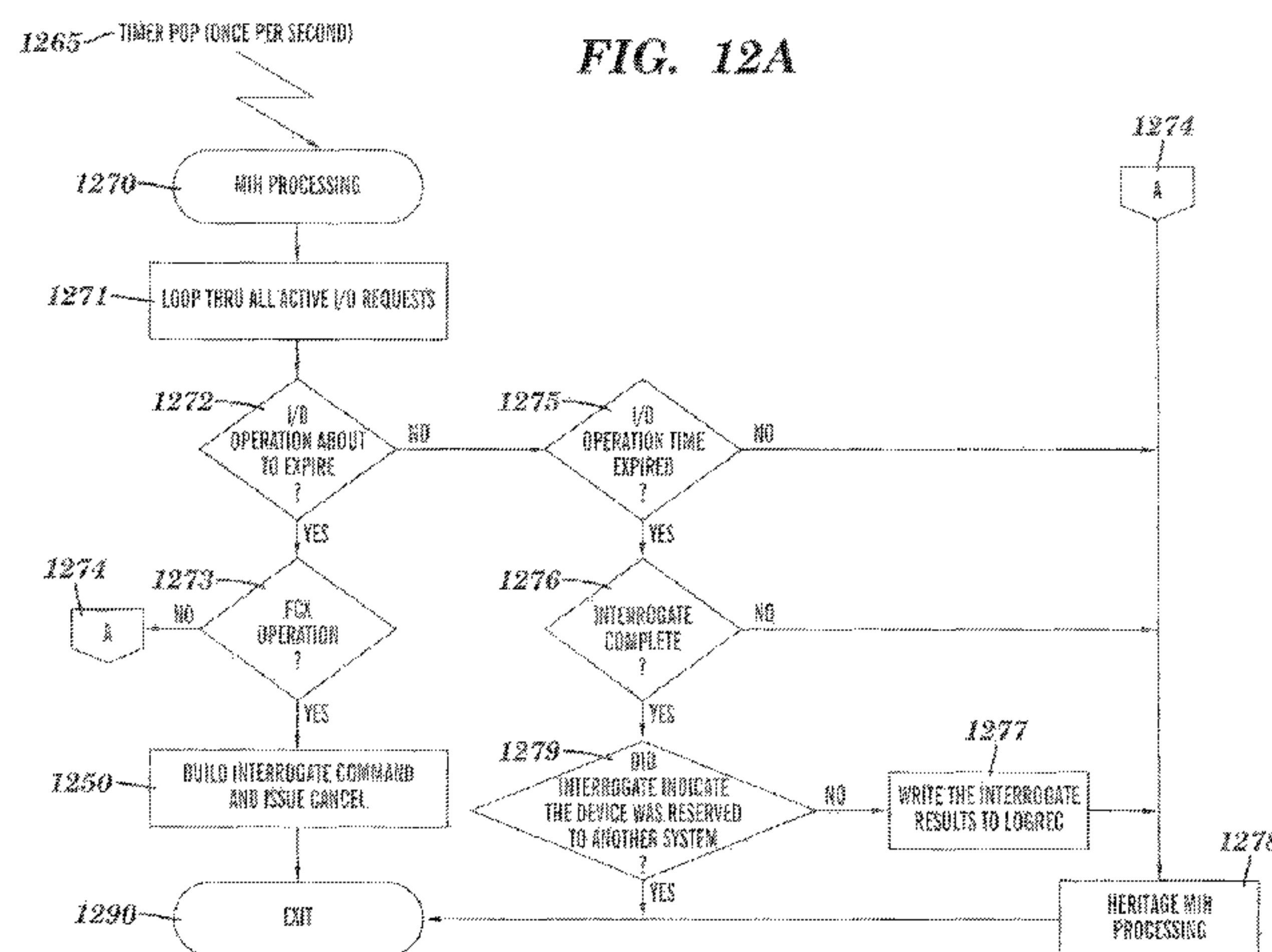
(57) **Abrégé(suite)/Abstract(continued)**:

executing the I/O operation. After a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, an interrogation request is received at the channel subsystem from the I/O operating system for determining the state of the I/O operation. An interrogation command is sent from the channel subsystem to the control unit. A response is received from the control unit, the response indicates a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

(10) International Publication Number
WO 2009/101056 A2

- Matthew** [US/US]; 5435 East Heatherwood Way, Tucson, Arizona 85718 (US). **RICCI, Louis** [US/US]; 5 Spruce Road, Hyde Park, New York 12538 (US). **RIEDY, Dale** [US/US]; 6 Oak Bend Road, Poughkeepsie, New York 12603 (US). **SITTMANN III, Gustav** [US/US]; 422 South Park Avenue, Webster Groves, Missouri 63119 (US). **UGOCHUKWU, Njoku** [US/US]; 426 Midland Avenue, Yonkers, New York 10704 (US). **YUDENFRIEND, Harry** [US/US]; 1 Nob Hill Road, Poughkeepsie, New York 12603-5545 (US).
- (74) **Agent: WILLIAMS, Julian, David**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

(54) Title: COMPUTER COMMAND AND RESPONSE FOR DETERMINING THE STATE OF AN I/O OPERATION



(57) Abstract: A state of an input/output (I/O) operation in an I/O processing system is determined. A request for performing the I/O operation is received from an I/O operating system at a channel subsystem and forwarded to a control unit controlling an I/O device for executing the I/O operation. After a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, an interrogation request is received at the channel subsystem from the I/O operating system for determining the state of the I/O operation. An interrogation command is sent from the channel subsystem to the control unit. A response is received from the control unit, the response indicates a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

WO 2009/101056 A2



MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), **Published:**

OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML,
MR, NE, SN, TD, TG).

— *without international search report and to be republished
upon receipt of that report (Rule 48.2(g))*

COMPUTER COMMAND AND RESPONSE FOR DETERMINING
THE STATE OF AN I/O OPERATION

FIELD OF THE INVENTION

The present disclosure relates generally to input/output processing, and in particular, to determining the state of an I/O operation.

BACKGROUND OF THE INVENTION

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 operating system may employ channel command words (CCWs) by passing them to the channel subsystem in order to transfer data between the I/O devices and memory. A CCW specifies the command to be executed. For commands initiating certain I/O operations, the CCW designates the memory area associated with the operation, the action to be taken whenever a transfer to or from the area is completed, and other options.

During I/O processing, a list of CCWs is fetched from memory by a channel. The channel parses each command from the list of CCWs and forwards a number of the commands, each command in its own entity, to a control unit coupled to the channel. The control unit then processes the commands. The channel tracks the state of each command and controls when the next set of commands is to be sent to the control unit for processing. The channel

ensures that each command is sent to the control unit in its own entity. Further, the channel obtains certain information associated with processing the response from the control unit for each command.

Depending on a link protocol used, an operating system may have difficulty making an informed decision regarding what action to take with an I/O operation that is taking a longer time than expected or allotted to complete. Accordingly, there is a need to provide the operating system with a way of determining the state of an I/O operation and determining an action to take for an I/O operation that is taking longer than the expected or allotted time to execute.

BRIEF SUMMARY OF THE INVENTION

Embodiments of the invention include a computer program product for determining a state of an input/output (I/O) operation in an I/O processing system. The computer program product comprises a tangible storage medium readable by a processing circuit and storing instructions for executing by the processing circuit for performing a method. The method comprises receiving, at a channel subsystem, a request for performing the I/O operation from an I/O operating system, and forwarding the request from the channel subsystem to a control unit controlling an I/O device for executing the I/O operation. The method further comprises, after a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, receiving, at the channel subsystem, an interrogation request from the I/O operating system for determining the state of the I/O operation, responsive to receipt of the interrogation request at the channel subsystem, sending an interrogation command from the channel subsystem to the control unit to determine the state of the I/O operation, and receiving, at the channel subsystem, a response from the control unit indicating the state of the I/O operation. The response includes information indicating a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

Additional embodiments include an apparatus adapted to communicate with a control unit controlling an input/output (I/O) device executing an I/O operation in an I/O processing

system. The apparatus comprises a channel subsystem for communication with an I/O operating system and the control unit. The channel subsystem performs a method comprising receiving a request for performing the I/O operation from the I/O operating system and forwarding the request to the control unit controlling the I/O device for executing the I/O operation. The method further comprises, after a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, receiving an interrogation request from the I/O operating system for determining a state of the I/O operation, responsive to receipt of the interrogation request, sending an interrogation command to the control unit to determine the state of the I/O operation, and receiving a response from the control unit indicating the state of the I/O operation. The response includes information indicating a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

Further embodiments include a method for determining a state of an input/output (I/O) operation in an I/O processing system. The method comprises receiving, at a channel subsystem, a request for performing the I/O operation from an I/O operating system and forwarding the request from the channel subsystem to a control unit controlling an I/O device for executing the I/O operation. The method further comprises, after a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, receiving, at the channel subsystem, an interrogation request from the I/O operating system for determining the state of the I/O operation, responsive to receipt of the interrogation request at the channel subsystem, sending an interrogation command from the channel subsystem to the control unit to determine the state of the I/O operation, and receiving, at the channel subsystem, a response from the control unit indicating the state of the I/O operation. The response includes information indicating a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

Other systems, methods, and/or computer program products according to embodiments will be or become apparent to one with skill in the art upon review of the following drawings and detailed description. It is intended that all such additional systems, methods, and/or articles

of manufacture be included within this description, be within the scope of the present invention, and be protected by the accompanying claims.

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. 2A depicts one example of a channel command word;

FIG. 2B depicts one example of a channel command word channel program;

FIG. 3 depicts one embodiment of a link protocol used in communicating between a channel and control unit to execute the channel command word channel program of FIG. 2B;

FIG. 4 depicts one embodiment of a transport control word channel program, in accordance with an aspect of the present invention;

FIG. 5 depicts one embodiment of a link protocol used to communicate between a channel and control unit to execute the transport control word channel program of FIG. 4, in accordance with an aspect of the present invention;

FIG. 6 depicts one embodiment of a link protocol used to communicate between a channel and control unit in order to execute four read commands of a channel command word channel program;

FIG. 7 depicts one embodiment of a link protocol used to communicate between a channel and control unit to process the four read commands of a transport control word channel program, in accordance with an aspect of the present invention;

FIG. 8 depicts one embodiment of a control unit and a channel, in accordance with an aspect of the present invention;

FIG. 9 depicts one embodiment of a Transport Control Word (TCW) including an Interrogate-TCW Address field in accordance with an aspect of the present invention;

FIG. 10 depicts one embodiment of an Interrogate DCW in accordance with an aspect of the present invention;

FIG. 11 depicts one embodiment of a Transport Response IU in accordance with an aspect of the present invention;

FIG. 12A depicts one embodiment of a process performed by the I/O operating system for deciding when to request the state of an I/O operation from the control unit in accordance with an aspect of the invention.

FIG. 12B depicts one embodiment of a process for interrogating a control unit to determine the state of an I/O operation in accordance with an aspect of the invention.

FIG. 13 depicts one embodiment of a computer program product incorporating one or more aspects of the present invention.

The detailed description explains the preferred embodiments of the invention, together with advantages and features, by way of example with reference to the drawings.

DETAILED DESCRIPTION OF THE INVENTION

In accordance with an aspect of the present invention, input/output (I/O) processing is facilitated. For instance, I/O processing is facilitated by readily enabling access to the information, such as status and measurement data, associated with I/O processing. Further, I/O processing is facilitated, in one example, by reducing communications between components of an I/O processing system used to perform the I/O processing. For instance, the number of exchanges and sequences between an I/O communications adapter, such as a channel, and a control unit is reduced. This is accomplished by sending a plurality of commands from the I/O communications adapter to the control unit as a single entity for

execution by the control unit, and by the control unit sending the data resulting from the commands, if any, as a single entity.

The plurality of commands is included in a block, referred to herein as a transport command control block (TCCB), an address of which is specified in a transport control word (TCW). The TCW is sent from an operating system or other application to the I/O communications adapter, which in turn forwards the TCCB in a command message to the control unit for processing. The control unit processes each of the commands absent a tracking of status relative to those individual commands by the I/O communications adapter. The plurality of commands is also referred to as a channel program, which is parsed and executed on the control unit rather than the I/O communications adapter.

In an exemplary embodiment, the control unit generates a response message including status and extended status information in response to executing the channel program. The control unit may also generate a response message without executing the channel program under a limited number of communication scenarios, e.g., to inform the I/O communications adapter that the channel program will not be executed. The control unit 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 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. The I/O communications adapter parses the response message, extracting the status and extended status information, and performs further calculations using the extracted information, such as determining an extended measurement word.

One example of an I/O processing system incorporating and using one or more aspects of the present invention is described with reference to FIG. 1. I/O processing system 100 includes, for instance, a main memory 102, one or more central processing units (CPUs) 104, a storage control element 106, a channel subsystem 108, one or more control units 110 and one or more I/O devices 112, each of which is described below.

Main memory 102 stores data and programs, which can be input from I/O devices 112. For example, the main memory 102 may include one or more operating systems 103 that are executed by one or more of the CPUs 104. The main memory 102 is directly addressable

and provides for high-speed processing of data by the CPUs 104 and the channel subsystem 108.

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 storage control element 106 via a connection 114, such as a bidirectional or unidirectional bus.

Storage control element 106 is coupled to the main memory 102 via a connection 116, such as a bus; to CPUs 104 via connection 114; and to channel subsystem 108 via a connection 118. Storage control element 106 controls, for example, queuing and execution of requests made by CPU 104 and channel subsystem 108.

Channel subsystem 108 is coupled to storage control element 106, as described above, and to each of the control units 110 via a connection 120, such as a serial link. Connection 120 may be implemented as an optical link, employing single-mode or multi-mode waveguides. Channel subsystem 108 directs the flow of information between I/O devices 112 and main memory 102. It relieves the CPUs 104 of the task of communicating directly with the I/O devices 112 and permits data processing to proceed concurrently with I/O processing. The channel subsystem 108 uses one or more channel paths 122 as the communication links in managing the flow of information to or from I/O devices 112. As a part of the I/O processing, channel subsystem 108 also performs the path-management functions of testing for channel path availability, selecting an available channel path 122 and initiating execution of the operation with the I/O devices 112.

Each channel path 122 includes a channel 124 (channels 124 are located within the channel subsystem 108, in one example, as shown in FIG. 1), one or more control units 110 and one or more connections 120. In another example, it is also possible to have one or more dynamic switches (not depicted) as part of the channel path 122. A dynamic switch is coupled to a channel 124 and a control unit 110 and provides the capability of physically interconnecting any two links that are attached to the switch. In another example, it is also possible to have multiple systems therefore multiple channel subsystems (not depicted) attached to control unit 110.

Also located within channel subsystem 108 are subchannels (not shown). One subchannel is provided for and dedicated to each I/O device 112 accessible to a program through the channel subsystem 108. A subchannel (e.g., a data structure, such as a table) represents the logical state of a device to the program. Each subchannel provides information concerning the associated I/O device 112 and its attachment to channel subsystem 108. The subchannel also provides information concerning I/O operations and other functions involving the associated I/O device 112. The subchannel is the means by which the channel subsystem 108 provides information about associated I/O devices 112 to operating systems running on CPUs 104, which obtain this information by executing I/O instructions.

Channel subsystem 108 is coupled to one or more control units 110. Each control unit 110 provides logic to operate and control one or more I/O devices 112 and adapts, through the use of common facilities, the characteristics of each I/O device 112 to the link interface provided by the channel 124. The common facilities provide for the execution of I/O operations, indications concerning the status of the I/O device 112 and control unit 110, control of the timing of data transfers over the channel path 122 and certain levels of I/O device 112 control.

Each control unit 110 is attached via a connection 126 (e.g., a bus) to one or more I/O devices 112. I/O devices 112 receive information or store information in main memory 102 and/or other memory. Examples of I/O devices 112 include card readers and punches, magnetic tape units, direct access storage devices, displays, keyboards, printers, pointing devices, teleprocessing devices, communication controllers and sensor based equipment, to name a few.

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. SA22-7832-05, 6th Edition, April 2007; U.S. Patent No. 5,461,721 entitled "System For Transferring Data Between I/O Devices And Main Or Expanded Storage Under Dynamic Control Of Independent Indirect Address Words (IDAWS)," Cormier et al., issued October 24, 1995; and U.S. Patent No. 5,526,484 entitled "Method And System For Pipelining The Processing Of Channel Command Words," Casper et al., issued June 11, 1996, each of which is hereby incorporated herein by reference in its entirety. 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.

In one embodiment, to transfer data between I/O devices 112 and memory 102, channel command words (CCWs) are used. A CCW specifies the command to be executed, and includes other fields to control processing. One example of a CCW is described with reference to FIG. 2A. A CCW 200 includes, for instance, a command code 202 specifying the command to be executed (e.g., read, read backward, control, sense and write); a plurality of flags 204 used to control the I/O operation; for commands that specify the transfer of data, a count field 206 that specifies the number of bytes in the storage area designated by the CCW to be transferred; and a data address 208 that points to a location in main memory that includes data, when direct addressing is employed, or to a list (e.g., contiguous list) of modified indirect data address words (MIDAWs) to be processed, when modified indirect data addressing is employed. Modified indirect addressing is further described in U.S. Application Serial Number 11/464,613, entitled "Flexibly Controlling The Transfer Of Data Between Input/Output Devices And Memory," Brice et al., filed August 15, 2006, which is hereby incorporated herein by reference in its entirety.

One or more CCWs arranged for sequential execution form a channel program, also referred to herein as a CCW channel program. The CCW channel program is set up by, for instance, an operating system, or other software. The software sets up the CCWs and obtains the addresses of memory assigned to the channel program. An example of a CCW channel program is described with reference to FIG. 2B. A CCW channel program 210 includes, for instance, a define extent CCW 212 that has a pointer 214 to a location in memory of define extent data 216 to be used with the define extent command. In this example, a transfer in channel (TIC) 218 follows the define extent command that refers the channel program to another area in memory (e.g., an application area) that includes one or more other CCWs, such as a locate record 217 that has a pointer 219 to locate record data 220, and one or more read CCWs 221. Each read CCW 220 has a pointer 222 to a data area 224. The data area includes an address to directly access the data or a list of data address words (e.g., MIDAWs or IDAWs) to indirectly access the data. Further, CCW channel program 210 includes a predetermined area in the channel subsystem defined by the device address called the subchannel for status 226 resulting from execution of the CCW channel program.

The processing of a CCW channel program is described with reference to FIG. 3, as well as with reference to FIG. 2B. In particular, FIG. 3 shows an example of the various exchanges and sequences that occur between a channel and a control unit when a CCW channel program is executing. The link protocol used for the communications is FICON (Fibre Connectivity), in this example. Information regarding FICON is described in "Fibre Channel Single Byte Command Code Sets-2 Mapping Protocol (FC-SB-3), T11/Project 1357-D/Rev. 1.6, INCITS (March 2003), which is hereby incorporated herein by reference in its entirety.

Referring to FIG. 3, a channel 300 opens an exchange with a control unit 302 and sends a define extent command and data associated therewith 304 to control unit 302. The command is fetched from define extent CCW 212 (FIG. 2B) and the data is obtained from define extent data area 216. The channel 300 uses TIC 218 to locate the locate record CCW and the read CCW. It fetches the locate record command 305 (FIG. 3) from the locate record CCW 217 (FIG. 2B) and obtains the data from locate record data 220. The read command 306 (FIG. 3) is fetched from read CCW 221 (FIG. 2B). Each is sent to the control unit 302.

The control unit 302 opens an exchange 308 with the channel 300, in response to the open exchange of the channel 300. This can occur before or after locate command 305 and/or read command 306. Along with the open exchange, a response (CMR) is forwarded to the channel 300. The CMR provides an indication to the channel 300 that the control unit 302 is active and operating.

The control unit 302 sends the requested data 310 to the channel 300. Additionally, the control unit 302 provides the status to the channel 300 and closes the exchange 312. In response thereto, the channel 300 stores the data, examines the status and closes the exchange 314, which indicates to the control unit 302 that the status has been received.

The processing of the above CCW channel program to read 4k of data requires two exchanges to be opened and closed and seven sequences. The total number of exchanges and sequences between the channel and control unit is reduced through collapsing multiple commands of the channel program into a TCCB. The channel, e.g., channel 124 of FIG. 1, uses a TCW to identify the location of the TCCB, as well as locations for accessing and

storing status and data associated with executing the channel program. The TCW is interpreted by the channel and is not sent or seen by the control unit.

One example of a channel program to read 4k of data, as in FIG. 2B, but includes a TCCB, instead of separate individual CCWs, is described with reference to FIG. 4. As shown, a channel program 400, referred to herein as a TCW channel program, includes a TCW 402 specifying a location in memory of a TCCB 404, as well as a location in memory of a data area 406 or a TIDAL 410 (i.e., a list of transport mode indirect data address words (TIDAWs), similar to MIDAWs) that points to data area 406, and a status area 408. TCWs, TCCBs, and status are described in further detail below.

The processing of a TCW channel program is described with reference to FIG. 5. The link protocol used for these communications is, for instance, Fibre Channel Protocol (FCP). In particular, three phases of the FCP link protocol are used, allowing host bus adapters to be used that support FCP to perform data transfers controlled by CCWs. FCP and its phases are described further in "Information Technology – Fibre Channel Protocol for SCSI, Third Version (FCP-3)," T10 Project 1560-D, Revision 4, September 13, 2005, which is hereby incorporated herein by reference in its entirety.

Referring to FIG. 5, a channel 500 opens an exchange with a control unit 502 and sends TCCB 504 to the control unit 502. In one example, the TCCB 504 and sequence initiative are transferred to the control unit 502 in a FCP command, referred to as FCP_CMND information unit (IU) or a transport command IU. The control unit 502 executes the multiple commands of the TCCB 504 (e.g., define extent command, locate record command, read command as device control words (DCWs)) and forwards data 506 to the channel 500 via, for instance, a FCP_Data IU. It also provides status and closes the exchange 508. As one example, final status is sent in a FCP status frame that has a bit active in, for instance, byte 10 or 11 of the payload of a FCP_RSP IU, also referred to as a transport response IU. The FCP_RES_IU payload may be used to transport FICON ending status along with additional status information, including parameters that support the calculation of extended measurement words and notify the channel 500 of the maximum number of open exchanges supported by the control unit 502.

In a further example, to write 4k of customer data, the channel 500 uses the FCP link protocol phases, as follows:

1. Transfer a TCCB in the FCP_CMND IU.
2. Transfer the IU of data, and sequence initiative to the control unit 502.

(FCP Transfer Ready Disabled)

3. Final status is sent in a FCP status frame that has a bit active in, for instance, byte 10 or 11 of the FCP_RSP IU Payload. The FCP_RES_INFO field or sense field is used to transport FICON ending status along with additional status information, including parameters that support the calculation of extended measurement words and notify the channel 500 of the maximum number of open exchanges supported by the control unit 502.

By executing the TCW channel program of FIG. 4, there is only one exchange opened and closed (see also FIG. 5), instead of two exchanges for the CCW channel program of FIG. 2B (see also FIG. 3). Further, for the TCW channel program, there are three communication sequences (see FIGs. 4-5), as compared to seven sequences for the CCW channel program (see FIGs. 2B-3).

The number of exchanges and sequences remain the same for a TCW channel program, even if additional commands are added to the program. Compare, for example, the communications of the CCW channel program of FIG. 6 with the communications of the TCW channel program of FIG. 7. In the CCW channel program of FIG. 6, each of the commands (e.g., define extent command 600, locate record command 601, read command 602, read command 604, read command 606, locate record command 607 and read command 608) are sent in separate sequences from channel 610 to control unit 612. Further, each 4k block of data (e.g., data 614-620) is sent in separate sequences from the control unit 612 to the channel 610. This CCW channel program requires two exchanges to be opened and closed (e.g., open exchanges 622, 624 and close exchanges 626, 628), and fourteen communications sequences. This is compared to the three sequences and one exchange for the TCW channel program of FIG. 7, which accomplishes the same task as the CCW channel program of FIG. 6.

As depicted in FIG. 7, a channel 700 opens an exchange with a control unit 702 and sends a TCCB 704 to the control unit 702. The TCCB 704 includes the define extent command, the two locate record commands, and the four read commands in DCWs, as described above. In response to receiving the TCCB 704, the control unit 702 executes the commands and sends, in a single sequence, the 16k of data 706 to the channel 700. Additionally, the control unit 702 provides status to the channel 700 and closes the exchange 708. Thus, the TCW channel program requires much less overhead to transfer the same amount of data as the CCW channel program of FIG. 6.

Turning now to FIG. 8, one embodiment of the control unit 110 and the channel 124 of FIG. 1 that support TCW channel program execution are depicted in greater detail. The control unit 110 includes CU control logic 802 to parse and process command messages containing a TCCB, such as the TCCB 704 of FIG. 7, received from the channel 124 via the connection 120. The CU control logic 802 can extract DCWs and control data from the TCCB received at the control unit 110 to control a devices, for instance, I/O device 112 via connection 126. The CU control logic 802 sends device commands and data to the I/O device 112, as well as receives status information and other feedback from the I/O device 112. For example, the I/O device 112 may be busy because of a previous reservation request targeting I/O device 112. To manage potential device reservation contention issues that can arise when the control unit 110 receives multiple requests to access the same I/O device 112, the CU control logic 802 keeps track of and stores device busy messages and associated data in a device busy queue 804.

The CU control logic 802 can access and control other elements within the control unit 110, such as CU timers 806 and CU registers 808. The CU timers 806 may include multiple timer functions to track how much time a sequence of I/O operations takes to complete. The CU timers 806 may further include one or more countdown timers to monitor and abort I/O operations and commands that do not complete within a predetermined period. The CU registers 808 can include fixed values that provide configuration and status information, as well as dynamic status information that is updated as commands are executed by the CU control logic 802. The control unit 110 may further include other buffer or memory elements (not depicted) to store multiple messages or status information associated with communications between the channel 124 and the I/O device 112. The CU registers 808

may include a maximum control unit exchange parameter that defines the maximum number of open control unit exchanges that the control unit 110 supports.

The channel 124 in the channel subsystem 108 includes multiple elements to support communication with the control unit 110. For example, the channel 124 may include CHN control logic 810 that interfaces with CHN subsystem timers 812 and CHN subsystem registers 814. In an exemplary embodiment, the CHN control logic 810 controls communication between the channel subsystem 108 and the control unit 110. The CHN control logic 810 may directly interface to the CU control logic 802 via the connection 120 to send commands and receive responses, such as transport command and response IUs. Alternatively, messaging interfaces and/or buffers (not depicted) can be placed between the CHN control logic 810 and the CU control logic 802. The CHN subsystem timers 812 may include multiple timer functions to track how much time a sequence of I/O operations takes to complete, in addition to the time tracked by the control unit 110. The CHN subsystem timers 812 may further include one or more countdown timers to monitor and abort command sequences that do not complete within a predetermined period. The CHN subsystem registers 814 can include fixed values that provide configuration and status information, as well as dynamic status information, updated as commands are transported and responses are received.

The FICON command response (CMR) frame from the control unit is not part of the Fibre Channel Extension (FCX), transport mode protocol. Removing the CMR from the transport mode protocol helps to improve the performance of FCX. The CMR in FICON informs the channel that the control unit has received and does the channel send executing the command. When the FICON channel receives the CMR, the channel marks the subchannel as "SubChannel and Active Device".

In all computing environments, interrupts at various I/O devices may occur. If an OS that requested an operation at an I/O device fails to detect an interrupt, this may cause operations in a data processing system to slow down and ultimately cease. A Missing Interrupt Handler (MIH) is a mechanism included, e.g., in the OS 103 that is useful in detecting lost interrupts by timing I/O operations that are in progress and determining whether the time taken by an I/O device to execute an operation has exceeded a predetermined "normal" amount of time allotted or set for execution of the operation. If the MIH time is reached, and the I/O device

has not completed execution of the operation, this is an indication that an interrupt may have been missed, a link failure occurred, an adapter failure occurred, a control unit error occurred, or a reserve was held by a sharing system longer than expected.

When the operating system MIH times out for FICON, i.e., the MIH time is reached, it looks to see if the sub-channel was or was not marked "SubChannel and Device Active" to determine what action to take next. For FCX, the subchannel stays "Start pending" during the entire operation. So, with FCX, when the MIH times out, the I/O operating system cannot tell the state of the I/O operation because the sub-channel state stays "Start pending" for the entire operation.

In accordance with an aspect of the present invention, just before a missing interrupt timeout, e.g., one second before the MIH time is reached, the operating system uses an interrogate command to determine the state of the I/O operation at the control unit. The interrogate command may be initiated with a cancel subchannel instruction before the time allotted for completion of the execution of the I/O operation elapses, and the I/O operation has not completed.

There are several benefits of the interrogate command. For example, the interrogate command is executed when a MIH timeout is about to occur, thereby removing the requirement for a CMR on every I/O operation. Removing the requirement for the CMR on every I/O operation improves the FCX performance by reducing fabric traffic and channel and adapter overhead. Also, the interrogate command transfers information to the control unit about the OS for logging by the control unit if a timeout does occur. Another advantage is that the control unit provides detailed state information about the I/O operation back to the OS, whereas the CMR for FICON only indicates that the control unit is currently executing the I/O. Also, if an I/O operation is lost, the information exchanged by the interrogate command is very useful for problem determination.

Implementation of the interrogation described herein involves a cancel subchannel instruction and an Interrogate -TCW Address field in a TCW and is described from the channel subsystem perspective, an interrogate command and response from the channel subsystem perspective, and an interrogate command and response from the control unit perspective. Each of these is described below.

An exemplary embodiment of a transport control word (TCW) 900 is depicted in FIG. 9. The TCW 900 is utilized by the channel 124 to set up the I/O operation and is not sent to the control unit 110. The TCW depicted in FIG. 9 is for the implementation of the interrogation from the channel subsystem perspective.

In an exemplary TCW 900 depicted in FIG. 9, a format field 902 equal to, e.g., "00b" indicates that what follows is a TCW 900. The TCW 900 also includes a flags field 906. The first five bits of the flags field 906 are reserved for future use and are set to zero. The sixth bit of the flags field 906 is a TIDAL data address flag. In an exemplary embodiment, the TIDAL data address flag is set to one when the data address field 914 contains an address of a TIDAL. If the TIDAL data address flag is set to zero, then the data address field 914 contains a data address. The seventh bit of the flags field 906 is a TCCB TIDAL flag. In an exemplary embodiment, the TCCB TIDAL flag is set to one when the TCCB address field 922 contains an address of a TCCB TIDAL. If the TCCB TIDAL flag is set to zero, then the TCCB address field 922 directly addresses the TCCB. The eighth through twenty-fourth bits of the flags field 906 are reserved for future use. Field 907 may be reserved for future use.

The TCW 900 also includes a TCCB length field 910, which indirectly represents the length of the TCCB and may be utilized to determine the actual length of the TCCB. A R/W 910 field includes read/write bits utilized to indicate whether data is being read and/or written as a result of executing the TCW 900. In an exemplary embodiment, the read bit in the read/write bits is set to one to indicate that input data is being transferred from an I/O device 112 to system storage (e.g., main memory 102) in the host system 101 as a result of executing the TCW 900. The write bit in the read/write bits is set to one to indicate that output data is being transferred from system storage (e.g., main memory 102) in the host system 101 to an I/O device as a result of executing the TCW 900. Field 912 may be reserved for future use.

Address field 914 may include a direct address or an indirect address per the flags field bit 6. The contents of the address field 914 may be an address of a TIDAL (a list of transport mode indirect data address words) for output data or the actual address of the output data. The contents of the address field 914 may be an address of a TIDAL for input data or the actual address of the input data. In an exemplary embodiment, the output data address and the

input data address are included in a single field 914, and a field 916 is reserved for future use. Alternatively, the output data address and the input data address may be split between fields 914 and 916.

The TCW 900 also includes a transport-status-block address field 920. A portion (e.g., the extended status part) of a completion status in a transport response IU for an I/O operation is stored at this address. The TCCB address field 922 in the TCW 900 includes an address where the TCCB is located in system storage. This is the control block where the DCWs to be executed for the TCW 900 reside. Also as described in the flags field bit 7, the contents of the TCCB address field 922 may be an address of a TIDAL for the TCCB or the actual address of the TCCB. A data byte count field 924 in the TCW 900 indicates the amount of output data to be transferred by the TCW for an output operation or the amount of input data to be transferred by the TCW for an input operation. Field 926 may be reserved for future use. Alternatively, the output data count and the input data count information may be split between fields 924 and 926. Several additional fields in the TCW 900 are reserved: reserved field 928, reserved field 930 and reserved field 932.

According to an aspect of the invention, the TCW 900 is expanded, e.g., from 32 bytes to 64 bytes, to allow more space for future functions. One such function is an interrogation function, made possible by an interrogate-TCW address field 934 that contains an interrogation value indicating whether an interrogation should be performed if an I/O fails to complete in an allotted time period. The interrogate-TCW address field 934 may contain the address of another TCW and may be used by the channel 124 to interrogate the state of an operation under the initiative of a cancel sub-channel I/O instruction, explained in detail below.

The TCW 900 may be set up by software to be used by the channel to drive I/O operations. The TCW depicted in FIG. 9 is one example of how a command word can be configured. Other configurations are possible where additional fields are included and/or fields depicted in FIG. 9 are not included.

According to an aspect of the present invention, a cancel subchannel instruction is executed to determine the state of the control unit if the FCX start subchannel has already been sent to a channel. If the subchannel is "start pending", and the start subchannel has been sent to the

channel, and the value of the interrogate TCW address field in the TCW is not zero, then the cancel instruction queues an interrogate command in the subchannel that may be sent to the control unit by the channel subsystem. This interrogate command may cause information to be returned from the control unit about the state of the operation being interrogated, in a data transfer phase or in the extended status part of the transport response IU. The protocol of the interrogate operation may be implemented as follows.

If a FCX I/O operation is active, the Interrogate TCW Address in the TCW for the I/O operation is used to point to an interrogate TCW. If the channel encounters a zero value Interrogate TCW Address, the channel will not initiate the Interrogate. Prior to a Missing Interrupt (MIH) time out, the OS updates the Interrogate TCW Address word in the TCW with the address of the Interrogate TCW if the OS wants to interrogate the I/O device. If the OS only wants to send the cancel instruction without conducting an interrogation, then the OS leaves the Interrogate TCW Address in the TCW set to zero. The interrogate initiative is then passed to the channel subsystem with the cancel instruction. The cancel instruction performs the current architected Cancel, but if the subchannel is “start pending” with a FCX start subchannel, and the Start has been passed to the channel, then the channel subsystem is given the initiative to interrogate the control unit.

According to aspects of the invention, if the subchannel is idle, interrupt pending with primary or alert status, or is device active only, the initiative to issue the interrogate command is discarded. If the channel receives the interrogate initiative at a “start pending” subchannel, and the start is still queued in the channel, the channel discards the interrogate initiative. If the channel receives the interrogate initiative, and the channel already has an interrogate operation in progress, the channel discards the new interrogate initiative.

If the channel receives the interrogate initiative to a UA that is start pending and has an exchange open to the control unit, the channel subsystem executes the interrogation.

According to an aspect of the invention, in executing the interrogation, the channel subsystem does the following. The channel subsystem fetches the Interrogate TCW address in the current TCW to get the pointer value to fetch the Interrogate TCW. If the pointer is all zeros, the channel discards the interrogation. In this case, the OS wants to do a cancel instruction but not an interrogation. If the pointer is valid, the channel subsystem continues. The channel subsystem opens a new exchange and sends the Interrogate DCW inside a

TCCB, addressed by the interrogate TCW, to the control unit in a transport command IU. This operation is timed by the channel for completion. If the interrogate operation does not complete in the amount of time set by the channel, the channel aborts both the interrogate operation and the operation that is being interrogated. The subchannel is then returned back to the OS with interface control check status.

The control unit receives and executes the interrogate command, transferring the interrogate information about the UA back to the channel subsystem in the transport response IU or as a data IU based on the command in the transport command IU. The original operation that is active on the UA that is being interrogated is not affected. The I/O subsystem may generate an intermediate status interrupt with an interrogate complete bit set to a one that reports the completion of the interrogation to the OS.

According to an aspect of the present invention, the interrogate command is a unique command that may be the same for all control unit types that support FCX. The transport command IU for an interrogation contains only one DCW. This interrogate DCW command may have up to 232 bytes of control data associated with it, which may be information that is passed to the control unit indicating why the interrogation is being executed.

The interrogate command transports information included in the interrogate DCW 1010 and the interrogate control data 1020 that are part of an interrogate command 1000 as shown in FIG. 10. The control unit responds with the transport response IU that returns information about the I/O operation back to the OS. The format of the information returned in the transport response IU in accordance with an aspect of the invention is shown as item 1100 in FIG. 11.

FIG. 10 depicts one embodiment of a DCW 1000 in accordance with an aspect of the present invention. In an exemplary embodiment, the DCW 1000 is eight bytes in length plus the length in the control data count field 1014. The DCW includes a command field 1011, a flags field 1012, a reserved field 1013, a control data (CD) count field 1014, and a data byte count field 1015, 1016, 1017 and 1018. The DCW command field 1011 is one byte in length and is the same as the CCW command byte utilized in a CCW (but may include additional command codes not utilized by a CCW). The flags field 1012 includes eight bits. In an exemplary embodiment, the second bit is a chain command to the next DCW 1000 in a

TCCB. When this flag bit is set to zero, it indicates that this is the last DCW 1000 of the DCW program in the TCCB. The other bits in the flag field 1016 are reserved and set to zero. An interrogate DCW 1000 has a DCW command-code field 1011 containing a 40 hex. The control data count field 1014 indicates the amount of Interrogate control data that is included with the DCW. Fields 1015-1018 include the 4 byte data count of read data that may be transferred by the interrogate DCW.

If the control data count 1014 of the interrogate DCW is greater than zero, then interrogate control data is specified in the DCW. The interrogate control data 1020 sent to the control unit is for device-dependent logging purposes and is used to aid in debugging I/O timeouts.

According to an aspect of the invention, the interrogate control data 1020 has the format described below. Referring to FIG. 10, byte 0 of word 0 {Fmt field 1021} of the interrogate control data 1020 contains an unsigned integer value that defines the layout or format (FMT) of the interrogate data. Byte 1 of word 0 (RC field 1022) contains an unsigned integer value or reason code (RC) that indicates the reason an interrogate operation was initiated by the OS. The meaning of RC values may be as follows.

- 0 Interrogate reason is not specified.
- 1 Timeout: Program-detected timeout for the operation being interrogated.
- 2-255 Reserved.

Byte 2 of word 0 of the interrogate control data (RCQ field 1023) contains an unsigned integer value that indicates additional information about the reason the interrogate operation was initiated, referred to as the Reason-Code Qualifier (RCQ). When the RC field 1022 contains the value one, the meaning of RCQ values may be as follows:

- 0 Interrogate reason qualifier not specified.
- 1 Primary: The timeout is detected by the primary program.
- 2 Secondary: The timeout is detected by the secondary program.
- 3-255 Reserved.

When the RC field 1022 does not contain the value one, the RCQ may have no meaning.

Byte 3 of word 0 {LPM field 1024} contains the Logical-Path Mask (LMP) that was used when the operation being interrogated was initiated by a start subchannel command.

Referring to word 1 of the interrogate control data, byte 0 of word 1 {PAM field 1025} contains a value of a Path-Available Mask (PAM) at the time the interrogate operation is initiated. Byte 1 of word 1 (PIM field 1026) contains a value of a Path-Installed Mask (PIM) at the time the interrogate operation is initiated. Bytes 2-3 of word 1 (Timeout field 1027) are indicative of a timeout value indicating a time allotted for completion of the I/O operation. When the RC field 1022 contains the value of one and the RCQ field 1023 contains the value of one or two, bytes 2-3 of word 1 contain the timeout interval used by the program in unsigned integer seconds.

Referring to word 2 of the interrogate control data 1020, byte 0 (Flags field 1028) contains flags that have information about the interrogation. The meaning of each flag bit may be given as follows:

Bit 0	Multipath mode.
Bit 1	Program path recovery. The interrogate is issued during path recovery by the program.
Bit 2	Critical. The device is a critical device for the program.
Bits 3-7	Reserved.

As shown in FIG. 10, bytes 1-3 of word 2 {field 1029} and bytes 0-3 of word 3 {field 1030} may also be reserved for future use. Words 4-5 (Time field 1040) may contain information regarding the time the interrogate operation was initiated. Words 6-7 {field 1050} may contain a program identifier identifying the program initiating the interrogate operation. The content of this field may be program-dependent. Words 8-N (field 1060) may contain program-dependent information.

The ending status information for the interrogate command may be set by the control unit in a transport response IU payload as shown in FIG. 11. In the transport response IU 1100,

words 0-7 may contain a status payload 1110, including ending status and status flags.

Words 8-23 may contain an extended status payload 1120, which may be stored at the status block address in z memory per the transport status block address in the TCW for the interrogation.

As shown in FIG. 11, word 8 byte 0 (ES field 1121), contains the extended status (ES) Length, which indicates the size of the ES payload. Word 8, byte 1 (ES flags field 1122) includes ES flags. Flag bits 5-7 indicate the type code. The type code defines the format of the status area of the ES payload. The type code for an Interrogate is 3. The three bit encode defining the status area may be given as follows:

- 0 Type Code 0. No information in the Status Areas.
- 1 Type code 1. Valid ending I/O status.
- 2 Type Code 2. Error terminated status.
- 3 Type Code 3. the Extended Status is an Interrogate Response, and the format is show in FIG. 11.
- 4-7 Reserved.

Word 8, bytes 2 and 3 {status field 1123} and word 9 (status field 1124) contain information indicating a status of the I/O operation. For the I/O operation, the value in these fields may be zero for an interrogation.

Words 11-23 of the transport extended status payload 1120 may be the interrogate status area. Byte 0 of word 11 (Format field 1126) contains an unsigned integer value that defines the layout of the interrogate status area. If the value of this field is zero, the contents of the interrogate status are meaningless. The following definitions of the interrogate status area apply when the format byte is set to a (01h).

Byte 1 of word 11 (Flags field 1127) contains information about the interrogate status area. The meaning of each flag bit may be given as follows:

- Bit 0 Control-width state valid: When bit 0 is one, the control unit state field

contains meaningful information. When bit 0 is zero, the control unit state field has no meaning.

Bit 1 Device-state valid: When bit 1 is one, the device state field contains meaningful information. When bit 1 is zero, the device state field has no meaning.

Bit 2 Operation-state valid: When bit 2 is one, the operation-state field contains meaningful information. When bit 2 is zero, the operation-state field has no meaning.

Bits 3-7 Reserved.

Byte 2 of word 11 (control unit state field 1128) contains an 8-bit unsigned integer that indicates a current state of the control unit for the I/O device. The meaning of each value may be given as follows:

0 Busy: The control unit is busy, and the device-dependent data field may contain additional information about the busy state.

1 Recovery: The control unit is performing a recovery process, and the device-dependent data field may contain additional information about the recovery state.

2 Interrogate maximum: The control unit is executing the maximum number of interrogate operations that it supports

3-127 Reserved.

128-255 Device dependent meanings.

Byte 3 of word 11 (device state field 1129) contains an 8-bit unsigned integer that indicates a current state of the I/O device. The meaning of this byte may be given as follows:

0 Path-group identification: The state-dependent-information field contains information identifying a path group that has the device reserved.

1 Long busy: The control unit is in a long-busy state. The meaning of long busy is device dependent, and the device-dependent field may contain additional information about the long-busy state.

2 Recovery: The device is performing a recovery process.

3-127. Reserved.

128-255 Device-dependent meanings.

Byte 0 of word 12 (operation state field 1030) contains an 8-bit unsigned integer that indicates whether an IO operation is present at the device and, when present, the state of the operation. The meaning of this byte value may be given as follows.

0 No I/O operation is present.

1 An I/O operation is present and executing.

2 An I/O operation is present and waiting for completion of an I/O operation that was initiated by another configuration.

3 An I/O operation is present and waiting for completion of an I/O operation that was initiated from the same device extent.

4 An I/O operation is present and waiting to perform a device-dependent operation.

5-127 Reserved.

128-255 Device dependent meanings.

Field 1031 may be reserved for future use.

Words 13-15 (field 1140) may contain state-dependent information. The contents of this field are device dependent. Whether this field has meaning is designated by the CS, DS, and OS fields 1128, 1129, and 1130, respectively.

Word 16 (field 1150) may contain a device-level identifier or token that identifies the implementation level of the device.

Words 17-23 (field 1160) may contain device dependent information. Whether this field has meaning may be designated by the CS, DS and OS fields 1128, 1129, and 1130, respectively.

According to exemplary embodiments, from the control unit state, device state, and operating state information returned in the transport response IU for an interrogation, the OS 103 can make an informed decision on what action to take with regard to an I/O operation that is taking a longer than an allotted time to complete.

FIG. 12A illustrates a method that the operating system uses for deciding when to request the state of an I/O operation from the control unit in accordance with aspects of the invention. When a timer pop occurs at step 1265, the operating system Missing Interrupt Handler (MIH) receives control at step 1270. For example, the MIH receives control once every second after a timer pop occurs. The MIH scans through every active I/O operation that has been issued by all applications, middleware and subsystems running in the operating system at step 1271. A determination is made at step 1272 whether an I/O operation is about to expire, e.g., within one second of the time limit for the I/O operation. If so, a determination is made whether the I/O operation is for transport mode (FCX) at step 1273. If the I/O operation is for transport mode, then an interrogate DCW is constructed, and the active TCW is updated to point to the interrogate command at step 1250. Otherwise, the process proceeds from step 1274 to step 1278 at which conventional "heritage" MIH processing is performed. From step 1250, MIH processing continues at step 1271 or terminates at step 1290 until a timer pop occurs, e.g., one second later at step 1265. If, at step 1272, it is determined that the I/O operation is not about to expire, then a check is made to see if the I/O operation time has exceeded its allotted time at step 1275. If the I/O operation has not exceeded its allotted time, then the traditional heritage (conventional) MIH processing occurs at step 1278. If the I/O operation time has indeed expired, then a check is made to see if the interrogate command has been issued at step 1276. If the interrogate command has not been issued, this is an indication that the command was not a transport mode command (FCX), and heritage MIH recovery processing is performed 1278. If, however, it is determined at step 1276 that the interrogate command has been issued and has

successfully completed, then the interrogate results are examined at step 1279 to determine whether the device is reserved for another system. If the interrogate information indicates that the device is reserved for another system 1140, then there is no error, and a missing interrupt has not occurred. Thus, processing terminates 1290. If the device is not reserved for some other system, the information returned by interrogate is placed into a record to be written to the system LOGREC dataset for diagnostic purposes at step 1277. Processing then continues with the heritage MIH processing at step 1278.

FIG. 12B illustrates a method for determining a state of an I/O operation according to aspects of the invention. A request for initiating an I/O operation is sent from an OS 103 to the channel subsystem 108 at step 1205 and on to a control unit 110 at step 1207. At step 1210, the request is received and processed at the control unit 110. If the I/O operation approaches the end of its allotted execution time, as described in the previous paragraph, an interrogate operation is initiated by the operating system at step 1250 shown in both FIG 12A and FIG 12B.

The interrogation begins at step 1250 at which the OS 103 sets up the interrogate control blocks in system memory 102 and sends the cancel subchannel to the channel subsystem 108. The channel subsystem 108 determines if the interrogate is to be sent to the control unit. If the conditions as described above are not met to do an interrogate, the heritage MIH processing terminates the I/O operation and simulates an error back to the initiator of the I/O. If the interrogate conditions are met, the interrogation request is sent from the channel subsystem 108 to the control unit 110 at step 1225. At step 1230, the control unit 110 receives the interrogation request. At step 1235, the control unit 110 sends an interrogation response to the channel subsystem 108, indicating the state of the I/O operation, the control unit 110, and the I/O device 112 executing the interrogate I/O operation. At step 1240, the interrogation response is received at the channel subsystem 108, which generates an interrupt to the OS. The OS receives the interrogate response at step 1245, creates a LOGREC entry to record the state information at the control unit and then proceeds with the heritage MIH processing as described above.

It should be appreciated that not all of the steps shown in FIG. 12 need be performed to determine the state of an I/O operation. Further, the order of steps shown in FIG. 12A and

12B are examples of how the processes may be performed. Also, additional steps that are not shown in FIGs. 12A and 12B may be performed.

As described above, embodiments can be embodied in the form of computer-implemented processes and apparatuses for practicing those processes. In exemplary embodiments, the invention is embodied in computer program code executed by one or more network elements. Embodiments include a computer program product 1300 as depicted in FIG. 13 on a computer usable medium 1302 with computer program code logic 1304 containing instructions embodied in tangible media as an article of manufacture. Exemplary articles of manufacture for computer usable medium 1302 may include floppy diskettes, CD-ROMs, hard drives, universal serial bus (USB) flash drives, or any other computer-readable storage medium, wherein, when the computer program code logic 1304 is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention.

Embodiments include computer program code logic 1304, for example, whether stored in a storage medium, loaded into and/or executed by a computer, or transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via electromagnetic radiation, wherein, when the computer program code logic 1304 is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. When implemented on a general-purpose microprocessor, the computer program code logic 1304 segments configure the microprocessor to create specific logic circuits.

While the invention has been described with reference to exemplary embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof without departing from the scope of the invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the invention without departing from the essential scope thereof. Therefore, it is intended that the invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out this invention, but that the invention will include all embodiments falling within the scope of the appended claims. Moreover, the use of the terms first, second, etc., do not denote any order or importance, but rather the terms first, second, etc. are used to distinguish one element from another. Furthermore, the use of the terms a, an, etc., do not denote a limitation of quantity, but rather denote the presence of at least one of the referenced item.

CLAIMS

1. A computer program product for determining a state of an input/output (I/O) operation in an I/O processing system, comprising:

a tangible storage medium readable by a processing circuit and storing instructions for executing by the processing circuit for performing a method comprising:

receiving, at a channel subsystem, a request for performing the I/O operation from an I/O operating system;

forwarding the request from the channel subsystem to a control unit controlling an I/O device for executing the I/O operation;

after a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, receiving, at the channel subsystem, an interrogation request from the I/O operating system for determining the state of the I/O operation;

responsive to receipt of the interrogation request at the channel subsystem, sending an interrogation command from the channel subsystem to the control unit to determine the state of the I/O operation; and

receiving, at the channel subsystem, a response from the control unit indicating the state of the I/O operation, wherein the response includes information indicating a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

2. The computer program product of claim 1, wherein the I/O operation is associated with a time period for completion, and the predetermined amount of time is less than the time period for completion.

3. The computer program product of claim 1, wherein the interrogation command includes information indicating why and when the interrogation command was sent.

4. The computer program product of claim 1, wherein the information included in the response indicating the state of the control unit indicates whether the control unit is busy, is performing a recovery operation, or is responding to other interrogation requests.

5. The computer program product of claim 1, wherein the information included in the response indicating the state of the I/O device indicates whether the I/O device is reserved by a path group, whether the control unit is in a long busy state, or whether the I/O device is performing a recovery process.

6. The compute program product of claim 1, wherein the information included in the response indicating the state of the I/O operations indicates whether an operation is present, executing, waiting for completion of another I/O operation, or waiting for performance of a device-dependent operation.

7. The computer program product of claim 1, wherein the information included in the response from the control unit indicates whether the I/O device is reserved by another I/O processing system.

8. An apparatus adapted for communicating with a control unit controlling an input/output (I/O) device for executing an I/O operation in an I/O processing system, the apparatus comprising:

a channel subsystem for communication with an I/O operating system and the control unit, the channel subsystem performing a method comprising:

receiving a request for performing the I/O operation from the I/O operating system;

forwarding the request to the control unit controlling the I/O device for executing the I/O operation;

after a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, receiving an interrogation request from the I/O operating system for determining a state of the I/O operation;

responsive to receipt of the interrogation request, sending an interrogation command to the control unit to determine the state of the I/O operation; and

receiving a response from the control unit indicating the state of the I/O operation, wherein the response includes information indicating a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

9. The apparatus of claim 8, wherein the I/O operation is associated with a time period for completion, and the predetermined amount of time is less than the time period for completion.
10. The apparatus of claim 8, wherein the interrogation command includes information indicating why and when the interrogation command was sent.
11. The apparatus of claim 8, wherein the information included in the response indicating the state of the control unit indicates whether the control unit is busy, is performing a recovery operation, or is responding to other interrogation requests.
12. The apparatus of claim 8, wherein the information included in the response indicating the state of the I/O device indicates whether the I/O device is reserved by a path group, whether the control unit is in a long busy state, or whether the I/O device is performing a recovery process.
13. The apparatus of claim 8, wherein the information included in the response indicating the state of the I/O operations indicates whether an operation is present, executing, waiting for completion of another I/O operation, or waiting for performance of a device-dependent operation.
14. The apparatus of claim 8, wherein the information included in the response from the control unit indicates whether the I/O device is reserved by another I/O processing system.
15. A method for determining a state of an input/output (I/O) operation in an I/O processing system, comprising:

receiving, at a channel subsystem, a request for performing the I/O operation from an I/O operating system;

forwarding the request from the channel subsystem to a control unit controlling an I/O device for executing the I/O operation;

after a predetermined amount of time passes without receiving indication from the control unit that the I/O operation is completed, receiving, at the channel subsystem, an interrogation request from the I/O operating system for determining the state of the I/O operation;

responsive to receipt of the interrogation request at the channel subsystem, sending an interrogation command from the channel subsystem to the control unit to determine the state of the I/O operation; and

receiving, at the channel subsystem, a response from the control unit indicating the state of the I/O operation, wherein the response includes information indicating a state of the I/O device executing the I/O operation, a state of the control unit controlling the I/O device executing the I/O operation, and the state of the I/O operation being executed.

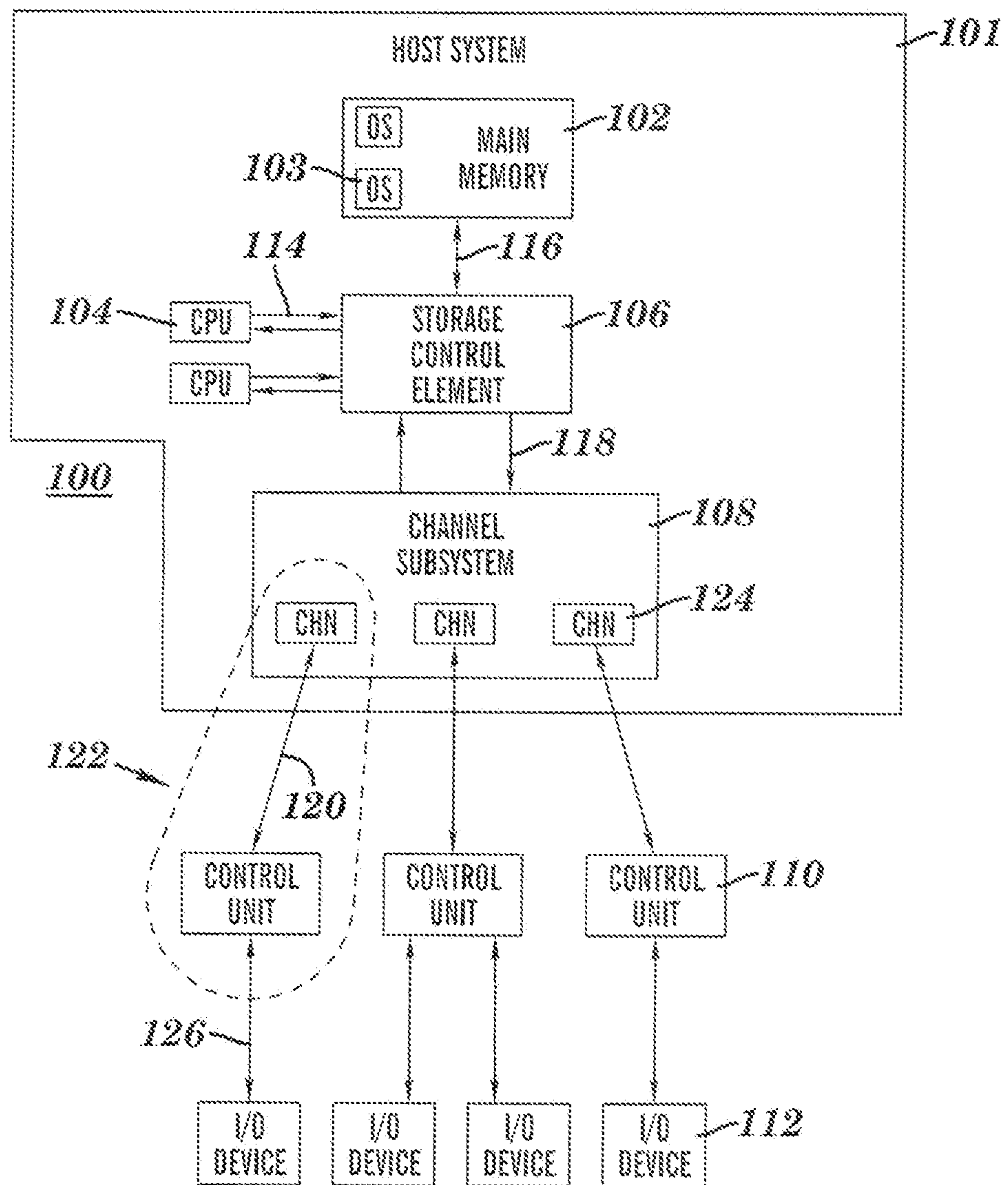
16. The method of claim 15, wherein the I/O operation is associated with a time period for completion, and the predetermined amount of time is less than the time period for completion.

17. The method of claim 15, wherein the interrogation command includes information indicating why and when the interrogation command was sent.

18. The method of claim 15, wherein the information included in the response indicating the state of the control unit indicates whether the control unit is busy, is performing a recovery operation, or is responding to other interrogation requests.

19. The method of claim 15, wherein the information included in the response indicating the state of the I/O device indicates whether the I/O device is reserved by a path group or another I/O processing system, whether the control unit is in a long busy state, or whether the I/O device is performing a recovery process.

20. The method of claim 15, wherein the information included in the response indicating the state of the I/O operations indicates whether an operation is present, executing, waiting for completion of another I/O operation, or waiting for performance of a device-dependent operation.

**FIG. 1**

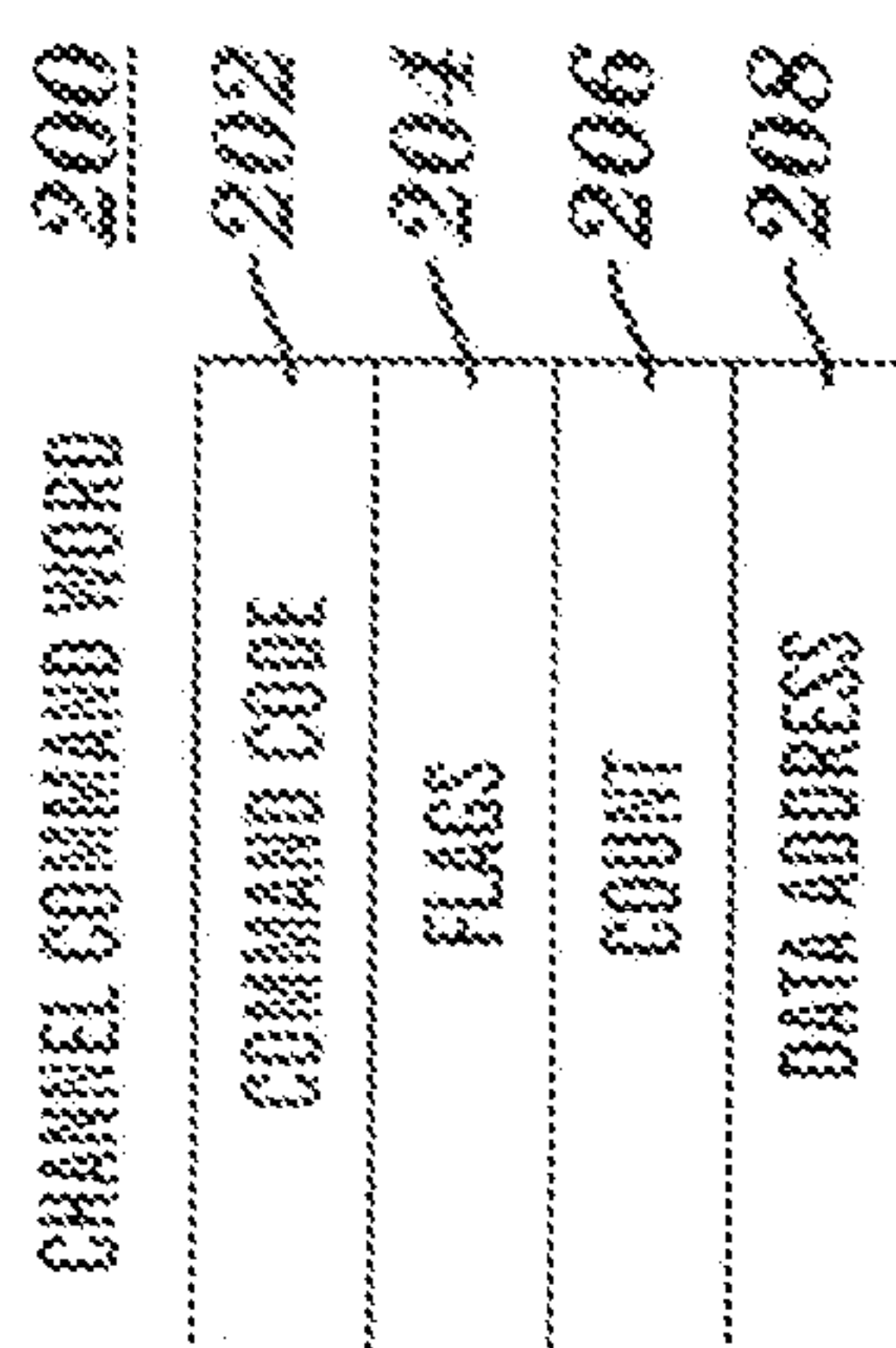


FIG. 2A
PRIOR ART

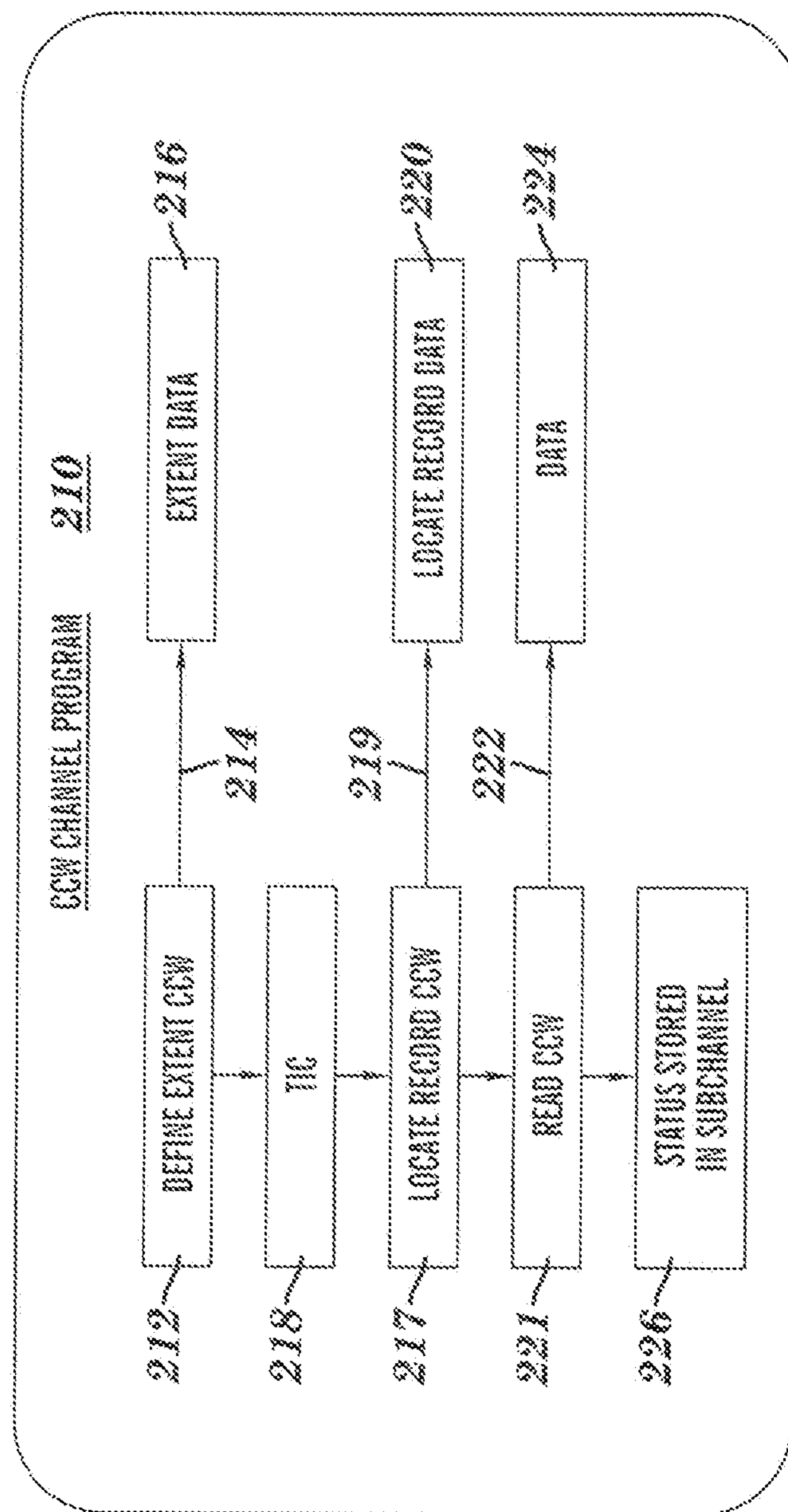


FIG. 2B
PRIOR ART

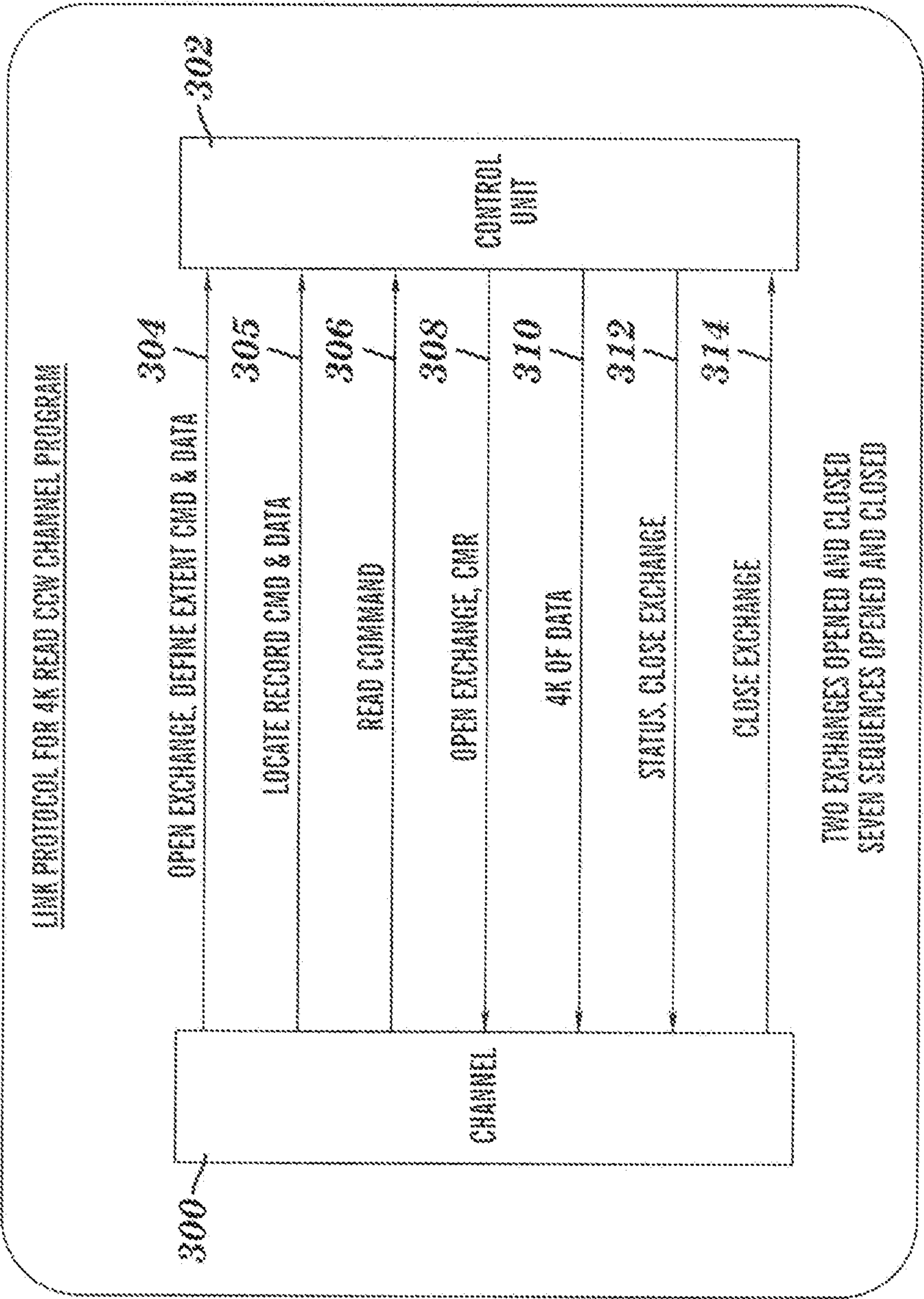
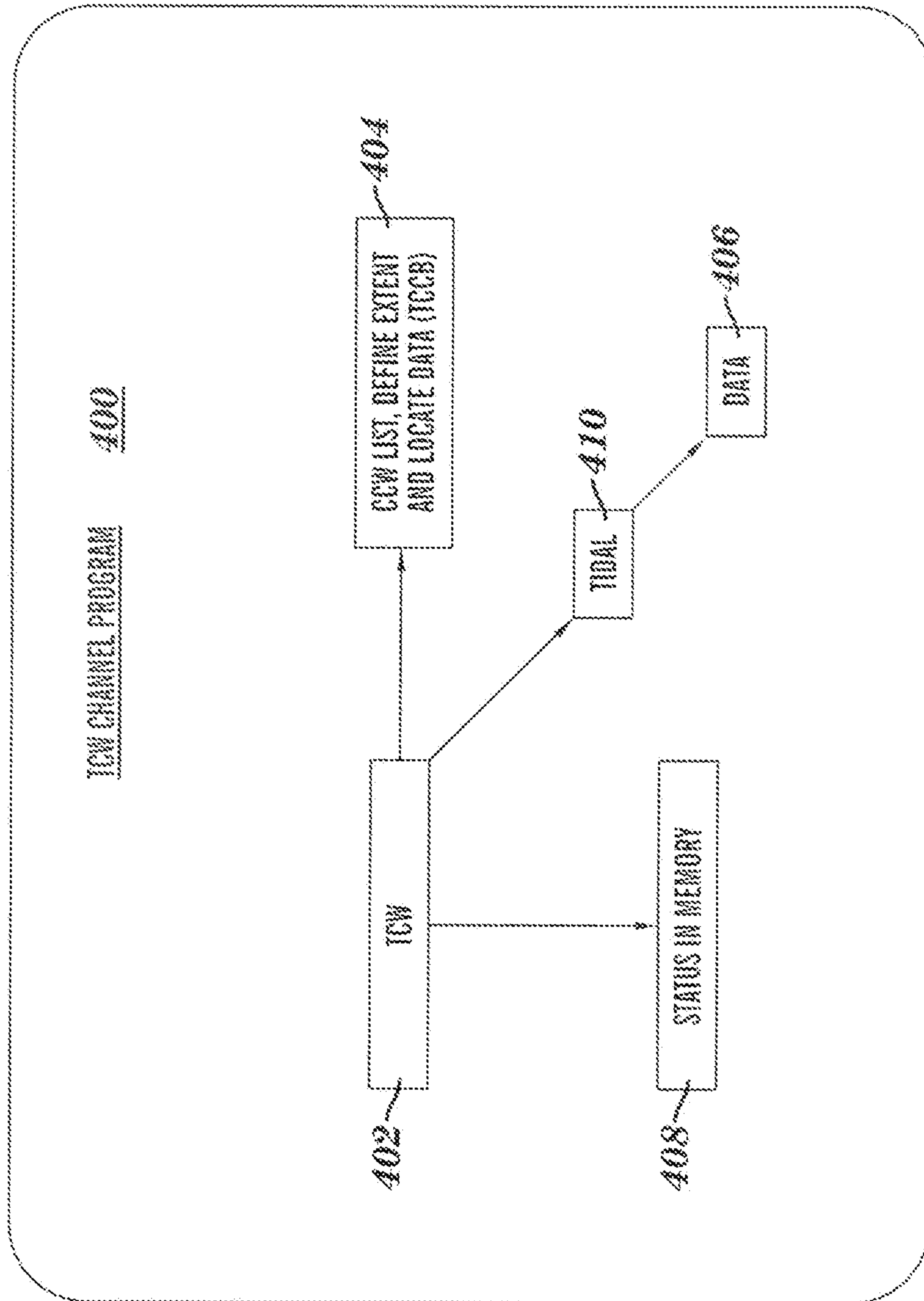


FIG. 3
PRIOR ART

**FIG. 4**

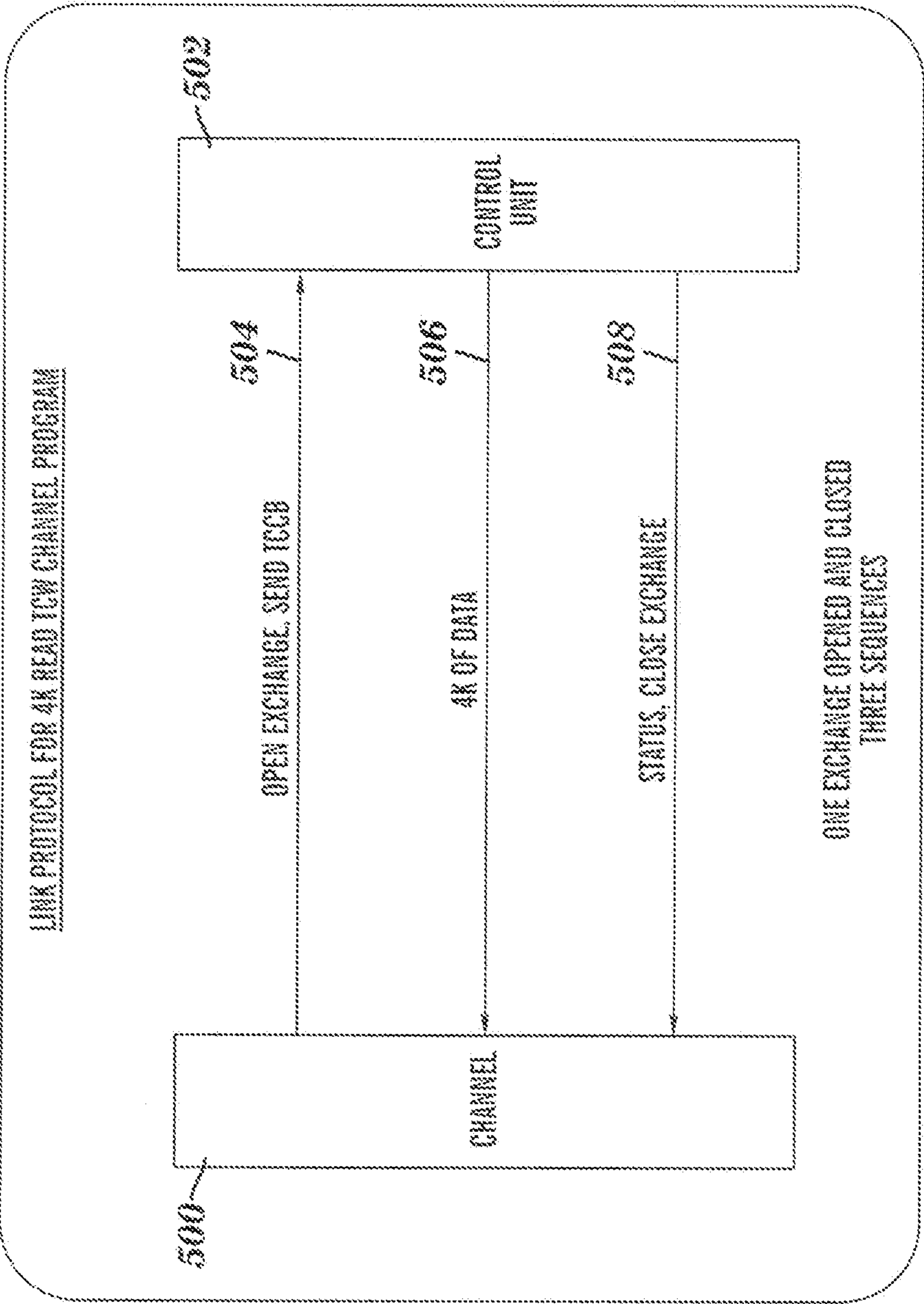


FIG. 5

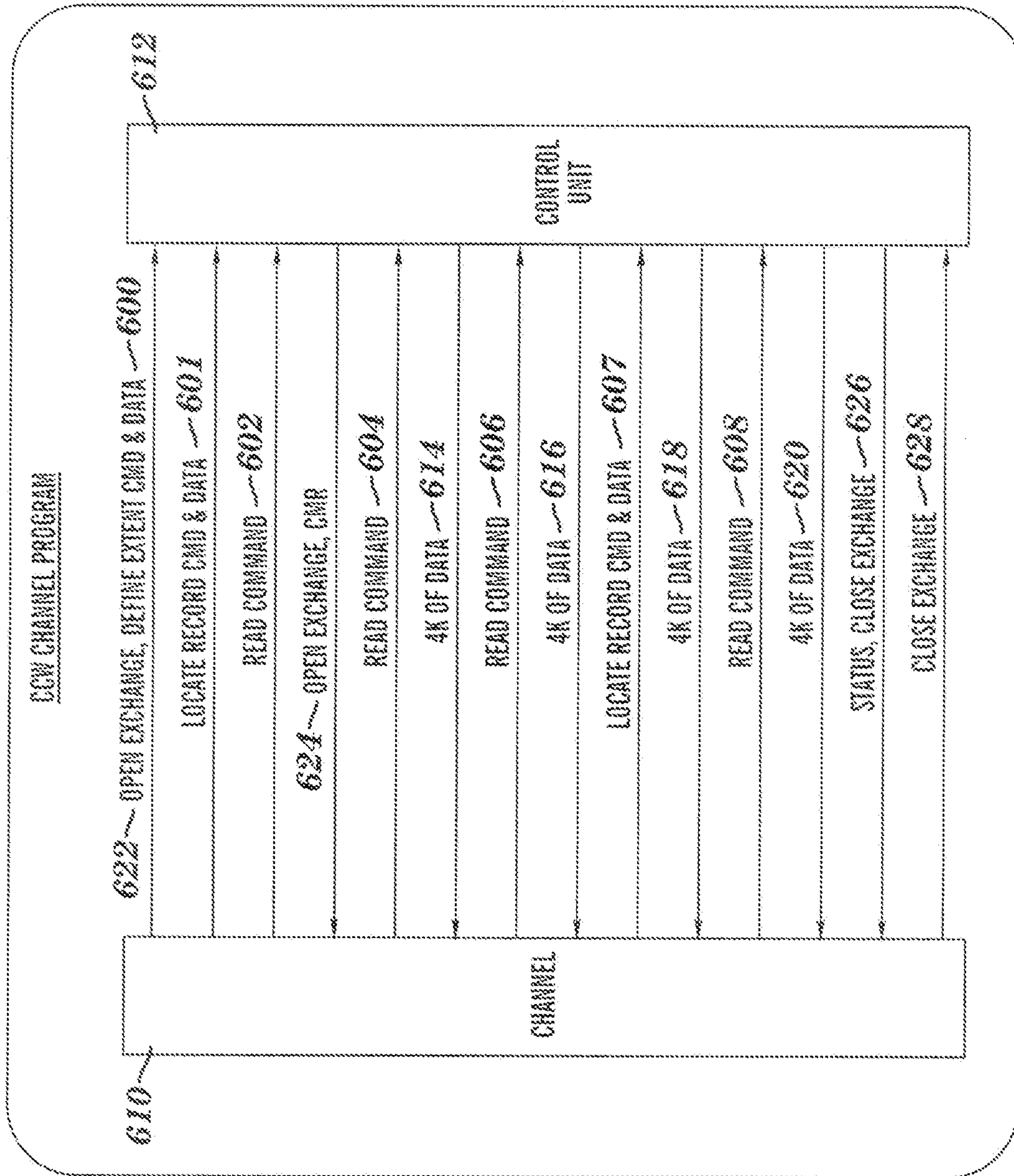
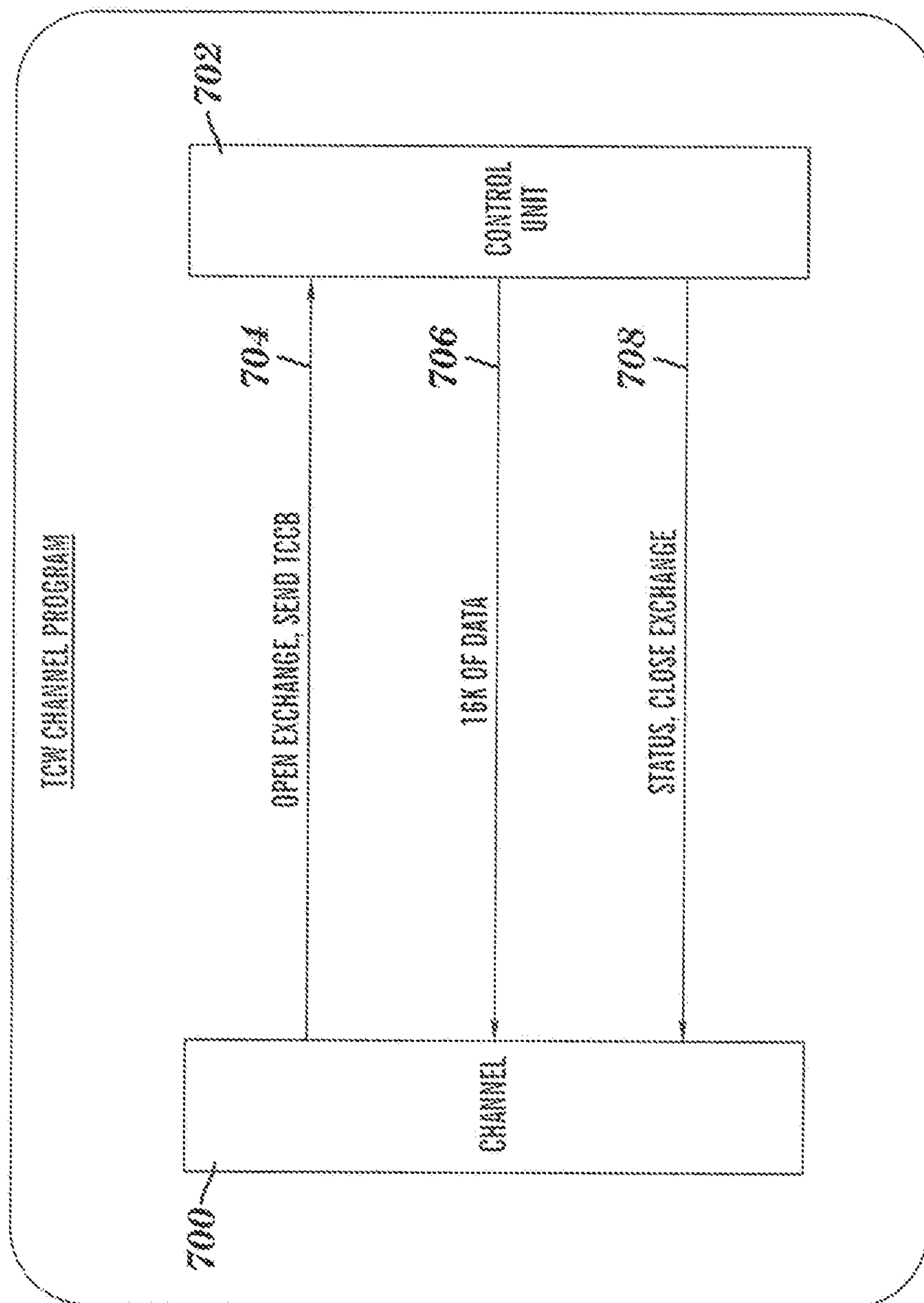
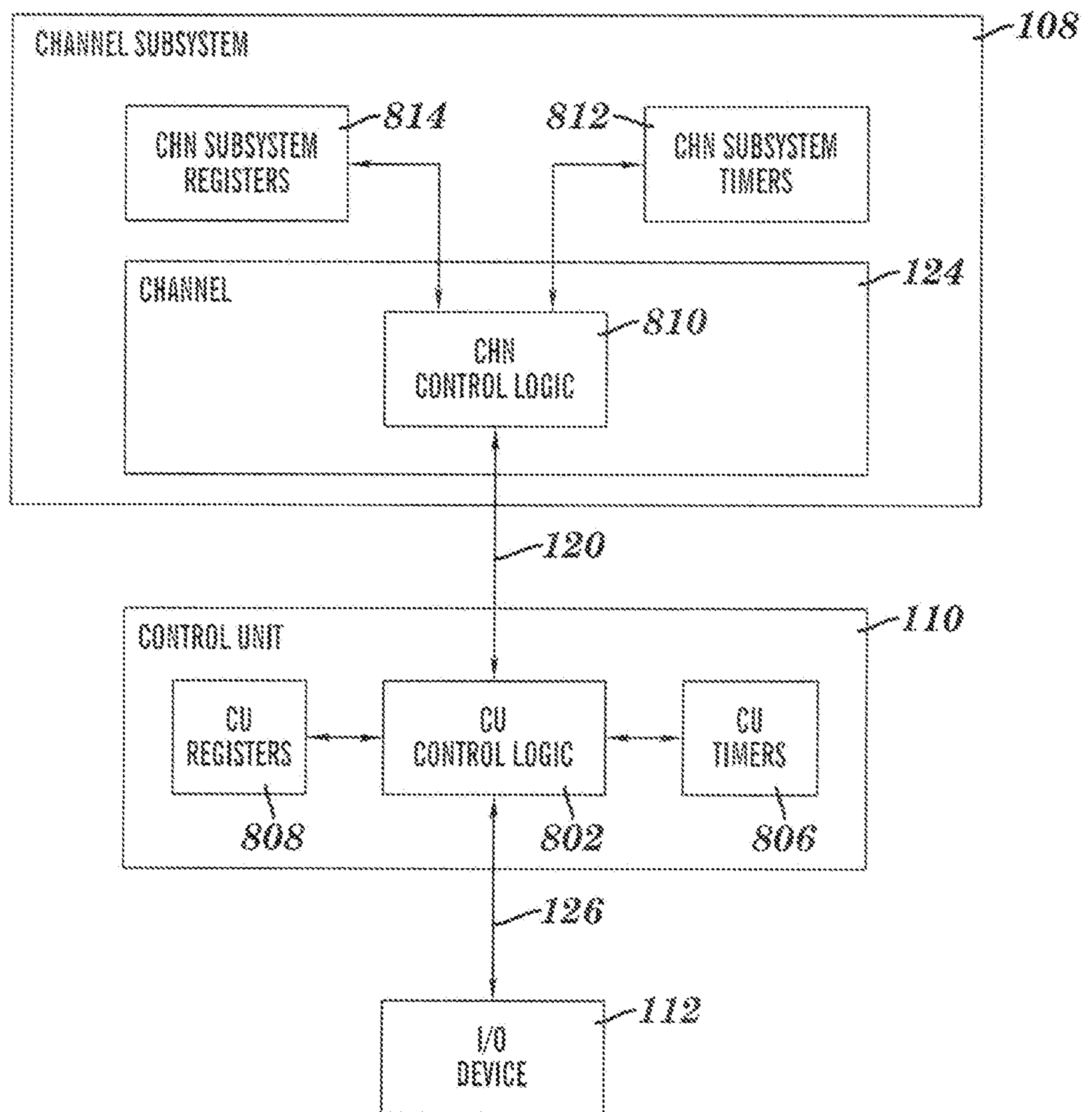


FIG. 6
PRIOR ART

**FIG. 7**

**FIG. 8**

900 ↗

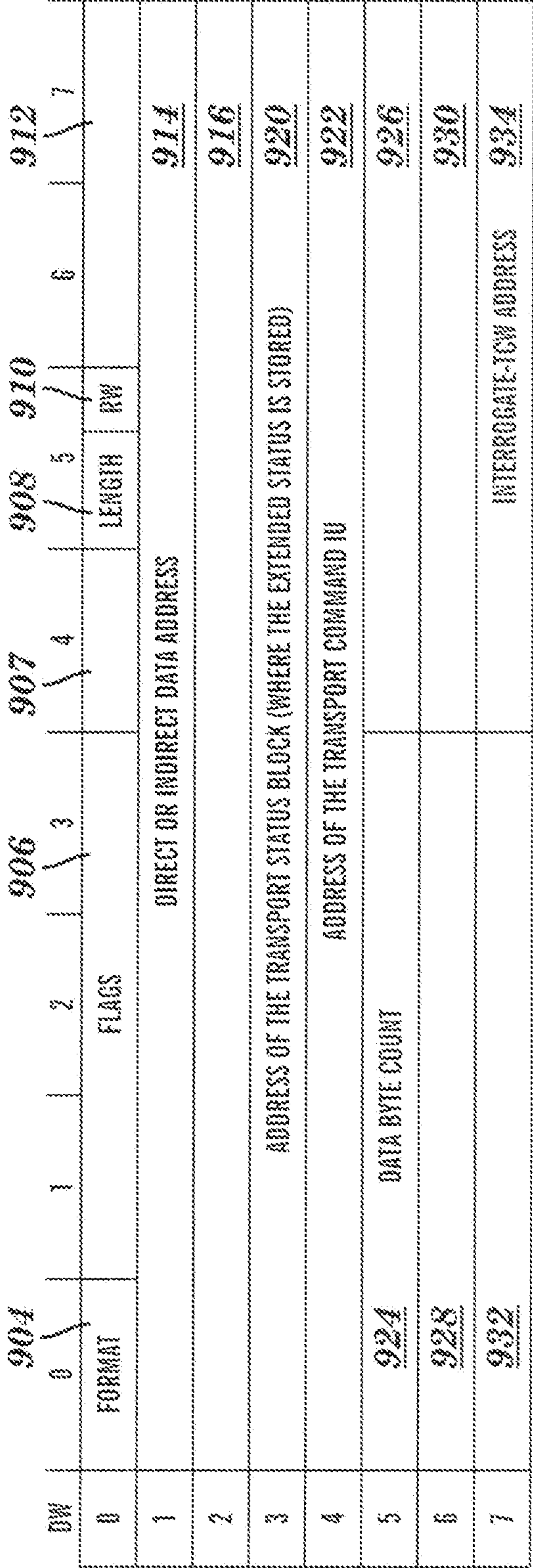


FIG. 9

1000 ↗

NAME	WORD	0	1	2	3
DCW		<u>1011</u>	<u>1012</u>	<u>1013</u>	<u>1014</u>
	0	40	00	00	CONTROL DATA COUNT <= 232
	1	00 <u>1015</u>	00 <u>1016</u>	00 <u>1017</u>	00 <u>1018</u>
INTERROGATE CONTROL DATA	0	FMT <u>1021</u>	RC <u>1022</u>	RCD <u>1023</u>	LPM <u>1024</u>
	1	PAM <u>1025</u>	PM <u>1026</u>		TIMEOUT <u>1027</u>
	2	FLAGS <u>1028</u>		RESERVED	<u>1029</u>
	3		RESERVED		<u>1030</u>
	4		TIME		<u>1040</u>
	5				
	6			PROGRAM IDENTIFIER	<u>1050</u>
	7				
	8				
	10			PROGRAM DEPENDENT DATA	<u>1060</u>
	8				

1010

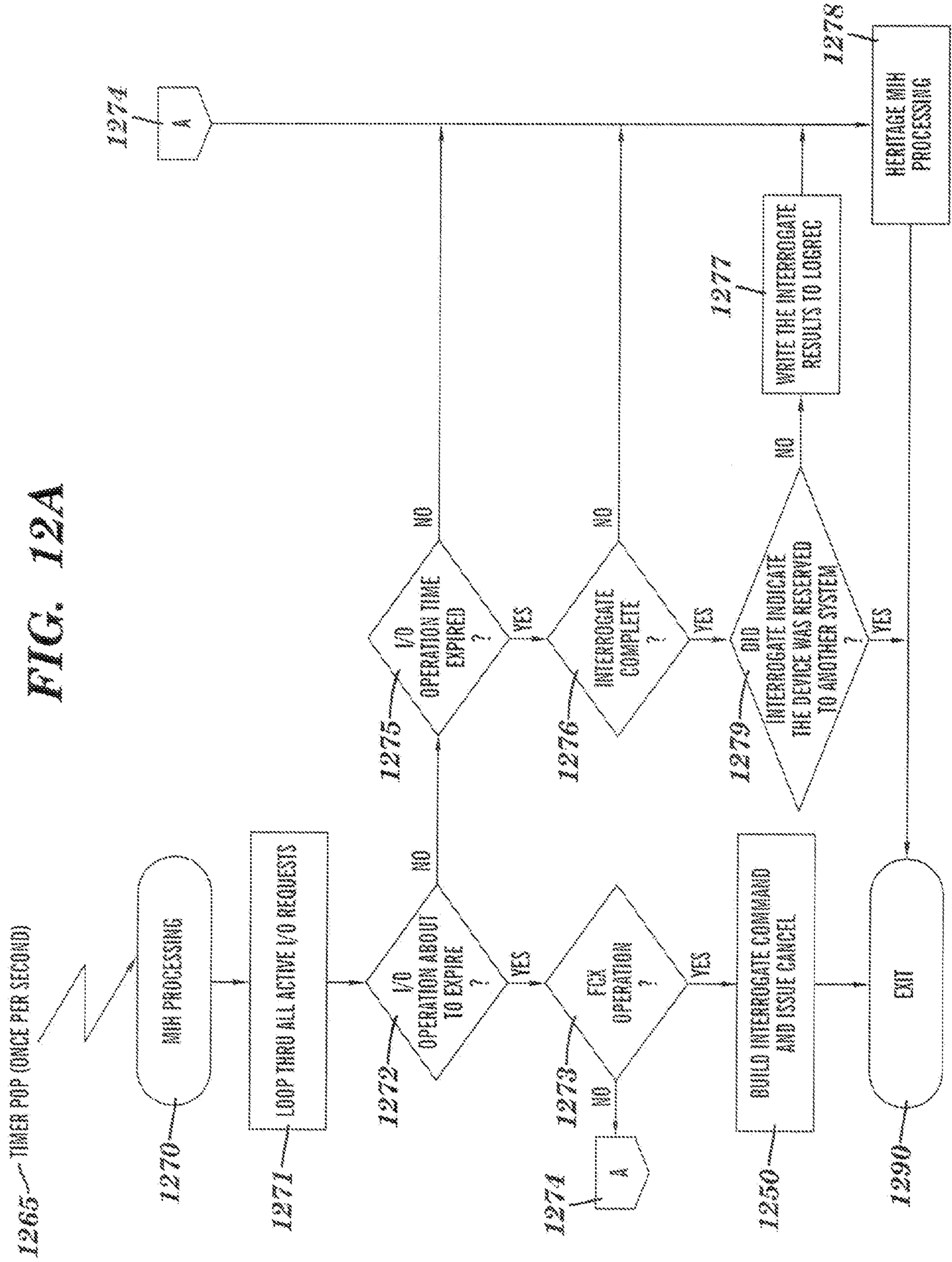
1020

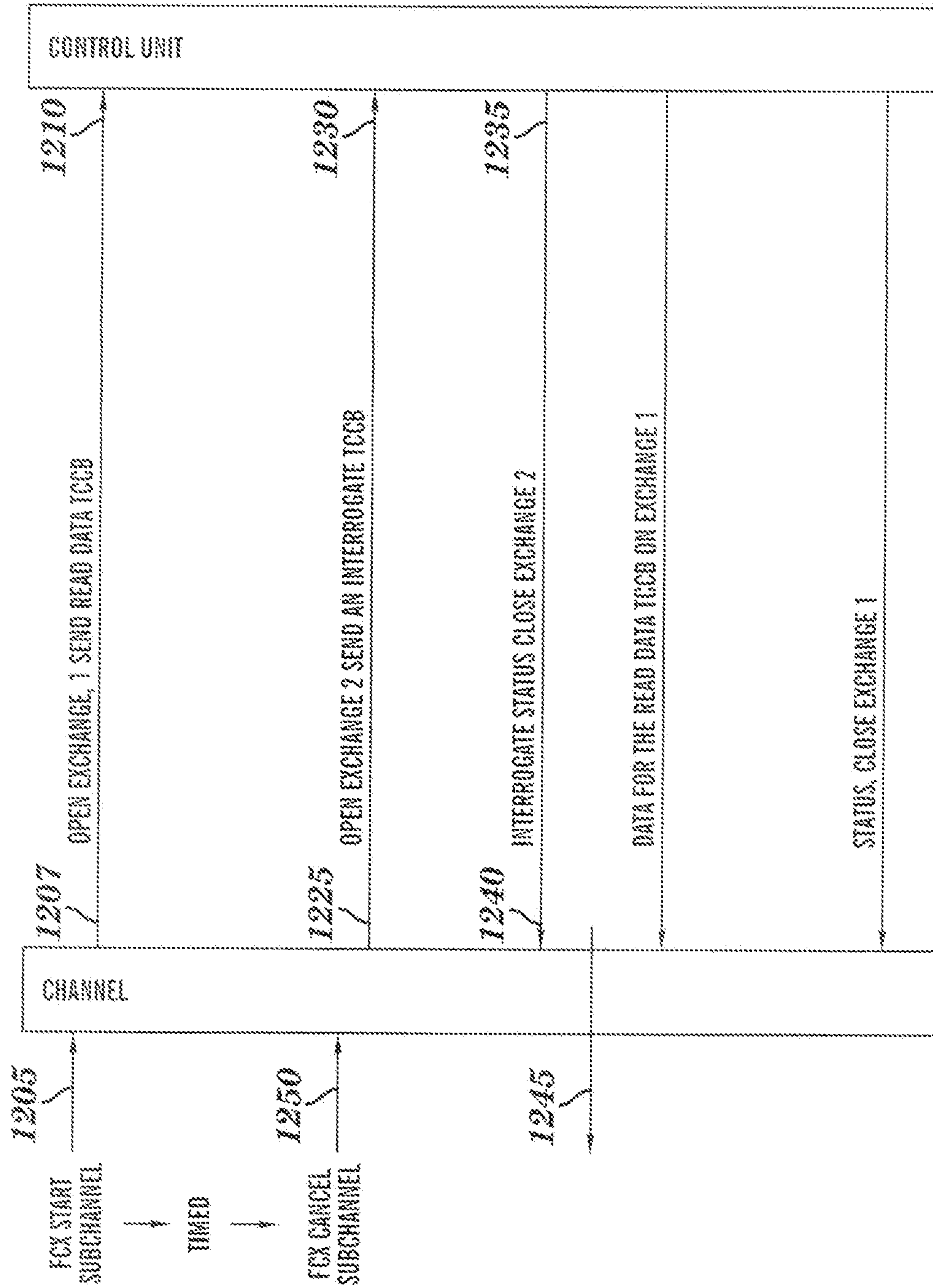
FIG. 10

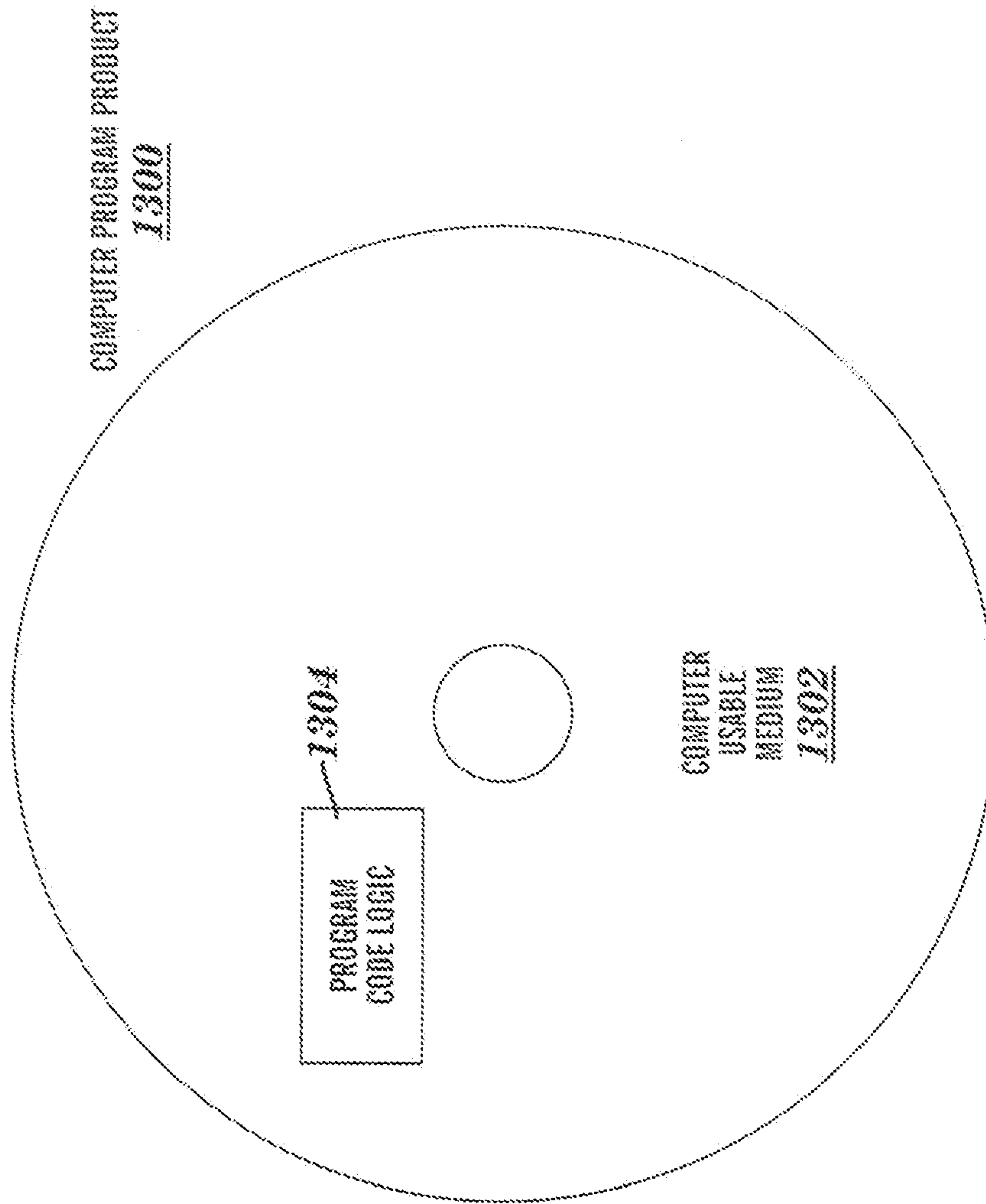
1100

DEFINITION	WORD	BYTE 0	1	2	3
1110<STATUS	00	STATUS FOR THE I/O OPERATION			
	01				
	02				
	03				
	04				
	05				
	06				
	07				
1120<EXTENDED STATUS	08	ES LENGTH 1121	ES FLAGS 1122	STATUS FIELD FOR THE I/O OPERATION	1123
	09	STATUS FIELD FOR THE I/O OPERATION			
	10	RESERVED			
	11	FORMAT 1126	FLAGS 1127	CONTROL UNIT STATE 1128	DEVICE STATE 1129
	12	OPERATION STATE 1130	RESERVED		
	13	STATE DEPENDENT INFORMATION			
	14				
	15				
	16	DEVICE LEVEL IDENTIFIER			
	17	1150			
	18	DEVICE DEPENDENT DATA			
	19				
	20				
	21				
	22				
	23				
	1160				

FIG. 11

FIG. 12A

**FIG. 12B**

**FIG. 13**

1265 — TIMER POP (ONCE PER SECOND)

FIG. 12A

