(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 November 2003 (13.11.2003)

PCT

(10) International Publication Number
WO 03/094403 A2

(51) International Patent Classification[7]: H04L

(21) International Application Number: PCT/US03/13049

(22) International Filing Date: 28 April 2003 (28.04.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/135,681 30 April 2002 (30.04.2002) US
10/135,612 30 April 2002 (30.04.2002) US

(71) Applicant: TRANSWITCH CORPORATION [US/US]; 3 Enterprise Drive, Shelton, CT 06484 (US).

(72) Inventors: SHANLEY, Timothy, M.; 654, Chestnut Ridge Road, Orange, CT 06477 (US). PRESTON, Thomas, M.; 209 Bioski Road, Middlebury, CT 06762 (US). PARRELLA, Eugene, L.; 8 Jennings Lane, Whitehouse Station, NJ 08889 (US). SRINIVASAN, Desikan, V.; 151, Andrew Avenue, Apt. 200, Naugatuck, CT 06770 (US).

(74) Agents: GORDON, David, P. et al.; 65 Woods End Road, Stamford, CT 06905 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/094403 A2

(54) Title: AN ATM DEVICE INCORPORATING METHODS AND APPARATUS FOR INCREASING THE NUMBER OF UTOPIA PORTS AND METHOD AND APPARATUS FOR AVOIDING HEAD OF LINE BLOCKING

(57) Abstract: An ATM device includes means for increasing the number of UTOPIA ports and means for avoiding head of line blocking. Data for a plurality of UTOPIA PHY ports are multiplexed over a first UTOPIA PHY port and backpressure information is provided to the ATM device via a second UTOPIA PHY port. The backpressure information is preferably formatted in a single 56-byte UTOPIA cell. The means for avoiding head of line blocking includes a scheduler, at least one multicast queue, at least one unicast queue, a multicast session table, a multicast timer, and a problem PHY vector. Scheduling is alternated between multicast queue(s) and unicast queue(s). If a PHY device in a multicast session is inactive, it is skipped and the next PHY in the session is serviced. When the session has serviced all of the active PHYs and there remain only inactive PHYs in the session table, the session is ended.

AN ATM DEVICE INCORPORATING METHODS AND APPARATUS FOR INCREASING THE NUMBER OF UTOPIA PORTS AND METHOD AND APPARATUS FOR AVOIDING HEAD OF LINE BLOCKING

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to telecommunications. More particularly, the present invention relates to the passing of high speed Asynchronous Transfer Mode (ATM) data or packet data over a standardized Universal Test and Operations Physical Interface for ATM (UTOPIA) bus and to methods and apparatus for buffering ATM cells, particularly during multicasting.

2. State of the Art

ATM provides a mechanism for removing performance limitations of local area networks (LANs) and wide area networks (WANs) and provides data transfers at a speed of on the order of gigabits/second. Within the ATM technology, a commonly used interface specification between chips on a board for passing ATM cells is the UTOPIA (Universal Test & Operations PHY Interface for ATM) interface. The UTOPIA interface is specified in ATM Forum standard specifications, including: af-phy-0017.000 (UTOPIA Level 1, Version 2.01 March 21, 1994); af_phy_0039.000 (UTOPIA Level 2, Version 1, June 1995); and af-phy-00136.000 (UTOPIA 3 Physical Layer Interface November 1999) which are hereby incorporated by reference herein in their entireties. A typical application of the UTOPIA interface is supporting the connection between an ATM network processor and various PHY devices such as a DSL chip set and/or a SONET framer. UTOPIA is also used as the interface between a switch fabric and an ATM network processor.

UTOPIA supports three operation modes: single PHY operation mode, Multiple PHY (MPHY) with Direct Status Indication operation mode and MPHY with Multiplexed Status Polling operation mode. In the single PHY mode, the UTOPIA interface includes a data bus

and a control bus. The operation of UTOPIA in the single PHY mode is relatively simple and straightforward. In MPHY operation mode, the UTOPIA interface includes a data bus, a control bus and an address bus. MPHY with Multiplexed Status Polling is used in most applications.

The MPHY UTOPIA transmit interface includes the following signals: transmit data (TxData); transmit address (TxAddr); and the transmit control signals transmit cell available (TxClav), transmit enable (TxEnb*) and transmit start of cell (TxSOC). The receive interface includes the following signals: receive data (RxData); receive address (RxAddr); and the receive control signals receive cell available (RxClav), receive enable (RxEnb*) and receive start of cell (RxSOC). A MPHY device may consist of multiple PHY ports, each PHY port having a one-to-one correspondence with a PHY Port address that is related to a UTOPIA address and Clav (Cell buffer available) signal.

Prior art Figure 1 illustrates an example of a UTOPIA Level 2 interface supporting MPHY with Multiplexed Status Polling operation. As shown in Figure 1, a transmit clock signal (TxClk) is used to clock control signals and data signals in the transmit direction (from the ATM device to the PHY devices). The TxData[15:0] signal is a 16-bit UTOPIA transmit data bus. The assertion of TxEnb* is coincident with the start of the cell transfer. TxSOC is used to indicate the start of cell position. TxClav is used to indicate that the PHY layer device is ready to receive a cell from the ATM layer device. TxAddr[4:0] is the UTOPIA address and is used to poll and select the appropriate MPHY device.

At the UTOPIA transmit interface, the ATM layer device polls the TxClav status of a PHY layer device by placing a specified address on the TxAddr bus for one clock cycle. The PHY layer device which is associated with the address on the TxAddr bus drives TxClav high (or low) during the next clock cycle during which the ATM device places a null address (1F) on the TxAddr bus. The ATM layer device checks TxClav at a certain time after it issues TxAddr. Based on polled TxClav information, the ATM layer device can select a PHY device and transfer data to this PHY device by driving TxEnb* and TxSOC signals.

Similarly, RxClk is the receive clock signal that is used to clock control signals and data in the receive direction (from the PHY device to the ATM device). RxData[15:0] is a 16-bit UTOPIA Receive bus. The assertion of RxEnb* is coincident with the start of the cell transfer. RxSOC is used to indicate the start of cell position. RxClav is used to indicate that the PHY layer device is ready to Receive a cell from the ATM layer device. RxAddr[4:0] is the UTOPIA address of the PHY device and is used by the ATM device to poll and select the appropriate PHY device in the receive direction.

At the UTOPIA receive interface, the ATM layer device polls the RxClav status of a PHY layer device by placing a specified address on RxAddr bus for one clock cycle. The PHY layer device which is associated with the address on the RxAddr bus drives RxClav high (or low) during the next clock cycle during which the ATM device places a null address (1F) on the RxAddr bus. The ATM layer device checks RxClav at a certain time after it issues RxAddr. Based on polled RxClav information, the ATM layer device can select a PHY device and receive data from this PHY device by driving the RxEnb* signal.

The number of PHY ports supported by a UTOPIA interface is generally fixed in the design of the device incorporating the UTOPIA interface. For example, the ASPEN® access processor device from Transwitch Corporation, Shelton, CT provides a UTOPIA interface for sixteen PHY layer devices to a CellBus® ATM switch.

In certain applications, it is desirable to use a particular ATM device which does not provide the desired number of UTOPIA PHY ports. In these situations, it would be desirable to provide a way to increase the number of PHY ports without significantly altering the ATM device.

ATM, by nature, is "bursty". Consequently, buffers must be provided in ATM devices so that cell loss is minimized. If one buffer is shared by more than one physical destination (PHY), an adverse effect known as "head of line blocking" can occur. Head of line blocking occurs when a cell at the head of a buffer cannot be transmitted to its PHY because of

any number of reasons. This cell then blocks the transmission of all of the cells behind it. Head of line blocking can be avoided by providing separate buffers for each PHY in an ATM device. However, this can be costly and space consuming.

In an ATM network, it is often desirable to effect a multicast of ATM cells; i.e., to transport ATM cells from a source terminal to a plurality of different destinations. Each of the destinations of the multicast will typically have its own address. Thus, it is necessary to duplicate the ATM cells, provide different headers for each of the cells, and send the cells out on different virtual circuits (VCs). The different VCs may be located at different PHYs in the case of a spatial multicast or the same PHY in the case of a logical multicast. In the case of spatial multicast, extensive buffering may be necessary in order to accommodate all of the copies of each incoming multicast cell. It will be appreciated that the outgoing buffers will rapidly fill with copies of each single incoming multicast cell. In order to reduce the amount of buffer space required for multicasting, it is known to use a single buffer for multicast incoming cells and to replicate the cells just as they are ready to be transmitted out of the switch. Although this saves buffer space, it makes head of line blocking a more likely occurrence.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to provide methods and apparatus for increasing the number of UTOPIA PHY ports in an ATM device.

It is also an object of the invention to provide methods and apparatus for increasing the number of UTOPIA PHY ports in an ATM device without significantly modifying the device.

It is another object of the invention to provide methods and apparatus for preventing head of line blocking in an ATM device.

Still another object of the invention is to provide methods and apparatus for preventing head of line blocking in an ATM device which do not require extensive use of buffer memory.

In accord with these objects which will be discussed in detail below, the methods of the present invention include multiplexing up to sixty-four UTOPIA PHY ports over a single UTOPIA PHY port. In order to prevent cell loss, the methods of the invention include providing backpressure information to the ATM device via a dedicated UTOPIA PHY port. The backpressure information is preferably formatted in a single 56-byte UTOPIA cell.

The presently preferred apparatus of the invention includes a sixty-four port UTOPIA Level 2 interface for coupling to up to sixty-four PHY devices, a two port UTOPIA Level 2 interface for coupling to the ATM device, and various buffers and controllers for controlling the flow of data between the two UTOPIA Level 2 interfaces. One of the ports in the two port UTOPIA Level 2 interface is used for configuration and control and the other is used for data. The various buffers and controllers include three rate decoupling FIFOs, a congestion status cell buffer, a multicast session table, an enqueueing control, an SRAM control, and a round robin scheduler with queue status. The apparatus is preferably implemented as a field programmable gate array (FPGA) or application specific integrated circuit (ASIC) and is provided with an external (32Kx32) SRAM as well as inlet and outlet clocks.

Data entering the apparatus through the sixty-four port UTOPIA interface is buffered in a four cell rate decoupling FIFO (RDF). When a cell enters the RDF, a two byte routing tag is prepended to the front of the cell identifying the source port ID. The ATM device is immediately notified (as soon as the entire cell has been stored) that a cell is available to be read out from the RDF. Cells written into the RDF are preferably immediately available to be clocked out to the ATM device. Preferably, UTOPIA address 0 is used for the data port and UTOPIA address 1 is used for the control port. This insures control path integrity under

heavy traffic load conditions. If the RDF fills, the sixty-four port UTOPIA interface is notified to stop requesting cells from the PHYs until a cell slot becomes available in the RDF.

According to an embodiment of the invention, the ATM device is provided with sufficient RAM and programmed to maintain 256 unicast service category queues (4 per port), and 256 multicast queues. The ATM device is also programmed to maintain a congestion table which indicates congestion status for the multicast queue, and unicast queues. The invention is illustrated with reference to the aforementioned ASPEN® ATM device. The cells entering the ASPEN® ATM device from the CellBus® interface are automatically enqueued by a Tandem Routing Header. Using outlet queue state and the congestion status supplied by the apparatus of the invention, the ASPEN® rate processor (RP) dequeues cells toward apparatus of the invention. The cells pass through the outbound processor (OP) where they go through connection table lookup, header translation, and statistics maintenance before having a routing tag prepended for use by apparatus of the invention. The apparatus of the invention uses the routing tag for final port queuing before forwarding traffic to the appropriate PHY device.

A preferred apparatus of the present invention includes a UTOPIA interface, a scheduler, at least one multicast queue, at least one unicast queue, a multicast session table, a multicast timer, and a problem PHY vector. The methods of the invention include alternate scheduling between multicast queue(s) and unicast queue(s). In particular, the PHY devices are serviced in round robin or other fair scheduling order. According to one embodiment, which is not the presently preferred embodiment, as each PHY is serviced, it is determined whether there exists a unicast cell or a multicast cell or both for this PHY. If both unicast and multicast cells are scheduled for this PHY, scheduling is alternated between them. If only unicast or multicast cells are scheduled for this PHY, alternation is not necessary.

For purposes of this invention, the act of replicating a multicast cell to plural PHY destinations is referred to as a multicast "session" and the identities of the PHY destinations are stored in a multicast session table for each session. The copying of the cell to one of the PHYs in the multicast session is referred to as a "leaf" in the session. According to the first

embodiment, a multicast timer is started when a multicast cell reaches the head of the multicast queue. The timer is a count down timer preferably based on the slowest PHY device. If a PHY device in a session is inactive, it is skipped and the next PHY in the session is serviced. The session ends when one of three events occurs: all PHYs in the session table have been serviced, the timer expires, or the only PHYs remaining in the session are PHYs listed in the problem PHY vector. At the end of each multicast session, the problem PHY vector is updated. The problem PHY vector includes a list of all of the PHYs which are deemed to be presently inactive based on the last multicast session and all previous multicast sessions. The problem PHY vector is updated whenever an inactive PHY becomes active, either in a unicast or a multicast cell transfer. The problem PHY vector is preferably used to shorten the multicast session before the timer expires. It may also be used by an external device to modify multicast session tables.

According to a presently preferred embodiment of the invention, the servicing of PHYs is driven by the status of the queues. In a background process, the status of the unicast and multicast queue is repeatedly updated. The status of PHYs is obtained through the UTOPIA interface. If the multicast queue is not empty and the last queue serviced was a unicast queue, the multicast queue is serviced by copying the head of line cell to the next active PHY in the multicast session (i.e. the next leaf of the session). If the multicast queue is empty or if the last cell serviced was not a unicast cell, the next (in round robin) available unicast queue is serviced. As used herein, "available unicast queue" means a queue with a cell ready to be sent to an active PHY which is not part of the current multicast session. During the multicast session, if the only PHYs remaining in the session table are PHYs which are in the problem PHY vector, the session is ended and the problem PHY vector is updated. If the only PHYs remaining in the multicast session include inactive PHYs which are not in the problem PHY vector, the multicast timer is started. The scheduler continues to attempt to complete the multicast session until the timer expires or until the only PHYs left are in the problem PHY vector. When the timer expires, the session is ended and the problem PHY vector is updated.

Additional objects and advantages of the invention will become apparent to those skilled in the art upon reference to the detailed description taken in conjunction with the provided figures.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a simplified block diagram of a prior art UTOPIA interface;

Figure 2 is a simplified block diagram of an apparatus according to the invention;

Figure 3 is a simplified block diagram of an apparatus according to the invention together with an ASPEN® ATM device and associated RAM;

Figure 4 is a high level block diagram illustrating how the principles of the invention may be applied to any ATM layer device to increase the number of UTOPIA ports so that more PHY devices may be serviced;

Figure 5 is a high level schematic block diagram illustrating a portion of apparatus according to the invention for avoiding head of line blocking;

Figure 6 is a high level simplified flow chart illustrating scheduling methods according to the one embodiment of the invention;

Figure 7 is a high level simplified flow chart illustrating multicast session handler methods according to one embodiment of the invention; and

Figure 8 is a high level simplified flow chart illustrating the presently preferred scheduling methods of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to Figure 2, a UTOPIA port expander 10 according to the invention includes a sixty-four port UTOPIA Level 2 interface 12 for coupling to up to sixty-four PHY devices and a two port UTOPIA Level 2 interface 14 for coupling to an ATM device. The presently preferred port expander also includes three rate decoupling FIFOs 16, 18, 20, a congestion status cell buffer 22, a multicast session table 24, an enqueueing control 26, an SRAM control 28, a round robin scheduler with queue status 30, an inlet and outlet global clock distributor 32, and a multiplexer 34. The apparatus 10 is preferably provided with external (32Kx32) RAM 36 as well as transmit and receive clock sources (not shown). The sixty-four port UTOPIA Level 2 interface 12 receives input from the three cell rate decoupling FIFO 18 and provides output to the four cell rate decoupling FIFO 20. The two port UTOPIA Level 2 interface 14 receives input from both the four cell rate decoupling FIFO 20 and the congestion/status cell buffer 22 via the multiplexer 34 and provides output to the three cell rate decoupling FIFO 16. The enqueuing control 26 receives port ID from the FIFO 16 and communicates with the multicast session table 24 as described in the previously incorporated co-owned application. The FIFO 16 and the enqueuing control 26 provide input to the SRAM controller 28 which is coupled to the external RAM 36 where individual queues are set up as described in more detail below with reference to Figure 3. The SRAM controller 28 communicates with the round robin scheduler and queue status 30 which schedules cells from queues to the rate decoupling FIFO 18 and delivers backpressure information cells to the congestion status cell buffer 22.

For the purpose of discussion herein, data flow in the direction from the sixty-four port UTOPIA Level 2 interface 12 to the two port UTOPIA Level 2 interface 14 shall be referred to as the "upstream" data flow and data flow in the opposite direction shall be referred to as the "downstream" data flow.

Turning now to Figure 3, the UTOPIA port expander 10 according to the invention is illustrated together with an ASPEN® ATM device 40 and associated RAM 36, 42. The

ASPEN® ATM device 40 includes a sixteen port UTOPIA Level 2 interface 44, a CellBus® interface 46, an inbound processor 48 with an associated rate decoupling FIFO 50, an outbound processor 52 with two associated rate decoupling FIFOs 54, 56, a rate processor 58 and an internal bus 60. Two of the sixteen UTOPIA ports 44 are coupled to the two port UTOPIA interface 14 of the apparatus 10. Preferably, UTOPIA address 0 is used for the data ports and UTOPIA address 1 is used for the control ports. The CellBus® interface 46 is used to couple the ASPEN® ATM device 40 to one or more other ATM devices (not shown). The inbound processor 48 is responsible for header lookup, header translation, backpressure message routing, usage parameter control (UPC), statistics, and overhead and maintenance (OAM). The outbound processor 52 is responsible for header translation, assignment of routing tags, statistics and OAM. The rate processor 58 is responsible for inlet scheduling, outlet multicast scheduling, and outlet scheduling for the sixty-four ports 12.

According to an embodiment of the invention, the ASPEN® ATM device 40 is provided with sufficient RAM 42 and programmed to maintain 256 unicast service category outlet queues (four per port, each of the four representing a different class of service), 256 multicast outlet queues, and four shared service class inlet queues. The ASPEN® ATM device 40 is also programmed to maintain a congestion table (not shown) which indicates congestion status for the downstream multicast queues and unicast queues in the RAM 36 of the port expander device 10. .

Upstream data from all of the ports 12 is buffered in the four cell rate decoupling FIFO 20. When a cell enters the FIFO 20, a two byte routing tag is prepended to the front of the (fifty-four byte) cell identifying the source port ID. Although the tag is two bytes, the first ten bits are padded zeros and the last six bits identify one of the sixty-four (0-63) ports. The ASPEN® ATM device 40 is immediately notified (as soon as the entire cell has been stored in the FIFO 20) that a cell is available to be read out from the FIFO 20. Cells written into the FIFO 20 are preferably immediately available to be clocked out to the ASPEN® ATM device 40. If the FIFO 20 fills, the sixty-four port UTOPIA interface is notified to stop requesting cells from the PHYs until a cell slot becomes available in the FIFO 20. Upstream data enters

the ASPEN® ATM device 40 to the rate decoupling FIFO 50 and passes through the inlet processor 48. The inlet processor 48 forwards backpressure messages to the rate processor 58, discards cells which were misdelivered. The inlet processor 48 also reads the cell header ionformation and forwards cells to the appropriate PHY via the CellBus® switch fabric 46.

Data in the downstream direction from the CellBus® interface 46 is automatically enqueued by a Tandem Routing Header pursuant to the CellBus® protocol. Based on outlet queue state and the congestion status supplied by the backpressure control 22, the rate processor 58 dequeues cells from the RAM 42 to the rate decoupling FIFO 54. The cells pass through the outbound processor 52 where they go through connection table lookup, header translation, and statistics maintenance before having a two-byte routing tag (or multicast session ID) prepended. The apparatus 10 of the invention uses the routing tag (or multicast session ID) for final port queuing before forwarding traffic to the appropriate PHY device. The two-byte tag used in the downstream direction is similar but not identical in format to the tag used in the upstream direction. In both the downstream and upstream direction, the six least significant bits of the second byte of the two-byte tag indicate the PHY ID. In the downstream direction, the least significant bit of the first byte of the tag is a multicast indicator. If that bit is set to "1", all eight bits of the second byte are used for the multicast session ID.

According to the invention, the two port UTOPIA interface 14 of the UTOPIA port expander 10 acts in slave mode to the master mode of the UTOPIA interface 44 of the ASPEN® ATM device 40. The UTOPIA interface 12 of the apparatus 10 acts in master mode relative to the PHY devices (not shown).

As mentioned above, the apparatus 10 periodically generates a backpressure message which is formatted in a (fifty-six byte) UTOPIA cell. Table 1 below illustrates the format of the backpressure message.

0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Padding							M	PHY ID/ Multicast Session ID							
1	GFC				VPI								VCI(15:12)			
2	VCI(11:0)												PTI			CLP
3	HEC								Padding							
4																
5	Message ID								Message Sub ID							
6																
7	MUSE	MUPE	SUSE	SUPE	Port# multicast timeout							PMCD	mcast port cells sent			
8	Port 7 cells sent				Port 6 cells sent				Port 5 cells sent				Port 4 cells sent			
9	Port 3 cells sent				Port 2 cells sent				Port 1 cells sent				Port 0 cells sent			
10	Port 15 cells sent				Port 14 cells sent				Port 13 cells sent				Port 12 cells sent			
11	Port 11 cells sent				Port 10 cells sent				Port 9 cells sent				Port 8 cells sent			
12	Port 23 cells sent				Port 22 cells sent				Port 21 cells sent				Port 20 cells sent			
13	Port 19 cells sent				Port 18 cells sent				Port 17 cells sent				Port 16 cells sent			
14	Port 31 cells sent				Port 30 cells sent				Port 29 cells sent				Port 28 cells sent			
15	Port 27 cells sent				Port 26 cells sent				Port 25 cells sent				Port 24 cells sent			
16	Port 39 cells sent				Port 38 cells sent				Port 37 cells sent				Port 36 cells sent			
17	Port 35 cells sent				Port 34 cells sent				Port 33 cells sent				Port 32 cells sent			
18	Port 47 cells sent				Port 46 cells sent				Port 45 cells sent				Port 44 cells sent			
19	Port 43 cells sent				Port 42 cells sent				Port 41 cells sent				Port 40 cells sent			

20	Port 55 cells sent				Port 54 cells sent				Port 53 cells sent				Port 52 cells sent			
21	Port 51 cells sent				Port 50 cells sent				Port 49 cells sent				Port 48 cells sent			
22	Port 63 cells sent				Port 62 cells sent				Port 61 cells sent				Port 60 cells sent			
23	Port 59 cells sent				Port 58 cells sent				Port 57 cells sent				Port 56 cells sent			
24	P63	P62	P61	P60	P59	P58	P57	P56	P55	P54	P53	P52	P51	P50	P49	P48
25	P47	P46	P45	P44	P43	P42	P41	P40	P39	P38	P37	P36	P35	P34	P33	P32
26	P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16
27	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00

TABLE 1

As illustrated in Table 1, the first word (0) of the UTOPIA cell includes the PHY address or the multicast session ID as described above. The next five words (1-5) of the cell contain ATM routing information. Word (6) is not used. Word (7) includes one bit indicators for MUSE, MUPE, SUSE, and SUPE, a seven bit Port# multicast timeout, a one bit PMCD indicator and a four bit indicator of multicast port cells sent. MUSE refers to "Master Utopia SOC error(s)" occurred. This bit remains asserted until the host clears the SOC counter. MUPE refers to "Master Utopia Parity error(s)" occurred. This bit remains asserted until the host clears the parity counter. SUSE refers to "Slave Utopia SOC error(s)" occurred. This bit remains asserted until the host clears the SOC counter. SUPE refers to "Slave Utopia Parity error(s)" occurred. This bit remains asserted until the host clears the parity counter. The Port# multicast timeout identifies the last port number to experience a multicast timeout error. Port numbers 0-3Fh are valid port numbers. Port number 7Fh indicates that no discard occurred between the last two backpressure messages. PMCD refers to "Port multicast discard(s)" occurred. This bit remains asserted until host clears the multicast discard counter.

The "mcast port cells sent" is a 4-bit rollover counter that increments each time a cell is dequeued downstream. Similarly, words (8) through (23) of the cell contain 4-bit rollover

counters for each of the sixty-four downstream port queues. The ASPEN® scheduler in the rate processor (58 in Figure 3) maintains its "sent" counts and compares them to the counts provided in the backpressure cells to determine the port queue fill levels. These counters are also be incremented if a cell is discarded due to congestion. They each have an initial value=0.

Words (24) through (27) of the cell include one bit cell discard indicators for each of the sixty-four downstream ports. These bits are asserted when a port discards a cell and remain asserted until the host clears the discard counter associated with the port.

Backpressure cells are generated periodically by the apparatus (10 in Figure 3) to support a closed-loop scheduler between the ASPEN® device 40 and the downstream UTOPIA ports (12 in Figure 3). The UTOPIA port expander (10 in Figure 3) is designed to handle up to 8Mb/s data rates for each of the sixty-four ports. A worst case analysis with a maximum data rate of 10 Mb results in a cell transfer rate of approximately 23,585 cells per second which is approximately 42.4×10^{-6} seconds per cell. With a buffer of 16 cells deep for each port, a backpressure message update should be sent every 8 cells or 339×10^{-6} seconds. Backpressure cells are sent to the ASPEN® device through UTOPIA port 1 with a PHY ID=00h, an ATM header of unassigned cell(VPI=0,VCI=0,CLP=0), Message ID=0, and Message Sub ID=0.

As mentioned above, configuration, control, and status communication is also passed through UTOPIA port 1 via the outbound processor (52 in Figure 3). These messages are also contained in 56-byte UTOPIA cells.

Referring now to Figure 4, those skilled in the art will appreciate that the methods and apparatus of the invention can be applied to virtually any ATM layer device to increase the number of UTOPIA ports of the device. Figure 4 illustrates how a port expander device 110 according to the invent can be coupled to an ATM layer device 140 and a plurality of PHY devices 112a-112n. The ATM layer device 140 (e.g. ATM traffic processor) will typically

include a plurality of ATM traffic queues (not shown) which do not utilize the previously described tandem routing header and a switch fabric (not shown) which does not utilize the previously described CellBus® technology. The device 140 will typically also include upstream and ddownstream cell processors (not shown) which are different from the inbound and outbound processors of the previously described ASPEN® device and an ATM cell scheduler (not shown) which is different from the rate processor of the previously described ASPEN® device.

Referring now to Figure 5, an ATM device 210 incorporating the scheduling aspect of the invention includes one multicast queue 212 and a plurality of unicast queues 214a, 214b, ..., 214n which are implemented as FIFO buffers in RAM. The queues are multiplexed by multiplexer 216 onto a UTOPIA level 2 interface 216a, 216b to a plurality of PHY devices (also known as ports, not shown). According to the presently preferred embodiment, sixty-four unicast queues are supported, one for each destination PHY. A scheduler 218 is coupled to the multiplexer 216 and arbitrates transmission of cells from the queues onto the UTOPIA level 2 data path 216a based on port status received via UTOPIA level 2 polling results 216b and queue status 215 received from the queues 212, 214a, 214b,...,214n. The scheduler 218 is preferably implemented as a state machine. According to the a first, and not presently preferred, embodiment of the scheduling methods of the invention, the queues are serviced in a round robin fashion according to destination PHY. Unicast cells are sent from their queues to the output 216a if their destination port is available as indicated by the port status at 216b. If there exists both a multicast cell and a unicast cell for a particular PHY, access to that PHY is alternated between the unicast flow and the multicast flow.

When servicing the multicast queue, the scheduler 218 utilizes a multicast session table 220, a multicast timer 222, and a problem PHY vector 224. The multicast session table is preferably implemented in RAM and includes two hundred fifty-six session entries. Each session entry is preferably a sixty-four bit string indicating which of the sixty-four ports are participating in the multicast session. A multicast session is defined as the process of copying a cell from the multicast queue to all of the PHYs (if possible) indicated by the corresponding

multicast session table entry. Each cell in the multicast queue includes a pre-pended word indicating which multicast session table entry is to be used for copying the cell to multiple PHYs. The multicast timer 222 is a count down timer which is started each time a multicast cell reaches the head of the multicast queue 212. The duration of the timer 222 is preferably based on the data rate of the slowest destination PHY. The problem PHY vector is preferably a sixty-four bit string which indicates which PHYs are presently inactive.

The methods of the first embodiment of the method utilized by the scheduler 218 for servicing the multicast and unicast queues are illustrated by way of example in the flow charts of Figures 6 and 7.

Referring now to Figure 6, for each PHY, starting at 2100, the scheduler determines at 2102 whether there is a multicast cell in the multicast queue which is scheduled for this PHY. This determination is made by determining whether the multicast queue is empty, and if it is not empty, looking up the entry in the multicast session table corresponding to the cell in the multicast queue. If there is not any multicast cell destined for this PHY, the scheduler determines at 2104 whether there is a unicast cell in the unicast queue corresponding to this PHY and whether the PHY is responding. If it is determined at 2104 that there is a unicast cell and the PHY is ready to receive, the scheduler causes the transmission of the unicast cell and removes it from the queue at 2106, the proceeds to the next PHY at 2100. If it is determined at 2104 that there is no unicast cell in the queue or that the PHY is not ready to receive, the scheduler proceeds to the next PHY at 2100.

If it is determined at 2102 that a multicast cell is in the multicast queue and its session table entry includes this port, the scheduler determines at 2108 whether the last cell sent to this port was a unicast cell. If it was not, the scheduler determines at 2110 whether there is a unicast cell in the queue for this port and whether the port is ready to receive. If it is determined at 2110 that there is a unicast cell ready to be sent, the scheduler sends the cell and removes it from the queue at 2106, then proceeds to the next PHY at 2100. If it is determined at 2108 that the last cell sent to this PHY was a unicast cell, or if it is determined at 2110 that

no unicast cell is available to send, a multicast handler is called at 2112. This is done even though the port may not be ready to receive a cell because the multicast handler needs to take note of which ports are inactive. The scheduler then waits at 2114 until the multicast handler has completed its task for this port before proceeding to the next at 2100.

Figure 7 illustrates the operation of the multicast handler. After starting at 2200, the multicast handler waits at 2202 to be called upon by the scheduler described above with reference to Figure 6. When it is determined at 2202 that a multicast cell is ready to be sent, it is first determined at 2204 whether a multicast session is already in progress. If this is the start of a new session, the session table is read and the session is set up at 2206. Once a session is set up or is in progress, the multicast timer is checked at 2208 to see if it has expired. When the timer is expired, the pending cell is removed from the multicast queue at 2210, the problem PHY vector is updated at 2212, and the scheduler is notified at 2214 that the task is complete before returning to 2200 to wait to be called again by the scheduler.

If it is determined at 2208 that the timer has not expired, it is then determined at 2216 whether the PHY to which the multicast cell is to be copied is ready to receive. If it is, the cell is copied to the PHY at 2218. After the cell is sent, or if it is determined at 2216 that the PHY is not responding (is not able to receive a cell), it is then determined whether the session should be ended. In particular, it is determined at 2220 whether the only PHYs remaining in the session are in the problem PHY vector. If that is the case, the session is ended by removing the cell from the queue at 2210, updating the problem PHY vector at 2212, and returning control to the scheduler at 2214. The session is also terminated if it is determined at 2222 that all of the PHYs in the session have been serviced.

According to the invention, the problem PHY vector can be used by an external device to alter the multicast session tables and/or to change the duration of the multicast session timer.

One of the advantages of the problem PHY vector is that a problem PHY causes a time-out only once. Thereafter, it is listed in the problem PHY vector and will be treated as if it were not listed in the session table entry.

Turning now to Figure 8, the presently preferred methods of the invention schedule cells based primarily on queue status and secondarily on PHY status. Starting at 2300, queue status is obtained at 2302 and it is determined at 2304 whether the multicast queue is empty. If there is no cell at the head of the multicast queue, it is determined at 2306 whether there is a unicast cell ready to be sent. The determination at 2306 includes determining which unicast queues have cells to be sent, which PHYs are active, and which unicast queue was last serviced. If it is determined that there are unicast cells ready to be sent, the appropriate cell is dequeued at 2308. According to the presently preferred embodiment, the cell dequeued at 2308 is the cell from the next unicast queue (in round robin) which has a cell ready to be sent to an active PHY which is not a PHY in a pending multicast session. If there is no unicast cell ready as determined at 2306, the scheduler returns to 2302. If a unicast cell is ready as determined at 2306, the cell is sent at 2308 and the scheduler returns to 2302 and processes the next queue.

If it is determined at 2304 that the multicast queue is not empty, it is then determined at 2310 whether the last cell sent was a unicast cell. According to the preferred embodiment of the invention, when the multicast queue is not empty, unicast cells are multiplexed 1:1 with multicast cells by the scheduler. Thus, if the last cell was not a unicast cell, it is determined whether a unicast cell is ready to be sent at 2307. If it is ready, the unicast queue is serviced at 2308 before the multicast queue is serviced. If the last cell sent was a unicast cell as determined at 2310, or if no unicast cells are ready as determined at 2307, a multicast session table is opened at 2312, if one is not already in progress. Although the multicast timer may not yet have been set, for simplicity, Figure 8 shows the timer being checked at 2314. If the timer is not expired, it is determined at 2316 whether all of the PHYs remaining in the session are inactive. If there are active PHYs remaining in the session, the multicast cell is copied at 318 to the next active PHY in the session and, if the timer had been running, it is stopped, but

not reset. At 2320, it is determined whether all of the leafs in the multicast session have been serviced. If they have, the session is closed and the problem PHY vector is updated at 2322 before the process returns to 2302. If leafs remain in the session as determined at 2320, the session is not closed and the process returns to 2302.

If, during a multicast session, it is determined at 2316 that the only PHYs remaining in the session are inactive, it is then determined at 2324 whether all of these inactive PHYs are listed in the problem PHY vector. If all of the remaining inactive PHYs are listed in the problem PHY vector, the session is ended at 2322. If at least one of the inactive PHYs remaining in the session is not listed in the problem PHY vector, the multicast timer is started at 2326, if it is not already running and the process returns to 2302. As mentioned above, if, during a multicast session, the multicast timer expires as determined at 2314, the session is ended and the problem PHY vector is updated to include the inactive PHY(s) which remained on the session table when the timer expired.

According to the presently preferred embodiment, during a multicast session, ports in the session table are "checked off" when they are serviced in order to make the determinations at 2316 and 2324. Though not shown in Figure 8, the problem PHY vector is also updated whenever a PHY in the vector displays a UTOPIA CLAV signal.

There have been described and illustrated herein methods and apparatus for increasing the number of UTOPIA ports in an ATM device. While particular embodiments of the invention have been described, it is not intended that the invention be limited thereto, as it is intended that the invention be as broad in scope as the art will allow and that the specification be read likewise. Thus, while particular hardware and software have been disclosed, it will be appreciated that other hardware and software could be utilized so long as the functional requirements of the invention are met. Also, while the apparatus of the invention has been shown in conjunction with an ASPEN® ATM device, it will be recognized that the invention could be used to increase the number of UTOPIA ports in other types of ATM devices. Moreover, while particular configurations have been disclosed in reference to the number of

UTOPIA ports provided by the invention, it will be appreciated that other configurations could be used as well to support more or fewer ports. It will therefore be appreciated by those skilled in the art that yet other modifications could be made to the provided invention without deviating from its spirit and scope as so claimed.

There have also been described and illustrated herein several embodiments of a methods and apparatus for avoiding head of line blocking in an ATM device. While particular embodiments of the invention have been described, it is not intended that the invention be limited thereto, as it is intended that the invention be as broad in scope as the art will allow and that the specification be read likewise. Thus, while particular method steps have been disclosed in particular order, it will be appreciated that some variation in the order of the steps will produce substantially the same results. It will be appreciated that, depending on the hardware implementation, some steps may be performed simultaneously. Also, while a specific number of queues and table entries have been shown, it will be recognized that other numbers of queues and table entries could be used with similar results obtained. It will therefore be appreciated by those skilled in the art that yet other modifications could be made to the provided invention without deviating from its spirit and scope as so claimed.

Claims:

1. A method for increasing the number of UTOPIA ports in an ATM device, said method comprising:

a) multiplexing data for n ports over a first UTOPIA PHY port from the ATM device; and

b) providing backpressure information for each of the n ports over a second UTOPIA PHY port to the ATM device.

2. The method according to claim 1, wherein:

n=sixty-four.

3. The method according to claim 1, wherein:

the backpressure information is formatted in a UTOPIA cell.

4. The method according to claim 1, further comprising:

c) providing a buffer for each of the n ports.

5. The method according to claim 4, wherein:

the backpressure information includes an indication of the number of cells dequeued from each buffer.

6. The method according to claim 5, wherein:

the backpressure information includes an indication of the number of cells discarded from each of the buffers.

7. The method according to claim 4, further comprising:

d) providing a single multicast buffer to be shared by up to n number of the n number of ports.

8. The method according to claim 7, wherein:

the backpressure information includes an indication of the number of cells dequeued from the multicast buffer.

9. The method according to claim 7, wherein:

the backpressure information includes an indication of whether a cell in the multicast buffer has been discarded.

10. The method according to claim 7, wherein:

the backpressure information includes an identification of the last port to experience a multicast timeout error.

11. An apparatus for increasing the number of UTOPIA ports in an ATM device, said method comprising:

a) a first UTOPIA interface adapted to be coupled to n PHY devices;

b) a second UTOPIA interface adapted to be coupled to the ATM device, said second UTOPIA interface having a first port for receiving data from the ATM device and a second port for sending backpressure information about each of the n ports to the ATM device.

12. The apparatus according to claim 11, wherein:

n=sixty-four.

13. The apparatus according to claim 11, wherein:

the backpressure information is formatted in a UTOPIA cell.

14. The apparatus according to claim 11, further comprising:

c) n buffers, one buffer for each of the ports.

15. The apparatus according to claim 14, wherein:

the backpressure information includes an indication of the number of cells dequeued from each of the buffers.

16. The apparatus according to claim 15, wherein:

the backpressure information includes an indication of the number of cells discarded from each of the buffers.

17. The apparatus according to claim 14, further comprising:

d) a single multicast buffer to be shared by up to n number of the ports.

18. The apparatus according to claim 17, wherein:

the backpressure information includes an indication of the number of cells dequeued from the multicast buffer.

19. The apparatus according to claim 17, wherein:

the backpressure information includes an indication of whether a cell in the multicast buffer has been discarded.

20. The apparatus according to claim 17, wherein:

the backpressure information includes an identification of the last port to experience a multicast timeout error.

21. An apparatus for avoiding head of line blocking in an ATM device, comprising:

a) a multiplexer;

b) at least one unicast queue coupled to said multiplexer;

c) at least one multicast queue coupled to said multiplexer;

d) a scheduler coupled to said multiplexer; and

e) a multicast session table accessible by said scheduler, said multicast session table including a list of PHY devices to which a multicast cell is to be copied, wherein

said scheduler alternates between unicast and multicast queues for cell transmission and inactive PHY devices in a multicast session are ignored.

22. An apparatus according to claim 21, wherein:

a multicast session is closed and its associated cell removed from the multicast queue when all of the PHY devices in the session have been serviced or all of the PHY devices remaining in the session are inactive.

23. An apparatus according to claim 21, further comprising:

f) a multicast timer coupled to said scheduler, wherein

said timer is started when a cell reaches the head of the multicast queue and the multicast session is closed when said timer expires or all of the PHY devices in the session have been serviced.

24. An apparatus according to claim 21, further comprising:

f) a multicast timer coupled to said scheduler, wherein

said timer is started when the only PHY devices remaining in the multicast session are inactive and the multicast session is closed when said timer expires or all of the PHY devices in the session have been serviced whichever occurs first.

25. An apparatus according to claim 21, further comprising:

f) a problem PHY vector coupled to said scheduler, wherein

said problem PHY vector contains a list of inactive PHY devices and is updated at the end of each multicast session.

26. An apparatus according to claim 25, wherein:

said problem PHY vector is updated whenever an inactive PHY becomes active.

27. An apparatus according to claim 25, wherein:

the multicast session is closed when all of the PHYs remaining to be serviced are listed in the problem PHY vector.

28. An apparatus according to claim 23, further comprising:

g) a problem PHY vector coupled to said scheduler, wherein

said problem PHY vector contains a list of inactive PHY devices and is updated at the end of each multicast session.

29. An apparatus according to claim 28, wherein:

said problem PHY vector is updated whenever an inactive PHY becomes active.

30. An apparatus according to claim 28, wherein:

the multicast session is closed when all of the PHYs remaining to be serviced are listed in the problem PHY vector.

31. An apparatus according to claim 24, further comprising:

g) a problem PHY vector coupled to said scheduler, wherein

said problem PHY vector contains a list of inactive PHY devices and is updated at the end of each multicast session.

32. An apparatus according to claim 31, wherein:

said problem PHY vector is updated whenever an inactive PHY becomes active.

33. An apparatus according to claim 31, wherein:

the multicast session is closed when all of the PHYs remaining to be serviced are listed in the problem PHY vector.

34. A method for avoiding head of line blocking in an ATM device, comprising:

a) servicing destination ports according to an arbitration scheme;

b) alternating between unicast and multicast queues for cell transmission when both a unicast cell and a multicast cell are scheduled for the same port; and

c) ignoring inactive PHY devices in a multicast session.

35. A method according to claim 34, further comprising:

d) closing a multicast session and removing its associated cell from the multicast queue when all of the PHY devices in the session have been serviced or all of the PHY devices remaining in the session are inactive.

36. A method according to claim 34, further comprising:

d) starting a timer when a multicast cell reaches the head of the multicast queue;and

e) closing the multicast session when the timer expires or all of the PHY devices in the session have been serviced.

37. A method according to claim 34, further comprising:

d) maintaining a problem PHY vector which contains a list of inactive PHY devices and is updated at the end of each multicast session.

38. A method according to claim 37, further comprising:

e) closing the multicast session when all of the PHYs remaining in the session are listed in the problem PHY vector.

39. A method according to claim 38, further comprising:

f) updating the problem PHY vector whenever an inactive PHY becomes active.

40. A method according to claim 36, further comprising:

f) maintaining a problem PHY vector which contains a list of inactive PHY devices and is updated at the end of each multicast session.

41. A method according to claim 40, further comprising:

g) closing the multicast session when all of the PHYs remaining in the session are listed in the problem PHY vector.

42. A method according to claim 41, further comprising:

h) updating the problem PHY vector whenever an inactive PHY becomes active.

43. A method for avoiding head of line blocking in an ATM device, comprising:

a) servicing unicast and multicast queues according to an arbitration scheme;

b) alternating between unicast and multicast queues for cell transmission; and

c) ignoring inactive PHY devices in a multicast session.

44. A method according to claim 43, further comprising:

d) closing a multicast session and removing its associated cell from the multicast queue when all of the PHY devices in the session have been serviced or all of the PHY devices remaining in the session are inactive.

45. A method according to claim 43, further comprising:

d) starting a timer when the only PHY devices remaining in a multicast session are inactive; and

e) closing the multicast session when the timer expires or all of the PHY devices in the session have been serviced.

46. A method according to claim 43, further comprising:

d) maintaining a problem PHY vector which contains a list of inactive PHY devices and is updated at the end of each multicast session.

47. A method according to claim 46, further comprising:

e) closing the multicast session when all of the PHYs remaining in the session are listed in the problem PHY vector.

48. A method according to claim 47, further comprising:

f) updating the problem PHY vector whenever an inactive PHY becomes active.

49. A method according to claim 45, further comprising:

f) maintaining a problem PHY vector which contains a list of inactive PHY devices and is updated at the end of each multicast session.

50. A method according to claim 49, further comprising:

g) closing the multicast session when all of the PHYs remaining in the session are listed in the problem PHY vector.

51. A method according to claim 50, further comprising:

h) updating the problem PHY vector whenever an inactive PHY becomes active.

52. An apparatus for avoiding head of line blocking in an ATM device, comprising:

a) a multiplexer;

b) at least one unicast queue coupled to said multiplexer;

c) at least one multicast queue coupled to said multiplexer;

d) a scheduler coupled to said multiplexer;

e) a multicast session table accessible by said scheduler, said multicast session table including a list of PHY devices to which a multicast cell is to be copied; and

f) a problem PHY vector which includes an indication of PHY devices which are not responding, wherein

a multicast session is closed and its associated cell removed from the multicast queue when the only PHYs remaining to be serviced are indicated in the problem PHY vector.

53. An apparatus according to claim 52, wherein:

a multicast session is closed and its associated cell removed from the multicast queue when all of the PHY devices in the session have been serviced or all of the PHY devices remaining in the session are inactive.

54. An apparatus according to claim 52, further comprising:

g) a multicast timer coupled to said scheduler, wherein

said timer is started when the only PHY devices remaining in the multicast session are inactive PHY devices and at least one of the remaining PHY devices is not indicated in the problem PHY vector and the multicast session is closed when said timer expires or all of the PHY devices in the session have been serviced.

55. An apparatus according to claim 52, wherein:

said problem PHY vector is updated whenever an inactive PHY becomes active.

56. A method for avoiding head of line blocking in an ATM device, comprising:

a) servicing multicast and unicast queues according to an arbitration scheme;

b) servicing multicast destinations (PHY devices) according to an entry in a multicast session table;

c) maintaining a problem PHY vector which indicates the PHY devices which are not responding;

d) terminating a multicast session when the only destinations remaining to be serviced are indicated in the problem PHY vector.

57. A method according to claim 56, further comprising:

e) closing a multicast session and removing its associated cell from the multicast queue when all of the PHY devices in the session have been serviced or all of the PHY devices remaining in the session are inactive.

58. A method according to claim 56, further comprising:

e) starting a timer when the only PHY devices remaining in a multicast session are inactive and at least one of the PHY devices is not indicated in the problem PHY vector;and

f) closing the multicast session when the timer expires or all of the PHY devices in the session have been serviced.

59. A method according to claim 56, further comprising:

e) updating the problem PHY vector whenever an inactive PHY becomes active.

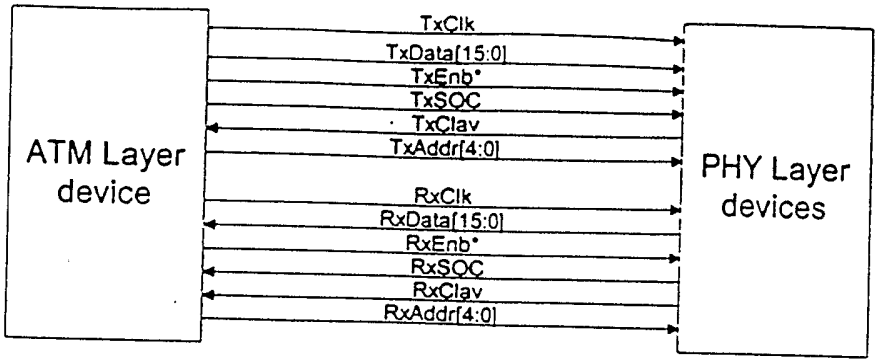


FIGURE 1. An Example of UTOPIA interface

PRIOR ART
FIG. 1

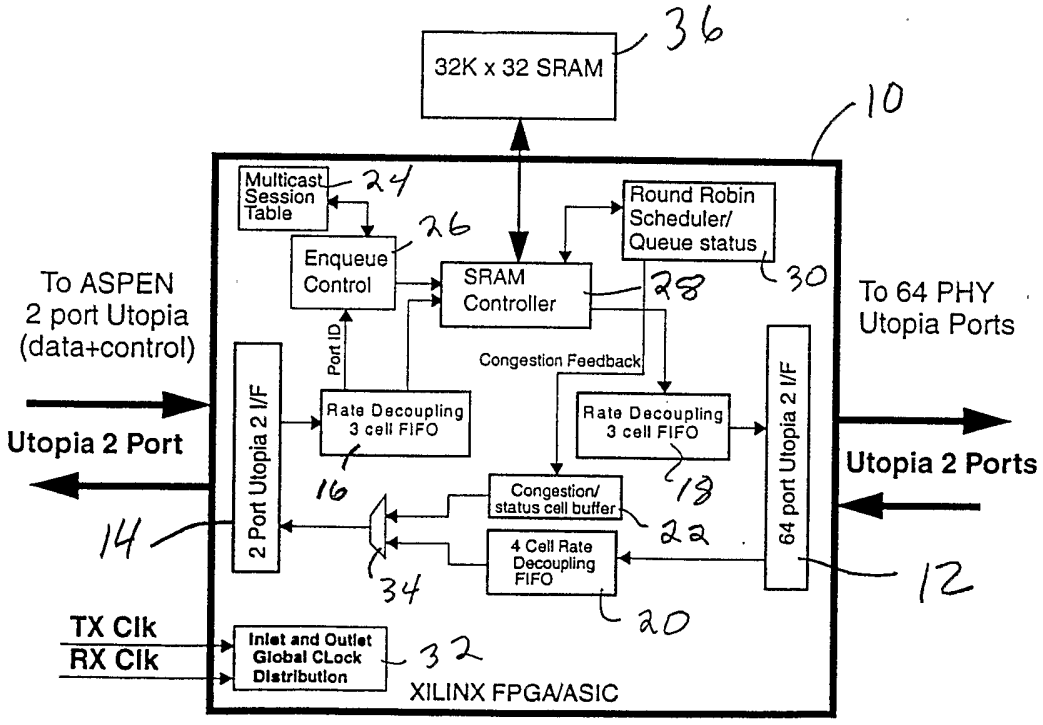


FIG. 2

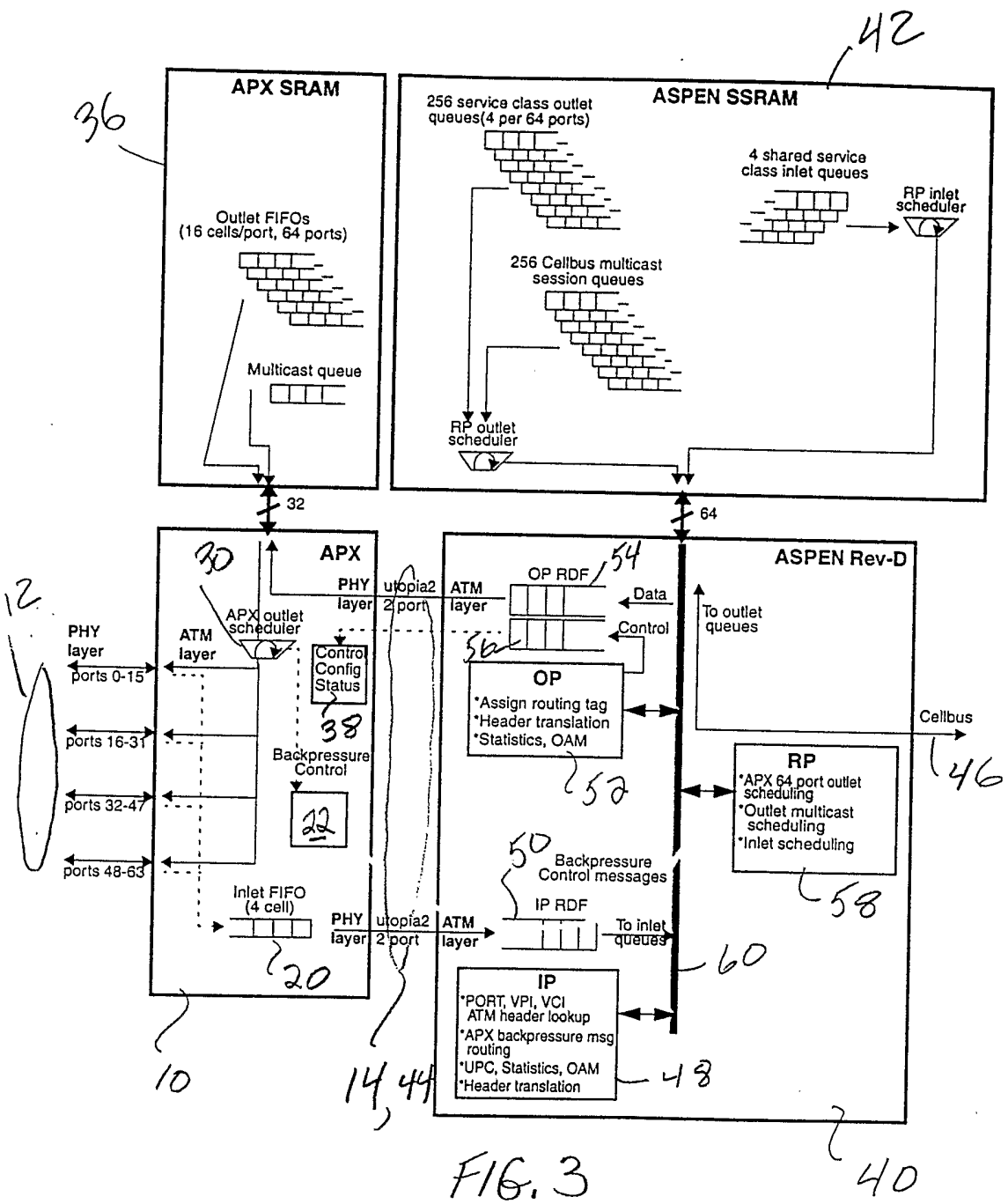

42
APX SRAM
36
ASPEN SSRAM
256 service class outlet queues(4 per 64 ports)
4 shared service class inlet queues
RP inlet scheduler
Outlet FIFOs (16 cells/port, 64 ports)
256 Cellbus multicast session queues
Multicast queue
RP outlet scheduler
32
64
APX
30
ASPEN Rev-D
54
PHY layer
utopia2 2 port
ATM layer
OP RDF
Data
Control
To outlet queues
APX outlet scheduler
Contro Config Status
38
PHY layer ports 0-15
ATM layer
ports 16-31
ports 32-47
ports 48-63
Backpressure Control
22
OP
*Assign routing tag
*Header translation
*Statistics, OAM
52
Cellbus
46
RP
*APX 64 port outlet scheduling
*Outlet multicast scheduling
*Inlet scheduling
58
50
Backpressure Control messages
IP RDF
To inlet queues
Inlet FIFO (4 cell)
20
60
IP
*PORT, VPI, VCI ATM header lookup
*APX backpressure msg routing
*UPC, Statistics, OAM
*Header translation
48
12
10
14, 44
40

FIG. 3

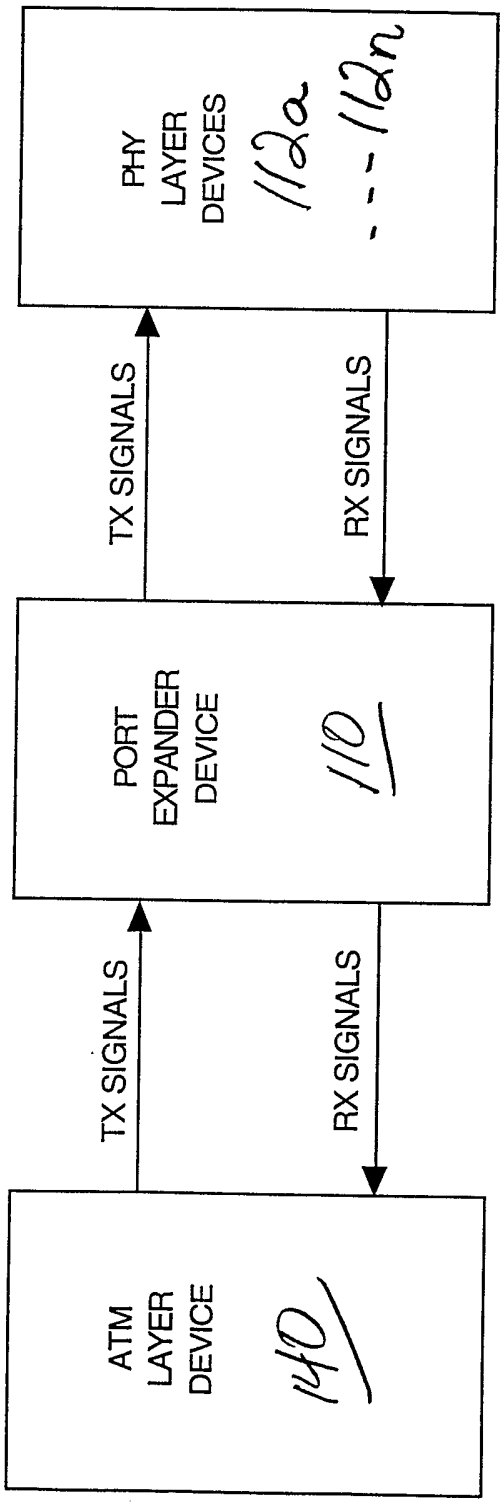

PHY LAYER DEVICES
112a
112n
TX SIGNALS
RX SIGNALS
PORT EXPANDER DEVICE
110
TX SIGNALS
RX SIGNALS
ATM LAYER DEVICE
140

Figure 4

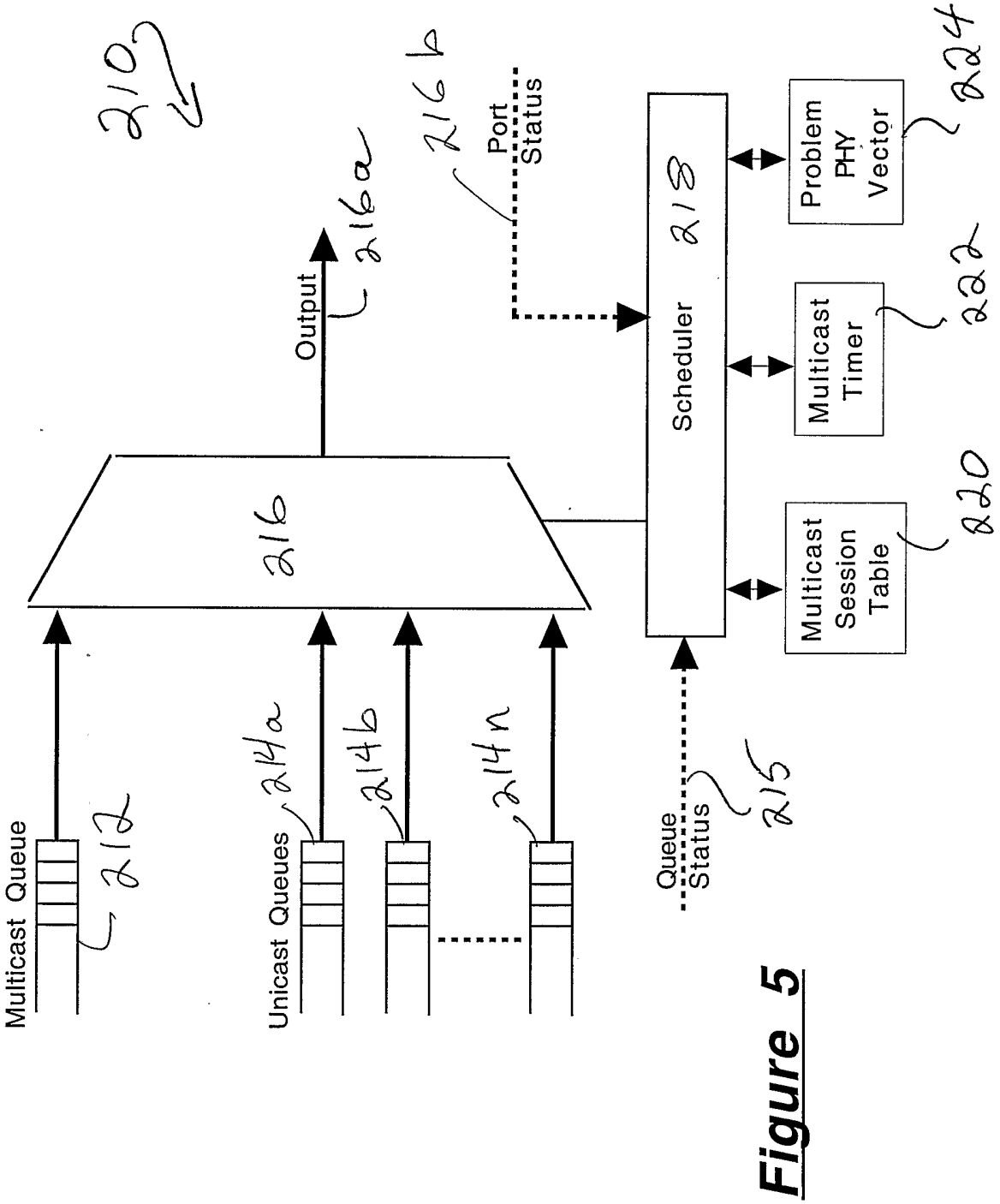

210
216b
Port Status
Output
216a
216
Scheduler 218
Problem PHY Vector
224
Multicast Timer
222
Multicast Session Table
220
Multicast Queue
212
Unicast Queues
214a
214b
214n
Queue Status
215

Figure 5

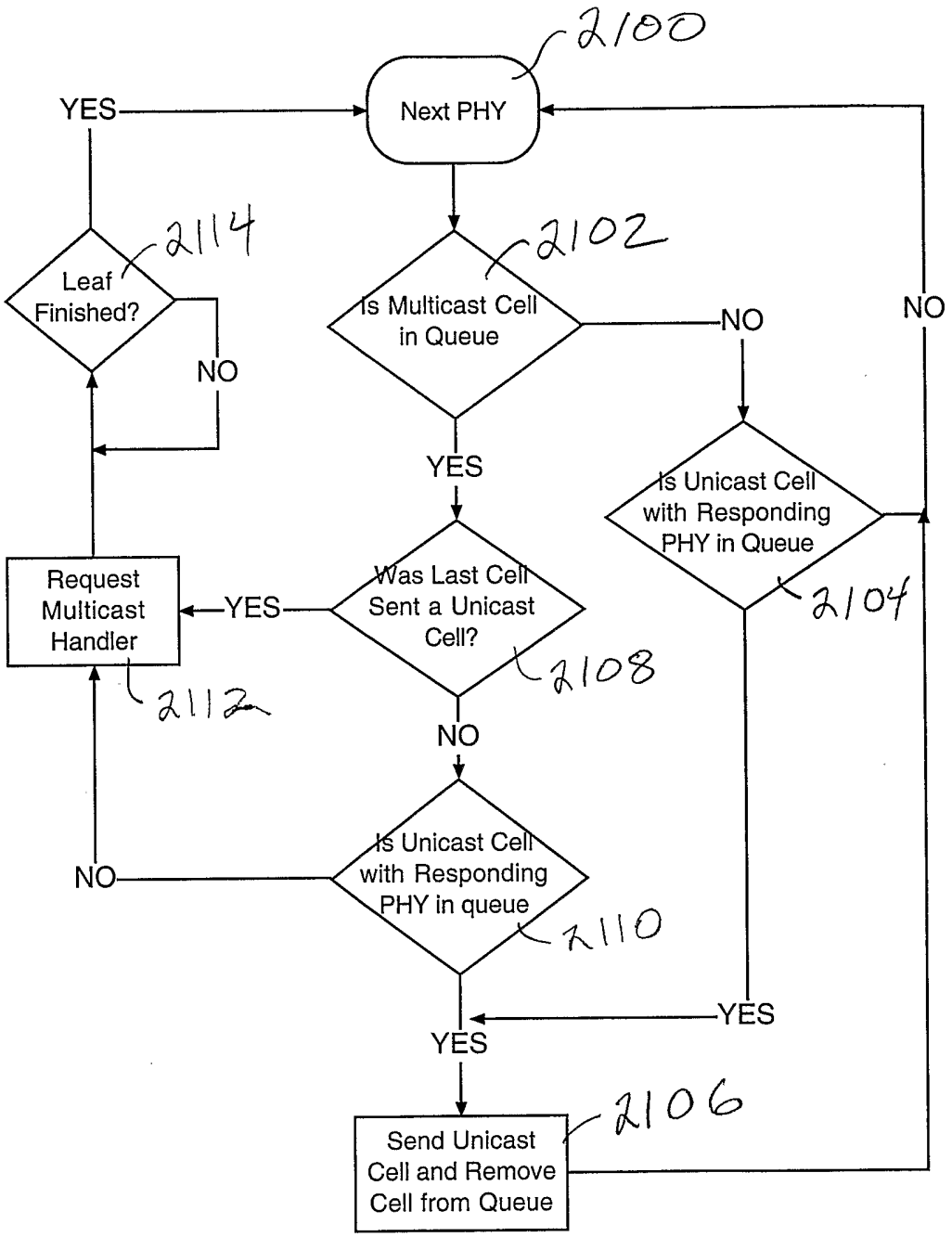

2100
Next PHY
YES
2102
Is Multicast Cell in Queue
NO
NO
2114
Leaf Finished?
NO
YES
Is Unicast Cell with Responding PHY in Queue
2104
Request Multicast Handler
YES
Was Last Cell Sent a Unicast Cell?
2108
2112
NO
Is Unicast Cell with Responding PHY in queue
NO
2110
YES
YES
2106
Send Unicast Cell and Remove Cell from Queue

Figure 6

Figure 7

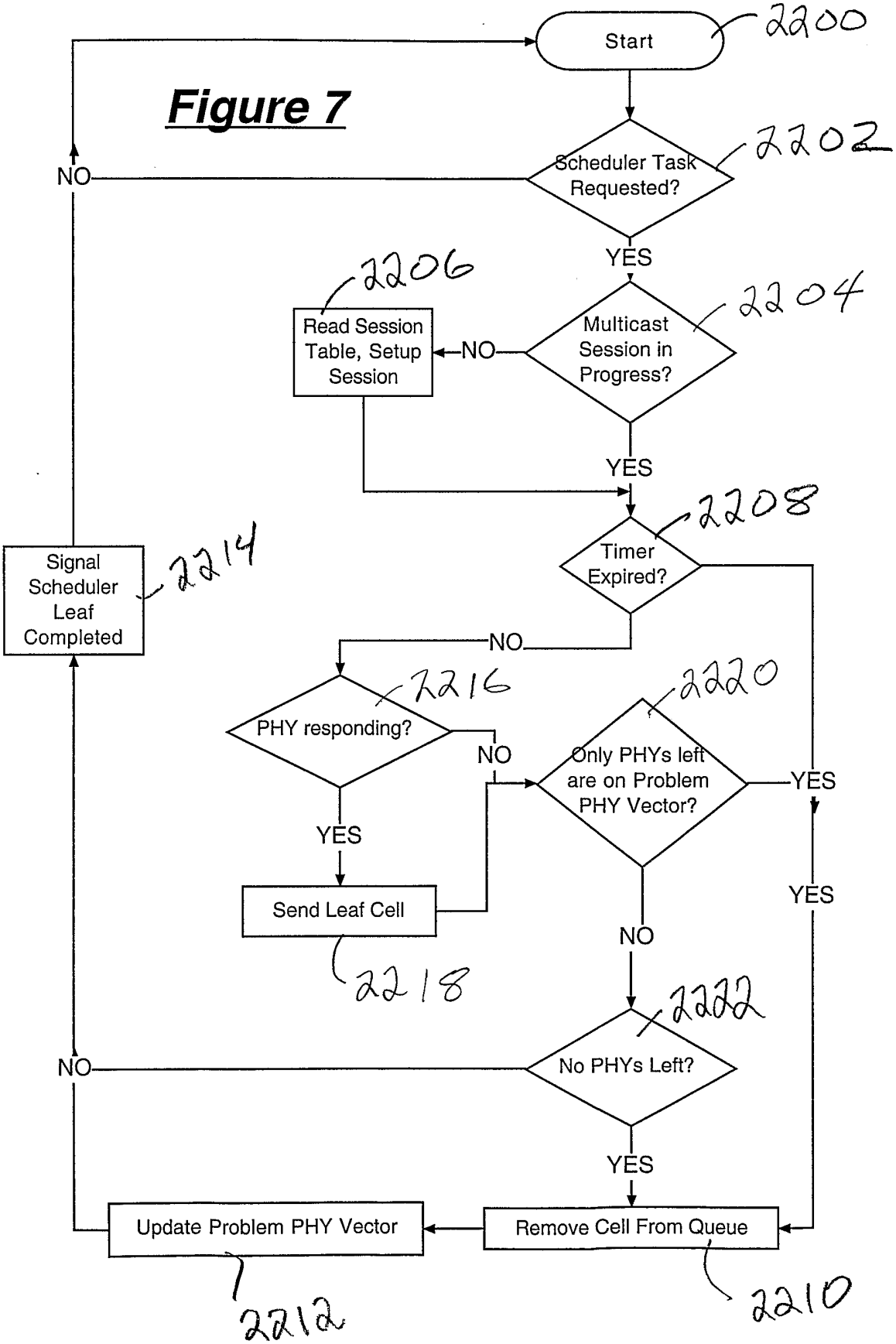
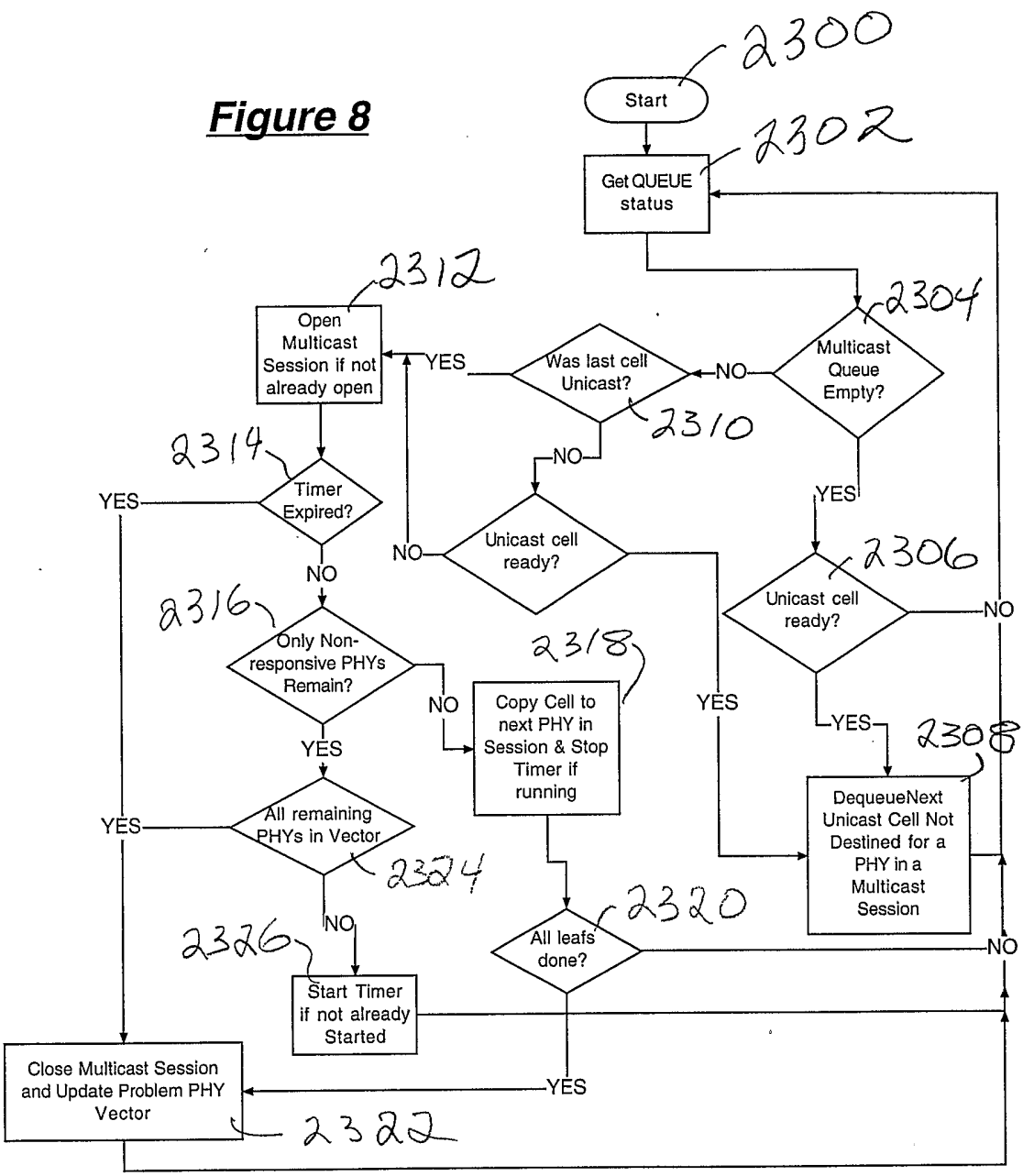

Start
2200
Scheduler Task Requested?
2202
NO
YES
Multicast Session in Progress?
2204
NO
Read Session Table, Setup Session
2206
YES
Timer Expired?
2208
NO
Signal Scheduler Leaf Completed
2214
PHY responding?
2216
NO
YES
Send Leaf Cell
2218
Only PHYs left are on Problem PHY Vector?
2220
YES
YES
NO
No PHYs Left?
2222
NO
YES
Update Problem PHY Vector
2212
Remove Cell From Queue
2210

Figure 8


2300
Start
2302
Get QUEUE status
2304
Multicast Queue Empty?
NO
YES
2306
Unicast cell ready?
NO
YES
2308
DequeueNext Unicast Cell Not Destined for a PHY in a Multicast Session
2310
Was last cell Unicast?
YES
NO
2312
Open Multicast Session if not already open
2314
Timer Expired?
YES
NO
2316
Only Non-responsive PHYs Remain?
NO
YES
2318
Copy Cell to next PHY in Session & Stop Timer if running
2320
All leafs done?
NO
YES
2322
Close Multicast Session and Update Problem PHY Vector
2324
All remaining PHYs in Vector
YES
NO
2326
Start Timer if not already Started
Unicast cell ready?
NO
YES