IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

EVERTZ MICROSYSTEMS LTD.,	)	
	)	
Plaintiff,	)	
	)	
v.	)	C.A. No. 19-302-MN
	)	
LAWO INC., LAWO NORTH AMERICA CORP., and LAWO AG,	)	**JURY TRIAL DEMANDED**
	)	
	)	
Defendants.	)	

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Evertz Microsystems Ltd. ("Evertz") files this First Amended Complaint for Patent Infringement against Defendants Lawo Inc., Lawo North America Corp., and Lawo AG (collectively, "Lawo"), and alleges as follows:

NATURE OF THIS ACTION

1. This is a patent infringement action based on each Lawo defendant's continued infringement of various Evertz patents. In particular, this is a patent infringement action based on each Lawo defendant's continued infringement of i) U.S. Patent No. 8,537,838, issued September 17, 2013, and titled "Packet Based Transmission of Multiple Data Signals" (the "'838 Patent"), ii) U.S. Patent No. 9,100,217, issued August 4, 2015, and titled "Apparatus, Systems and Methods for Packet Based Transmission of Multiple Data Signals" (the "'217 Patent"), iii) U.S. Patent No. 9,473,322, issued October 18, 2016, and titled "Apparatus, Systems and Methods for Packet Based Transmission of Multiple Data Signals" (the "'322 Patent"), iv) U.S. Patent No. 8,270,398, issued September 18, 2012, and titled "System and Method for Signal Processing" (the "'398 Patent"), v) U.S. Patent No. 9,654,391, issued May 16, 2017, and titled "Video Router" (the "'391 Patent"), vi) U.S. Patent No. 9,942,139, issued April 10, 2018, and titled "Video Router" (the "'139 Patent"),

and vii) U.S. Patent No. 10,164,877, issued December 25, 2018, and titled "Video Router" (the "'877 Patent") (collectively, the "Patents-In-Suit").

PARTIES

2. Plaintiff Evertz is a corporation organized under the laws of the Province of Ontario, Canada, and has a principal place of business at 5292 John Lucas Drive, Burlington, Ontario L7L 5Z9, Canada.

3. Upon information and belief, Lawo Inc. is a Delaware corporation, and has a principal place of business at 99 Hudson Street, 5th Floor, New York, New York 10013. Lawo Inc. can be served through its registered agent, The Corporation Trust Company, 1209 Orange Street, Wilmington, Delaware 19801.

4. Upon information and belief, Lawo North America Corp. is a Canadian corporation, and has a principal place of business at 2041 McCowan Road, Unit 1, Toronto, Ontario M1S 3Y6, Canada.

5. Upon information and belief, Lawo AG is a German corporation, and has a principal place of business at AM Oberwald 8, 76437 Rastatt, Germany.

JURISDICTION AND VENUE

6. This is an action for patent infringement arising under the United States Patent Act, 35 U.S.C. § 100, et seq.

7. This Court has subject matter jurisdiction over this action under 28 U.S.C. §§ 1331 and 1338(a).

8. This Court has personal jurisdiction over Defendant Lawo Inc. because Lawo Inc. is incorporated in Delaware and, on information and belief, has continuous and systematic contacts with the State of Delaware, including continuous contacts with, offers to sell, and/or sales to, customers in Delaware. Further, on information and belief, Lawo AG manufactures and supplies

products that it offers to sell, sells, and/or imports into the United States, including through its subsidiaries Lawo Inc. and Lawo North America Corp. On information and belief, Lawo Inc., Lawo North America Corp, and Lawo AG have committed acts within the District of Delaware giving rise to this action, including using, offering for sale, selling, and/or importing into the United States products that infringe one or more of the claims of the Patents-In-Suit.

9. Lawo Inc. is incorporated in Delaware, Lawo North America Corp. is a Canadian corporation, and Lawo AG is a German company. Venue is proper in this District under 28 U.S.C. §§ 1391(b), 1391(c), and/or 1400(b).

FACTUAL BACKGROUND

Evertz Microsystems Ltd.

10. Evertz is a Canadian technology company originally founded as DynaQuip Electron Devices Limited in 1966. Evertz designs, manufactures, and markets solutions for the production, post-production, broadcast, and telecommunications markets in the United States, Canada, and internationally. Evertz's innovations center on video and audio infrastructure solutions for television, telecommunications, and new-media industries, including on-demand services and mobile devices.

11. Last year, Evertz invested approximately $80 million (CAD) on research and development. In the past five years, Evertz has spent more than $340 million (CAD) on research and development. Evertz is recognized as a global innovator and a leading solutions provider to the broadcast and media industries, and has received a number of awards for its novel infrastructure solutions and technologies, including Emmy Awards from the National Academy of Television Arts and Sciences in 2008 and 2017.

12. Among other things, Evertz's award-winning solutions convert traditional broadcast transmissions to internet protocol ("IP") based transmissions, thereby optimizing

broadcast speed and quality through software-defined networking, system architecture, and product solutions.

13. Because they are valuable business assets, Evertz takes steps to protect the intellectual property assets that result from its investments in innovation, including by securing utility patent protection in North America.

Evertz Patents Relating to Packet-Based Transmission of Multiple Data Signals

14. The United States Patent and Trademark Office ("USPTO") issued the '838 Patent on September 17, 2013. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the '838 Patent. A true and correct copy of the '838 Patent is attached as Exhibit A.

15. The USPTO issued the '217 Patent on August 4, 2015. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the '217 Patent. A true and correct copy of the '217 Patent is attached as Exhibit B.

16. The USPTO issued the '322 Patent on October 18, 2016. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the '322 Patent. A true and correct copy of the '322 Patent is attached as Exhibit C.

17. Rakesh Thakor Patel and Romolo Magarelli, are the named inventors of the '838 Patent, the '217 Patent, and the '322 Patent (collectively, the "First Patent Family").

18. The First Patent Family relates generally to a system for transmitting and distributing video and audio signals. In simplified terms, the inventions, among other things, permit numerous different video/audio feeds to be displayed in one or more windows on one or more monitors, in various formats.

19. More specifically, the First Patent Family relates to a system for efficiently receiving various input signals (e.g., video, audio, and/or data), generating packetized signals for

transmission to an output processor, and providing output signals based on the packetized signals to various output devices. One or more master controllers generate control signals to, for example, define characteristics for generating the packetized signals and providing output signals. Among other things, the control signals may specify the location and dimensions for display of video input signals and correlation with specific audio equalization or noise cancellation. The inventions claimed by the First Patent Family facilitate generating one or more formatted output signals from one or more input signals, without the need for excessive cabling, signal regeneration/replication, and corresponding signal degradation, thereby proving indispensable in applications where multiple video, audio, and/or data inputs must be monitored, such as in connection with the broadcast of news or sporting events.

20. Evertz makes and sells numerous systems that use the claimed inventions of the First Patent Family, including products in Evertz's MVP, Maestro II, VUE, and VistaLINK PRO product lines. The inventions of the First Patent Family embodied in the various Evertz product lines have garnered industry praise and have had substantial commercial success.

Evertz Patent Relating to Signal Processing

21. The USPTO issued the '398 Patent on September 18, 2012. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the '398 Patent. A true and correct copy of the '398 Patent is attached as Exhibit D.

22. Romolo Magarelli, Rakesh Thakor Patel, Eric Fankhauser, and Daniel G. Turow are the named inventors of the '398 Patent.

23. The '398 Patent relates generally to a system that allows for a plurality of input signals to be routed to one or more output processors, where each output processor processes the one or more input signals it receives and provides one or more output signals based on the received

input signals. A controller controls the configuration of a cross point switch to switch a subset of input signals to particular output processors.

24. Evertz makes and sells numerous systems that use the claimed invention of the ’398 Patent, including products in Evertz’s EQX product lines. The inventions of the ’398 Patent embodied in Evertz’s EQX product lines have garnered industry praise and have had substantial commercial success.

Evertz Patents Relating to Video Routing

25. The USPTO issued the ’391 Patent on May 16, 2017. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the ’391 Patent. A true and correct copy of the ’391 Patent is attached as Exhibit E.

26. The USPTO issued the ’139 Patent on April 10, 2018. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the ’139 Patent. A true and correct copy of the ’139 Patent is attached as Exhibit F.

27. The USPTO issued the ’877 Patent on December 25, 2018. Evertz is the lawful owner by assignment of all rights, title, and interest in and to the ’877 Patent. A true and correct copy of the ’877 Patent is attached as Exhibit G.

28. Rakesh Thakor Patel is the named inventor of the ’391 Patent, the ’139 Patent, and the ’877 Patent (collectively, the “Second Patent Family”).

29. The Second Patent Family relates generally to a system for transmitting and distributing video, audio, and/or data signals. In simplified terms, the inventions, among other things, provide a data communication network to route video, audio, and/or data signals to and from devices.

30. More specifically, the Second Patent Family relates to a data communication network that includes a plurality of line cards and network switches to route various input signals across the network. One or more controllers generate control signals to, for example, control the operation of the line cards and network switches. The inventions claimed by the Second Patent Family facilitate efficient routing of video, audio, and/or data signals, thereby proving indispensable in applications where a large number of signals must be efficiently routed, such as in connection with the broadcast of news or sporting events.

31. Evertz makes and sells numerous systems that use the claimed invention of the Second Patent Family, including products in Evertz's EXE product lines. The inventions of the Second Patent Family embodied in Evertz's EXE product line have garnered industry praise and have had substantial commercial success.

32. Evertz's MVP, Maestro II, VUE, VistaLINK PRO, EXE and EQX product lines are collectively referred to herein as the "Evertz Covered Products."

Providius, Former Evertz Employees, and Lawo

33. On October 29, 2012, while the application that matured into the '838 Patent was pending at the USPTO, Tony Zare, Ayman Al Khatib, and Jackson Wiegman founded a Canadian corporation named Mayana Media Corp. in Ontario. In May 2013, Mayana was renamed Providius Corp. Providius directly competed with Evertz in the design, manufacture, marketing, and sales of systems for transmitting and distributing video and audio signals in broadcasting industries. Providius and Evertz attended the same trade shows in the United States and, on information and belief, offered to sell and/or sold products for use by the same customers and in the same types of applications as those of the Evertz Covered Products.

34. Mr. Zare is a design engineer and was an Evertz employee from April 15, 2002, until March 6, 2015. Mr. Zare owed various duties to Evertz as conditions of his employment and was party to a confidentiality agreement he entered with Evertz. At the same time he was working at Evertz, public records reveal that Mr. Zare was a director of Providius from its founding until April 29, 2013. Further, Mr. Zare signed an annual return filed by Providius on January 28, 2015, pursuant to the Canada Business Corporations Act. On information and belief, Mr. Zare concealed his work with Providius from Evertz to maintain access to Evertz's confidential research and development activities. At the time of his departure from Evertz in 2015, almost two and a half years after cofounding Providius, he was a director-level Evertz employee working on digital compression systems, including technologies incorporated in the Evertz Covered Products. While at Evertz, Mr. Zare worked on teams with Evertz's Mr. Patel, one of the aforementioned inventors of the '838 Patent. Further, Mr. Zare is identified as an inventor in another Evertz patent family (U.S. Patent No. 9,620,131B2, filed on April 8, 2011, issued on April 11, 2017; and U.S. Patent Application 15/445,605, filed on February 28, 2017).

35. Mr. Al Khatib is a production engineer and was an Evertz employee from May 7, 2007, until November 22, 2016. Mr. Al Khatib owed various duties to Evertz as conditions of his employment and was party to a confidentiality agreement he entered with Evertz. When he left his employment with Evertz, Mr. Al Khatib was a Director of International Business and—on information and belief—well-versed in Evertz's competitive business relationship with Lawo. Public records evidence that, contemporaneous with his employment at Evertz, Mr. Al Khatib was also a director of Providius from its founding until at least April 29, 2013. On information and belief, he too concealed his work with Providius from Evertz to maintain access to Evertz's confidential innovations, business opportunities, and technology plans.

36. Mr. Wiegman was an Evertz employee from October 29, 2007, until October 19, 2012, approximately ten days before he cofounded Providius. Mr. Wiegman owed various duties to Evertz as conditions of his employment and was party to a confidentiality agreement he entered with Evertz. At the time he left his employment with Evertz, Mr. Wiegman was a director-level Evertz product manager, who—on information and belief—was not only familiar with Evertz's competitive business relationship with Lawo, but also with Evertz's product line plans and future product planning.

37. On information and belief, sometime in 2016, Providius launched a product called "BMG Solution" that was based on and substantially similar to proprietary Evertz solutions, including the Evertz Covered Products.

38. On information and belief, between 2016 and 2018, Lawo acquired Providius and/or the BMG Solution, and Lawo started promoting Providius' BMG Solution as its own product in competition with Evertz and the Evertz Covered Products. In 2018, Lawo and Evertz attended the same trade show in the United States and, on information and belief, offered to sell and/or sold products for use by the same customers and in the same types of applications as those of the Evertz Covered Products.

39. On information and belief, Messrs. Zare and Wiegman continue to be senior technical employees of Providius and are also involved with technology and product development for Lawo. For example, Mr. Zare is identified on Lawo's website as a Senior Director of Product Management and he appears in a promotional video for Lawo "SMART" scope (*see*, LAWO, www.lawo.com/products/network-monitoring/smartscope.html, last visited March 11, 2019).

40. Public records show that on December 23, 2016, Messrs. Zare and Al Khatib rejoined Providius' Board of Directors. Further, Messrs. Zare and Wiegman continue to serve on Providius' Board alongside Philipp Lawo, the CEO of Lawo.

41. On information and belief, Messrs. Zare, Al Khatib, and/or Wiegman knew of Evertz's efforts to commercialize the Evertz Covered Products, to prosecute the Patents-in-Suit, and/or the issuance of each of the Patents-in-Suit. On further information and belief, Messrs. Zare, Al Khatib, and Wiegman willfully and without authorization transferred confidential and proprietary information about Evertz innovations, including but not limited to Evertz Covered Products and innovations covered by the Patents-in-Suit, to Lawo, and Lawo used this information knowing it to be proprietary to Evertz and/or covered by the Patents-in-Suit to offer to sell, sell, and/or import into the United States products that infringe the Patents-in-Suit, as further described below.

42. On May 16, 2018, Evertz sued Lawo, Providius, and Messrs. Zare, Al Khatib, and Wiegman, among others, in the Ontario Superior Court of Justice, Court File No. CV1800597979000, alleging theft of confidential information, breach of confidence, conspiracy, and unjust enrichment, in addition to other related claims (the "Ontario Civil Action").

43. As explained in Evertz's Ontario Civil Action complaint, and on information and belief, Messrs. Zare, Al Khatib and Wiegman misappropriated Evertz's confidential information about Evertz Covered Products, including, among other things, software, architecture, product solution, roadmap, materials, work flow, and pricing information, to build an IP-based broadcast network product for Providius and Lawo using information from Evertz (Ontario Civil Action Complaint, p. 20, Paras. 91-92).

44. As a result, Providius was, on information and belief, able to release a product line (the BMG Solution) with attributes and functionality similar to Evertz Covered Products, despite being a new company with few resources (Ontario Civil Action Complaint, p. 19, Para. 81). In comparison, it had taken Evertz many years and tens of millions of dollars in research and development investment to develop and release the Evertz Covered Products.

45. Each Lawo defendant has, on information and belief, continued to willfully copy and profit from Evertz's confidential and proprietary information, and to willfully infringe the Patents-In-Suit. To that end, Lawo has introduced a product called V_matrix, which Lawo touts on its website as a "new IP broadcast video core infrastructure." However, Lawo's V_matrix product is not "new." To the contrary, on information and belief, it makes substantial use of confidential, proprietary, and patented Evertz technology knowingly transferred without authorization by Messrs. Zare, Al Khatib, and Wiegman to Lawo.

Lawo Infringes the Patents-In-Suit

46. Lawo offers to sell, sells, uses, and imports into the United States a system of products it calls V_matrix. Lawo's V_matrix systems infringe at least claim 1 of each of the '838 Patent, the '217 Patent, the '322 Patent, '398 Patent, the '391 Patent, the '139 Patent, and the '877 Patent.

47. A true and correct copy of Lawo's English language brochure for its V_matrix system, downloaded from Lawo's website at <www.lawo.com/fileadmin/content/Products/V__matrix/Lawo_V__matrix_EN.pdf>, is attached as Exhibit H.

48. As described in detail below in Counts I - VII, Lawo offers a series of products that infringe the Patents-in-Suit, including the V_matrix system, which may comprise a plurality of

modules. The accused product comprises one or more V_matrix modules that receive one or more input signals, generate packetized signals, transmit the packetized signals, and produce one or more output signals.

COUNT I
INFRINGEMENT OF U.S. PATENT NO. 8,537,838

49. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 48 above as if fully set forth herein.

50. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the ’838 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling (directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

51. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the ’838 Patent that reads:

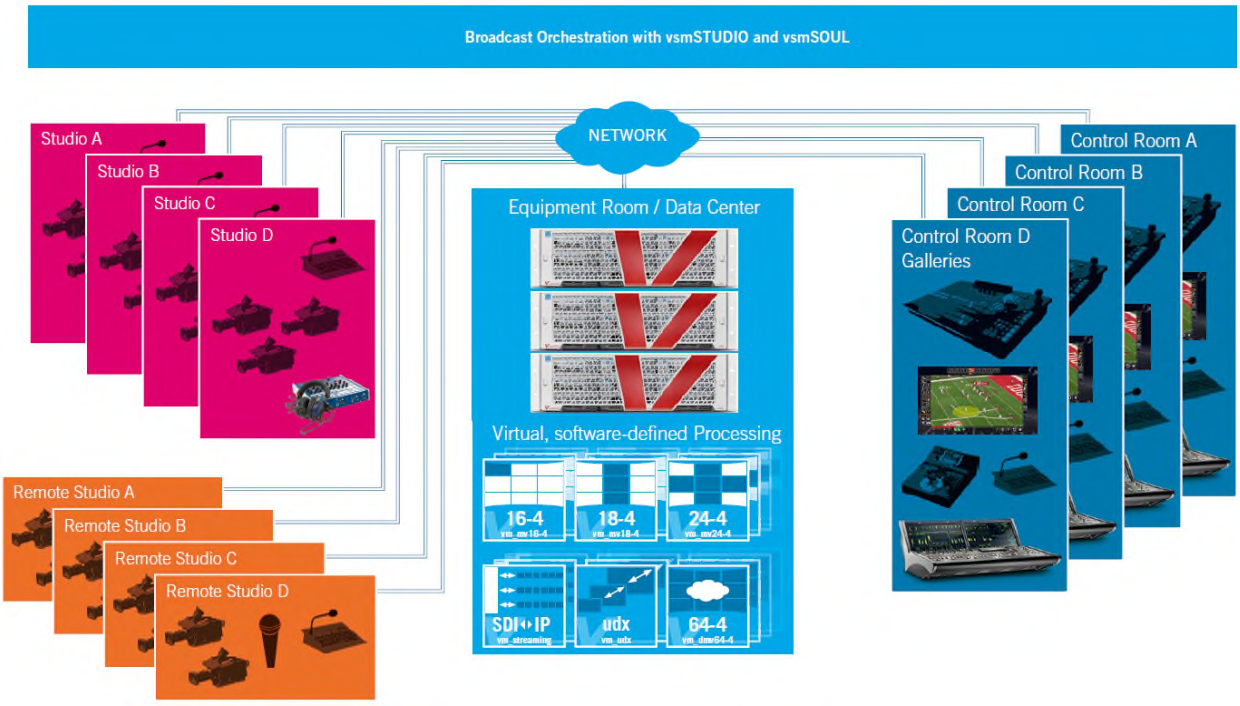
> A system for receiving one or more input signals and for producing one or more output signals, the system comprising:
>
> > (a) a master controller for generating input processor control signals and output processor control signals, and for assigning a unique global identification code to each of a plurality of packet source signals;
> >
> > (b) an input processor having: (i) one or more input ports for receiving the input signals; (ii) one or more input signal processors for processing the input signals to provide one or more processed signals (iii) an input processor memory system for buffering the input signals and the processed signals, wherein at least some of the buffered signals are designated as packet source signals; (iv) one or more packetized signal output ports; (v) one or more packetized signal output stages for retrieving one or more of the packet source signals from the input processor memory system and for producing one or more packetized signals at the packetized signal output ports, wherein each of the packetized signals includes a series of packetized signal packets, wherein each of the packetized signal packets contains the unique global identification code corresponding to one of the packet source signals and data corresponding to the same packet source signal; and (vi) an input processor local controller for controlling the operation

> of at least the signal processors and the packetized signal output stages in response to the input processor control signals;
>
> (c) an output processor having: (i) one or more packetized signal input ports for receiving the packetized signals; (ii) one or more packetized signal input stages for extracting data corresponding to each of the packet source signals from each of the packetized signals and for storing data corresponding to each of the packet source signals in a separate buffer in the output processor memory system as an output source signal based on the unique global identification code in the packetized signal packets of each packetized signal; (iii) one or more output signal generators for providing one or more output signals, each of the output signals corresponding to one or more of the output source signals; (iv) an output processor local controller for controlling the operation of the packetized signal input stages and the output signal generators in response to the output processor control signals; and
>
> (d) a communications link coupled between the one or more packetized signal output ports and the one or more packetized signal input ports.

52. The Lawo products including, but not limited to, the Lawo V_matrix system perform each and every limitation of ’838 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

53. By way of example, a series of Lawo products individually and collectively provide broadcast video core infrastructure for broadcast facilities. Lawo refers to this system as “V_matrix.” The V_matrix system may receive one or more input signals, such as recorded video signals from a studio, and may output one or more output signals, such as video for monitoring in a control room. The V_matrix system may comprise a plurality of modules (e.g., C100 modules and/or virtual machines running on one or more C100 modules).

54. The following figure provides an overview of the V_matrix "ecosystem" as presented by Lawo:



V_matrix System Overview (Ex. H, at 4-5.)

55. The accused product comprises a master controller that generates input processor control signals and output processor control signals. For example, vsmSTUDIO and/or vsmSOUL provides a "unified orchestration, control and monitoring system" that generates input processor control signals and output processor control signals. The accused product accomplishes IP routing using vsmSOUL:

vsmSOUL Description (Ex. H, at 20.)

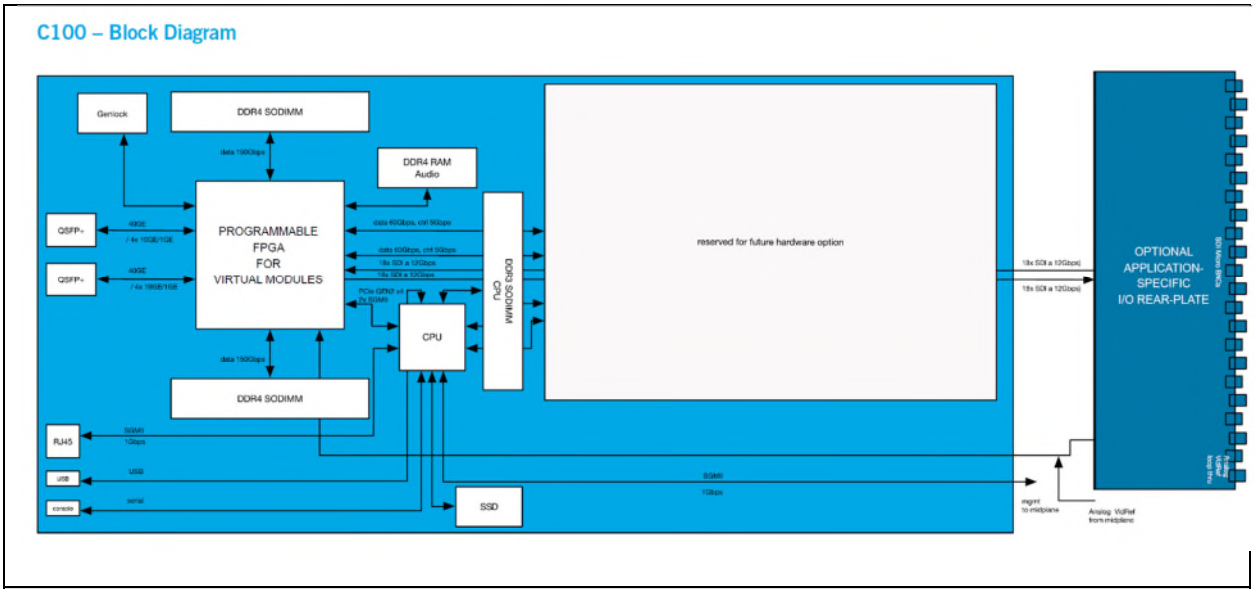
56. The accused product comprises an input processor with one or more input ports, one or more input signal processors, an input processor memory system, one or more packetized signal output ports, one or more packetized signal output stages, and an input processor local controller.

57. For example, the input processor of the accused product comprises one or more components of a C100 module and/or a virtual machine module running on the C100 module (e.g., vm_dmv64-4). The one or more input ports may be data inputs, such as SDI inputs and/or IP inputs. The input processor may utilize one or more input signal processors to provide processed signals. For example, a vm_dmv64-4 may generate downscaled versions of a received signal.

Every vm_dmv64-4 has an input stage capable of receiving up to 24 sources of any combination of 4K/3G/HD/SD which is limited only by the physical (up to 18 SDI inputs) or network (2x 40GbE) I/O. These sources are downscaled by the vm_mv64-4 and returned to the network as IP (RFC 4175) encapsulated mipmaps.

vm_dmv64-4 Description (Ex. H, at 16.)

58. The input processor of the accused product also comprises an input processor memory system for buffering input signals and/or processed signals. For example, each C100 module comprises DDR4 SODIMMs with signal buffering functionality. Processed signals may be buffered using the memory.



C100 – Block Diagram (Ex. H, at 6.)

For example, the accused product buffers received input signals and downscaled versions of received signals before transmitting at least some of the buffered signals as packet source signals in packetized signals.

59. The input processor of the accused product comprises one or more packetized signal output stages for producing and transmitting packetized signals. For example, a C100 module, when configured with a vm_dmv64-4 virtual module, comprises an FPGA and QSFP+ ports for producing packetized signals packets corresponding to the packet source signals, and for transmitting the packetized signal packets. The accused product may utilize IP transmissions to send the packet source signals using one or more packetized signal packets.

V_matrix C100 Core Processing Module (Ex. H, at 6.)

60. The master controller of the accused product assigns a unique global identification code to the packetized signal packets. For example, the master controller may assign a unique multicast address to each of the plurality of packet source signals. In this manner, each packet source signal has a unique global identification code (e.g., a multicast address) that identifies that particular packet source signal.

61. The input processor of the accused product comprises an input processor local controller for controlling the signal processors and the signal output stages based on input processor control signals from the master controller. For example, the FPGA and CPU of a C100 module control operation of the signal processors and the signal output stages. The FPGA and CPU may receive input processor control signals via RJ45 ports from the master controller (e.g., from the vsmSOUL).

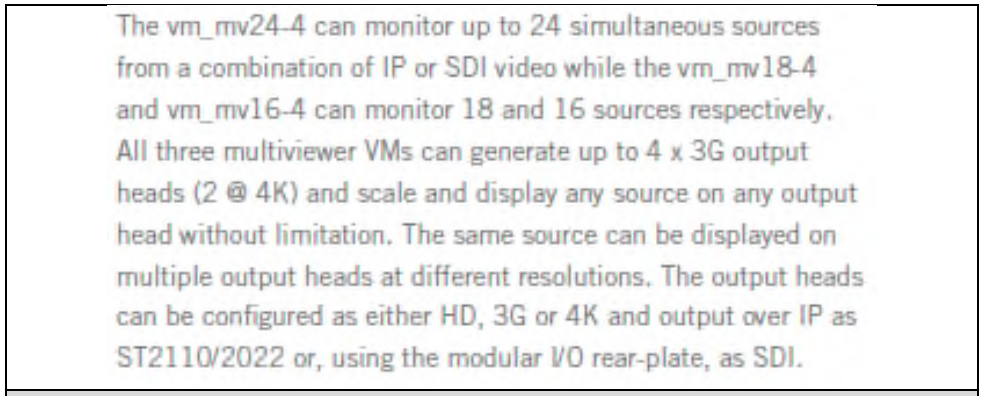
62. The accused product comprises an output processor with one or more packetized signal input ports, one or more packetized signal input stages, one or more output signal generators, and an output processor local controller.

63. The output processor of the accused product comprises one or more packetized signal input ports for receiving packetized signals from the input processor. For example, a vm_mv24-4 module operating on a C100 module may receive packetized signals at one or more

QSFP+ ports through a communications link. The module may identify a particular packet source signal within a packetized signal based on a unique global identification code (e.g., a multicast address).

64. The output processor of the accused product comprises one or more packetized signal input stages. For example, the accused product extracts data corresponding to each packet source signal from packetized signals (e.g., based on a multicast address) via a buffer (e.g., a FIFO buffer utilizing the DDR4 SODIMMs).

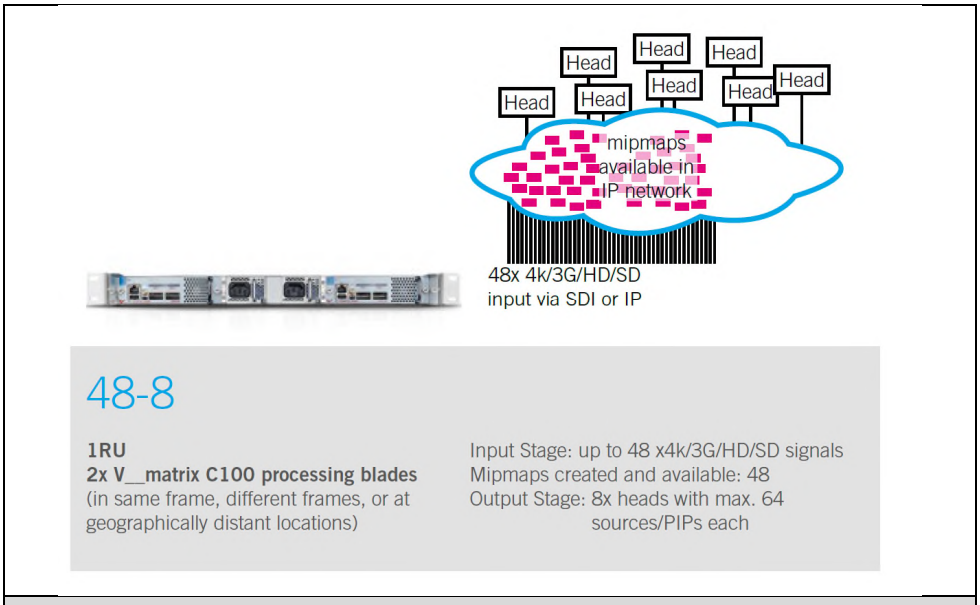
65. The output processor of the accused product comprises an output signal generator that provides one or more output signals. For example, a vm_mv24-4 module provides up to four output signals, which may each contain a mosaic of buffered packet source signals.



vm_mv24-4 Description (Ex. H, at 14.)

66. The output processor of the accused product comprises a local controller for controlling the packetized signal input stages and the output signal generators based on output processor control signals from the master controller. For example, the FPGA and CPU of a C100 module control operation of the signal input stages and the output signal generators. The FPGA and CPU may receive output processor control signals via RJ45 ports from the master controller (e.g., from the vsmSOUL).

67. The accused product comprises a communications link between the packetized signal output ports and the packetized signal input ports. The accused product is designed to utilize an IP infrastructure to connect one or more modules. For example, a C100 module executing a vm_dmv_64-4 module may comprise QSFP+ ports that are communicatively coupled, via IP communication, to QSFP+ ports of a second C100 module executing a vm_mv24-4 module. Downscaled versions of a received signal (e.g., mipmaps) may thus be sent from an input processor of one module, via a communication link, to an output processor of a second module. In some instances, one or more other communication links, such as an internal backplane and/or switch for communication within a frame, may substitute for and/or augment IP communication via QSFP+ ports.



Virtual Module (Ex. H, at 17.)

68. Evertz has not licensed or granted permission to Lawo to use the subject matter of the ’838 Patent.

69. Each Lawo defendant is aware of the ‘838 Patent and its infringement at least as of the date of this Amended Complaint and has willfully infringed the ’838 Patent.

70. Evertz has been damaged and continues to be damaged by Lawo's infringement of the '838 Patent.

71. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo's wrongful acts in an amount subject to proof at trial.

72. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo's infringement of the '838 Patent. By way of example, Lawo's infringing products compete with Evertz's products and Lawo's infringing products were willfully derived from Evertz's proprietary information that is now embodied in '838 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

COUNT II
INFRINGEMENT OF U.S. PATENT NO. 9,100,217

73. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 72 above as if fully set forth herein.

74. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the '217 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling (directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

75. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the '217 Patent that reads:

> A method of producing a packetized signal comprising:
>
> receiving one or more input signals;
>
> determining which of the one or more input signals and signals derived from the one or more input signals are required to generate the packetized signal;

> upon determining the one or more input signals required to generate the packetized signal, buffering the one or more input signals required to generate the packetized signal in a memory system;
>
> upon determining the signals derived from the one or more input signals required to generate the packetized signal, processing at least one of the one or more input signals to provide a corresponding processed signal, wherein the corresponding processed signal is required to generate the packetized signal, and buffering the corresponding processed signal in the memory system;
>
> designating the one or more input signals buffered in the memory system and the corresponding processed signal buffered in the memory system as packet source signals and assigning each of the packet source signals a unique global identification code; and
>
> generating the packetized signal wherein the packetized signal includes a series of packetized signal packets, wherein each of the packetized signal packets contains the unique global identification code of one of the packet source signals and data corresponding to the same packet source signal.

76. Each Lawo defendant performs each and every limitation of ’217 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

77. The accused product receives one or more input signals and determines which of the one or more input signals and signals derived from the one or more input signals are required to generate the packetized signal. For example, the accused product comprises an input processor with one or more input ports for receiving one or more input signals. The input processor of the accused product comprises one or more components of a C100 module and/or a virtual machine module running on the C100 module (e.g., vm_dmv64-4). The input ports may be data inputs, such as SDI inputs and/or IP inputs (Ex. H, at 6, 10). The accused product may determine the received input signals and downscaled versions of received signals in packetized signals (Ex. H, at 6).

78. The accused product processes at least one of the one or more input signals, and buffers the input signals and the processed signals. For example, the input processor may utilize

one or more input signal processors to provide processed signals. For example, a vm_dmv64-4 may generate downscaled versions of a received input signal (Ex. H, at 10).

79. The input processor of the accused product also comprises an input processor memory system for buffering input signals and/or processed signals. For example, each C100 module comprises DDR4 SODIMMs with signal buffering functionality. The accused product buffers received input signals and downscaled versions of received signals before including at least some of the buffered signals as packet source signals in packetized signals (Ex. H, at 6).

80. The accused product designates some of the buffered signals as packet source signals and assigns a unique global identification code to each packet source signal. For example, a unique multicast address is assigned to each of the plurality of packet source signals in the accused product. In this manner, each packet source signal has a unique global identification code (e.g., a multicast address) that identifies that particular packet source signal (Ex. H, at 16, 17).

81. The input processor of the accused product produces packetized signals. For example, a C100 module, when configured with a vm_dmv64-4 virtual module, comprises an FPGA and QSFP+ ports for producing packetized signals packets corresponding to the packet source signals (Ex. H, at 11).

82. Evertz has not licensed or granted permission to Lawo to use the subject matter of the ’217 Patent.

83. Each Lawo defendant is aware of the ‘217 Patent and its infringement at least as of the date of this amended Complaint and has willfully infringed the ’217 Patent.

84. Evertz has been damaged and continues to be damaged by Lawo’s infringement of the ’217 Patent.

85. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo's wrongful acts in an amount subject to proof at trial.

86. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo's infringement of the '217 Patent. By way of example, Lawo's infringing products compete with Evertz's products and Lawo's infringing products were willfully derived from Evertz's proprietary information that is now embodied at least '217 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

COUNT III
INFRINGEMENT OF U.S. PATENT NO. 9,473,322

87. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 86 above as if fully set forth herein.

88. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the '322 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling (directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

89. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the '322 Patent that reads:

> A method of producing a packetized signal comprising:
>
> receiving one or more input signals;
>
> automatically assigning each of the one or more input signals a unique global identification code;
>
> deriving one or more derived signals from at least one of the input signals;
>
> determining which of the one or more input signals and derived signals are required to generate the packetized signal;
>
> upon determining the derived signals required to generate the packetized signal, processing at least one input signals to provide a corresponding processed signal,

> wherein the corresponding processed signal is required to generate the packetized signal, and buffering the corresponding processed signal in the memory system; and
>
> upon determining the one or more input signals required to generate the packetized signal, buffering the one or more input signals required to generate the packetized signal in a memory system.

90. Each Lawo defendant performs each and every limitation of ’322 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

91. The accused product comprises an input processor with one or more input ports for receiving one or more input signals. For example, the input processor of the accused product comprises one or more components of a C100 module and/or a virtual machine module running on the C100 module (e.g., vm_dmv64-4). The one or more input ports may be data inputs, such as SDI inputs and/or IP inputs (Ex. H, at 6, 10).

92. The accused product automatically assigns unique global identification code to at least one received input signal. For example, in the accused product, a unique multicast address is assigned to each of the plurality of buffered signals, including at least one received input signal. In this manner, each buffered signal has a unique global identification code (e.g., a multicast address) that identifies that buffered signal (Ex. H, at 16, 17).

93. The accused product derives a signal from the input signals. For example, a vm_dmv64-4 may generate downscaled versions of a received signal (Ex. H, at 10).

94. The accused product determines which of the input signals and derived signals are required to generate the packetized signal. For example, the accused product may transmit pre-determined received input signals and downscaled versions of received signals in packetized signals (Ex. H, at 6).

95. Based on the determination, the accused product processes at least one input signal, and buffers the input signals and the processed signals. For example, the input processor may utilize one or more input signal processors to provide downscaled versions of a received signal (Ex. H, at 10).

96. The input processor of the accused product also comprises an input processor memory system for buffering input signals and/or processed signals. For example, each C100 module comprises DDR4 SODIMMs with signal buffering functionality. The accused product buffers received input signals and downscaled versions of received signals before including at least some of the buffered signals in packetized signals (Ex. H, at 6).

97. Evertz has not licensed or granted permission to Lawo to use the subject matter of the ’322 Patent.

98. Each Lawo defendant is aware of the ‘322 Patent and its infringement at least as of the date of this Amended Complaint and has willfully infringed the ’322 Patent.

99. Evertz has been damaged and continues to be damaged by Lawo’s infringement of the ’322 Patent.

100. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo’s wrongful acts in an amount subject to proof at trial.

101. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo’s infringement of the ’322 Patent. By way of example, Lawo’s infringing products compete with Evertz’s products and Lawo’s infringing products were willfully derived from Evertz’s proprietary information that is now embodied in the ’322 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

COUNT IV
INFRINGEMENT OF U.S. PATENT NO. 8,270,398

102. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 101 above as if fully set forth herein.

103. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the '398 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling (directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

104. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the '398 Patent that reads:

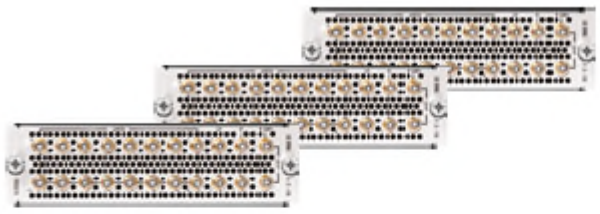
> A modular system for processing signals comprising:
>
> > a plurality of frame input terminals for receiving a plurality of input signals;
> >
> > a plurality of input modules coupled to the frame input terminals to receive the input signals;
> >
> > a cross point switch coupled to the input modules at a plurality of cross point switch input terminals to receive the input signals, wherein the cross point switch is configurable to couple one or more of the input signals to each of a plurality of processor input terminals;
> >
> > a plurality of output modules, each output module including one or more output processors, wherein each of at least some of the output processors is coupled to one or more of the processor input terminals to receive one or more of the input signals, and wherein each output module has one or more frame output terminals; and
> >
> > a controller coupled to:
> >
> > > the cross point switch to couple at least some of the frame input terminals to one of the processor input terminals whereby the input signals received at the frame input terminals are provided at the corresponding processor input terminals; and
> > >
> > > at least one of the output processors to controllably configure the at least one output processor to processor input signals received at the one or more processor input terminals of the at least one output processor to provide one or more output signals at the one or more frame output terminals of the at least one output processor.

105. The Lawo products including, but not limited to, the Lawo V_matrix system perform each and every limitation of ’398 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

106. The accused product comprises a plurality of frame input terminals, a plurality of input modules, a cross point switch coupled to the plurality of input modules, a plurality of output modules and a controller coupled to the cross point switch and at least one of the plurality of output processors.

107. By way of an example, in one implementation of the accused product, the accused product may comprise at least four C100 modules (e.g., four C100 modules installed in a 2RU or a 3RU frame, or four C100 modules installed in multiple frames) (Ex. H, at 6, 16, 18).

108. The accused product comprises frame input terminals for receiving a plurality of input signals. For example, each C100 module is connected to a rear-mounted I/O interface module to be able to receive SDI video inputs, and the C100 module comprises QSFP+ ports to be able to receive IP video inputs. (Ex. H, at 6, 7, 19). The rear-plate I/O modules may be configured to include 10 SDI input and 10 SDI output connections, 18 SDI input and 2 SDI output connections, 2 SDI input and 18 SDI output connections, and 2 SDI input, 2 SDI output and 18 bidirectional input/output connections (Ex. H, at 22).



V__matrix REAR-PLATE I/O MODULES

The V__matrix rear-plate I/O modules provide additional interfaces on the back of the processing blade. These rear-plates house a variety of application specific physical interface connectors in order to provide connectivity to legacy broadcast equipment such as baseband video and audio components. All

V_matrix Rear-Plate I/O Modules (Ex. H, at 7.)

109. The accused product comprises a plurality of input modules coupled to the frame input terminals to receive the input signals. For example, each input module of the accused product may comprise one or more components of a C100 module and/or a virtual machine module running on the C100 module (e.g., vm_dmv64-4). In this implementation, at least two C100 modules receive SDI or IP input signals from the frame input terminals.

Every vm_dmv64-4 has an input stage capable of receiving up to 24 sources of any combination of 4K/3G/HD/SD which is limited only by the physical (up to 18 SDI inputs) or network (2x 40GbE) I/O. These sources are downscaled by the vm_mv64-4 and returned to the network as IP (RFC 4175) encapsulated mipmaps.

vm_dmv64-4 Description (Ex. H, at 16.)

110. The accused product comprises a cross point switch coupled to the input modules to couple one or more input signals to each of a plurality of processor input terminals. For example, upon information and belief, the V_matrix frame of the accused product may include an internal

switching fabric for routing video signals from one C100 module to another C100 module installed in the same frame. The internal switching fabric may receive routing control signals from a controller (e.g., from the vsmSOUL) (Ex. H, at 6, 20-21).

111. The accused product comprises a plurality of output modules where each output module includes one or more output processors. For example, each output module of the accused product may comprise one or more components of a C100 module and/or a virtual machine module running on the C100 module (e.g., vm_dmv64-4, vm_mv24-4, etc.). The vm_dmv64-4 or vm_mv24-4 module operating on a C100 module may receive one or more input signals (e.g., at one or more QSFP+ ports from the V_matrix frame's internal switching fabric), and may generate output signals (e.g., at one or more QSFP+ ports).

112. The accused product comprises a controller to control the cross point switch and at least one output processor. The V_matrix frame may provide power and connectivity for control and monitoring to the C100 modules installed in the frame (Ex. H, at 6). For example, each input module, output module, and internal switching fabric in the V_matrix system is coupled to vsmSTUDIO and/or vsmSOUL through a network. VsmSTUDIO and vsmSOUL provide a "unified orchestration, control and monitoring system" that generates control signals for the C100 modules and the V_matrix frame (Ex. H, at 4, 5, 19).

113. The accused product may comprise C100 modules in various arrangements. For example, two C100 modules may be installed in one frame and two C100 modules may be installed in a different frame. In another example, one C100 module may be installed in one frame and three C100 modules may be installed in another frame. In yet another example, each of four C100 modules may be installed in four different frames. At least four C100 modules (six modules, eight modules, etc.) may be installed across any number of frames. The C100 modules may be installed

in different frames, and may be connected through an external switching fabric and cables connecting the different V_matrix frames (Ex. H, at 6, 16, 17). In various different arrangements involving at least two V_matrix frames, the external switching fabric connecting the V_matrix frames additionally routes video signals between the various C100 modules.

114. Evertz has not licensed or granted permission to Lawo to use the subject matter of the ’398 Patent.

115. Each Lawo defendant is aware of the ’398 Patent and its infringement at least as of the date of this Amended Complaint and has willfully infringed the ’398 Patent.

116. Evertz has been damaged and continues to be damaged by Lawo’s infringement of the ’398 Patent.

117. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo’s wrongful acts in an amount subject to proof at trial.

118. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo’s infringement of the ’398 Patent. By way of example, Lawo’s infringing products compete with Evertz’s products and Lawo’s infringing products were willfully derived from Evertz’s proprietary information that is now embodied at least ’398 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

COUNT V
INFRINGEMENT OF U.S. PATENT NO. 9,654,391

119. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 118 above as if fully set forth herein.

120. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the ’391 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling

(directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

121. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the ’391 Patent that reads:

> A video router comprising:
>
> a backplane including a plurality of static point-to-point backplane connections;
>
> a plurality of line cards, each line card including:
>
> a plurality of input ports and output ports, each input port and output port is coupled to a respective external signal through the backplane;
>
> a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals such that a first plurality of input and output switch terminals are coupled to the respective plurality of input and output ports and a second plurality of input and output switch terminals are coupled to the plurality of backplane connections;
>
> a line card controller coupled to the line card cross-point switch to selectively couple some of the input switch terminals to the output switch terminals;
>
> one or more fabric cards, each fabric card including:
>
> a fabric card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals coupled to the plurality of backplane connections;
>
> a fabric card controller coupled to the fabric card cross-point switch to selectively couple some of the input switch terminals to the output switch terminals; and
>
> a controller communication network coupled to each of the line cards and fabric cards to control the operation of the fabric card controllers and the line card controllers.

122. The Lawo products including, but not limited to, the Lawo V_matrix system perform each and every limitation of ’391 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

123. The accused product comprises a backplane, a plurality of line cards, one or more fabric cards and a controller. By way of an example, in one implementation, the accused product

comprises two C100 modules installed in a single V_matrix frame (e.g. in 1RU frame) (Ex. H, at 6, 16, 18).

124. In this implementation, a V_matrix frame of the accused product comprises the backplane. The V_matrix frame provides power and connectivity for control and monitoring to the C100 modules installed in the frame. (Ex. H, at 6). The V_matrix frame, on information and belief, may have internal static connections for transmitting and receiving video signals to and from the C100 modules (Ex. H, at 6).

The V__matrix frames provide power and protected housing for the V__matrix processing blades. Each frame has a dedicated 1GE management port that provides connectivity for control and monitoring to all installed processing modules of the frame.
V_matrix FRAMES Description (Ex. H, at 6.)

125. Each line card of the accused product includes a plurality of input ports and output ports. For example, each line card of the accused product comprises one or more components of a C100 module and/or a virtual machine module running on the C100 module (e.g., vm_dmv64-4). Each C100 module is connected to a rear-mounted I/O interface module to be able to receive SDI video inputs, and the C100 module comprises QSFP+ ports to be able to receive IP video inputs (Ex. H, at 6, 7, 19). Similarly, each C100 module is connected to a rear-mounted I/O interface module to be able to provide SDI video outputs, and the C100 module comprises QSFP+ ports to be able to provide IP video outputs (Ex. H, at 6, 7, 19).

126. Each line card of the accused product also includes a line card cross-point switch having a plurality of input switch terminals and output switch terminals. For example, each C100 module comprises an FPGA with signal routing functionality, where the FPGA is configured to receive or transmit SDI or IP video signals to or from the rear-mounted I/O interface module and

the QSFP+ ports. At least some of the SDI or IP video signals, on information and belief, may be transmitted to or received from the internal connections and/or traces in the V_matrix frame (Ex. H, at 6).

127. Each line card of the accused product includes a line card controller coupled to the line card cross-point switch. For example, the CPU of a C100 module controls the operation of the FPGA to facilitate video signal routing. The FPGA and CPU may receive control signals from a controller (e.g., from the vsmSOUL) (Ex. H, at 6).

128. The accused product comprises one or more fabric cards where each fabric card includes a fabric card cross-point switch having input switch terminals and output switch terminals and a fabric card controller coupled to the fabric card cross-point switch. For example, upon information and belief, the V_matrix frame of the accused product may include an internal switching fabric which may route the video signals to and/or from the internal static connections that comprise the backplane (Ex. H, at 6).

129. The accused product comprises a controller communication network to control the operation of each line card and fabric card. For example, upon information and belief, each line card and the internal switching fabric in the V_matrix frame is coupled to vsmSTUDIO and/or vsmSOUL through a network. vsmSTUDIO and VSMSOUL provides a “unified orchestration, control and monitoring system” that generates control signals for the C100 modules and the V_matrix frame (Ex. H, at 4, 5, 19). The accused product accomplishes IP routing using vsmSOUL (Ex. H, at 19, 20).

130. In another implementation of the accused product, at least two C100 modules are installed in different frames (e.g., one C100 module is installed in one frame and another C100 module is installed in another frame). In this implementation, the two C100 modules are connected

through an external switching fabric and/or cables connecting the C100 modules or the V_matrix frames to the external switching fabric (Ex. H, at 6, 16, 17).

131. In this implementation, the backplane comprises the static connections within V_matrix frames housing the two C100 modules as well as the cables connecting the C100 modules or the V_matrix frames to the external switching fabric (Ex. H, at 6).

132. The accused product comprises a plurality of line cards with each line card including input and output ports, a line card cross-point switch and a line card controller. For example, a line card of the accused product comprises a C100 module, where the C100 module is connected to a rear-mounted I/O interface module for receiving and/or transmitting SDI video signals and is connected to QSFP+ ports to be able to receive and transmit IP video signals (Ex. H, at 6, 7, 19).

133. Each line card comprises a cross-point switch. For example, each C100 module comprises an FPGA with signal routing functionality, where the FPGA is configured to receive or transmit SDI or IP video signals to or from the rear-mounted I/O interface module and/or the QSFP+ ports. At least some of the SDI or IP video signals may be transmitted to or received from the cables connecting the V_matrix frames, the internal static connections in the V_matrix frames, or both (Ex. H, at 6).

134. Each line card of the accused product includes a line card controller coupled to the line card cross-point switch. For example, the CPU of a C100 module controls the operation of the FPGA to facilitate video signal routing. The FPGA and CPU may receive control signals from a controller (e.g., from the vsmSOUL) (Ex. H, at 6).

135. The accused product comprises at least one fabric card with each fabric card having a fabric card cross-point switch and a fabric card controller. For example, in this implementation,

the external switching fabric coupled between the two V_matrix frames in the accused product route the video signals between the C100 modules. The V_matrix frame of the accused product, on information and belief, may also include an internal switching fabric for routing the video signals between the internal static connections that comprise the backplane (Ex. H, at 6).

136. The accused product comprises a controller communication network to control the operation of each line card and fabric card. For example, upon information and belief, each line card, the internal switching fabric in the V_matrix frames, and the external switching fabric are coupled to the vsmSTUDIO and/or vsmSOUL through a network. vsmSTUDIO and vsmSOUL provide a "unified orchestration, control and monitoring system" that generates control signals for the C100 modules, the V_matrix frame and the external switching fabric connecting multiple V_matrix frames.

137. Evertz has not licensed or granted permission to Lawo to use the subject matter of the '391 Patent.

138. Each Lawo defendant is aware of the '391 Patent and its infringement at least as of the date of this Amended Complaint and has willfully infringed the '391 Patent.

139. Evertz has been damaged and continues to be damaged by Lawo's infringement of the '391 Patent.

140. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo's wrongful acts in an amount subject to proof at trial.

141. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo's infringement of the '391 Patent. By way of example, Lawo's infringing products compete with Evertz's products and Lawo's infringing

products were willfully derived from Evertz's proprietary information that is now embodied at least '391 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

COUNT VI
INFRINGEMENT OF U.S. PATENT NO. 9,942,139

142. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 141 above as if fully set forth herein.

143. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the '139 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling (directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

144. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the '139 Patent that reads:

> A data transmission system, comprising:
>
> a plurality of video routers;
>
> a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising instructions to selectively configure the one or more routers; and
>
> a controller communication network for coupling the plurality of video routers and the supervisory system,
>
> wherein, each video router comprises:
>
> a backplane including a plurality of backplane connections,
>
> at least one line card, the line card comprising:
>
> a plurality of input ports and output ports, each input port and output port being coupled to a respective external signal through the backplane, and
>
> a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein a first plurality of input and output switch terminals are coupled to a respective plurality of input and output ports and a second plurality of input and output switch terminals are coupled to a respective plurality of backplane connections, and

at least one fabric card, each fabric card comprising:

a fabric card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein the plurality of input and output switch terminals are coupled to a respective plurality of backplane connections, and

wherein, each line card and fabric card comprises a card controller, the card controller being coupled to one or more cross-point switches and configured to selectively couple one or more input switch terminals of a cross-point switch to one or more output switch terminals of that cross-point switch, the cross-point switch being a fabric card cross-point switch or a line card cross-point switch,

wherein, the controller communication network is communicably coupled to each card controller to control the operation of each line card and fabric card.

145. The Lawo products including, but not limited to, the Lawo V_matrix system perform each and every limitation of ’139 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

146. The accused product comprises a plurality of video routers, a supervisory system and a controller communication network coupling the plurality of video routers and the supervisory system.

147. For example, the accused product may be used in a broadcast facility where multiple V_matrix frames and C100 modules are used to transmit video, audio and/or data signals. The multiple V_matrix frames may be connected to each other via an external switching fabric as well as cables connecting the C100 modules or the V_matrix frames to the external switching fabric (Ex. H, at 4 - 5 and 16 – 17).

148. The accused product comprises a supervisory system that transmits router configuration signals to the video routers. For example, vsmSTUDIO and/or vsmSOUL provides a control and monitoring system that generates control signals for configuring the various video routers (Ex. H, at 4, 5).

149. Evertz has not licensed or granted permission to Lawo to use the subject matter of the ’139 Patent.

150. Each Lawo defendant is aware of the ‘139 Patent and its infringement at least as of the date of this Amended Complaint and has willfully infringed the ’139 Patent.

151. Evertz has been damaged and continues to be damaged by Lawo’s infringement of the ’139 Patent.

152. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo’s wrongful acts in an amount subject to proof at trial.

153. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo’s infringement of the ’139 Patent. By way of example, Lawo’s infringing products compete with Evertz’s products and Lawo’s infringing products were willfully derived from Evertz’s proprietary information that is now embodied at least ’139 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

COUNT VII
INFRINGEMENT OF U.S. PATENT NO. 10,164,877

154. Evertz repeats and re-alleges the allegations contained in Paragraphs 1 through 153 above as if fully set forth herein.

155. Each Lawo defendant, individually and collectively, has directly infringed and continues to directly infringe, literally or under the doctrine of equivalents, one or more claims of the ’877 Patent in violation of 35 U.S.C. § 271(a) by making, using, offering to sell, selling (directly or through intermediaries), and/or importing into the United States, Lawo products including, but not limited to, the Lawo V_matrix system.

156. For example, each Lawo defendant, individually and collectively, infringes at least claim 1 of the ’877 Patent that reads:

A priority based transmission system comprising:

a plurality of data signals;

a plurality of video routers;

a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising a data signal path;

a controller communication network for coupling the plurality of video routers and the supervisory system;

wherein, each video router comprises:

a backplane including a plurality of backplane connections,

at least one line card, the line card comprising:

a plurality of input ports and output ports, each input port and output port being coupled to a respective data signal through the backplane, and

a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein a first plurality of input and output switch terminals are coupled to a respective plurality of input and output ports and a second plurality of input and output switch terminals are coupled to a respective plurality of backplane connections, and

at least one fabric card, each fabric card comprising: a fabric card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein the plurality of input and output switch terminals are coupled to a respective plurality of backplane connections,

wherein the data signal path comprises an input switch terminal, one or more cross-point switches from one or more video routers, and an output switch terminal, and

wherein, each line card and fabric card comprises a card controller, the card controller being coupled to one or more cross-point switches and configured to determine the path of one or more data signals based on the router configuration signals.

157. The Lawo products including, but not limited to, the Lawo V_matrix system perform each and every limitation of ’877 Patent claim 1. The following paragraphs explain the infringement in detail, with particular reference to Exhibit H.

158. The accused product comprises a plurality of data signals, a plurality of video routers, a supervisory system and a controller communication network coupling the plurality of video routers and the supervisory system.

159. For example, the accused product may be used in a broadcast facility where multiple V_matrix frames and C100 modules are used to transmit video, audio and/or data signals. The multiple V_matrix frames may be connected to each other via an external switching fabric as well as cables connecting the C100 modules or the V_matrix frames to the external switching fabric (Ex. H, at 4 - 5 and 16 – 17).

160. The accused product comprises a supervisory system that transmits router configuration signals to the video routers where the router configuration signals comprise a data signal path. For example, vsmSTUDIO and/or vsmSOUL provides a control and monitoring system that generates control signals for configuring the various video routers in order to provide a signal path for the incoming video, audio and/or data signals (Ex. H, at 4, 5).

161. The signal path of the incoming video, audio and/or data signals comprises two or more line cards, one or more V_matrix frames and optionally an external switching fabric as well as cables connecting the C100 modules or the V_matrix frames to the external switching fabric.

162. Evertz has not licensed or granted permission to Lawo to use the subject matter of the ’877 Patent.

163. Each Lawo defendant is aware of the ‘877 Patent and its infringement at least as of the date of this Amended Complaint and has willfully infringed the ’877 Patent.

164. Evertz has been damaged and continues to be damaged by Lawo’s infringement of the ’877 Patent.

165. Evertz is entitled to recover from Lawo the damages sustained by Evertz as a result of Lawo's wrongful acts in an amount subject to proof at trial.

166. Evertz is suffering and will continue to suffer irreparable harm for which there is no adequate remedy at law as a result of Lawo's infringement of the '877 Patent. By way of example, Lawo's infringing products compete with Evertz's products and Lawo's infringing products were willfully derived from Evertz's proprietary information that is now embodied at least '877 Patent claim 1. Unless enjoined, Lawo will continue its infringing conduct.

PRAYER FOR RELIEF

WHEREFORE, Evertz respectfully requests that this Court enter judgment against each Lawo defendant, granting Evertz the following relief:

A. A judgment holding each Lawo defendant liable for direct infringement of the Patents-In-Suit, and that all such infringements have been willful;

B. All damages available under 35 U.S.C. § 284 resulting from Lawo's willful infringement of the Patents-In-Suit in an amount to be proven at trial, but no less than a reasonable royalty, including treble damages, based on any infringement found to be willful and egregious, together with pre-judgment interest and post-judgment interest;

C. An order and judgment permanently enjoining each Lawo defendant and those acting in active and concert participation with them from further acts of infringement of the Patents-In-Suit;

D. A judgment holding this to be an exceptional case, and an award to Evertz for its attorneys' fees, costs, and expenses incurred prosecuting this action pursuant to 35 U.S.C. § 285; and

E. Such other and further relief as the Court deems just and equitable.

DEMAND FOR JURY TRIAL

Evertz demands a trial by jury of all issues so triable.

POTTER ANDERSON & CORROON LLP

OF COUNSEL:

Binal J. Patel
Erik S. Maurer
BANNER & WITCOFF, LTD.
71 S. Wacker Drive – Suite 3600
Chicago, IL 60606
(312) 463-5000

Joshua L. Davenport
BANNER & WITCOFF, LTD.
1100 13th Street, NW – Suite 1200
Washington, D.C. 20005
(202) 824-3000

By: */s/ Jonathan A. Choa*
Jonathan A. Choa (#5319)
Alan Silverstein (#5066)
Hercules Plaza
P.O. Box 951
Wilmington, DE 19899
(302) 984-6000
jchoa@potteranderson.com
asilverstein@potteranderson.com

Attorneys for Plaintiff
Evertz Microsystems Ltd.

March 11, 2019
6106637

EXHIBIT A

US008537838B2

(12) **United States Patent**
Patel et al.

(10) **Patent No.: US 8,537,838 B2**
(45) **Date of Patent: Sep. 17, 2013**

(54) **PACKET BASED TRANSMISSION OF MULTIPLE DATA SIGNALS**

(75) Inventors: **Rakesh Thakor Patel**, Oakville (CA); **Romolo Magarelli**, Woodbridge (CA)

(73) Assignee: **Evertz Microsystems Ltd.**, Burlington (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 979 days.

(21) Appl. No.: **10/816,841**

(22) Filed: **Apr. 5, 2004**

(65) **Prior Publication Data**

US 2004/0208163 A1 Oct. 21, 2004

Related U.S. Application Data

(60) Provisional application No. 60/459,964, filed on Apr. 4, 2003.

(51) **Int. Cl.**
H04L 12/28 (2006.01)

(52) **U.S. Cl.**
USPC **370/399**; 370/423; 370/429

(58) **Field of Classification Search**
USPC 370/412–418, 422–424, 428, 429; 455/132, 133, 137
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,682,195	A	10/1997	Hendricks et al.	
6,430,183	B1 *	8/2002	Satran et al.	370/389
6,519,223	B1 *	2/2003	Wager et al.	370/216
6,937,291	B1 *	8/2005	Gryskiewicz	348/581
7,190,695	B2 *	3/2007	Schaub et al.	370/392
7,269,841	B1 *	9/2007	Hendricks et al.	725/94
2003/0063611	A1	4/2003	Schaub et al.	
2003/0156535	A1 *	8/2003	Lebizay et al.	370/229
2003/0198226	A1 *	10/2003	Westberg	370/393

FOREIGN PATENT DOCUMENTS

FR	2655451	6/1991
FR	2726723	5/1996
FR	2793042	11/2000
FR	2793043	11/2000

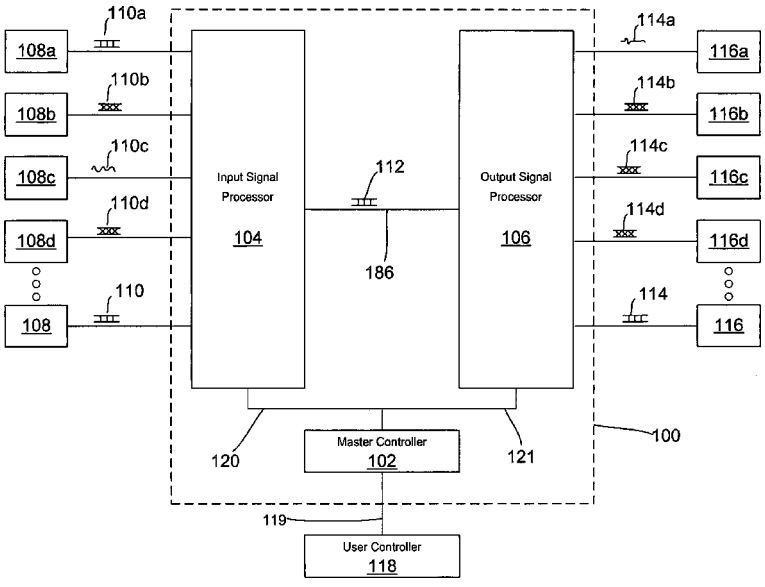
* cited by examiner

Primary Examiner — Alvin Zhu
(74) *Attorney, Agent, or Firm* — Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.

(57) **ABSTRACT**

Apparatus, systems and methods for receiving one or more input signals and providing output signals in various video, audio, data and mixed formats are described. One or more input processors receive the input signals. Each of the input processors provides one or more packetized signals corresponding to one or more of the input signals received at the input processor. Each output processor can receive one or more packetized signals and generate or more output signals. The output signals correspond to one or more of the input signals, additional locally generated signals or data relating to the signals or any combination of such signals. Use of a packet router according to the invention allows input signals encoded as one set of packetized signals to be recombined to provide additional packetized signals incorporating the same or different combinations of the packetized signals.

6 Claims, 11 Drawing Sheets



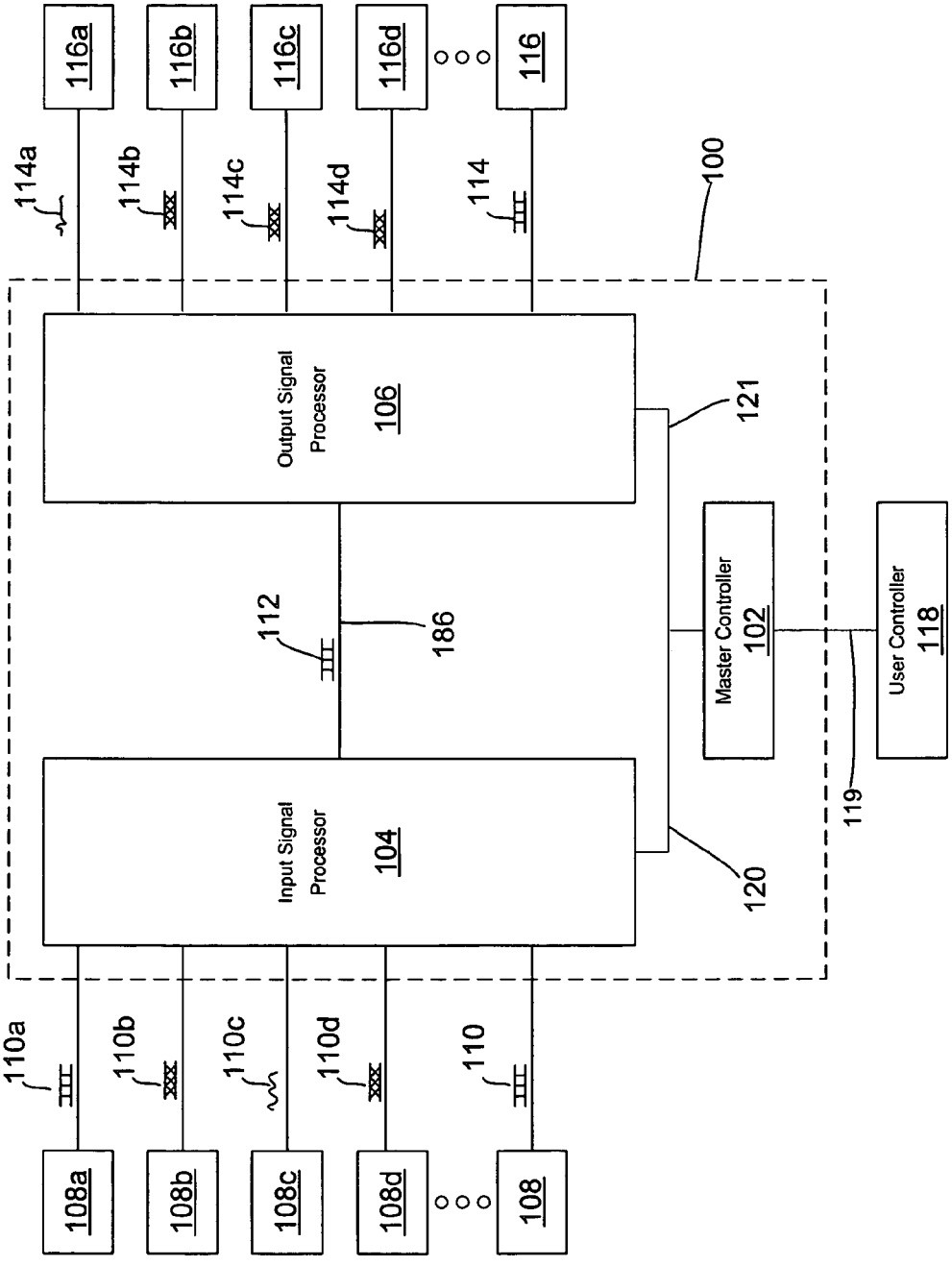

116a
116b
116c
116d
116
114a
114b
114c
114d
114
100
Output Signal Processor
106
121
112
186
Master Controller
102
User Controller
118
119
Input Signal Processor
104
120
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 1

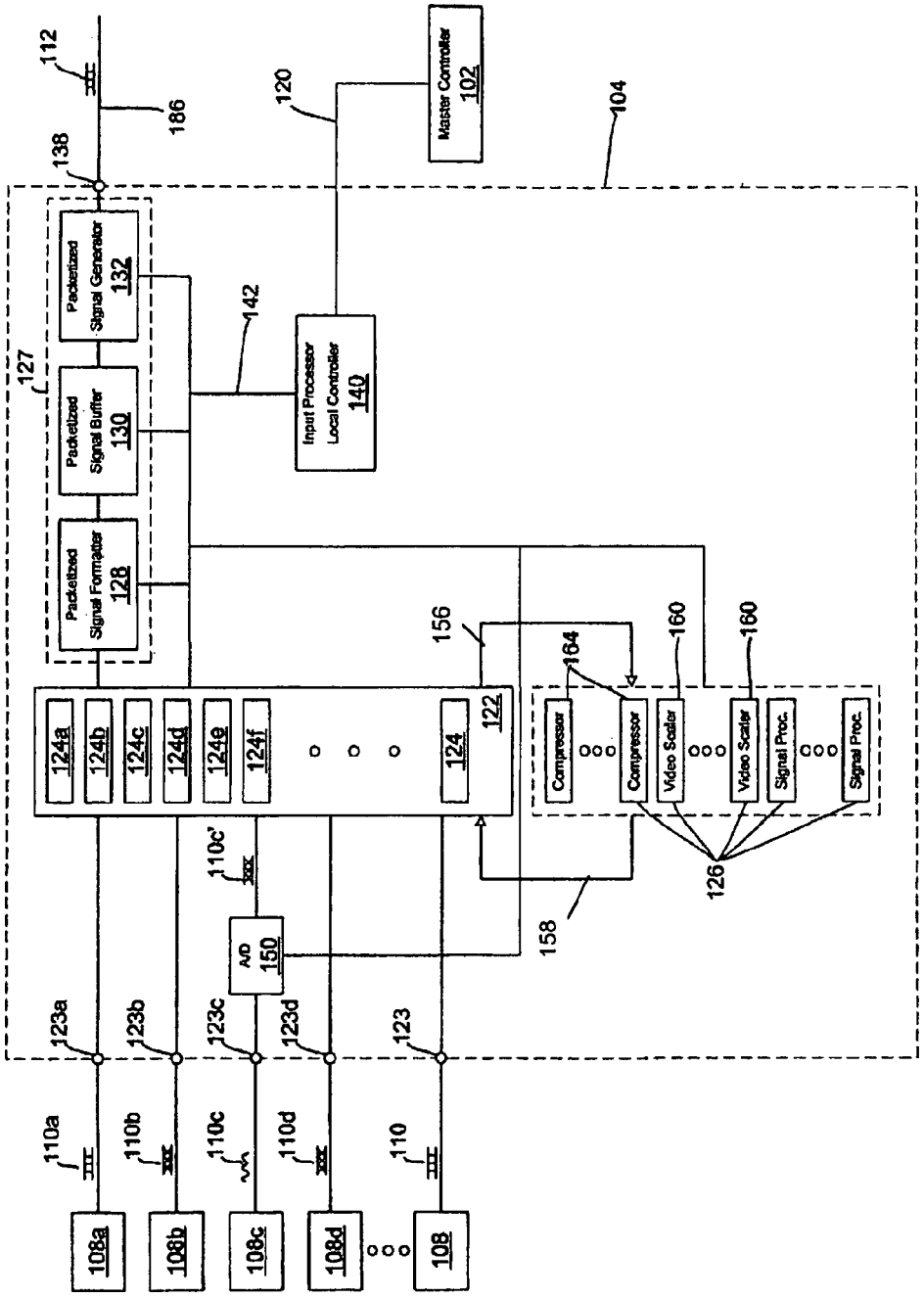

112
186
120
Master Controller 102
104
138
Packetized Signal Generator 132
142
127
Packetized Signal Buffer 130
Input Processor Local Controller 140
Packetized Signal Formatter 128
156
164
160
160
124a
124b
124c
124d
124e
124f
124
122
Compressor
Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
110c'
126
158
A/D 150
123a
123b
123c
123d
123
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 2

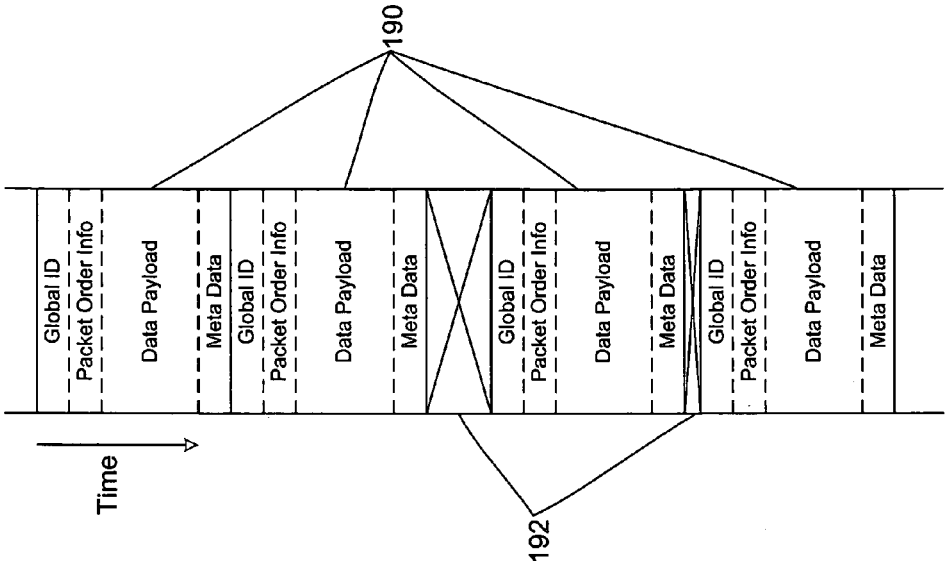


Figure 5

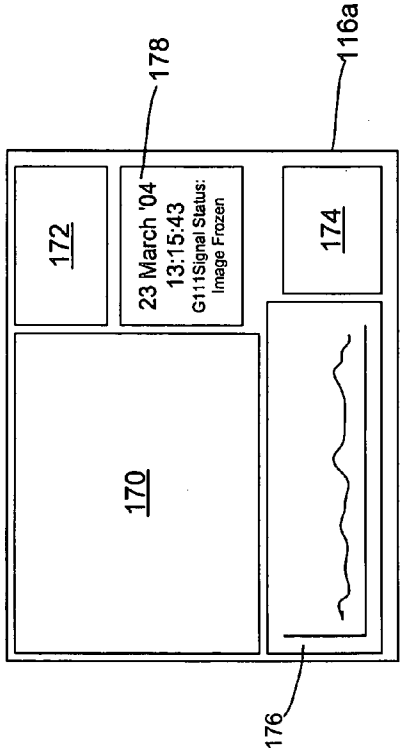


Figure 3

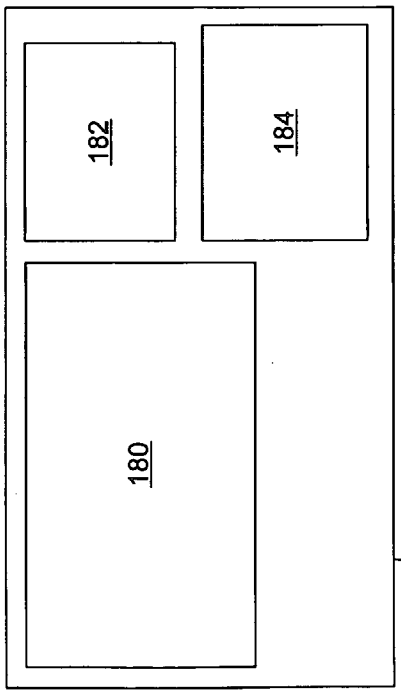


Figure 4

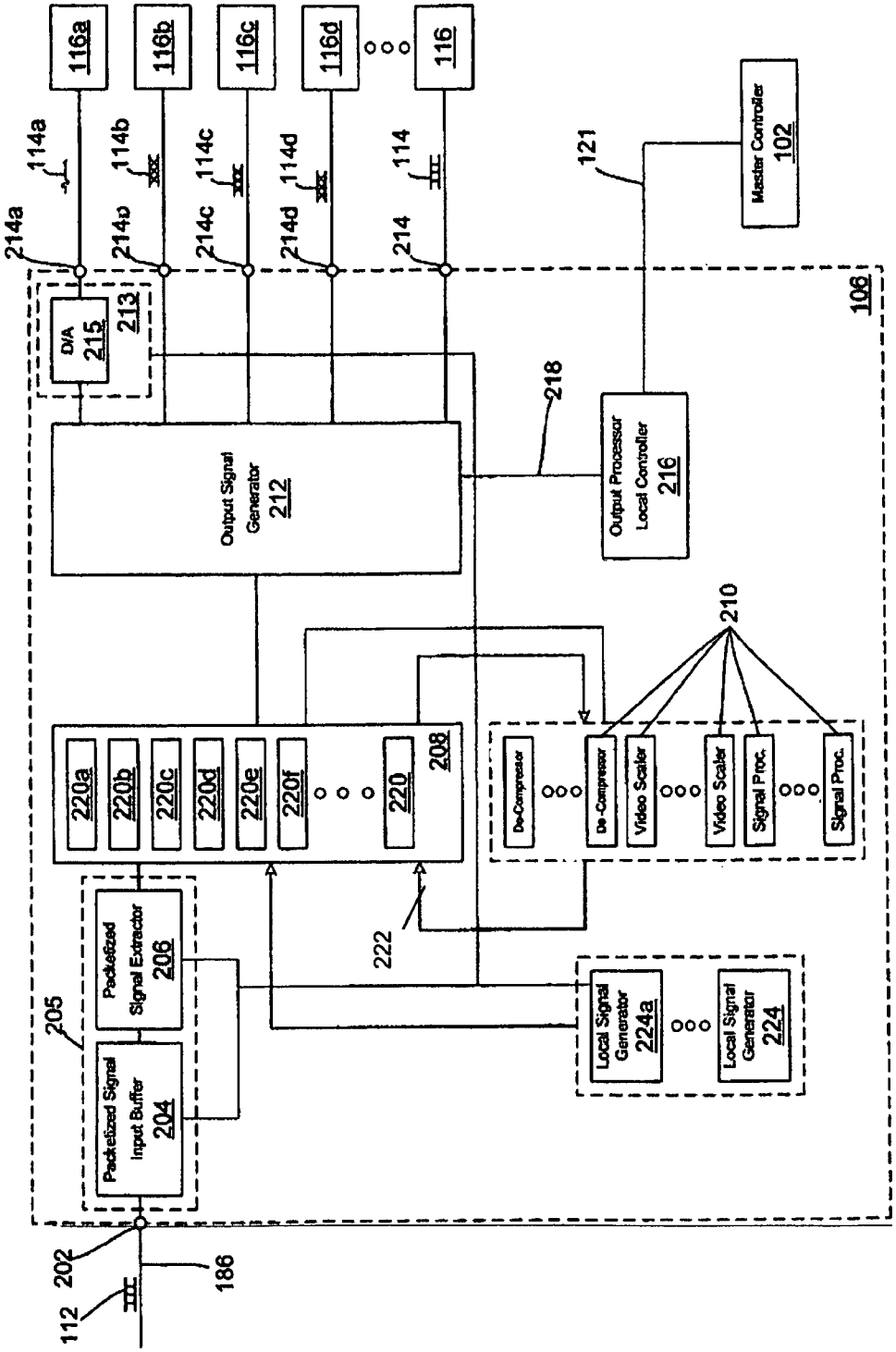

116a
116b
116c
116d
116
114a
114b
114c
114d
114
214a
214b
214c
214d
214
121
Master Controller 102
106
D/A 215
213
218
Output Signal Generator 212
Output Processor Local Controller 216
210
220a
220b
220c
220d
220e
220f
220
208
De-Compressor
De-Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
222
Packetized Signal Extractor 206
205
Packetized Signal Input Buffer 204
Local Signal Generator 224a
Local Signal Generator 224
202
112
186

Figure 6

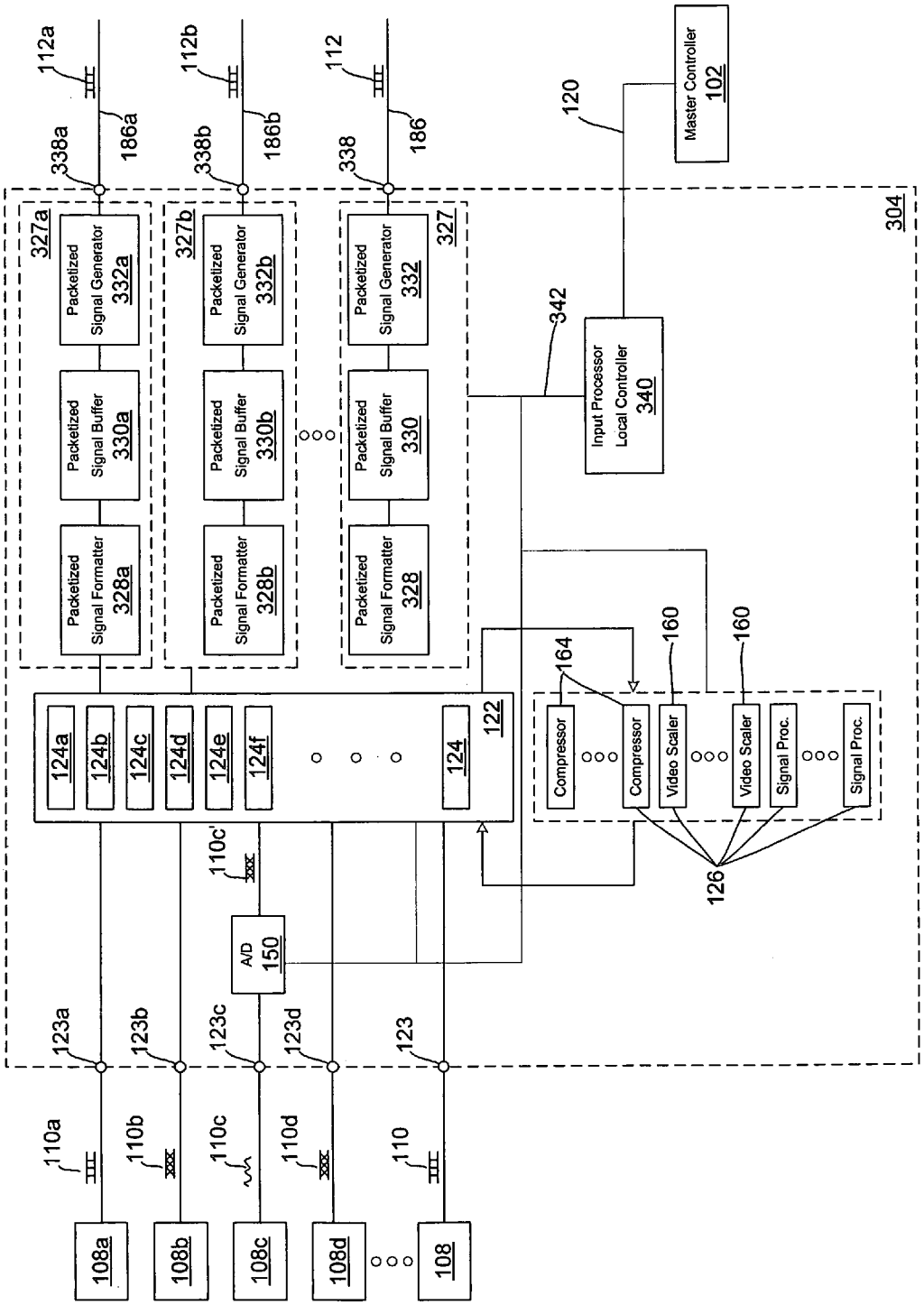

112a
112b
112
338a
338b
338
186a
186b
186
120
Master Controller 102
304
327a
327b
327
Packetized Signal Generator 332a
Packetized Signal Generator 332b
Packetized Signal Generator 332
Packetized Signal Buffer 330a
Packetized Signal Buffer 330b
Packetized Signal Buffer 330
Packetized Signal Formatter 328a
Packetized Signal Formatter 328b
Packetized Signal Formatter 328
342
Input Processor Local Controller 340
124a
124b
124c
124d
124e
124f
124
122
164
160
160
Compressor
Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
126
110c'
A/D 150
123a
123b
123c
123d
123
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 7

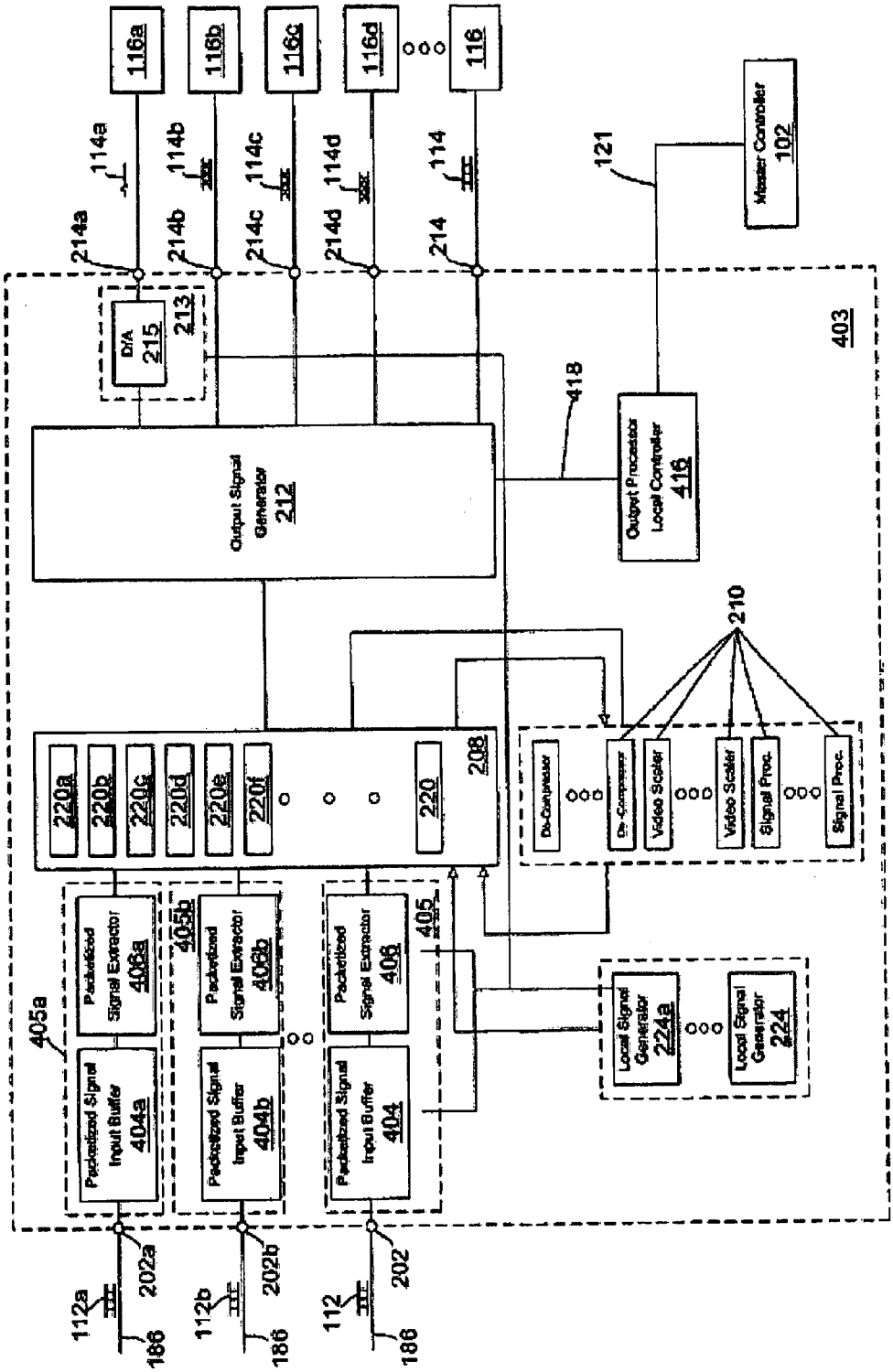

116a
116b
116c
116d
116
114a
114b
114c
114d
114
214a
214b
214c
214d
214
121
Master Controller 102
403
213
D/A 215
418
Output Signal Generator 212
Output Processor Local Controller 416
210
220a
220b
220c
220d
220e
220f
220
208
De-Compressor
De-Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
405a
405b
405
Packetized Signal Extractor 406a
Packetized Signal Extractor 406b
Packetized Signal Extractor 406
Packetized Signal Input Buffer 404a
Packetized Signal Input Buffer 404b
Packetized Signal Input Buffer 404
Local Signal Generator 224a
Local Signal Generator 224
112a
112b
112
202a
202b
202
186
186
186

Figure 6

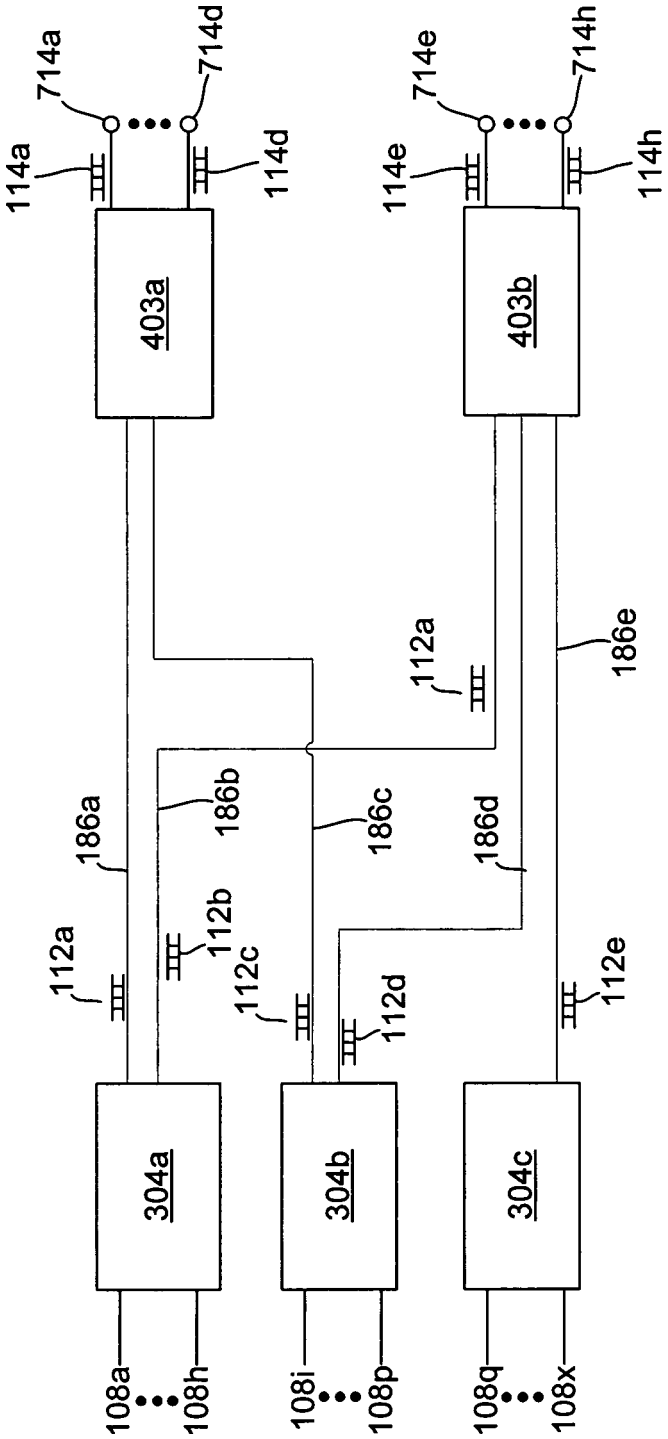

714a
714d
714e
714h
114a
114d
114e
114h
403a
403b
112a
186e
186a
186b
186c
186d
112a
112b
112c
112d
112e
304a
304b
304c
108a
108h
108i
108p
108q
108x

Figure 9

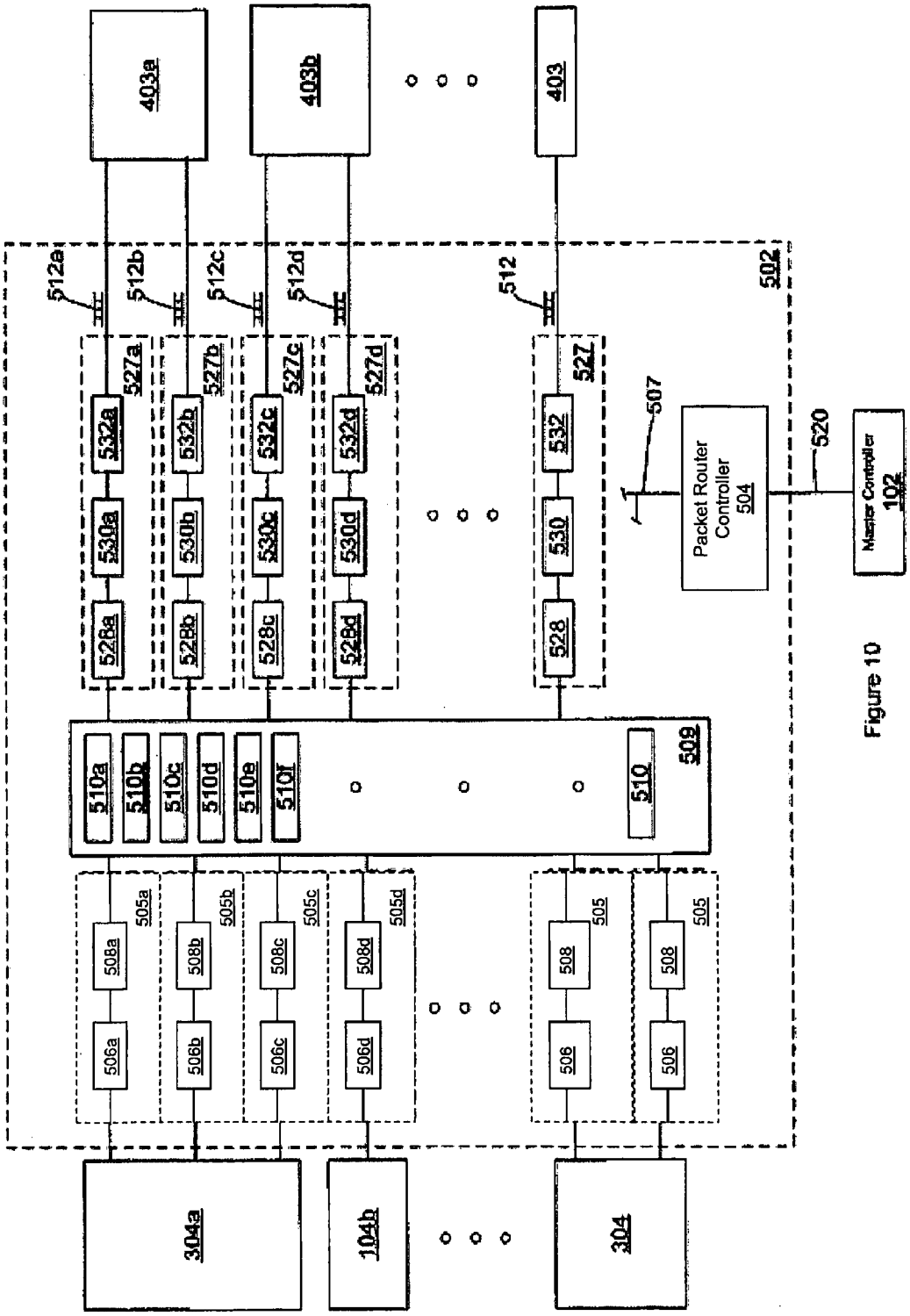

403a
403b
403
512a
512b
512c
512d
512
502
527a
527b
527c
527d
527
532a
532b
532c
532d
532
530a
530b
530c
530d
530
528a
528b
528c
528d
528
507
Packet Router Controller 504
520
Master Controller 102
510a
510b
510c
510d
510e
510f
510
509
505a
505b
505c
505d
505
508a
508b
508c
508d
508
506a
506b
506c
506d
506
304a
104b
304

Figure 10

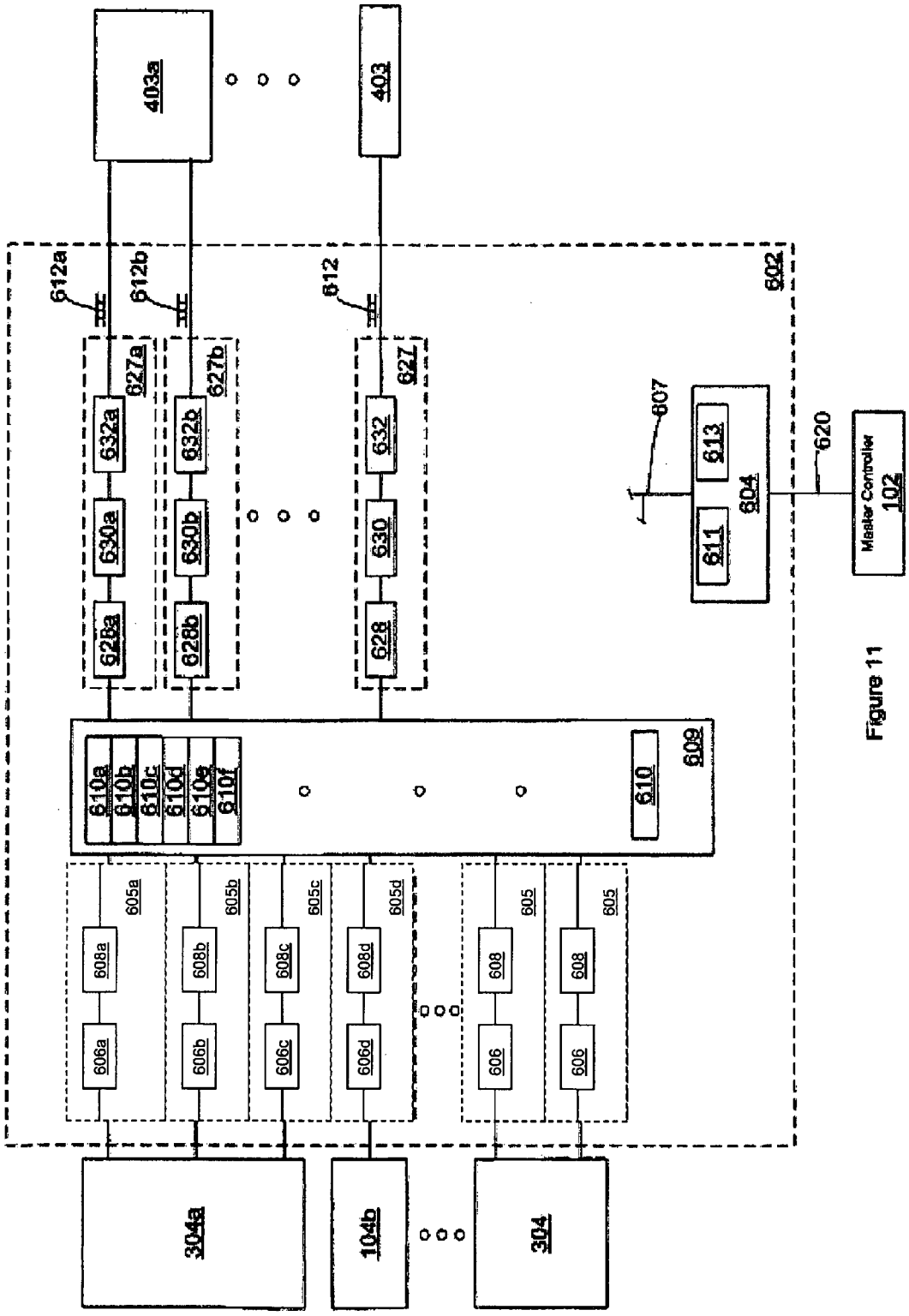

403a
403
612a
612b
612
602
627a
627b
627
632a
632b
632
630a
630b
630
628a
628b
628
607
613
611
604
620
Master Controller
102
610a
610b
610c
610d
610e
610f
610
609
605a
605b
605c
605d
605
608a
608b
608c
608d
608
606a
606b
606c
606d
606
304a
104b
304

Figure 11

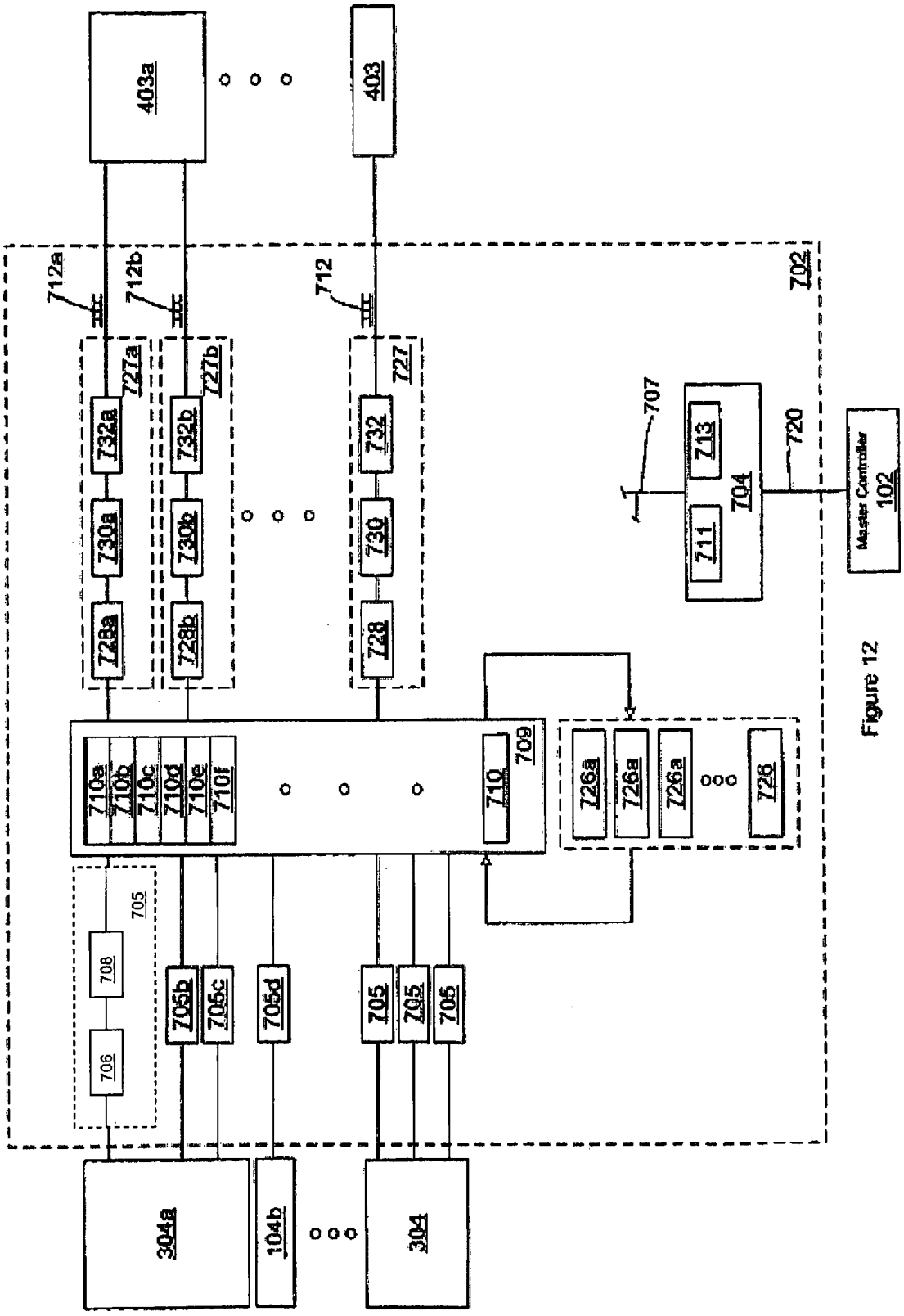

403a
403
712a
712b
712
702
727a
727b
727
732a
732b
732
730a
730b
730
728a
728b
728
707
713
711
704
720
Master Controller
102
710a
710b
710c
710d
710e
710f
710
709
726a
726a
726a
726
705
708
706
705b
705c
705d
705
705
705
304a
104b
304

Figure 12

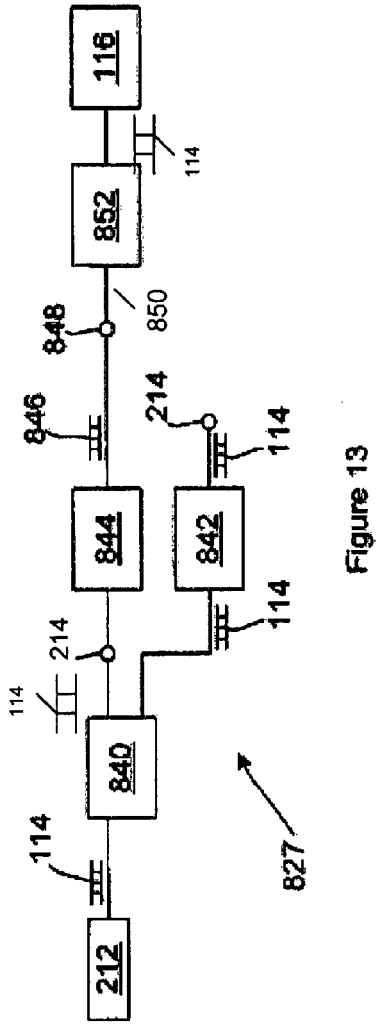


Figure 13

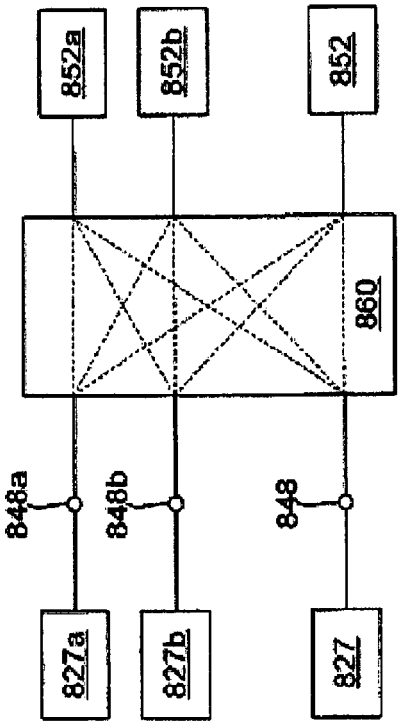


Figure 14

PACKET BASED TRANSMISSION OF MULTIPLE DATA SIGNALS

FIELD OF THE INVENTION

This invention relates to systems for transmitting and distributing audio data, video data and other types of data.

BACKGROUND OF THE INVENTION

Recent advances in video monitor technology have resulted in the development of large format, high quality video displays that are capable of displaying multiple video signals simultaneously. In television studios and other locations where many different video sources must be monitored, these video displays have begun to displace traditional individual monitors that displayed a single video source to which they were physically coupled.

Modern signal processing equipment allows video and other data to be routed to different display monitors, however, this equipment can still require that for a particular signal to be used in multiple locations on multiple display devices it must be replicated and coupled to equipment in the different locations. This results in excessive cabling requirements, multiple signal regeneration and replication stages, and can result in degraded signals and multiple failure points within the signal path.

There is a need for an improved efficient system for receiving various input signals, including video, audio and data signals, formatting the received signals and routing the formatted signals to various output devices.

SUMMARY OF THE INVENTION

In one embodiment, the present invention provides a system that includes a master controller, one or more input processors, one or more output processors, and one or more user controllers. The system may also include additional master controllers that serve as back-up master controllers.

One or more input devices are coupled, directly or indirectly, to each of the input processors. Each input device provides one or more input signals to the input processors. One or more output devices are coupled to the output processors. Each output device receives an output signal from an output processor. Each of the input processors generates one or more packetized signals. Each packetized signal is transported across a communications link to one or more of the output processors. Each output processor may receive one or more packetized signals.

The master controller receives user control signals from one or more user controllers indicating which input signals are to be routed to which output devices. The user control signals may also indicate the format in which the input signal is to be presented at the output device.

For example, for a video input signal the user control signals may specify the position and dimensions of a window on an output display device in which the input video signal is to be displayed. The user control signals may also define other characteristics for the displayed video signal, such as color balance, white balance, color to black & white conversion or the addition of a border or any other characteristics that a video signal may be processed to have.

For an audio input, the user control signals may specify a particular equalization (ie. jazz, rock, classical), volume balancing to some specified maximum or average volume, left—right signal balance, encoding in or conversion to a particular noise reduction and/or a multi-channel audio standard. For a data signal, the user controls may specify characteristics that are appropriate to the data.

In response to the user control signals, the master controller generates input processor control signals to operate the input processors and output processor control signals to operate the output processors. The input and output processor control signals may be transmitted to the various input and output processors using any type of communications link. The master controller coordinates the operations of the various input and output processors (and other elements of the system) to provide the output signals requested by a user or users who operate the user controllers to generate the user control signals.

If the system includes more than one master controller, one of the master controllers may be designated as a primary master controller and the remaining master controllers designated as backup master controllers. Each of the master controllers is coupled to the user controllers to receive the user control signals and is capable of generating the input processor control signals and the output processor control signals. The primary master controller actively generates the input processor control signals and output processor control signals. If the master controller fails or is disabled for any reason, one of the backup master controllers may be designated as the primary master controller.

Each input processor has an input processor local controller which receives the input processor control signals for that input processor. In response to the input processor control signals, the input processor local controller configures the various components of the input processor to receive and process the input signals coupled to that input processor and to generate one or more packetized signals, as requested by the master controller.

Each input processor includes a plurality of data buffers to store each input signal in a digital form. If an input signal is received in an analog form, an analog-to-digital converter is dynamically coupled between an input port at which the input signal is received and a data buffer to digitize the signal. Some input signals may be received in a processed manner, meaning that the signal has been processed in some manner. If an input signal is received in a processed manner then an unprocessor may be dynamically coupled between an input port at which the input signal is received and a data buffer in which the input signal is stored. Additionally each input processor may include one or more data analyzers that can be dynamically coupled to each input signal to provide information about the input signal. The resulting information is also stored in a data buffer. The buffered signals are then read out and processed by signal processors to format them as indicated by the user control signals. The processed signals are also buffered in data buffers. Each input signal may be processed multiple times to create different versions of the input signal, to extract various information regarding the input signal for use on different output devices, or for use on the same output device in different versions, formats or sizes.

The data buffers in the input processor, the output processor and other components and embodiments of the invention are used to temporarily store data that is received from a source and is subsequently read out by one or more recipient or destination elements or devices. Data may be read out in the order in which it is received, in which case a first-in/first-out buffer may be used to store the data. Alternatively, the data may be read out in any order by the recipient elements. In each case, the data buffer is configured to retain any datum until it has been read by all recipient elements that use the datum.

Each of the buffered signals (including the input signals and the processed signals) is assigned a global identification code. One or more of the buffered signals are converted into a packetized signal by a packetized signal output stage in each input processor. Each packetized signal contains a series of packets. Each packet contains a part of the data from the buffered signal along with the global identification code of the buffered signal. An input processor may have more than one packetized signal output stages to produce more than one packetized signal.

A packetized signal may be converted into and transmitted as a bitstream, or it may be transmitted using any communications protocol.

Each output processor receives one or more packetized signals. Each packetized signal is buffered as it is received. As complete packets corresponding to each global identification code are received, they are assigned a local identification code and are buffered in a separate data buffer. The isolated packets in data buffer correspond to a particular version of an input signal received at one of the input processors. The isolated stream may be processed to reverse any signal processing step or steps applied in the input processor or in an input device or combination of devices that combine to produce an input signal coupled to the input processor, such as a data compression step, or to apply additional signal processing steps. Any such processed signal is buffered again and assigned a new local identification code. One or more these buffered signals is then combined to form each output signal.

In one embodiment of the invention, a packet router is coupled between a plurality of input processors and a plurality of output processors. The packet router receives packetized signals from the input processors and isolates the packets corresponding to each global identification code. The packetized router then assembles new packetized signals corresponding to a combination of the global identification codes. The packetized router operates under the control of the master controller to route packets with the appropriate global identification code to the appropriate output processor. The packet router allows an input signal received at any of the input processors to be formatted and routed to any of the output processors.

In another embodiment of a packet router according to the invention, packets from one or more incoming packetized signals are stored in packet storage locations within a memory system. The packets are then read by one or more packetized signal output stages, each of which produces an outgoing packetized signal corresponding to a selected set of global identification codes. Storage of incoming packets and distribution of the packets to the packetized signal output stages is controlled by a router controller. In another embodiment of a packet router, one or more signal processors, such as video scalers or delay elements, are provided to process the incoming packets to provide processed packets that form a processed signal. Each processed signal is assigned a unique global identification code and may be included in an outgoing packetized signal.

These and other aspects of the invention and its various embodiments are described in greater details below,

BRIEF DESCRIPTION OF THE DRAWINGS

Several exemplary embodiments of the present invention will now be described in detail with reference to the drawings, in which:

FIG. **1** illustrates a system according to the present invention;

FIG. **2** illustrates a first input processor according to the present invention;

FIG. **3** illustrates the display of a first video monitor;

FIG. **4** illustrates the display of a second video monitor;

FIG. **5** illustrates one form of a packetized signal according to the present invention;

FIG. **6** illustrates a first output processor according to the present invention;

FIG. **7** illustrates a second input processor according to the present invention;

FIG. **8** illustrates a second output processor according to the present invention;

FIG. **9** illustrates an exemplary coupling between a plurality of input processors and a plurality of output processors;

FIG. **10** illustrates a first packet router according to the present invention;

FIG. **11** illustrates a second packet router according to the present invention;

FIG. **12** illustrates a third packet router according to the present invention;

FIG. **13** illustrates an alternative output stage for an output processor according to the present invention; and

FIG. **14** illustrates a switch for use with the alternate output stage of FIG. **13**.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Reference is first made to FIG. **1**, which illustrates a signal processing system **100**. The signal processing system includes a master controller **102**, an input processor **104** and an output processor **106** according to the present invention. The input processor **104** receives a plurality of input signals **110** from various signal sources **108** and provides a packetized signal **112**. The packetized signal **112** corresponds to some or all of the input signals **110** or to data or processed signals derived from the input signals. The packetized signal **112** is transported by a communications link **186** to the output processor **106**. The output processor **106** receives the packetized signal **112** and produces one or more output signals **114**, which are processed by output devices **116**. The output signals **114** correspond, at least in part, to one or more of the input signals **110**.

The system may additionally include backup master controllers (not shown).

The input processor **104** and output processor **106** operate under the control of the master controller **102**. The master controller **102** is coupled to one or more user controllers **118**, from which the master controller receives user control signals **119**. The master controller **102** and the user controllers **118** may be combined in a single unit, or may be assembled in a single assembly, or they may be separate units that are coupled together.

A user or multiple users (not shown) human or some other type of device (for example automated monitoring and control systems) operate the user controllers **118** to generate the user control signals **119**. The user control signals **119** indicate which input signals or signals derived from the input signals **110** the user would like included in the output signal **114** provided to each output device. Each user may have control over one or more output devices **116** in whole or in part. The user control signals **119** may also indicate additional characteristics about the output signal **114** provided to each output device **116**. The master controller **102** translates the users control signals **119** into input processor control signals **120** and output processor control signals **121** to control the operation of the input processor **104** and the output processor **106**

respectively so that the output signals **114** are provided in accordance with the user control signals **119**.

The input signals **110** may be base-band, compressed, time division multiplexed audio signals, video signals (which may also include audio information), metadata, or other data signals. Similarly, the output signals **114** may be audio signals, video signals, or data signals. Typically, each output signal **114** will correspond to one or more of the input signals and or information derived from the input signal. A particular output signal may include a combination of audio, video or data input signals or signals produced by input signal analyzers or any combination of these types of signals. The nature of each output signal **114** is appropriate for the output device **116** that receives the output signal **114**. Some of the output devices **116** may be video monitors, such as analog video monitor **116***a* and digital video monitor **116***b*, for displaying output video signals. Some of the output devices **116** may be sound systems, such as sound amplification and broadcast system **116***c*, for further processing or playing output audio signals. Some of the output devices may be data processing systems, such as computer system **116***d*, for further processing or displaying the output data signals. In any particular embodiment of the present invention, the output signals **114** may be of the same or different types, depending on the usage of the embodiment. In an alternative embodiment of the invention, the output processor may provide only a single output signal. The type of any particular signal may change depending on the usage of the signal, under the control of the master controller **102**.

Reference is next made to FIG. **2**, which illustrates the input processor **104** in greater detail. Input processor **104** includes a plurality of input ports **123**, a plurality of input signal analyzers or processors, such as A/D converter **150**, a plurality of data buffers **124**, which are part of a memory system **122**, one or more signal processors **126**, a packetized signal output stage **127**, a packetized signal output port **138** and an input processor local controller **140**. The packetized signal output stage **127** includes a packetized signal formatter **128**, a packetized signal buffer **130** and a packetized signal generator **132**.

Memory system **122** may be a local memory device or memory space within the input processor **104** or it may be located on an attached storage device or other medium. Data buffers **124** will typically comprise memory space allocated within memory system **122**.

The input processor local controller **140** receives the input processor control signals **120**. The input processor local controller **140** controls the operation of the various elements of the input processor **104** through control lines **142** in response to the input processor control signals **120**.

Each input signal **110** is coupled to one of the input ports **123**. Each of the input ports **123** is coupled to the memory system **122**. Each input signal **110** is buffered in a data buffer **124** in memory system **122**. Analog input signals **110** are converted to a digital version and the digital version is buffered in a data buffer **124**. For example, if input signal **110***c* is an analog input signal, then an analog-to-digital (A/D) converter **150** is dynamically coupled between input port **110***c* and memory system **122** to convert input signal **110***c* into a corresponding signal **110***c*' comprising a stream of packets according to a digital signal standard. For example, if input signal **110***c* is a standard 1V peak-to-peak audio signal, it may be sampled and converted by A/D converter **150** into a WAV digital audio signal **110***c*', as is commonly done to record music on CD-ROM media. For example, if input signal **110***c* is a compressed video stream a de-compressor (one of the signal processors) may be dynamically coupled between input port **110***c* and memory system **122** to convert input signal **110***c* into a corresponding signal **110***c*' comprising a stream of data according to a digital signal standard.

In the present embodiment, the input processor contains a bank of A/D converters, each of which may dynamically be coupled between any of the input ports **123** (or a group of the input ports) and memory system **122**. The input processor local controller **140** controls the coupling of any particular A/D converter between any particular input port **123** and memory space **122**. A particular A/D converter **150** may be shared by two or more input ports **123** under the control of input processor local controller **140**. Alternatively, a dedicated A/D converter **150** may be provided for some or all of the input ports **123**. The A/D converter **150** may be activated by the input process local controller **140** if the input signal **110** received at a port **123** is an analog signal. In another alternative embodiment, some or all of the input ports **123** may be designated as analog input ports and an A/D converter may be permanently coupled between those ports and the memory system **122**.

The input ports may be any type of communication port, such as an Ethernet, BNC, optical, telephone line or any port suitable with any type of communication system. The input signals may be in any communication standard or protocol, including, including TCP/IP. In this case, the coupling between the input device and the processor may be a LAN, WAN, the Internet or another TCP/IP communication system.

The input processor **104** may also contain a bank of input signal analyzers (not shown). The input signal analyzers may be dynamically or statically coupled to an input port in the same manner as A/D converter **150**. For example if the input signal is a video signal, an input signal analyzer may extract performance and signal content metrics from or about the input signal such as blackness of the signal, the amount of motion within the signal, bit or formatting errors in the signal. The metadata produced by the data analyzer is stored in a data buffer and is considered and treated as a processed signal that can be packetized and coupled to the output processor over a communication link.

Each of the input signals **110** may be retrieved from the corresponding data buffer **124** as a buffered signal **156**.

The input processor may also include other input signal processing elements that may be coupled between an input port **123** and memory system **122**. The signal processing elements may include video scalers, video de-interlacers, data compressors, data de-compressors, data format converters or any other type of signal processor, including the signal processing elements described below. For example, if one of the input signals is an analog NTSC video signal, then a video signal digitizer may be dynamically coupled between an input port at which the signal is received to convert the input signal into a MPEG2 digital video signal. The input processor may contain a bank of input signal processing elements and analyzers, which may be dynamically coupled between any input port and memory system **122**. Input processing elements or analyzers may also (or alternatively) be coupled to only one port for selective use with that port. Input processing elements may also (or alternatively) be permanently coupled to one or more of the input ports.

The signal processors **126** are coupled to memory system **122** to retrieve the buffered signals **156** from memory system **122**, process the buffered data signals to generated processed signals **158**, which are then buffered in data buffers **124** in memory system **122**. A processed signal **158** is stored in a different data buffer **124** than the input signal **110** from which the processed signal is derived. The signal processors **126** are illustrated in a dotted box and the coupling between the

memory system **122** and the signal processor extends to the dotted box to indicate that any of the signal processors may be dynamically coupled to any of the data buffers **124** to retrieve a buffered signal and to store a processed signal.

A particular input signal **110** may be processed to generate more than one processed signal **158** and each of the resulting processed signals **158** are stored in different data buffers **124** in memory system **122**. As a result, the original input signal **110** and any versions of the original input signal **110** that are generated as processed signals **158** are available from memory system **122** as buffered signals **156**.

In the present embodiment, the signal processors **126** include video scalers **160**, embedded audio extractors, ancillary data extractors, signal content analysers and data compressors **164**. The signal processors **126** may also include data de-compressors, image rotation devices, special effects processors, image invertors, spatial filters, edge enhancement processors, color space converters, audio sweetening processors, digital audio decompressors, and digital audio processors. A signal processor may be used to process two or more input signals (or processed signals) by time-division-multiplexing the signal processor between the data buffers used to buffer the two or more input signal (or processed signals).

Each video input signal **110** will have height and width dimensions, usually defined in pixels. For example, a video image may comprise a series of frames that are 640 pixels wide by 400 pixels high. A video scaler **160** is capable of rescaling a video signal from its original dimensions to different dimensions. In the present embodiment, the input processor **104** includes a plurality or bank of video scalers **160**. Each video scaler **160** receives control instructions from the input processor local controller **140** to extract a particular video input signal **110** from the appropriate data buffer **124** and rescale the video input signal to specified dimensions and to store the resulting processed signal **158** in another data buffer **124**. A video scaler **160** may be configured to retain or change the aspect ratio of an input data signal or to crop the input data signal in the processed signal and to provide any other function that a conventional scaler is capable of providing. For example, a video scaler may be configured to crop the input data signal to select a portion of it, and then scale the cropped video image to specified dimensions.

A particular video scaler **160** may be instructed to scale a video input signal **110** to more than one set of new dimensions and may generate two or more processed signals **158**, each of which is separately buffered in separate data buffers **124**. In addition, a particular video scaler **160** may be shared (or multiplexed) between two or more video input signals **110** to generate two or more corresponding processed signals **158**, each of which is separately buffered in separate data buffers **124**. One video input signal **110** may also be retrieved by two or more video scalers **160** (or other signal processors **126**) to produce two or more corresponding processed signals **158**, which are similarly buffered in separate data buffers **124**.

Data compressors **164** are used to generate a processed signal **158** that is a compressed version of any signal stored in a data buffer **124**. For example a video input signal **110** in a DVI format may be compressed into an MPEG-2 format to reduce the amount of data required to transmit the signal. The resulting MPEG-2 format video signal is stored in a data buffer **124** as a processed signal **158**. The data compressors **164** may include a plurality of compression elements, which may be hardware or software elements, designed to compress audio, video or data signals into various types of compressed signals. The data compressors may provide a lossy or lossless compression. In each case, the compressed data signal produced by a data compressor **164** is stored as a processed signal **158**. A particular embodiment of an input processor **104** may include any number and type of data compressors **164**.

The data compressors **164** may include horizontal or vertical line filters that produce a processed video data signal comprising a portion of the video data from a video input data signal. For example, a horizontal line filter may be configured to horizontally compress a 640×400 pixel video signal into a 320×400 pixel video signal by discarding every other pixel in each line of the video signal. A vertical line filter may be configured to compress a 640×400 pixel video signal into a 640×200 pixel video signal by discarding every other line in the video signal. A horizontal/vertical line filter may be configured to compress a 640×400 pixel video signal into a 160×100 pixel video signal by discarding three of every four lines of the video signal and discarding three of every four pixels in each line that is retained.

The input signals **110** will typically be asynchronous with respect to one another (unless they are from a synchronized source). The size and timing of packets in each input signal will depend on the signal standard used to encode the signal. For example, input signal **110***a* may be formed of uniformly sized packets that are spaced equally apart in time. Input signal **110***b* may be formed of differently sized packets that do not arrive at equally spaced intervals. As a result of the differences between the input signals **110**, data buffers **124** may be allocated with a large or smaller memory space to properly buffer each input signal **110**.

In addition, the size and timing of packets in processed signals **158** may also vary depending on the data standard used to encode the processed signals **158**. The data buffers **124** used to buffer a processed signal **158** may similarly be dynamically allocated a memory space of a suitable size.

The master controller **102** (FIG. **1**) controls the operation of the input processor **104**. The master controller **102** assigns a unique global identification code to each signal that is buffered in a data buffer **124**. This global identification code is used to identify the signal in both the input processor and the output processor. Each version of a particular input signal **110** that is buffered in a data buffer **124** is assigned a different global identification code. For example, video input signal **110***a* is first buffered in the form in which it is received. The input signal **110***a* may be scaled to new dimensions using a video scaler **160** to produce a scaled signal **110***aa*, which is separately buffered. The input signal **110***a* may also be scaled to a second set of dimensions to produce a second scaled signal **110***ab*, which is also separately buffered. The second scaled signal **110***ab* may then be compressed to produce a scaled and compressed signal **110***ac*, which is also separately buffered. Each of the different versions **110***a*, **110***aa*, **110***ab*, **110***ac* may be separately retrieved from its data buffer and may be identified using its unique global identification code.

To further explain the invention and the present embodiment, an example of the use of this embodiment will be described. In the example, the input processor **104** receives three digital video input signals **110***a*, **110***b* and **110***d* and one analog video input signal **110***e*. Analog video input signal **110***e* is digitized using a A/D converter **150** to produce a digital signal **110***e*' corresponding to analog signal **110***e*. Signals **110***a*, **110***b*, **110***d* and **110***e*' are buffered in separate data buffers **124**.

Reference is made to FIG. **3**, which illustrates output device **116***a*, which is an analog standard definition 4:3 format video monitor capable of displaying images with a resolution of 640×480 pixels. The display of video monitor **116***a* is used to display information in five different parts or windows: video windows **170**, **172**, **174** and **176** and graphics window **178**.

A user configures the video and other information shown on each video monitor using user controller **118**. User controller **118** may provide a graphical or other interface allowing the user to define windows and other elements on a video monitor and assign specific input signals or other information to be displayed in each window or other element. The user has defined the parts of the display on video monitor **116***a* as follows:

Window/Element	Position (relative to top left corner)	Dimensions	Contents
Video window 170	10, 10	400 × 300	A version of video input signal 110a
Video window 172	420, 10	200 × 113	A version of video input signal 110b
Video window 174	460, 340	160 × 120	A version of video input signal 110d
Video window 176	10, 320	440 × 140	Rejected packets data for video signal 110b
Graphics window 178	420, 150	200 × 150	Date/Time/ Metadata Information

Reference is made to FIG. **4**, which illustrates output device **116***b*, which is a digital high definition 16:9 format video monitor with a resolution of 1920×1080 pixels. The display of video monitor **116***b* is used to display information in three different parts or windows: video windows **180**, **182** and **184**. The user has defined the parts of the display on display monitor **116***b* as follows:

Window/Element	Position (relative to top left corner)	Dimensions	Contents
Video window 180	60, 60	1140 × 640	A version of video input signal 110b
Video window 182	1280, 60	560 × 420	A version of video input signal 110e
Video window 184	1280, 540	610 × 460	A version of video input signal 110a

In an alternate example, window positions maybe such that some or all of the windows are overlapping, or arranged in a cascaded manner.

The video windows have been described as containing "a version of" one of the video input signals **110**. The user will typically specify the position and dimension of a window on a video monitor and the input signal **110** that the user would like displayed in each window. An appropriate version of the input signal is prepared by the input processor **104** and provided to the output processor **106** for display on the video monitor. Alternatively, the user may specify certain signal processing steps to be performed on an input signal before it is displayed in a window. For example, if the signal processors **126** (FIG. **2**) include a color/black & white converter, then a user may specify that a color input signal be converted into a black & white signal and that the black & white version of the input signal (or a version of the black & white signal) be displayed in a particular window.

Reference is again made to FIG. **1**. The user controller **118** transmits the user's instructions for each output device **116** to the master controller as user control signals **119**. The user's instructions relating each output device **116** will typically depend on the nature of the output device **116**. For example, if an output device **116** is an audio processing system capable of receiving and switching between multiple audio signals, then the user may specify that one or more audio input signals **110**, or the audio components of video input signals **110**, be directed to the sound output device **116**. If an output device **116** is only capable of receiving a single audio signal and then amplifying and broadcasting the audio signal, the user may specify that a particular input audio signal or the audio component of a particular video input signal **110** be directed to the sound output device **116**. Similarly, a user may specify that any particular output device **116** can receive any combination of information that the output device is capable of receiving.

Referring again to FIG. **3**, the windows **176** and **178** contain information that is not present in any input signal **110**. The user controller is configured to allow the user to select any information that may be generated within system **100** and which is suitable for a particular display device. The rejected packets information displayed in video window **176** may be determined by a signal analyzer (not shown) that analyzes input signal **110***b* to determine the number of defective packets received as part of the input signal **110***b*. The signal analyzer is one of the signal processors **126**. The signal analyzer then generates a video signal illustrating this information in a standard video signal format and stores the video signal in a data buffer **124** as a processed signal **158**.

In response to the user control signals **119**, the master controller transmits input processor control signals **120** to the input processor local controller **140** indicating the final version of each input signal **110** that will be required by the output processor **106** to produce the output signals **114** for the output devices **116**. For each required version, the master controller **102** also indicates the top left pixel at which that version will be displayed.

For the example input signals **110** and output video monitors **116** described above, the master controller instructs the input processor to prepare the following signals:

i. 400×300 pixel scaled version of video input signal **110***a;*
ii. 610×460 pixel scaled version of video input signal **110***a;*
iii. 200×113 pixel scaled version of video input signal **110***b;*
iv. 1140×640 pixel scaled version of video input signal **110***b;*
v. 160×120 pixel scaled version of video input signal **110***d;*
vi. 560×420 pixel scaled version of video input signal **110***e*; and
vii. 440×140 pixel video image illustrating rejected packet information for video signal **110***b.*

The master controller **102** does not instruct the input processor to produce a signal showing the date, time and analyzed information, which is required for graphics window **178** on video monitor **116***a*. This signal is produced in the output processor and is described below.

In response to the input processor control signals **120**, the input processor local controller **140** determines how the required versions of each input signal **110** can be produced and configures and couples the input ports **123**, A/D converters **150**, data buffers **124** and signal processors **126** to produce the required versions of each input signal. As described above, every signal stored in a data buffer **124** is assigned a unique global identification code.

In the present example, the input processor local controller **140** configures the input processor **104** as follows:

i. Store input signal **110***a* in data buffer **124***a*. Assign global identification code G**101** to the stored signal.
ii. Store input signal **110***b* in data buffer **124***b*. Assign global identification code G**102** to the stored signal.

iii. Store input signal **110***d* in data buffer **124***c*. Assign global identification code G**103** to the stored signal.

iv. Couple an A/D converter **150** between input port **123***e* at which input signal **110***e* is received to produce a digital version **110***e*' of input signal **110***e*. Store digital signal **110***e*' in data buffer **124***e*. Assign global identification code G**104** to the stored signal.

v. Couple video scaler **160***a* to memory system **122** to retrieve signal G**101** and produce a scaled version of 400×300 pixel scaled version of signal G**101**. The scaled version is stored in data buffer **124***f* and is assigned global identification code G**105**.

vi. Couple video scaler **160***b* to memory system **122** to retrieve signal G**101** and produce a 610×460 pixel scaled version of signal G**101**. The scaled version is stored in data buffer **124***g* and is assigned global identification code G**106**.

vii. Couple video scaler **160***c* to memory system **122** to retrieve signal G**102** and produce a 200×113 pixel scaled version of signal G**102**. The scaled version is stored in a memory buffer **124***h* and is assigned global identification code G**107**.

viii. Couple video scaler **160***d* to memory system **122** to retrieve signal G**102** and produce an 1140×640 pixel scaled version of signal G**102**. The scaled version is stored in data buffer **124***i* and is assigned global identification code G**108**.

ix. Couple video scaler **160***e* to memory system **122** to retrieve signal G**103** and produce a 160×120 pixel scaled version of signal G**103**. The scaled version is stored in data buffer **124***j* and is assigned global identification code G**109**.

x. Couple video scaler **160***f* to memory system **122** to retrieve signal G**104** and produce a 560×420 pixel scaled version of signal G**104**. The scaled version is stored in data buffer **124***k* and is assigned global identification code G**110**.

xi. Couple a signal analyzer (one of the signal processors **126**, as described above) to the memory system **122** to retrieve and analyze signal G**102**. The signal analyzer produces a video signal with a standard size of 320×200 pixels and metadata. The output of the signal analyzer is stored in data buffer **124***m* and is assigned global identification code G**111**.

xii. Couple a video scaler **160***g* to memory system **122** to retrieve signal G**111** and produce a 440×140 pixel scaled version of signal G**111**. The scaled version is stored in data buffer **124***n* and is assigned global identification code G**112**.

During the operation of input processor **104**, successive packets of each signal stored in a data buffer **124** are stored in the data buffer and previously stored packets are read out and then discarded. Some signals, such as input signal **110***a* are read by more than one device. Input signal **110***a*, identified by its global identification code G**101**, is read out by video scalers **160***a* and **160***b*. The data buffer **124***a* in which input signal **110***a* is buffered is configured to discard each packet in the input signal only after the packet has been read by both of the video scalers.

Signals G**105**-G**110** and G**112** are required to produce the output signals **114** for video monitors **116**. These signals are combined into packetized signal **112** using packetized signal formatter **128**, packetized signal buffer **130** and packetized signal generator **132**. The signals that are used to produce a packetized signal **112** are referred to herein as packet source signals for that packetized signal.

Reference is next made to FIG. **5**, which illustrates the format of the packetized signal **112**. In the present embodiment, the packetized signal **112** comprises a series of packets **190**, each of which contains the following fields:

i. global identification code for the signal from which the data in the packet was obtained;

ii. packet ordering information;

iii. a data payload;

iv. optional error detection and correction information and other metadata.

The packet ordering information, which may comprise a sequential packet number for each packet with the same global identification code, allows packets derived from the same packet source signal to be isolated from other packets, allowing the data in the packet source signal or a version of the data in the packet source signal to be re-assembled in the output processor **106**, as is described below.

In addition to packet ordering information, a video data packet may optionally also include frame ordering information, identifying the particular frame of video signal to which the packet corresponds.

The content and format of the data payload in each packet **190** will depend on the type of the data contained in the packet.

For example, if a packet source signal comprises a stream of data that is not organized as packets of information, then each packet **190** formed from that packet source signal contains a fixed amount of data in the data payload field. For example, if a packet source signal is a continuous stream of video data, then each corresponding packet **190** contains up to 320 bytes of the video data. In other embodiments, the amount of data in a particular packet may be fixed at a different size or may be variable.

If a packet source signal is organized as a series of packets of information, as in case of MPEG-2 encoded video or MP3 encoded audio or AES encoded audio, then the data payload may comprise the entire packet from the packet source signal.

Referring to FIG. **3**, video window **170** is a 400×300 pixel window in which signal G**105** will be displayed. Signal G**105** is created by video scaler **160***a* in a digital video standard that comprises a stream of video data that is not separated into packets. If each pixel in the 400×300 pixel window **170** requires one byte of video data from signal G**105**, then an entire frame of video information for the window requires 120,000 bytes of data. If the standard according to which the signal is encoded provides that one complete horizontal line of information will be encoded in a single packet, then each frame will be encoded in 300 packets in data buffer **124***f*.

Packet signal formatter **128** retrieves the successive packets in data buffer **124***f* that encode each frame of video signal G**105** and produces a series of packets **190** that correspond to the retrieved packets. In the following discussion, pixel numbers are set out as n,m where n is the number of the pixel in a window in a horizontal line of a window or frame and m is the number of the line in the window or frame. Pixels and lines are numbered starting at 1. The packets **190** corresponding to one frame of the 400×300 pixel window include pixel data for the following ranges of pixels:

Packet	Pixel range
1	1.1-320.1
2	321.1-240.2 (i.e. pixels 321-400 on line 1 and pixels 1-240 on line 2)

-continued

Packet	Pixel range
3	241.2-160.3
4	161.3-80.4
5	81.4-400.4
6	1.5-320.5
.	.
.	.
.	.
373	241,298-160,299
374	161,299-80,300
375	81,300-400,300

Similarly, the packetized signal formatter reads the successive packets in data buffer **124***h* that encode each frame of video signal G**107** and produces a series of packets **190**. The packets **190** corresponding to one frame of the 200×113 pixel window include pixel data for the following ranges of pixels:

Packet	Pixel range
1	1.1-120.2
2	121.2-40.4
3	41.4-160.5
.	.
.	.
.	.
70	81,111-300,112
71	1,113-200,113

The last packet **190** used to packetize each frame of video signal G**107** contains data for only 200 pixels. The remaining data space is filled with null characters by the packetized signal formatter **128**. Alternatively, the last packet may have a shortened data payload length.

The packetized signal formatter **128** produces packets **190** corresponding to the data in the packet source signals. Depending on the data formats used for the packet source signals, the packetized signal formatter **128** may produce one or more packets **190** that correspond to the data in one packet of a packet source signal. For example, if packet source signal G**105** is encoded using a digital video standard that includes a complete frame of video in a single packet, then the packetized signal formatter **128** will produce 375 packets **190** corresponding to each packet in the packet source signal.

A single packet **190** may correspond to data from more than one packet of a packet source signal. For example, if packet source signal G**107** is encoded using a digital video standard that encodes a single line of a frame in each packet, then the packetized signal formatter will generate packets **190** corresponding to more than one packet in the packet source signal, since each of the packets **190** can contain data for 320 pixels and since each line in packet source signal G**107** is only 200 pixels wide.

The packetized signal formatter **128** proceeds to generate packets **190** for each of the packet source signals for the packetized signal **112**, as packets from the packet source signals are available from the corresponding data buffers **124**. As packetized signal formatter **128** produces packets **190**, it stores them in packetized signal buffer **130**. Packetized signal buffer **130** is a data buffer and may include memory space in memory system **122**.

Packetized signal generator **132** retrieves the packets **190** stored in packetized signal buffer **130** and generates packetized signal **112** at packetized signal output port **138**. Packetized signal **112** may be a synchronous signal. For example, in the present embodiment, the packetized signal is a synchronous signal transmitted at 2.5 Gbits/second. Referring to FIG. **5**, if there are no packets **190** in the packetized signal buffer **130**, the packetized signal generator transmits null characters **192** between packets. In other embodiments, the packetized signal generator may transmit the packetized signal **112** at any bit rate, depending on requirements and capabilities of the system **100**.

Reference is made to FIG. **1**. In the present embodiment, the packetized signal output port **138** will typically be coupled to the output processor (FIG. **1**) through a communication link **186**, which may be a data cable such as an electrical or optical cable. The data rate and other aspects of the data protocol used to transmit the packetized signal **112** correspond to the ability of the communication link **186**.

In other alternative embodiments, the packetized signal generator **132** may transmit the buffered packets **190** as an asynchronous stream of packets to the output processor using any communication protocol, including TCP/IP. In this case, the communication link **186** may be a cable or may be a LAN, WAN, the Internet or another communication system.

Reference is next made to FIG. **6**, which illustrates the output processor **106**. The output processor **106** has a packetized signal input port **202**, a packetized signal input stage **205**, a memory system **208**, a plurality of signal processor **210**, an output signal generator **212**, a bank **213** of digital-to-analog (D/A) converters **215**, a plurality of output ports **214** and one or more local signal generators **224**. Each packetized signal input stage **205** comprises a packetized signal input buffer **204**, a packetized signal extractor **206**. The display devices **116** are coupled to the output ports **214**. The output processor **106** also includes an output processor local controller **216** that receives output processor control signals **121** from the master controller **102** (FIG. **1**). The output processor local controller **216** is coupled to the various components of the output controller **106** through control lines **218** and controls the operation of those components in response to the output processor control signals **121**.

Memory system **208** includes a plurality of data buffers **220**.

The output processor control signals **121** received by the output processor local controller **216** indicate:

i. which signals (by their global identification codes) are required for each output signal **114**; and

ii. the format of each output signal **114** and, if the output signal is a video signal, the layout of the display including the position and dimensions of each window on the display, in accordance with the user control signals (FIG. **1**).

In the present embodiment, the output processor local controller **216** translates each global identification code into a local identification code.

The packetized signal **112** is received at input port **202** and is buffered in packetized stream input buffer **204**. As complete packets **190** are stored in buffer **204**, they are retrieved by packetized signal extractor **206**. The packetized signal extractor **206** determines the global identification code of each packet, translates the global identification code into the corresponding local identification code assigned by output processor local controller and stores packets **190** corresponding to each local identification code in a different data buffer **220**. Through this process, the data from each source signal for the packetized signal is isolated in a different data buffer **220**. Each isolated signal corresponds to one of the packet source signals for the packetized signal. The packet ordering information from each packet **190** is used to organize the packets

190 into their original sequence. Each isolated signal is referred to herein as an output source signal.

The local identification codes are used within the output processor **106** in place of the global identification code to distinguish between the different local source signals encoded in the packetized signal. In alternative embodiments, the global identification code may be used to identify the different local source signals within the output processor **106**.

The signal processors **210** may be used to reverse any compression or other signal processing operation applied in the input processor **104** (FIG. **2**) using the signal processors **126**. Depending on the signal processing operations performed in the input processor **104**, a reversing step may or may not be required. For example, if one of the input signals **110** was compressed using a standard compression format that may be directly used to produce an output signal **114**, then it is not necessary to reverse the compression. However, if the result of the compression step produced data that cannot be directly used to produce an output signal **114**, then a decompressor may be used to reverse the compression step. For example, one of the signal processor described above was a horizontal line filter, which compresses an input video signal **110** by discarding a portion of the video signal. This compression step may be reversed by interpolating the discarded data from the retained data. The resulting processed signal **222** is stored in a data buffer as an output source signal and is assigned a unique local identification code by the output processor local controller **216**.

In addition to reversing signal processing operations applied in the input processor, a signal processor **210** may be used to apply any other signal processing operations to a signal buffered in a data buffer **220** to produce an output source signal.

Reference is made to FIG. **3**. Graphics display window **178** on video monitor **116***a* contains a display of the current date, time and warning messages based on metadata extracted from the input packetized signal received with global identification code G**111**. The date and time information is generated by a local signal generator **224***a*, which operates under the control of the output processor local controller. Each local signal generator **224** produces an output source signal containing information and formatted for the use in an output signal. In this example, the local signal generator **224***a* generates a 200×150 pixel window containing the date and time. The output processor **106** may include other local signal generators **224** that produce other video, audio or data signals for inclusion in the output signals. In each case, the signal produced by the local signal generator is stored in a separate data buffer **220** and is assigned a local identification code. The metadata maybe generated using a signal analyzer among the signal processors **210**. The metadata produced by the signal analyzer is combined with the data and time information by a video signal generator in the output signal generator **212**. Similar metadata could also be generated in the input processor **104** (FIG. **1**) using a signal processor **126**.

The output signal generator **212** can generate a variety of digital output signals that may be used directly, or after conversion through a D/A converter **215**, by output device **116**. The output signal generator **212** may include one or more digital video signal generators, one or more digital audio signal generators or one or more data signal generators or any combination of video, audio and data signal generators. The data signal generators may include TCP/IP signal generators that produce an output signal **114** suitable for transmission using a communications link to a remote computer system, where the output signal may be decoded and used by a video, audio or data system. Similarly, the data signal generator may generate signals in any data format.

The output signal generator **212** extracts the data required for each output signal **114** from the appropriate data buffers **220** and generates the output signal **114**. For example, a video output signal generator receives instructions from the output processor local controller **216** identifying the output source signals (by their local identification code and the data buffer **220** in which they are buffered) required for an output signal, the layout of the output video signal in terms of the position and dimensions of each window, and the output source signal for each window. The video output signal generator extracts the video information for each frame from the corresponding data buffers **220** and generates each frame for the output signal **114**. If the video signal includes audio components, these audio components are similarly retrieved as output source signals and added to the output video signal **114**.

Similarly, the audio and data output signal generators retrieve the output source signals from the appropriate buffers and produce their output signals.

If the device coupled to a particular output port **214** requires an analog output signal, then one of the D/A converters **215** may be dynamically coupled between the output signal generator and the output port **214** to convert the digital output signal into a corresponding analog output signal.

To produce the video signals for the example output video monitors **116***a* (FIG. **3**) and **116***b* (FIG. **4**), the output processor local controller configures the output processor **106** to operate as follows:

i. Packetized signal extractor **206** operates as follows:

a. Extract signal G**105** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**201** and store it as an output source signal in data buffer **220***a;*

b. Extract signal G**106** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**202** and store it as an output source signal in data buffer **220***b;*

c. Extract signal G**107** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**203** and store it as an output source signal in data buffer **220***c;*

d. Extract signal G**108** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**204** and store it as an output source signal in data buffer **220***d;*

e. Extract signal G**109** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**205** and store it as an output source signal in data buffer **220***e;*

f. Extract signal G**110** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**206** and store it as an output source signal in data buffer **220***f*; and

g. Extract signal G**112** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**207** and store it as an output source signal in data buffer **220***g*.

ii. Local signal generator **224** produces a 200×150 pixel data and time window as described above. Assign local identification code B**208** to this signal and store it as an output source signal in data buffer **220***h*.

iii. Output signal generator **212** generates two output signals as follows:

a. One output video signal generator **212***a* extracts local signals B**201**, B**203**, B**205**, B**207** and B**208** from the corresponding data buffers **220** and produces an output signal **114***a*.

b. A second output video signal generator **212***b* extracts local signals B**202** B**204** and B**206** from the corresponding data buffers **220** and produces an output signal **114***b*.

iv. A D/A converter is coupled between video signal generator **212***a* and output terminal to convert output signal **114***a* into an analog output signal, which is then displayed by video monitor **116***a*.

v. Output signal **114***b* is coupled directly to output port **214***b*. Video monitor **116***b* receives and displays the digital output signal **114***b*.

Referring to FIG. **1**, the input processor **104** receives a plurality of different input signals **110**, which are asynchronous with respect to one another to be received at the input processor. The input signals are processed using signal processor **126** to put them into a format that is required for the output signals **114** and resulting processed signal (the packet source signals) are combined into a single packetized signal **112**. If an input signal **110** does not require any processing to be used as part of an output signal, the input signal **110** may be a packet source signal. The input processor allows a plurality of asynchronous data signals **110**, which may include video, audio and data signals, to be combined into a single packetized signal that may be transmitted using a single communication link **186**.

The output processor **106** receives the packetized signal **112** and isolates the different packet source signals and stores them in buffers **220** as output source signals. Local signal processor **210** in the output processor **106** may be used to reverse any signal processing operation performed in the input processor, if necessary or desired, to produce the output source signals. In addition, local signal generators **224** in the output processor **106** may be used to produce additional output source signals. One or more of the output source signals is used by a set of output signal generators **212** produce output signals **214**. If necessary, a D/A converter may dynamically be coupled between an output signal generator and an output port to convert the corresponding output signal into an analog form.

Together, the input processor **104** and output processor **106** allow a plurality of input signals to be transported from the input ports **108**, combined in a manner controlled by a user through the user controller **118** and then provided in the final combined manner to the output devices **116**. The input processor **104** and output processor **106** are coupled together using a single communication link **186**, eliminating the need to couple each of the input signals separately to the output processor **106**.

In the embodiment of FIGS. **1** to **6**, the input processor **104** includes video scalers **160** to scale video input signals **110** from their original dimension to other dimensions required for the output signals **114**. In some cases, this may required that the input video signal may be expanded to large dimensions, resulting in a packet source signal that requires a larger portion of the packetized signal bandwidth to transmit than the original input signal **110**. To reduce this increased usage of bandwidth, another embodiment of the invention may be configured to ensure that the scalers **160** in the input processor **104** are only used to reduce an input signal **110** to smaller dimensions. Video scalers may be included in the output processor as signal processors **210** to scale any input signal that must be enlarged before it is incorporated into an output signal.

Reference is next made to FIG. **7**, which illustrates a second input processor **304**. Input processor **304** is similar to input processor **104** (FIG. **2**) and similar components are given similar reference numbers. The input processor local controller **340** is coupled to the various components of the input processor **304**. These couplings are not illustrated to simplify the Figure. Input processor **304** has a plurality of packetized signal output stages **327**, each of which comprises a packetized signal formatter **328**, packetized signal buffer **330** and packetized signal generator **332**. Each packetized signal output stage **327** is capable of generating a packetized signal **112**. Each packetized signal **112** may include information of any one or more of the input signals **110**. Input processor **304** may be used to provide packetized signals to different output processors **106** (FIG. **7**). Each output processor can receive a packetized signal containing only information from packet source signals that are required to produce the output signals **114** produced by that specific output processor.

The number of packet source signals (which are generally different versions of input signals **110**) that can be transmitted in a single packetized signal may be limited by the amount of data in each signal and the bandwidth of the packetized signal. Particularly in the case of audio and video signals, which may be required to be received in real time at the output processor **106** in order to be properly displayed on an output device **116**. Input processor **304** allows each input source **108** to be coupled to a single input port on a single input processor and then be combined in different combinations for transmission to different output processors **106**. In one embodiment, an input processor includes four output stages to provide four packetized signals **112**, which may be coupled to four different output processors **106**.

Reference is next made to FIG. **8**, which illustrates a second output processor **403**. Output processor **403** is similar to output processor **106** (FIG. **6**) and similar components are identified with similar reference numbers. Output processor **403** has a plurality of packetized signal input stages **405**, each of which comprises a packetized signal input buffer **404** and a packetized signal extractor **406**. Each input stage **405** receives a packetized signal **112** at a packetized signal input port **202** and stores the data for each source signal for each packetized signal in a separate data buffer in memory system **208**. This allows output processor **403** to receive a larger number of source signal than could be transmitted in a single packetized signal. Output processor **403** operates in the same manner to further process and generate output signals **114**, which may incorporate data from one or both of the packetized signals.

Reference is next made to FIG. **9**, which illustrates three input processors **304** and two output processors **403**. Input processor **304***a* receives eight input signals from eight sources **108***a*-**108***h* and generates two packetized signals **112***a* and **112***b*. Input processor **304***b* receives eight input signals from eight sources **108***i*-**108***p* and generates two packetized signals **112***c* and **112***d*. Input processor **304***c* receives eight input signals **108***q*-**108***x* and generates one packetized signal **112***e*. Output processor **403***a* receives packetized signals **112***a* and **112***c* and produces four output signals **114***a*-**114***d* at output terminals **714***a*-**714***d*. These output signals may include information from any of the sixteen input signals **108***a*-**108***p*. Output processor **403***b* receives packetized signals **112***b*, **112***d* and **112***e* and produces four output signals **114***e*-**114***h* at terminals **714***e*-**714***h*. The output signals **114***a*-**114***h* may include information from any of the twenty-four input signals **108***a*-**108***x*. In each case, each input source is

coupled to only one input processor, but may be combined with the other input sources in the output signals.

A single packetized signal **112** produced by a input processor **104** or **304** may be coupled to more than one output processor by first routing the packetized signal **112** through a signal replicating device. For example, the packetized signal **112** may be replicated using a cable driver with multiple duplicate outputs or other signal replication device and transmitted on multiple communications links to more than one output processor.

Reference is next made to FIG. **10**, which illustrates a first packet router **502** coupled between a plurality of input processors **104** and **304** and a plurality of output processors **403**. Each of the input processors produces one or more packetized signals **112** that are received by the packet router **502**. Packet router **502** includes a packet router controller **504**, a plurality of packetized signal buffers **506**, a plurality of packetized signal extractors **508**, a plurality of data buffers **510** and a plurality of packetized link output stages. Packet router controller **504** controls the operation of packet router **502** through control lines **507**, which couple the packet router controller **504** to the other elements of packet router **502** (connections are not shown to simplify the Figure). Each packetized signal **112** is buffered in a packetized signal buffer **506**. As complete packets **190** of a packetized signal **112** arrive, a packetized signal extractor **508** determines the global identification code of each packet **190** and stores all packets **190** corresponding to the same global identification code in a single data buffer **510**. The packetized signal extractor operates under the control of the packet router controller, which designates the particular data buffer in which the packets **190** having the same global identification code are stored. Through this process, all packets having the same global identification code are isolated in a data buffer. The actual content of the packets **190** is not altered.

Each packetized signal output stage **527** includes a packet selector **528**, a packetized signal buffer **530** and a packetized signal generator **532**. The packet selector **528** operates under the control of the packet router controller **504** to extract packets **190** from one or more of the data buffers **510** and place them in packetized signal buffer **530**. The packetized router controller receives packet router control instructions **520** from the master controller **102** to generate one or more packetized signals containing corresponding to a set of specified global identification codes. For each requested packetized signal, the packet router controller instructs the packet selector **528** in one of the packetized signal output stages **527** to extract packets from the data buffers **510** corresponding to the specified global identification codes for that requested packetized signal. As the packets become available in the data buffers **510**, the packet selector **528** extracts them and stores them in the packetized signal buffer **530**. Packetized signal generator **532** operates in the same manner as packetized signal generator **132** to generate a new packetized signal **512**.

Each packetized signal output stage **527** operates independently of the others. Any number of packetized signals **512** generated by the packetized signal output stages may include packets from the same data buffer **510** (corresponding to a particular global identification code). Each data buffer is operated to ensure that each packet in the data buffer are not discarded until each packet has been read by every packetized signal output stage that requires the packet.

Through this operation, the packet router receives a plurality of packetized signals **112** and generates a new set of packetized signals **512**. The created packetized signals may comprise packets with any combination of global identification codes, allowing input signals received at different input processors to be combined in a single packetized signal **512** for delivery to an output processor **403**. Each output processor may receive multiple packetized signals **512** from a packet router **502** and may also receive one or more packetized signals **112** directly from an input processor.

Inserting packet router **502** between a plurality of input processors **104** and **304** and output processors **106** and **403** allows an input signal **110** received at any one of the input processors to be routed (possibly after being processed in the input processor by signal processor **126**) to any of the output processors for use by any of the output devices **116** coupled to an output processor. Each input signal is received in only one location, but may be used in multiple formats (by creating appropriate versions of the input signal using signal processor **126**) at multiple output devices **116**.

Reference is next made to FIG. **11**, which illustrates a second packet router **602**. Like packet router **502**, packet router **602** receives one or more packetized signals **112** from one or more input processors, and provides one or more packetized signals **612** to one or more output processors. Packet router **602** includes a packet router controller **604**, a plurality of packetized signal buffers **606**, a plurality of packetized signal extractors **608**, a memory system **609** including a plurality of packet storage locations **610** and a plurality of packetized signal output stages **627**. Packet router controller **604** controls the operation of packet router **602** through control lines **607**. Packet router controller **604** uses the packet storage locations **610** to temporarily store packets **190** from the packetized signal **112**.

Each packetized signal **112** is buffered in a packetized signal buffer **606**. As each complete packet **190** of a packetized signal **112** arrives, a packetized signal extractor **608** stores the complete packet **190** in one of the packet storage locations **610**. The packet router controller **604** maintains a storage location table **611** indicating whether each packet storage location **610** is available to store a newly arrived packet. The packet router controller **604** selects an available packet storage location **610** and instructs the packetized signal extractor to store the newly arrived packet **190** in the selected packet storage location **610**. The packet router controller **604** then updates the storage location table **611** to indicate that the packet storage location **610** is not available to store another packet **190**.

The packet router controller **604** receives router control instructions **620** (similar to the router control instructions **520** received by packet router controller **502** (FIG. **10**)) from master controller **102** instructing the packet router controller to generate the packetized links **612** using packets **190** with specified global identification codes. The packet router controller **604** determines and assigns one of the packetized signal output stages **627** to generate each of the required packetized signals **612** and maintains a global identification code distribution table **613** correlating each global identification code with the packetized signal output stages **627** that require the global identification code. For example, a specified global identification code G**603** may be required for three of the outgoing packetized signals **612**. The three packetized signal output stage **627** used to generate those three packetized signals **612** are listed in the global identification code distribution table **613** in association with global identification code G**603**.

Each packetized signal output stage **627** includes a packet selector **628**, a packetized signal buffer **630** and a packetized signal generator **632**. Packet selector **628** reads packets **190** from the packet storage locations **610** as described below and stores the packets **190** in packetized signal buffer **630**. Packetized signal buffer **630** and packetized signal generator **632**

operate in the same manner as packetized signal buffer **130** (FIG. **2**) and packetized signal generator **132** to produce the packets signals **612**.

In the storage location table **611**, the router controller **604** maintains the status of each packet storage location **610** by recording the number of packetized link output stages **627** that must read a packet **190** stored in the packet storage location before the packet **190** may be discarded. In the present embodiment, when a newly received packet **190** is stored in a free packet storage location **610**, the router controller records the number of packetized signal output stages **627** that require the packet **190** to generate a packetized signal **612**. The router controller **604** then instructs each of the packetized signal output stages **627** to read the packet **190** from the packet storage location **610**. The packet selector **628** in each packetized signal output stage reads the packet from the packet storage location **610** and indicates to the router controller **604** that it has done so. The router controller **604** than decrements the number of packetized signal output stages **627** that still require the packet in that packet storage location **610**. When each of the packetized signal output stages **627** that require the packet **190** have indicated that they have read the packet **190**, the packet is no longer required (i.e. the number of packetized signal output stages still requiring the packet **190** is zero), and the router controller **604** treats the packet storage location **610** as free.

For example, each packet **190** with global identification code G**603** may be required by packetized signal output stages **627***a*, **627***c* and **627***d* to produce outgoing packetized signal **612***a*, **612***c* and **612***d*. When a complete packet **190** with global identification code G**603** is received, router controller **604** selects a free packet storage location **610***b* and instructs the appropriate packetized signal extractor **608** to store the packet **190** in packet storage location **610***b*. The router controller **604** then sets the status of packet storage location **610***b* to "3", indicating that the packet **190** must still be read by three packetized signal output stages. The router controller than instructs packetized signal output stages **627***a*, **627***c* and **627***d* to read the packet **190**. Each of packet selectors **628***a*, **628***c* and **628***d* reads the packet **190** and indicates to router controller **604** that it has done so. Router controller **604** decrements the status of the packet storage location **610***b* as it receives each indication and when the status returns to "0", the packet storage location **610***b* is again free to store another packet **190**.

Reference is next made to FIG. **12**, which illustrates a third packet router **702**. Packet router **702** is similar in structure and operation to packet router **602** and similar components are identified with similar reference numbers, increase by one hundred. Packet router **702** has a plurality of signals processors **726** coupled to memory system **709**. Signal processor **726** operate in a similar manner to signal processors **210** (FIG. **6**) under the control of router controller **704**. Router controller **704** receives instructions from master controller **102** to perform one or more signal processing steps on the signal encoded with a particular global identification code. For example, the master controller may indicate that a video signal with global identification code G**734** must be scaled to dimensions of 800×600 pixels and the resulting processed signal is to be assigned global identification code G**783** and must be included in packetized signals **712***b* and **712***c*. Router controller **704** then configures the global identification code distribution table **713** to route packets with global identification code G**734** to a video scaler (not shown) among the signal processors **726**. The router controller **704** may also route the same packet to one or more other signal processors **726** or packetized signal output stages **727**. The video scaler (not shown) is configured to perform the video scaling operation and produces packets **190** identified with global identification code G**783**. These packets are stored in free packet storage locations as designated by the router controller **702**. The packets are then distributed to packetized signal output stages **727***b* and **727***c* using global identification code distribution table **713** and storage location table **711**.

Reference is next made to FIG. **13**, which illustrates an output stage **827** for an output processor. Output stage **827** may be used for video output signals and includes a buffer **840** coupled to output signal generator **212** and an output terminal **214**. The output signal **114** generated by the output signal generator **212** is stored in data buffer **840**. The stored signal is extracted from the data buffer **840** by a local output generator **842** which makes the output signal **114** available at an output terminal **214**. Optionally a D/A converter may be coupled between the local output generator **842** and output terminal **214** to convert the output signal into a corresponding analog output signal for use by an analog device coupled to terminal **214**.

The buffered stream is also extracted from the data buffer **840** by a remote output generator **844**, which packetizes the video output signal **114** into a graphics packet stream **846**. Each packet in the graphics packet signal **846** contains video data for a fixed number of pixels in the output signal **114**. Each packet has the following fields:

i. packet ordering information, such as video positioning information indicating the first pixel at which the video data is to be displayed; and

ii. the video data.

Each packet may also contain additional metadata including error correction and detection information, frame numbering information and other information.

The graphics packet signal **846** is transmitted to a graphics packet signal ports **848**, from which it may be transmitted across a communication link **850** to display interface **852** capable of receiving the graphics packet signal **846**, reconstructing the output signal **114** and displaying the output signal on a display monitor **116**.

Output stage **827** allows an output signal **114** to be replicated on two different display monitors. The output signal **114** may be replicated on any number of display monitors by providing a remote output generator for each such monitor.

Reference is next made to FIG. **14**, which illustrates a switch **860** coupled between a plurality of graphics packet signal ports **848**, which may be part of one or more output stage **827** in one or more output processors, and a plurality of display interfaces **852**. The switch **860** may be implemented as a physical switch, which may be manually operable or automatically operable under the control of the master controller **102** (not shown in FIG. **12**). The switch **860** may be implemented using a field-programmable gate array (FPGA) or with any other switching or packet routing technology. Switch **860** allows any of the graphics packet signal ports **848** to be coupled to any display interface **852**, allows any of the output signals **114** available at any of the graphics packet signal ports **848** to be displayed at any display monitor coupled to a display adapter **852**.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the spirit and scope of the invention, which is limited only by the appended claims.

We claim:

1. A system for receiving one or more input signals and for producing one or more output signals, the system comprising:

(a) a master controller for generating input processor control signals and output processor control signals, and for assigning a unique global identification code to each of a plurality of packet source signals;

(b) an input processor having:

(i) one or more input ports for receiving the input signals;

(ii) one or more input signal processors for processing the input signals to provide one or more processed signals

(iii) an input processor memory system for buffering the input signals and the processed signals, wherein at least some of the buffered signals are designated as packet source signals;

(iv) one or more packetized signal output ports;

(v) one or more packetized signal output stages for retrieving one or more of the packet source signals from the input processor memory system and for producing one or more packetized signals at the packetized signal output ports, wherein each of the packetized signals includes a series of packetized signal packets, wherein each of the packetized signal packets contains the unique global identification code corresponding to one of the packet source signals and data corresponding to the same packet source signal; and

(vi) an input processor local controller for controlling the operation of at least the signal processors and the packetized signal output stages in response to the input processor control signals;

(c) an output processor having:

(i) one or more packetized signal input ports for receiving the packetized signals;

(ii) one or more packetized signal input stages for extracting data corresponding to each of the packet source signals from each of the packetized signals and for storing data corresponding to each of the packet source signals in a separate buffer in the output processor memory system as an output source signal based on the unique global identification code in the packetized signal packets of each packetized signal;

(iii) one or more output signal generators for providing one or more output signals, each of the output signals corresponding to one or more of the output source signals;

(iv) an output processor local controller for controlling the operation of the packetized signal input stages and the output signal generators in response to the output processor control signals; and

(d) a communications link coupled between the one or more packetized signal output ports and the one or more packetized signal input ports.

2. The system of claim **1** wherein the signal processors include one or more video scalers for providing a scaled version of the one or more input signals as one or more processed signals.

3. The system of claim **1** wherein the signal processors include one or more data compression elements for providing a scaled version of the one or more input signals as one or more processed signals.

4. The system of claim **1** further comprising one or more A/D converters coupled between one or more of the input ports and the input processor memory system.

5. The system of claim **3** wherein the data compression elements include one or more horizontal line filters.

6. The system of claim **3** wherein the data compression elements include one or more vertical line filters.

* * * * *

EXHIBIT B

US009100217B2

(12) **United States Patent**
Patel et al.

(10) **Patent No.: US 9,100,217 B2**
(45) **Date of Patent: Aug. 4, 2015**

(54) **APPARATUS, SYSTEMS AND METHODS FOR PACKET BASED TRANSMISSION OF MULTIPLE DATA SIGNALS**

(71) Applicant: **Evertz Microsystems Ltd.**, Burlington (CA)

(72) Inventors: **Rakesh Patel**, Mississauga (CA); **Romolo Magarelli**, Kleinburg (CA)

(73) Assignee: **Evertz Microsystems Ltd.** (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/026,039**

(22) Filed: **Sep. 13, 2013**

(65) **Prior Publication Data**

US 2014/0071999 A1 Mar. 13, 2014

Related U.S. Application Data

(63) Continuation of application No. 10/816,841, filed on Apr. 5, 2004, now Pat. No. 8,537,838.

(60) Provisional application No. 60/459,964, filed on Apr. 4, 2003.

(51) **Int. Cl.**

H04L 12/28	(2006.01)
H04L 12/54	(2013.01)
H04L 29/06	(2006.01)
H04N 7/52	(2011.01)
H04N 21/226	(2011.01)
H04N 21/234	(2011.01)
H04N 21/236	(2011.01)

(52) **U.S. Cl.**

CPC ***H04L 12/56*** (2013.01); ***H04L 29/06027*** (2013.01); ***H04L 65/607*** (2013.01); ***H04N 7/52*** (2013.01); ***H04N 21/226*** (2013.01); ***H04N 21/234*** (2013.01); ***H04N 21/236*** (2013.01)

(58) **Field of Classification Search**

USPC 370/412–418, 422–424, 428, 429; 455/132, 133, 137

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,682,195	A	10/1997	Hendricks et al.
6,430,183	B1	8/2002	Satran et al.
6,519,223	B1	2/2003	Wager et al.
6,937,291	B1	8/2005	Gryskiewicz

(Continued)

FOREIGN PATENT DOCUMENTS

FR	2655451	A1	6/1991
FR	2726723	A1	5/1996

(Continued)

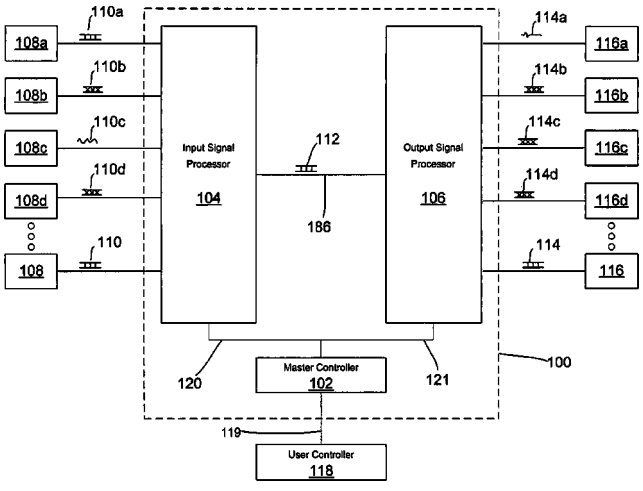
Primary Examiner — Alvin Zhu

(74) *Attorney, Agent, or Firm* — Bereskin & Parr LLP/S.E.N.C.R.L.,s.r.l.

(57) **ABSTRACT**

Apparatus, systems and methods for receiving one or more input signals and providing output signals in various video, audio, data and mixed formats are described. One or more input processors receive the input signals. Each of the input processors provides one or more packetized signals corresponding to one or more of the input signals received at the input processor. Each output processor can receive one or more packetized signals and generate one or more output signals. The output signals correspond to one or more of the input signals, additional locally generated signals or data relating to the signals or any combination of such signals. Use of a packet router according to the invention allows input signals encoded as one set of packetized signals to be recombined to provide additional packetized signals incorporating the same or different combinations of the packetized signals.

22 Claims, 11 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

7,190,695	B2	3/2007	Schaub et al.
7,269,841	B1	9/2007	Hendricks et al.
2003/0063611	A1	4/2003	Schaub et al.
2003/0156535	A1	8/2003	Lebizay et al.
2003/0198226	A1	10/2003	Westberg

FOREIGN PATENT DOCUMENTS

FR	2793042	A1	11/2000
FR	2793043	A1	11/2000

APPARATUS, SYSTEMS AND METHODS FOR PACKET BASED TRANSMISSION OF MULTIPLE DATA SIGNALS

FIELD OF THE INVENTION

This invention relates to systems for transmitting and distributing audio data, video data and other types of data.

BACKGROUND OF THE INVENTION

Recent advances in video monitor technology have resulted in the development of large format, high quality video displays that are capable of displaying multiple video signals simultaneously. In television studios and other locations where many different video sources must be monitored, these video displays have begun to displace traditional individual monitors that displayed a single video source to which they were physically coupled.

Modern signal processing equipment allows video and other data to be routed to different display monitors, however, this equipment can still require that for a particular signal to be used in multiple locations on multiple display devices it must be replicated and coupled to equipment in the different locations. This results in excessive cabling requirements, multiple signal regeneration and replication stages, and can result in degraded signals and multiple failure points within the signal path.

There is a need for an improved efficient system for receiving various input signals, including video, audio and data signals, formatting the received signals and routing the formatted signals to various output devices.

SUMMARY OF THE INVENTION

In one embodiment, the present invention provides a system that includes a master controller, one or more input processors, one or more output processors, and one or more user controllers. The system may also include additional master controllers that serve as back-up master controllers.

One or more input devices are coupled, directly or indirectly, to each of the input processors. Each input device provides one or more input signals to the input processors. One or more output devices are coupled to the output processors. Each output device receives an output signal from an output processors. Each of the input processors generates one or more packetized signals. Each packetized signal is transported across a communications link to one or more of the output processors. Each output processor may receive one or more packetized signals.

The master controller receives user control signals from one or more user controllers indicating which input signals are to be routed to which output devices. The user control signals may also indicate the format in which the input signal is to be presented at the output device.

For example, for a video input signal the user control signals may specify the position and dimensions of a window on an output display device in which the input video signal is to be displayed. The user control signals may also define other characteristics for the displayed video signal, such as color balance, white balance, color to black & white conversion or the addition of a border or any other characteristics that a video signal may be processed to have.

For an audio input, the user control signals may specify a particular equalization (ie. jazz, rock, classical), volume balancing to some specified maximum or average volume, left-right signal balance, encoding in or conversion to a particular noise reduction and/or a multi-channel audio standard. For a data signal, the user controls may specify characteristics that are appropriate to the data.

In response to the user control signals, the master controller generates input processor control signals to operate the input processors and output processor control signals to operate the output processors. The input and output processor control signals may be transmitted to the various input and output processors using any type of communications link. The master controller coordinates the operations of the various input and output processors (and other elements of the system) to provide the output signals requested by a user or users who operate the user controllers to generate the user control signals.

If the system includes more than one master controller, one of the master controllers may be designated as a primary master controller and the remaining master controllers designated as backup master controllers. Each of the master controllers is coupled to the user controllers to receive the user control signals and is capable of generating the input processor control signals and the output processor control signals. The primary master controller actively generates the input processor control signals and output processor control signals. If the master controller fails or is disabled for any reason, one of the backup master controllers may be designated as the primary master controller.

Each input processor has an input processor local controller which receives the input processor control signals for that input processor. In response to the input processor control signals, the input processor local controller configures the various components of the input processor to receive and process the input signals coupled to that input processor and to generate one or more packetized signals, as requested by the master controller.

Each input processor includes a plurality of data buffers to store each input signal in a digital form. If an input signal is received in an analog form, an analog-to-digital converter is dynamically coupled between an input port at which the input signal is received and a data buffer to digitize the signal. Some input signals may be received in a processed manner, meaning that the signal has been processed in some manner. If an input signal is received in a processed manner then an unprocessor may be dynamically coupled between an input port at which the input signal is received and a data buffer in which the input signal is stored. Additionally each input processor may include one or more data analyzers that can be dynamically coupled to each input signal to provide information about the input signal. The resulting information is also stored in a data buffer. The buffered signals are then read out and processed by signal processors to format them as indicated by the user control signals. The processed signals are also buffered in data buffers. Each input signal may be processed multiple times to create different versions of the input signal, to extract various information regarding the input signal for use on different output devices, or for use on the same output device in different versions, formats or sizes.

The data buffers in the input processor, the output processor and other components and embodiments of the invention are used to temporarily store data that is received from a source and is subsequently read out by one or more recipient or destination elements or devices. Data may be read out in the order in which it is received, in which case a first-in/first-out buffer may be used to store the data. Alternatively, the data may be read out in any order by the recipient elements. In each case, the data buffer is configured to retain any datum until it has been read by all recipient elements that use the datum.

Each of the buffered signals (including the input signals and the processed signals) is assigned a global identification code. One or more of the buffered signals are converted into a packetized signal by a packetized signal output stage in each input processor. Each packetized signal contains a series of packets. Each packet contains a part of the data from the buffered signal along with the global identification code of the buffered signal. An input processor may have more than one packetized signal output stages to produce more than one packetized signal.

A packetized signal may be converted into and transmitted as a bitstream, or it may be transmitted using any communications protocol.

Each output processor receives one or more packetized signals. Each packetized signal is buffered as it is received. As complete packets corresponding to each global identification code are received, they are assigned a local identification code and are buffered in a separate data buffer. The isolated packets in data buffer correspond to a particular version of an input signal received at one of the input processors. The isolated stream may be processed to reverse any signal processing step or steps applied in the input processor or in an input device or combination of devices that combine to produce an input signal coupled to the input processor, such as a data compression step, or to apply additional signal processing steps. Any such processed signal is buffered again and assigned a new local identification code. One or more these buffered signals is then combined to form each output signal.

In one embodiment of the invention, a packet router is coupled between a plurality of input processors and a plurality of output processors. The packet router receives packetized signals from the input processors and isolates the packets corresponding to each global identification code. The packetized router then assembles new packetized signals corresponding to a combination of the global identification codes. The packetized router operates under the control of the master controller to route packets with the appropriate global identification code to the appropriate output processor. The packet router allows an input signal received at any of the input processors to be formatted and routed to any of the output processors.

In another embodiment of a packet router according to the invention, packets from one or more incoming packetized signals are stored in packet storage locations within a memory system. The packets are then read by one or more packetized signal output stages, each of which produces an outgoing packetized signal corresponding to a selected set of global identification codes. Storage of incoming packets and distribution of the packets to the packetized signal output stages is controlled by a router controller. In another embodiment of a packet router, one or more signal processors, such as video scalers or delay elements, are provided to process the incoming packets to provide processed packets that form a processed signal. Each processed signal is assigned a unique global identification code and may be included in an outgoing packetized signal.

These and other aspects of the invention and its various embodiments are described in greater details below,

BRIEF DESCRIPTION OF THE DRAWINGS

Several exemplary embodiments of the present invention will now be described in detail with reference to the drawings, in which:

FIG. **1** illustrates a system according to the present invention;

FIG. **2** illustrates a first input processor according to the present invention;

FIG. **3** illustrates the display of a first video monitor;

FIG. **4** illustrates the display of a second video monitor;

FIG. **5** illustrates one form of a packetized signal according to the present invention;

FIG. **6** illustrates a first output processor according to the present invention;

FIG. **7** illustrates a second input processor according to the present invention;

FIG. **8** illustrates a second output processor according to the present invention;

FIG. **9** illustrates an exemplary coupling between a plurality of input processors and a plurality of output processors;

FIG. **10** illustrates a first packet router according to the present invention;

FIG. **11** illustrates a second packet router according to the present invention;

FIG. **12** illustrates a third packet router according to the present invention;

FIG. **13** illustrates an alternative output stage for an output processor according to the present invention; and

FIG. **14** illustrates a switch for use with the alternate output stage of FIG. **13**.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Reference is first made to FIG. **1**, which illustrates a signal processing system **100**. The signal processing system includes a master controller **102**, an input processor **104** and an output processor **106** according to the present invention. The input processor **104** receives a plurality of input signals **110** from various signal sources **108** and provides a packetized signal **112**. The packetized signal **112** corresponds to some or all of the input signals **110** or to data or processed signals derived from the input signals. The packetized signal **112** is transported by a communications link **186** to the output processor **106**. The output processor **106** receives the packetized signal **112** and produces one or more output signals **114**, which are processed by output devices **116**. The output signals **114** correspond, at least in part, to one or more of the input signals **110**.

The system may additionally include backup master controllers (not shown).

The input processor **104** and output processor **106** operate under the control of the master controller **102**. The master controller **102** is coupled to one or more user controllers **118**, from which the master controller receives user control signals **119**. The master controller **102** and the user controllers **118** may be combined in a single unit, or may be assembled in a single assembly, or they may be separate units that are coupled together.

A user or multiple users (not shown) human or some other type of device (for example automated monitoring and control systems) operate the user controllers **118** to generate the user control signals **119**. The user control signals **119** indicate which input signals or signals derived from the input signals **110** the user would like included in the output signal **114** provided to each output device. Each user may have control over one or more output devices **116** in whole or in part. The user control signals **119** may also indicate additional characteristics about the output signal **114** provided to each output device **116**. The master controller **102** translates the users control signals **119** into input processor control signals **120** and output processor control signals **121** to control the operation of the input processor **104** and the output processor **106**

respectively so that the output signals **114** are provided in accordance with the user control signals **119**.

The input signals **110** may be base-band, compressed, time division multiplexed audio signals, video signals (which may also include audio information), metadata, or other data signals. Similarly, the output signals **114** may be audio signals, video signals, or data signals. Typically, each output signal **114** will correspond to one or more of the input signals and or information derived from the input signal. A particular output signal may include a combination of audio, video or data input signals or signals produced by input signal analyzers or any combination of these types of signals. The nature of each output signal **114** is appropriate for the output device **116** that receives the output signal **114**. Some of the output devices **116** may be video monitors, such as analog video monitor **116***a* and digital video monitor **116***b*, for displaying output video signals. Some of the output devices **116** may be sound systems, such as sound amplification and broadcast system **116***c*, for further processing or playing output audio signals. Some of the output devices may be data processing systems, such as computer system **116***d*, for further processing or displaying the output data signals. In any particular embodiment of the present invention, the output signals **114** may be of the same or different types, depending on the usage of the embodiment. In an alternative embodiment of the invention, the output processor may provide only a single output signal. The type of any particular signal may change depending on the usage of the signal, under the control of the master controller **102**.

Reference is next made to FIG. **2**, which illustrates the input processor **104** in greater detail. Input processor **104** includes a plurality of input ports **123**, a plurality of input signal analyzers or processors, such as A/D converter **150**, a plurality of data buffers **124**, which are part of a memory system **122**, one or more signal processors **126**, a packetized signal output stage **127**, a packetized signal output port **138** and an input processor local controller **140**. The packetized signal output stage **127** includes a packetized signal formatter **128**, a packetized signal buffer **130** and a packetized signal generator **132**.

Memory system **122** may be a local memory device or memory space within the input processor **104** or it may be located on an attached storage device or other medium. Data buffers **124** will typically comprise memory space allotted within memory system **122**.

The input processor local controller **140** receives the input processor control signals **120**. The input processor local controller **140** controls the operation of the various elements of the input processor **104** through control lines **142** in response to the input processor control signals **120**.

Each input signal **110** is coupled to one of the input ports **123**. Each of the input ports **123** is coupled to the memory system **122**. Each input signal **110** is buffered in a data buffer **124** in memory system **122**. Analog input signals **110** are converted to a digital version and the digital version is buffered in a data buffer **124**. For example, if input signal **110***c* is an analog input signal, then an analog-to-digital (A/D) converter **150** is dynamically coupled between input port **110***c* and memory system **122** to convert input signal **110***c* into a corresponding signal **110***c*' comprising a stream of packets according to a digital signal standard. For example, if input signal **110***c* is a standard 1V peak-to-peak audio signal, it may be sampled and converted by A/D converter **150** into a WAV digital audio signal **110***c*', as is commonly done to record music on CD-ROM media. For example, if input signal **110***c* is a compressed video stream a de-compressor (one of the signal processors) may be dynamically coupled between input port **110***c* and memory system **122** to convert input signal **110***c* into a corresponding signal **110***c*' comprising a stream of data according to a digital signal standard.

In the present embodiment, the input processor contains a bank of A/D converters, each of which may dynamically be coupled between any of the input ports **123** (or a group of the input ports) and memory system **122**. The input processor local controller **140** controls the coupling of any particular A/D converter between any particular input port **123** and memory space **122**. A particular A/D converter **150** may be shared by two or more input ports **123** under the control of input processor local controller **140**. Alternatively, a dedicated A/D converter **150** may be provided for some or all of the input ports **123**. The A/D converter **150** may be activated by the input process local controller **140** if the input signal **110** received at a port **123** is an analog signal. In another alternative embodiment, some or all of the input ports **123** may be designated as analog input ports and an A/D converter may be permanently coupled between those ports and the memory system **122**.

The input ports may be any type of communication port, such as an Ethernet, BNC, optical, telephone line or any port suitable with any type of communication system. The input signals may be in any communication standard or protocol, including, including TCP/IP. In this case, the coupling between the input device and the processor may be a LAN, WAN, the Internet or another TCP/IP communication system.

The input processor **104** may also contain a bank of liquid signal analyzers (not shown). The input signal analyzers may be dynamically or statically coupled to an input port in the same manner as ND converter **150**. For example if the input signal is a video signal, an input signal analyzer may extract performance and signal content metrics from or about the input signal such as blackness of the signal, the amount of motion within the signal, bit or formatting errors in the signal. The metadata produced by data analyzer is stored in a data buffer and is considered and treated as a processed signal that can be packetized and coupled to the output processor over a communication link.

Each of the input signals **110** may be retrieved from the corresponding data buffer **124** as a buffered signal **156**.

The input processor may also include other input signal processing elements that may be coupled between an input port **123** and memory system **122**. The signal processing elements may include video scalers, video de-interlacers, data compressors, data de-compressors, data format converters or any other type of signal processor, including the signal processing elements described below. For example, if one of the input signals is an analog NTSC video signal, then a video signal digitizer may be dynamically coupled between an input port at which the signal is received to convert the input signal into a MPEG2 digital video signal. The input processor may contain a bank of input signal processing elements and analyzers, which may be dynamically coupled between any input port and memory system **122**. Input processing elements or analyzers may also (or alternatively) be coupled to only one port for selective use with that port. Input processing elements may also (or alternatively) be permanently coupled to one or more of the input ports.

The signal processors **126** are coupled to memory system **122** to retrieve the buffered signals **156** from memory system **122**, process the buffered data signals to generated processed signals **158**, which are then buffered in data buffers **124** in memory system **122**. A processed signal **158** is stored in a different data buffer **124** than the input signal **110** from which the processed signal is derived. The signal processors **126** are illustrated in a dotted box and the coupling between the

memory system **122** and the signal processor extends to the dotted box to indicate that any of the signal processors may be dynamically coupled to any of the data buffers **124** to retrieve a buffered signal and to store a processed signal.

A particular input signal **110** may be processed to generate more than one processed signal **158** and each of the resulting processed signals **158** are stored in different data buffers **124** in memory system **122**. As a result, the original input signal **110** and any versions of the original input signal **110** that are generated as processed signals **158** are available from memory system **122** as buffered signals **156**.

In the present embodiment, the signal processors **126** include video scalers **160**, embedded audio extractors, ancillary data exteriors, signal content analysers and data compressors **164**. The signal processors **126** may also include data decompressors, image rotation devices, special effects processors, image invertors, spatial filters, edge enhancement processors, color space converters, audio sweetening processors, digital audio decompressors, and digital audio processors. A signal processor may be used to process two or more input signals (or processed signals) by time-division-multiplexing the signal processor between the data buffers used to buffer the two or more input signal (or processed signals).

Each video input signal **110** will have height and width dimensions, usually defined in pixels. For example, a video image may comprise a series of frames that are 640 pixels wide by 400 pixels high. A video scaler **160** is capable of rescaling a video signal from its original dimensions to different dimensions. In the present embodiment, the input processor **104** includes a plurality or bank of video scalers **160**. Each video scaler **160** receives control instructions from the input processor local controoler **140** to extract a particular video input signal **110** from the appropriate data buffer **124** and rescale the video input signal to specified dimensions and to store the resulting processed signal **158** in another data buffer **124**. A video scaler **160** may be configured to retain or change the aspect ratio of an input data signal or to crop the input data signal in the processed signal and to provide any other function that a conventional scaler is capable of providing. For example, a video scaler may be configured to crop the input data signal to select a portion of it, and then scale the cropped video image to specified dimensions.

A particular video scaler **160** may be instructed to scale a video input signal **110** to more than one set of new dimensions and may generate two or more processed signals **158**, each of which is separately buffered in separate data buffers **124**. In addition, a particular video scaler **160** may be shared (or multiplexed) between two or more video input signals **110** to generate two or more corresponding processed signals **158**, each of which is separately buffered in separate data buffers **124**. One video input signal **110** may also be retrieved by two or more video scalers **160** (or other signal processors **126**) to produce two or more corresponding processed signals **158**, which are similarly buffered in separate data buffers **124**.

Data compressors **164** are used to generate a processed signal **158** that is a compressed version of any signal stored in a data buffer **124**. For example a video input signal **110** in a DVI format may be compressed into an MPEG-2 format to reduce the amount of data required to transmit the signal. The resulting MPEG-2 format video signal is stored in a data buffer **124** as a processed signal **158**. The data compressors **164** may include a plurality of compression elements, which may be hardware or software elements, designed to compress audio, video or data signals into various types of compressed signals. The data compressors may provide a lossy or lossless compression. In each case, the compressed data signal produced by a data compressor **164** is stored as a processed signal **158**. A particular embodiment of an input processor **104** may include any number and type of data compressors **164**.

The data compressors **164** may include horizontal or vertical line filters that produce a processed video data signal comprising a portion of the video data from a video input data signal. For example, a horizontal line filter may be configured to horizontally compress a 640×400 pixel video signal into a 320×400 pixel video signal by discarding every other pixel in each line of the video signal. A vertical line filter may be configured to compress a 640×400 pixel video signal into a 640×200 pixel video signal by discarding every other line in the video signal. A horizontal/vertical line filter may be configured to compress a 640×400 pixel video signal into a 160×100 pixel video signal by discarding three of every four lines of the video signal and discarding three of every four pixels in each line that is retained.

The input signals **110** will typically be asynchronous with respect to one another (unless they are from a synchronized source). The size and timing of packets in each input signal will depend on the signal standard used to encode the signal. For example, input signal **110***a* may be formed of uniformly sized packets that are spaced equally apart in time. Input signal **110***b* may be formed of differently sized packets that do not arrive at equally spaced intervals. As a result of the differences between the input signals **110**, data buffers **124** may be allocated with a large or smaller memory space to properly buffer each input signal **110**.

In addition, the size and timing of packets in processed signals **158** may also vary depending on the data standard used to encode the processed signals **158**. The data buffers **124** used to buffer a processed signal **158** may similarly be dynamically allocated a memory space of a suitable size.

The master controller **102** (FIG. **1**) controls the operation of the input processor **104**. The master controller **102** assigns a unique global identification code to each signal that is buffered in a data buffer **124**. This global identification code is used to identify the signal in both the input processor and the output processor. Each version of a particular input signal **110** that is buffered in a data buffer **124** is assigned a different global identification code. For example, video input signal **110***a* is first buffered in the form in which it is received. The input signal **110***a* may be scaled to new dimensions using a video scaler **160** to produce a scaled signal **110***aa*, which is separately buffered. The input signal **110***a* may also be scaled to a second set of dimensions to produce a second scaled signal **110***ab*, which is also separately buffered. The second scaled signal **110***ab* may then be compressed to produce a scaled and compressed signal **110***ac*, which is also separately buffered. Each of the different versions **110***a*, **110***aa*, **110***ab*, **110***ac* may be separately retrieved from its data buffer and may be identified using its unique global identification code.

To further explain the invention and the present embodiment, an example of the use of this embodiment will be described. In the example, the input processor **104** receives three digital video input signals **110***a*, **110***b* and **110***d* and one analog video input signal **110***e*. Analog video input signal **110***e* is digitized using a A/D converter **150** to produce a digital signal **110***e*' corresponding to analog signal **110***e*. Signals **110***a*, **110***b*, **110***d* and **110***e***40** are buffered in separate data buffers **124**.

Reference is made to FIG. **3**, which illustrates output device **116***a*, which is an analog standard definition 4:3 format video monitor capable of displaying images with a resolution of 640×480 pixels. The display of video monitor **116***a* is used to display information in five different parts or windows: video windows **170**, **172**, **174** and **176** and graphics window **178**.

A user configures the video and other information shown on each video monitor using user controller **118**. User controller **118** may provide a graphical or other interface allowing the user to define windows and other elements on a video monitor and assign specific input signals or other information to be displayed in each window or other element. The user has defined the parts of the display on video monitor **116***a* as follows:

Window/Element	Position (relative to top left corner)	Dimensions	Contents
Video window 170	10, 10	400 × 300	A version of video input signal 110a
Video window 172	420, 10	200 × 113	A version of video input signal 110b
Video window 174	460, 340	160 × 120	A version of video input signal 110d
Video window 176	10, 320	440 × 140	Rejected packets data for video signal 110b
Graphics window 178	420, 150	200 × 150	Date/Time/ Metadata Information

Reference is made to FIG. **4**, which illustrates output device **116***b*, which is a digital high definition 16:9 format video monitor with a resolution of 1920×1080 pixels. The display of video monitor **116***b* is used to display information in three different parts or windows: video windows **180**, **182** and **184**. The user has defined the parts of the display on display monitor **116***b* as follows:

Window/Element	Position (relative to top left corner)	Dimensions	Contents
Video window 180	60, 60	1140 × 640	A version of video input signal 110b
Video window 182	1280, 60	560 × 420	A version of video input signal 110e
Video window 184	1280, 540	610 × 460	A version of video input signal 110a

In an alternate example, window positions maybe such that some or all of the windows are overlapping, or arranged in a cascaded manner.

The video windows have been described as containing "a version of "one of the video input signals **110**. The user will typically specify the position and dimension of a window on a video monitor and the input signal **110** that the user would like displayed in each window. An appropriate version of the input signal is prepared by the input processor **104** and provided to the output processor **106** for display on the video monitor. Alternatively, the user may specify certain signal processing steps to be performed on an input signal before it is displayed in a window. For example, if the signal processors **126** (FIG. **2**) include a color / black & white converter, then a user may specify that a color input signal be converted into a black & white signal and that the black & white version of the input signal (or a version of the black & white signal) be displayed in a particular window.

Reference is again made to FIG. **1**. The user controller **118** transmits the user's instructions for each output device **116** to the master controller as user control signals **119**. The user's instructions relating each output device **116** will typically depend on the nature of the output device **116**. For example, if an output device **116** is an audio processing system capable of receiving and switching between multiple audio signals, then the user may specify that one or more audio input signals **110**, or the audio components of video input signals **110**, be directed to the sound output device **116**. If an output device **116** is only capable of receiving a single audio signal and then amplifying and broadcasting the audio signal, the user may specify that a particular input audio signal or the audio component of a particular video input signal **110** be directed to the sound output device **116**. Similarly, a user may specify that any particular output device **116** can receive any combination of information that the output device is capable of receiving.

Referring again to FIG. **3**, the windows **176** and **178** contain information that is not present in any input signal **110**. The user controller is configured to allow the user to select any information that may be generated within system **100** and which is suitable for a particular display device. The rejected packets information displayed in video window **176** may be determined by a signal analyzer (not shown) that analyzes input signal **110***b* to determine the number of defective packets received as part of the input signal **110***b*. The signal analyzer is one of the signal processors **126**. The signal analyzer then generates a video signal illustrating this information in a standard video signal format and stores the video signal in a data buffer **124** as a processed signal **158**.

In response to the user control signals **119**, the master controller transmits input processor control signals **120** to the input processor local controller **140** indicating the final version of each input signal **110** that will be required by the output processor **106** to produce the output signals **114** for the output devices **116**. For each required version, the master controller **102** also indicates the top left pixel at which that version will be displayed.

For the example input signals **110** and output video monitors **116** described above, the master controller instructs the input processor to prepare the following signals:

i. 400×300 pixel scaled version of video input signal **110***a;*
ii. 610×460 pixel scaled version of video input signal **110***a;*
iii. 200×113 pixel scaled version of video input signal **110***b;*
iv. 1140×640 pixel scaled version of video input signal **110***b;*
v. 160×120 pixel scaled version of video input signal **110***d;*
vi. 560×420 pixel scaled version of video input signal **110***e*; and
vii. 440×140 pixel video image illustrating rejected packet information for video signal **110***b.*

The master controller **102** does not instruct the input processor to produce a signal showing the date, time and analyzed information, which is required for graphics window **178** on video monitor **116***a*. This signal is produced in the output processor and is described below.

In response to the input processor control signals **120**, the input processor local controller **140** determines how the required versions of each input signal **110** can be produced and configures and couples the input ports **123**, A/D converters **150**, data buffers **124** and signal processors **126** to produce the required versions of each input signal. As described above, every signal stored in a data buffer **124** is assigned a unique global identification code.

In the present example, the input processor local controller **140** configures the input processor **104** as follows:

i. Store input signal **110***a* in data buffer **124***a*. Assign global identification code G**101** to the stored signal.
ii. Store input signal **110***b* in data buffer **124***b*. Assign global identification code G**102** to the stored signal.
iii. Store input signal **110***d* in data buffer **124***c*. Assign global identification code G**103** to the stored signal.

iv. Couple an ND converter **150** between input port **123***e* at which input signal **110***e* is received to produce a digital version **110***e*' of input signal **110***e*. Store digital signal **110***e*' in data buffer **124***f*. Assign global identification code G**104** to the stored signal.

v. Couple video scaler **160***a* to memory system **122** to retrieve signal G**101** and produce a scaled version of 400×300 pixel scaled version of signal G**101**. The scaled version is stored in data buffer **124***f* and is assigned global identification code G**105**.

vi. Couple video scaler **160***b* to memory system **122** to retrieve signal G**101** and produce a 610×460 pixel scaled version of signal G**101**. The scaled version is stored in data buffer **124***g* and is assigned global identification code G**106**.

vii. Couple video scaler **160***c* to memory system **122** to retrieve signal G**102** and produce a 200×113 pixel scaled version of signal G**102**. The scaled version is stored in a memory buffer **124***h* and is assigned global identification code G**107**.

viii. Couple video scaler **160***d* to memory system **122** to retrieve signal G**102** and produce an 1140×640 pixel scaled version of signal G**102**. The scaled version is stored in data buffer **124***b* and is assigned global identification code G**108**.

ix. Couple video scaler **160***e* to memory system **122** to retrieve signal G**103** and produce a 160×120 pixel scaled version of signal G**103**. The scaled version is stored in data buffer **124***j* and is assigned global identification code G**109**.

x. Couple video scaler **160***f* to memory system **122** to retrieve signal G**104** and produce a 560×420 pixel scaled version of signal G**104**. The scaled version is stored in data buffer **124***k* and is assigned global identification code G**110**.

xi. Couple a signal analyzer (one of the signal processors **126**, as described above) to the memory system **122** to retrieve and analyze signal G**102**. The signal analyzer produces a video signal with a standard size of 320×200 pixels and metadata. The output of the signal analyzer is stored in data buffer **124***m* and is assigned global identification code G**111**.

xii. Couple a video scaler **106***g* to memory system **122** to retrieve signal G**111** and produce a 440×140 pixel scaled version of signal G**111**. The scaled version is stored in data buffer **124***n* and is assigned global identification code G**112**

During the operation of input processor **104**, successive packets of each signal stored in a data buffer **124** are stored in the data buffer and previously stored packets are read out and then discarded. Some signals, such as input signal **110***a* are read by more than one device. Input signal **110***a*, identified by its global identification code G**101**, is read out by video scalers **160***a* and **160***b*. The data buffer **124***a* in which input signal **110***a* is buffered is configured to discard each packet in the input signal only after the packet has been read by both of the video scalers.

Signals G**105**-G**110** and G**112** are required to produce the output signals **114** for video monitors **116**. These signals are combined into packetized signal **112** using packetized signal formatter **128**, packetized signal buffer **130** and packetized signal generator **132**. The signals that are used to produce a packetized signal **112** are referred to herein as packet source signals for that packetized signal.

Reference is next made to FIG. **5**, which illustrates the format of the packetized signal **112**. In the present embodiment, the packetized signal **112** comprises a series of packets **190**, each of which contains the following fields:

i. global identification code for the signal from which the data in the packet was obtained;

ii. packet ordering information;

iii. a data payload;

iv. optional error detection and correction information and other metadata.

The packet ordering information, which may comprise a sequential packet number for each packet with the same global identification code, allows packets derived from the same packet source signal to be isolated from other packets, allowing the data in the packet source signal or a version of the data in the packet source signal to be re-assembled in the output processor **106**, as is described below.

In addition to packet ordering information, a video data packet may optionally also include frame ordering information, identifying the particular frame of video signal to which the packet corresponds.

The content and format of the data payload in each packet **190** will depend on the type of the data contained in the packet.

For example, if a packet source signal comprises a stream of data that is not organized as packets of information, then each packet **190** formed from that packet source signal contains a fixed amount of data in the data payload field. For example, if a packet source signal is a continuous stream of video data, then each corresponding packet **190** contains up to 320 bytes of the video data. In other embodiments, the amount of data in a particular packet may be fixed at a different size or may be variable.

If a packet source signal is organized as a series of packets of information, as in case of MPEG-2 encoded video or MP3 encoded audio or AES encoded audio, then the data payload may comprise the entire packet from the packet source signal.

Referring to FIG. **3**, video window **170** is a 400×300 pixel window in which signal G**105** will be displayed. Signal G**105** is created by video scaler **160***a* in a digital video standard that comprises a stream of video data that is not separated into packets. If each pixel in the 400×300 pixel window **170** requires one byte of video data from signal G**105**, then an entire frame of video information for the window requires 120,000 bytes of data. If the standard according to which the signal is encoded provides that one complete horizontal line of information will be encoded in a single packet, then each frame will be encoded in 300 packets in data buffer **124***f*.

Packet signal formatter **128** retrieves the successive packets in data buffer **124***f* that encode each frame of video signal G**105** and produces a series of packets **190** that correspond to the retrieved packets. In the following discussion, pixel numbers are set out as n,m where n is the number of the pixel in a window in a horizontal line of a window or frame and m is the number of the line in the window or frame. Pixels and lines are numbered starting at 1. The packets **190** corresponding to one frame of the 400×300 pixel window include pixel data for the following ranges of pixels:

Packet	Pixel range
1	1.1-320.1
2	321.1-240.2 (i.e. pixels 321-400 on line 1 and pixels 1-240 on line 2)
3	241.2-160.3
4	161.3-80.4
5	81.4-400.4

-continued

Packet	Pixel range
6	1.5-320.5
.	.
.	.
.	.
373	241.298-160.299
374	161.299-80.300
375	81.300-400.300

Similarly, the packetized signal formatter reads the successive packets in data buffer **124***h* that encode each frame of video signal G**107** and produces a series of packets **190**. The packets **190** corresponding to one frame of the 200×113 pixel window include pixel data for the following ranges of pixels:

Packet	Pixel range
1	1.1-120.2
2	121.2-40.4
3	41.4-160.5
.	.
.	.
.	.
70	81.111-300.112
71	1.113-200.113

The last packet **190** used to packetize each frame of video signal G**107** contains data for only 200 pixels. The remaining data space is filled with null characters by the packetized signal formatter **128**. Alternatively, the last packet may have a shortened data payload length.

The packetized signal formatter **128** produces packets **190** corresponding to the data in the packet source signals. Depending on the data formats used for the packet source signals, the packetized signal formatter **128** may produce one or more packets **190** that correspond to the data in one packet of a packet source signal. For example, if packet source signal G**105** is encoded using a digital video standard that includes a complete frame of video in a single packet, then the packetized signal formatter **128** will produce 375 packets **190** corresponding to each packet in the packet source signal.

A single packet **190** may correspond to data from more than one packet of a packet source signal. For example, if packet source signal G**107** is encoded using a digital video standard that encodes a single line of a frame in each packet, then the packetized signal formatter will generate packets **190** corresponding to more than one packet in the packet source signal, since each of the packets **190** can contain data for 320 pixels and since each line in packet source signal G**107** is only 200 pixels wide.

The packetized signal formatter **128** proceeds to generate packets **190** for each of the packet source signals for the packetized signal **112**, as packets from the packet source signals are available from the corresponding data buffers **124**. As packetized signal formatter **128** produces packets **190**, it stores them in packetized signal buffer **130**. Packetized signal buffer **130** is a data buffer and may include memory space in memory system **122**.

Packetized signal generator **132** retrieves the packets **190** stored in packetized signal buffer **130** and generates packetized signal **112** at packetized signal output port **138**. Packetized signal **112** may be a synchronous signal. For example, in the present embodiment, the packetized signal is a synchronous signal transmitted at 2.5 Gbits/second. Referring to FIG. **5**, if there are no packets **190** in the packetized signal buffer **130**, the packetized signal generator transmits null characters **192** between packets. In other embodiments, the packetized signal generator may transmit the packetized signal **112** at any bit rate, depending on requirements and capabilities of the system **100**.

Reference is made to FIG. **1**. In the present embodiment, the packetized signal output port **138** will typically be coupled to the output processor (FIG. **1**) through a communication link **186**, which may be a data cable such as an electrical or optical cable. The data rate and other aspects of the data protocol used to transmit the packetized signal **112** correspond to the ability of the communication link **186**.

In other alternative embodiments, the packetized signal generator **132** may transmit the buffered packets **190** as an asynchronous stream of packets to the output processor using any communication protocol, including TCP/IP. In this case, the communication link **186** may be a cable or may be a LAN, WAN, the Internet or another communication system.

Reference is next made to FIG. **6**, which illustrates the output processor **106**. The output processor **106** has a packetized signal input port **202**, a packetized signal input stage **205**, a memory system **208**, a plurality of signal processor **210**, an output signal generator **212**, a bank **213** of digital-to-analog (D/A) converters **215**, a plurality of output ports **214** and one or more local signal generators **224**. Each packetized signal input stage **205** comprises a packetized signal input buffer **204**, a packetized signal extractor **206**. The display devices **116** are coupled to the output ports **214**. The output processor **106** also includes an output processor local controller **216** that receives output processor control signals **121** from the master controller **102** (FIG. **1**). The output processor local controller **216** is coupled to the various components of the output controller **106** through control lines **218** and controls the operation of those components in response to the output processor control signals **121**.

Memory system **208** includes a plurality of data buffers **220**.

The output processor control signals **121** received by the output processor local controller **216** indicate:

i. which signals (by their global identification codes) are required for each output signal **114**; and

ii. the format of each output signal **114** and, if the output signal is a video signal, the layout of the display including the position and dimensions of each window on the display, in accordance with the user control signals (FIG. **1**).

In the present embodiment, the output processor local controller **216** translates each global identification code into a local identification code.

The packetized signal **112** is received at input port **202** and is buffered in packetized stream input buffer **204**. As complete packets **190** are stored in buffer **204**, they are retrieved by packetized signal extractor **206**. The packetized signal extractor **206** determines the global identification code of each packet, translates the global identification code into the corresponding local identification code assigned by output processor local controller and stores packets **190** corresponding to each local identification code in a different data buffer **220**. Through this process, the data from each source signal for the packetized signal is isolated in a different data buffer **220**. Each isolated signal corresponds to one of the packet source signals for the packetized signal. The packet ordering information from each packet **190** is used to organize the packets **190** into their original sequence. Each isolated signal is referred to herein as an output source signal.

The local identification codes are used within the output processor **106** in place of the global identification code to

distinguish between the different local source signals encoded in the packetized signal. In alternative embodiments, the global identification code may be used to identify the different local source signals within the output processor **106**.

The signal processors **210** may be used to reverse any compression or other signal processing operation applied in the input processor **104** (FIG. **2**) using the signal processors **126**. Depending on the signal processing operations performed in the input processor **104**, a reversing step may or may not be required. For example, if one of the input signals **110** was compressed using a standard compression format that may be directly used to produce an output signal **114**, then it is not necessary to reverse the compression. However, if the result of the compression step produced data that cannot be directly used to produce an output signal **114**, then a decompressor may be used to reverse the compression step. For example, one of the signal processor described above was a horizontal line filter, which compresses an input video signal **110** by discarding a portion of the video signal. This compression step may be reversed by interpolating the discarded data from the retained data. The resulting processed signal **222** is stored in a data buffer as an output source signal and is assigned a unique local identification code by the output processor local controller **216**.

In addition to reversing signal processing operations applied in the input processor, a signal processor **210** may be used to apply any other signal processing operations to a signal buffered in a data buffer **220** to produce an output source signal.

Reference is made to FIG. **3**. Graphics display window **178** on video monitor **116***a* contains a display of the current date, time and warning messages based on metadata extracted from the input packetized signal received with global identification code G**111**. The date and time information is generated by a local signal generator **224***a* , which operates under the control of the output processor local controller. Each local signal generator **224** produces an output source signal containing information and formatted for the use in an output signal. In this example, the local signal generator **224***a* generates a 200×150 pixel window containing the date and time. The output processor **106** may include other local signal generators **224** that produce other video, audio or data signals for inclusion in the output signals. In each case, the signal produced by the local signal generator is stored in a separate data buffer **220** and is assigned a local identification code. The metadata maybe generated using a signal analyzer among the signal processors **210**. The metadata produced by the signal analyzer is combined with the data and time information by a video signal generator in the output signal generator **212**. Similar metadata could also be generated in the input processor **104** (FIG. **1**) using a signal processor **126**.

The output signal generator **212** can generate a variety of digital output signals that may be used directly, or after conversion through a D/A converter **215**, by output device **116**. The output signal generator **212** may include one or more digital video signal generators, one or more digital audio signal generators or one or more data signal generators or any combination of video, audio and data signal generators. The data signal generators may include TCP/IP signal generators that produce an output signal **114** suitable for transmission using a communications link to a remote computer system, where the output signal may be decoded and used by a video, audio or data system. Similarly, the data signal generator may generate signals in any data format.

The output signal generator **212** extracts the data required for each output signal **114** from the appropriate data buffers **220** and generates the output signal **114**. For example, a video output signal generator receives instructions from the output processor local controller **216** identifying the output source signals (by their local identification code and the data buffer **220** in which they are buffered) required for an output signal, the layout of the output video signal in terms of the position and dimensions of each window, and the output source signal for each window. The video output signal generator extracts the video information for each frame from the corresponding data buffers **220** and generates each frame for the output signal **114**. If the video signal includes audio components, these audio components are similarly retrieved as output source signals and added to the output video signal **114**.

Similarly, the audio and data output signal generators retrieve the output source signals from the appropriate buffers and produce their output signals.

If the device coupled to a particular output port **214** requires an analog output signal, then one of the D/A converters **215** may be dynamically coupled between the output signal generator and the output port **214** to convert the digital output signal into a corresponding analog output signal.

To produce the video signals for the example output video monitors **116***a*(FIGS. **3**) and **116***b* (FIG. **4**), the output processor local controller configures the output processor **106** to operate as follows:

i. Packetized signal extractor **206** operates as follows:
   a. Extract signal G**105** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**201** and store it as an output source signal in data buffer **220***a*;
   b. Extract signal G**106** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**202** and store it as an output source signal in data buffer **220***b*;
   c. Extract signal G**107** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**203** and store it as an output source signal in data buffer **220***c*;
   d. Extract signal G**108** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**204** and store it as an output source signal in data buffer **220***d*;
   e. Extract signal G**109** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**205** and store it as an output source signal in data buffer **220***e*;
   f. Extract signal G**110** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**206** and store it as an output source signal in data buffer **220***f*; and
   g. Extract signal G**112** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**207** and store it as an output source signal in data buffer **220***g*.

ii. Local signal generator **224** produces a 200×150 pixel data and time window as described above. Assign local identification code B**208** to this signal and store it as an output source signal in data buffer **220**h.

iii. Output signal generator **212** generates two output signals as follows:
   a. One output video signal generator **212***a* extracts local signals B**201**, B**203**, B**205**, B**207** and B**208** from the corresponding data buffers **220** and produces an output signal **114***a*.
   b. A second output video signal generator **212***b* extracts local signals B**202** B**204** and B**206** from the corresponding data buffers **220** and produces an output signal **114***b*.

iv. A D/A converter is coupled between video signal generator **212***a*and output terminal to convert output signal **114***a* into an analog output signal, which is then displayed by video monitor **116***a*.

v. Output signal **114***b* is coupled directly to output port **214***b*. Video monitor **116***b* receives and displays the digital output signal **114***b*.

Referring to FIG. **1**, the input processor **104** receives a plurality of different input signals **110**, which are asynchronous with respect to one another to be received at the input processor. The input signals are processed using signal processor **126** to put them into a format that is required for the output signals **114** and resulting processed signal (the packet source signals) are combined into a single packetized signal **112**. If an input signal **110** does not require any processing to be used as part of an output signal, the input signal **110** may be a packet source signal. The input processor allows a plurality of asynchronous data signals **110**, which may include video, audio and data signals, to be combined into a single packetized signal that may be transmitted using a single communication link **186**.

The output processor **106** receives the packetized signal **112** and isolates the different packet source signals and stores them in buffers **220** as output source signals. Local signal processor **210** in the output processor **106** may be used to reverse any signal processing operation performed in the input processor, if necessary or desired, to produce the output source signals. In addition, local signal generators **224** in the output processor **106** may be used to produce additional output source signals. One or more of the output source signals is used by a set of output signal generators **212** produce output signals **214**. If necessary, a D/A converter may dynamically be coupled between an output signal generator and an output port to convert the corresponding output signal into an analog form.

Together, the input processor **104** and output processor **106** allow a plurality of input signals to be transported from the input ports **108**, combined in a manner controlled by a user through the user controller **118** and then provided in the final combined manner to the output devices **116**. The input processor **104** and output processor **106** are coupled together using a single communication link **186**, eliminating the need to couple each of the input signals separately to the output processor **106**.

In the embodiment of FIGS. **1** to **6**, the input processor **104** includes video scalers **160** to scale video input signals **110** from their original dimension to other dimensions required for the output signals **114**. In some cases, this may required that the input video signal may be expanded to large dimensions, resulting in a packet source signal that requires a larger portion of the packetized signal bandwidth to transmit than the original input signal **110**. To reduce this increased usage of bandwidth, another embodiment of the invention may be configured to ensure that the scalers **160** in the input processor **104** are only used to reduce an input signal **110** to smaller dimensions. Video scalers may be included in the output processor as signal processors **210** to scale any input signal that must be enlarged before it is incorporated into an output signal.

Reference is next made to FIG. **7**, which illustrates a second input processor **304**. Input processor **304** is similar to input processor **104** (FIG. **2**) and similar components are given similar reference numbers. The input processor local controller **340** is coupled to the various components of the input processor **304**. These couplings are not illustrated to simplify the Figure. Input processor **304** has a plurality of packetized signal output stages **327**, each of which comprises a packetized signal formatter **328**, packetized signal buffer **330** and packetized signal generator **332**. Each packetized signal output stage **327** is capable of generating a packetized signal **112**. Each packetized signal **112** may include information of any one or more of the input signals **110**. Input processor **304** may be used to provide packetized signals to different output processors **106** (FIG. **7**). Each output processor can receive a packetized signal containing only information from packet source signals that are required to produce the output signals **114** produced by that specific output processor.

The number of packet source signals (which are generally different versions of input signals **110**) that can be transmitted in a single packetized signal may be limited by the amount of data in each signal and the bandwidth of the packetized signal. Particularly in the case of audio and video signals, which may be required to be received in real time at the output processor **106** in order to be properly displayed on an output device **116**. Input processor **304** allows each input source **108** to be coupled to a single input port on a single input processor and then be combined in different combinations for transmission to different output processors **106**. In one embodiment, an input processor includes four output stages to provide four packetized signals **112**, which may be coupled to four different output processors **106**.

Reference is next made to FIG. **8**, which illustrates a second output processor **403**. Output processor **403** is similar to output processor **106** (FIG. **6**) and similar components are identified with similar reference numbers. Output processor **403** has a plurality of packetized signal input stages **405**, each of which comprises a packetized signal input buffer **404** and a packetized signal extractor **406**. Each input stage **405** receives a packetized signal **112** at a packetized signal input port **202** and stores the data for each source signal for each packetized signal in a separate data buffer in memory system **208**. This allows output processor **403** to receive a larger number of source signal than could be transmitted in a single packetized signal. Output processor **403** operates in the same manner to further process and generate output signals **114**, which may incorporate data from one or both of the packetized signals.

Reference is next made to FIG. **9**, which illustrates three input processors **304** and two output processors **403**. Input processor **304***a*receives eight input signals from eight sources **108***a*-**108***h* and generates two packetized signals **112***a* and **112***b*. Input processor **304***b* receives eight input signals from eight sources **108***i*-**108***p* and generates two packetized signals **112***c* and **112***d*. Input processor **304***c* receives eight input signals **108***q*- **108***x* and generates one packetized signal **112***e*. Output processor **403***a*receives packetized signals **112***a* and **112***c* and produces four output signals **114***a*-**114***d* at output terminals **714***a*-**714***d*. These output signals may include information from any of the sixteen input signals **108***a*-**108***p*. Output processor **403***b* receives packetized signals **112***b*, **112***d* and **112***e* and produces four output signals **114***e*-**114***h* at terminals **714***d*-**714***h*. The output signals **114***a*-**114***h* may include information from any of the twenty-four input signals **108***a*-**108***x*. In each case, each input source is coupled to only one input processor, but may be combined with the other input sources in the output signals.

A single packetized signal **112** produced by a input processor **104** or **304** may be coupled to more than one output processor by first routing the packetized signal **112** through a signal replicating device. For example, the packetized signal **112** may be replicated using a cable driver with multiple

duplicate outputs or other signal replication device and transmitted on multiple communications links to more than one output processor.

Reference is next made to FIG. **10**, which illustrates a first packet router **502** coupled between a plurality of input processors **104** and **304** and a plurality of output processors **403**. Each of the input processors produces one or more packetized signals **112** that are received by the packet router **502**. Packet router **502** includes a packet router controller **504**, a plurality of packetized signal buffers **506**, a plurality of packetized signal extractors **508**, a plurality of data buffers **510** and a plurality of packetized link output stages. Packet router controller **504** controls the operation of packet router **502** through control lines **507**, which couple the packet router controller **504** to the other elements of packet router **502** (connections are not shown to simplify the Figure). Each packetized signal **112** is buffered in a packetized signal buffer **506**. As complete packets **190** of a packetized signal **112** arrive, a packetized signal extractor **508** determines the global identification code of each packet **190** and stores all packets **190** corresponding to the same global identification code in a single data buffer **510**. The packetized signal extractor operates under the control of the packet router controller, which designates the particular data buffer in which the packets **190** having the same global identification code are stored. Through this process, all packets having the same global identification code are isolated in a data buffer. The actual content of the packets **190** is not altered.

Each packetized signal output stage **527** includes a packet selector **528**, a packetized signal buffer **530** and a packetized signal generator **532**. The packet selector **528** operates under the control of the packet router controller **504** to extract packets **190** from one or more of the data buffers **510** and place them in packetized signal buffer **530**. The packetized router controller receives packet router control instructions **520** from the master controller **102** to generate one or more packetized signals containing corresponding to a set of specified global identification codes. For each requested packetized signal, the packet router controller instructs the packet selector **528** in one of the packetized signal output stages **527** to extract packets from the data buffers **510** corresponding to the specified global identification codes for that requested packetized signal. As the packets become available in the data buffers **510**, the packet selector **528** extracts them and stores them in the packetized signal buffer **530**. Packetized signal generator **532** operates in the same manner as packetized signal generator **132** to generate a new packetized signal **512**.

Each packetized signal output stage **527** operates independently of the others. Any number of packetized signals **512** generated by the packetized signal output stages may include packets from the same data buffer **510** (corresponding to a particular global identification code). Each data buffer is operated to ensure that each packet in the data buffer are not discarded until each packet has been read by every packetized signal output stage that requires the packet.

Through this operation, the packet router receives a plurality of packetized signals **112** and generates a new set of packetized signals **512**. The created packetized signals may comprise packets with any combination of global identification codes, allowing input signals received at different input processors to be combined in a single packetized signal **512** for delivery to an output processor **403**. Each output processor may receive multiple packetized signals **512** from a packet router **502** and may also receive one or more packetized signals **112** directly from an input processor.

Inserting packet router **502** between a plurality of input processors **104** and **304** and output processors**106** and **403** allows an input signal **110** received at any one of the input processors to be routed (possibly after being processed in the input processor by signal processor **126**) to any of the output processors for use by any of the output devices **116** coupled to an output processor. Each input signal is received in only one location, but may be used in multiple formats (by creating appropriate versions of the input signal using signal processor **126**) at multiple output devices **116**.

Reference is next made to FIG. **11**, which illustrates a second packet router **602**. Like packet router **502**, packet router **602** receives one or more packetized signals **112** from one or more input processors, and provides one or more packetized signals **612** to one or more output processors. Packet router **602** includes a packet router controller **604**, a plurality of packetized signal buffers **606**, a plurality of packetized signal extractors **608**, a memory system **609** including a plurality of packet storage locations **610** and a plurality of packetized signal output stages **627**. Packet router controller **604** controls the operation of packet router **602** through control lines **607**. Packet router controller **604** uses the packet storage locations **610** to temporarily store packets **190** from the packetized signal **112**.

Each packetized signal **112** is buffered in a packetized signal buffer **606**. As each complete packet **190** of a packetized signal **112** arrives, a packetized signal extractor **608** stores the complete packet **190** in one of the packet storage locations **610**. The packet router controller **604** maintains a storage location table **611** indicating whether each packet storage location **610** is available to store a newly arrived packet. The packet router controller **604** selects an available packet storage location **610** and instructs the packetized signal extractor to store the newly arrived packet **190** in the selected packet storage location **610**. The packet router controller **604** then updates the storage location table **611** to indicate that the packet storage location **610** is not available to store another packet **190**.

The packet router controller **604** receives router control instructions **620** (similar to the router control instructions **520** received by packet router controller **502** (FIG. **10**)) from master controller **102** instructing the packet router controller to generate the packetized links **612** using packets **190** with specified global identification codes. The packet router controller **604** determines and assigns one of the packetized signal output stages **627** to generate each of the required packetized signals **612** and maintains a global identification code distribution table **613** correlating each global identification code with the packetized signal output stages **627** that require the global identification code. For example, a specified global identification code G**603** may be required for three of the outgoing packetized signals **612**. The three packetized signal output stage **627** used to generate those three packetized signals **612** are listed in the global identification code distribution table **613** in association with global identification code G**603**.

Each packetized signal output stage **627** includes a packet selector **628**, a packetized signal buffer **630** and a packetized signal generator **632**. Packet selector **628** reads packets **190** from the packet storage locations **610** as described below and stores the packets **190** in packetized signal buffer **630**. Packetized signal buffer **630** and packetized signal generator **632** operate in the same manner as packetized signal buffer **130** (FIG. **2**) and packetized signal generator **132** to produce the packets signals **612**.

In the storage location table **611**, the router controller **604** maintains the status of each packet storage location **610** by recording the number of packetized link output stages **627** that must read a packet **190** stored in the packet storage

location before the packet **190** may be discarded. In the present embodiment, when a newly received packet **190** is stored in a free packet storage location **610**, the router controller records the number of packetized signal output stages **627** that require the packet **190** to generate a packetized signal **612**. The router controller **604** then instructs each of the packetized signal output stages **627** to read the packet **190** from the packet storage location **610**. The packet selector **628** in each packetized signal output stage reads the packet from the packet storage location **610** and indicates to the router controller **604** that it has done so. The router controller **604** than decrements the number of packetized signal output stages **627** that still require the packet in that packet storage location **610**. When each of the packetized signal output stages **627** that require the packet **190** have indicated that they have read the packet **190**, the packet is no longer required (i.e. the number of packetized signal output stages still requiring the packet **190** is zero), and the router controller **604** treats the packet storage location **610** as free.

For example, each packet **190** with global identification code G**603** may be required by packetized signal output stages **627***a*, **627***c* and **627***d* to produce outgoing packetized signal **612***a*, **612***c* and **612***d*. When a complete packet **190** with global identification code G**603** is received, router controller **604** selects a free packet storage location **610**b and instructs the appropriate packetized signal extractor **608** to store the packet **190** in packet storage location **610***b*. The router controller **604** then sets the status of packet storage location **610***b* to "**3**", indicating that the packet **190** must still be read by three packetized signal output stages. The router controller than instructs packetized signal output stages **627***a*, **627***c* and **627***d* to read the packet **190**. Each of packet selectors **628***a*, **628***c* and **628***d* reads the packet **190** and indicates to router controller **604** that it has done so. Router controller **604** decrements the status of the packet storage location **610***b* as it receives each indication and when the status returns to "0", the packet storage location **610***b* is again free to store another packet **190**.

Reference is next made to FIG. **12**, which illustrates a third packet router **702**. Packet router **702** is similar in structure and operation to packet router **602** and similar components are identified with similar reference numbers, increase by one hundred. Packet router **702** has a plurality of signals processors **726** coupled to memory system **709**. Signal processor **726** operate in a similar manner to signal processors **210** (FIG. **6**) under the control of router controller **704**. Router controller **704** receives instructions from master controller **102** to perform one or more signal processing steps on the signal encoded with a particular global identification code. For example, the master controller may indicate that a video signal with global identification code G**734** must be scaled to dimensions of 800×600 pixels and the resulting processed signal is to be assigned global identification code G**783** and must be included in packetized signals **712***b* and **712***c*. Router controller **704** then configures the global identification code distribution table **713** to route packets with global identification code G**734** to a video scaler (not shown) among the signal processors **726**. The router controller **704** may also route the same packet to one or more other signal processors **726** or packetized signal output stages **727**. The video scaler (not shown) is configured to perform the video scaling operation and produces packets **190** identified with global identification code G**783**. These packets are stored in free packet storage locations as designated by the router controller **702**. The packets are then distributed to packetized signal output stages **727***b* and **727***c*using global identification code distribution table **713** and storage location table **711**.

Reference is next made to FIG. **13**, which illustrates an output stage **827** for an output processor. Output stage **827** may be used for video output signals and includes a buffer **840** coupled to output signal generator **212** and an output terminal **214**. The output signal **114** generated by the output signal generator **212** is stored in data buffer **840**. The stored signal is extracted from the data buffer **840** by a local output generator **842** which makes the output signal **114** available at an output terminal **214**. Optionally a D/A converter may be coupled between the local output generator **842** and output terminal **214** to convert the output signal into a corresponding analog output signal for use by an analog device coupled to terminal **214**.

The buffered stream is also extracted from the data buffer **840** by a remote output generator **844**, which packetizes the video output signal **114** into a graphics packet stream **846**. Each packet in the graphics packet signal **846** contains video data for a fixed number of pixels in the output signal **114**. Each packet has the following fields:

i. packet ordering information, such as video positioning information indicating the first pixel at which the video data is to be displayed; and

ii. the video data.

Each packet may also contain additional metadata including error correction and detection information, frame numbering information and other information.

The graphics packet signal **846** is transmitted to a graphics packet signal ports **848**, from which it may be transmitted across a communication link **850** to display interface **852** capable of receiving the graphics packet signal **846**, reconstructing the output signal **114** and displaying the output signal on a display monitor **116**.

Output stage **827** allows an output signal **114** to be replicated on two different display monitors. The output signal **114** may be replicated on any number of display monitors by providing a remote output generator for each such monitor.

Reference is next made to FIG. **14**, which illustrates a switch **860** coupled between a plurality of graphics packet signal ports **848**, which may be part of one or more output stage **827** in one or more output processors, and a plurality of display interfaces **852**. The switch **860** may be implemented as a physical switch, which may be manually operable or automatically operable under the control of the master controller **102** (not shown in FIG. **12**). The switch **860** may be implemented using a field-programmable gate array (FPGA) or with any other switching or packet routing technology. Switch **860** allows any of the graphics packet signal ports **848** to be coupled to any display interface **852**, allows any of the output signals **114** available at any of the graphics packet signal ports **848** to be displayed at any display monitor coupled to a display adapter **852**.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the spirit and scope of the invention, which is limited only by the appended claims.

We claim:

1. A method of producing a packetized signal comprising:

receiving one or more input signals;

determining which of the one or more input signals and signals derived from the one or more input signals are required to generate the packetized signal;

upon determining the one or more input signals required to generate the packetized signal, buffering the one or more input signals required to generate the packetized signal in a memory system;

upon determining the signals derived from the one or more input signals required to generate the packetized signal, processing at least one of the one or more input signals to provide a corresponding processed signal, wherein the corresponding processed signal is required to generate the packetized signal, and buffering the corresponding processed signal in the memory system;

designating the one or more input signals buffered in the memory system and the corresponding processed signal buffered in the memory system as packet source signals and assigning each of the packet source signals a unique global identification code; and

generating the packetized signal wherein the packetized signal includes a series of packetized signal packets, wherein each of the packetized signal packets contains the unique global identification code of one of the packet source signals and data corresponding to the same packet source signal.

2. The method of claim **1** wherein each of the packet source signals comprises a series of packet source signal packets, and wherein each of the packetized signal packets is formed by retrieving one or more of the packet source signal packets corresponding to a single packet source signal, extracting data from the retrieved one or more of the packet source signal packets, recording the unique global identification code of the single packet source signal and at least a portion of the extracted data in the packetized signal packets.

3. The method of claim **1** wherein each of the packet source signals comprises a series of packet source signal packets, and wherein each of the packetized signal packets is formed by retrieving one or more of the packet source signal packets corresponding to a single packet source signal and including the packet source signal packet within the packetized signal packet.

4. The method of claim **1** wherein each of the packetized signal packets includes a global identification code, packet sequencing information and a data payload.

5. The method of claim **4** wherein at least one of the packet source signals is a video signal and wherein the packetized signal corresponding to the packet source signals includes video data and position information indicating how the video data is to be displayed on a video display.

6. The method of claim **5** wherein the position information includes pixel information indicating a position within a window of the video display at which the video data is to be displayed.

7. The method of claim **1** wherein the processed signal is a scaled version of the at least one of the one or more input signals.

8. The method of claim **1** wherein the processed signal is a compressed version of the at least one of the one or more input signals.

9. A method of producing one or more output signals, the method comprising:

receiving one or more input signals;

determining which of the one or more input signals and signals derived from the one or more input signals are required to generate the packetized signal;

upon determining, buffering the one or more input signals required to generate the packetized signal in an input processor memory system, and

processing at least one of the one or more input signals to provide a processed signal wherein the at least one of the one or more input signals is required to generate the packetized signal, and buffering the processed signal in the input processor memory system;

designating the one or more input signals buffered in the memory system and the processed signal buffered in the memory system as packet source signals and assigning each of the packet source signals a unique global identification code;

generating the packetized signal wherein the packetized signal includes a series of packetized signal packets, wherein each of the packetized signal packets contains the unique global identification code of one of he packet source signals and data corresponding to the same packet source signal;

transmitting the packetized signal across a communications link;

receiving the packetized signal;

extracting each of the packetized signal packets from the packetized signal;

buffering the packetized signal packets containing a same unique global identification code in a separate data buffer in an output processor memory system and designating the packetized signal packets in each separate data buffer as an output source signal; and

producing the one or more output signals by retrieving one or more output source signals and combining the retrieved output source signals.

10. The method of claim **9** wherein each of the packet source signals comprises a series of packet source signal packets, and wherein the each of the packetized signal packets is formed by retrieving one or more of the packet source signal packets corresponding to a single packet source signal, extracting data from the retrieved one or more of the packet source signal packets, recording the unique global identification code of the single packet source signal and at least a portion of the extracted data in the packetized signal packets.

11. The method of claim **10** wherein at least one of the packet source signals is a video signal and wherein the packetized signal includes video data and position information indicating how the video data is to be displayed on a video display.

12. The method of claim **9** wherein the processed signal is a scaled version of the at least one of the one or more input signals.

13. The method of claim **9** wherein the processed signal is a compressed version of the at least one of the one or more input signals.

14. The method of claim **9** wherein the each of the packetized signal packets contains a global identification code.

15. An input processor comprising:

one or more input ports for receiving one or more input signals;

an input processor local controller configured to determine which of the one or more input signals and signals derived from the one or more input signals that are required to generate a packetized signal;

a memory system for buffering the one or more input signals that are required to generate a packetized signal;

one or more signal processors for retrieving at least one of the one or more input signals that are required to generate the packetized signal, for processing the at least one of the one or more input signals to generate processed signals, and for buffering the processed signals in the memory system;

an output port;

a packetized signal output stage for retrieving one or more of the buffered one or more input signals and the buffered processed signals, for designating the retrieved buffered one or more input signals and the buffered

processed signals as packet source signals, for generating the packetized signal, wherein the packetized signal includes a series of packetized signal packets, wherein each of the packetized signal packets contain a unique global identification code of one of the packet source signals and data corresponding to the same packet source signal, and for providing the packetized signal at the output port; and

the input processor local controller further configured to control the operation of the memory system, the one or more signal processors and the packetized signal output stage.

16. The input processor of claim **15** wherein the one or more signal processors include one or more video scalers for processing a scaled version of the input signal as a processed signal.

17. A method of generating one or more outgoing packetized signals, the method comprising:

receiving one or more incoming packetized signals, each of the packetized signals including one or more packetized signal packets, each packetized signal packet identified with a global identification code;

determining which packetized signal packet is required to generate the one or more outgoing packetized signals;

recording one or more packetized signal packets required to generate the one or more outgoing packetized signals in a packet storage location;

recording a number of outgoing packetized signals in which each of the one or more packetized signal packets will be included;

instructing a group of packetized signal output stages to read each of the one or more packetized signal packets, the number of packetized signal output stage corresponding to the recorded number of outgoing packetized signals.

18. The method of claim **17** further comprising a step of processing at least one of the one or more packetized signal packets that is required to generate the one or more outgoing packetized signals to provide a set of processed packets, assigning the set of processed packets a unique global identification code and storing the processed packets in the packet storage location.

19. A packet router comprising:

one or more input stages, each of the input stages configured to receive an incoming packetized signal, determine a unique global identification code of each packetized signal packet extracted from the packetized signal, and store packetized signal packets in a separate buffer in a packet router memory system based on the unique global identification code corresponding to the packetized signal packets of each packetized signal;

one or more output stages configured to read at least one of the packetized signal packets from the memory system to generate an outgoing packetized signal; and

a router controller for controlling the storage of the packetized signal packets in the memory system and the generation of the outgoing packetized signal in response to router control signals received from a master controller.

20. The packet router of claim **19** wherein the memory system includes a plurality of packet storage locations and wherein the router controller includes a storage location table to manage usage of the packet storage locations and a global identification code distribution table to manage distribution of packetized signal packets to particular output stages, and wherein the router controller is configured to instruct the input stages to store each packetized signal packet in a free packet storage location and to instruct each of the particular output stages to read the packetized signal packet from the packet storage location.

21. The packet router of claim **20** wherein the global identification code distribution table identifies the particular output stages to which packetized signal packets having a particular global identification code are distributed.

22. The packet router of claim **20** wherein the storage location table identifies a particular packet storage location as free if no output stage requires a packetized signal packet in the particular packet storage location.

* * * * *

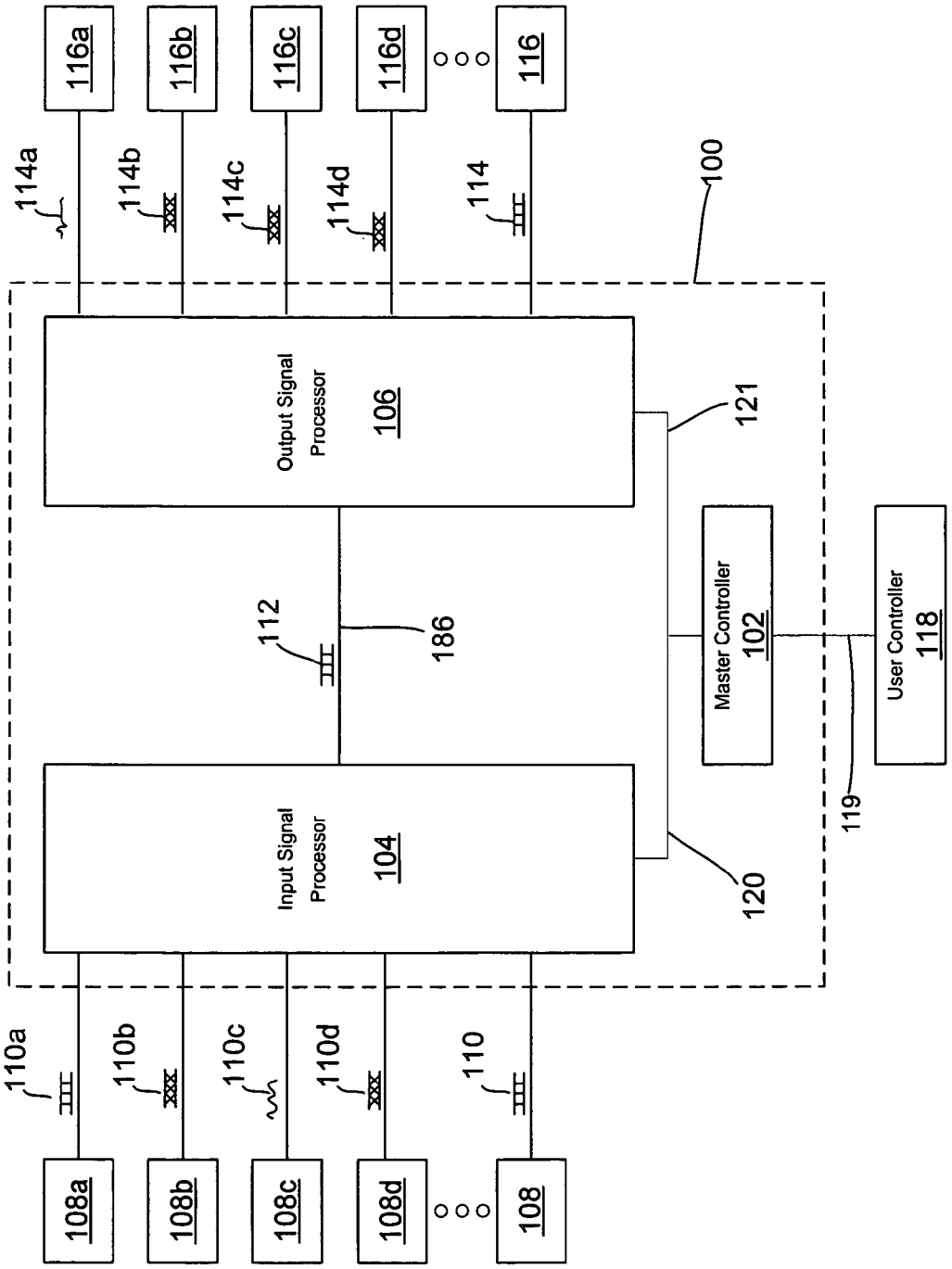

116a
116b
116c
116d
116
114a
114b
114c
114d
114
100
Output Signal Processor
106
121
112
186
Master Controller
102
User Controller
118
119
Input Signal Processor
104
120
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 1

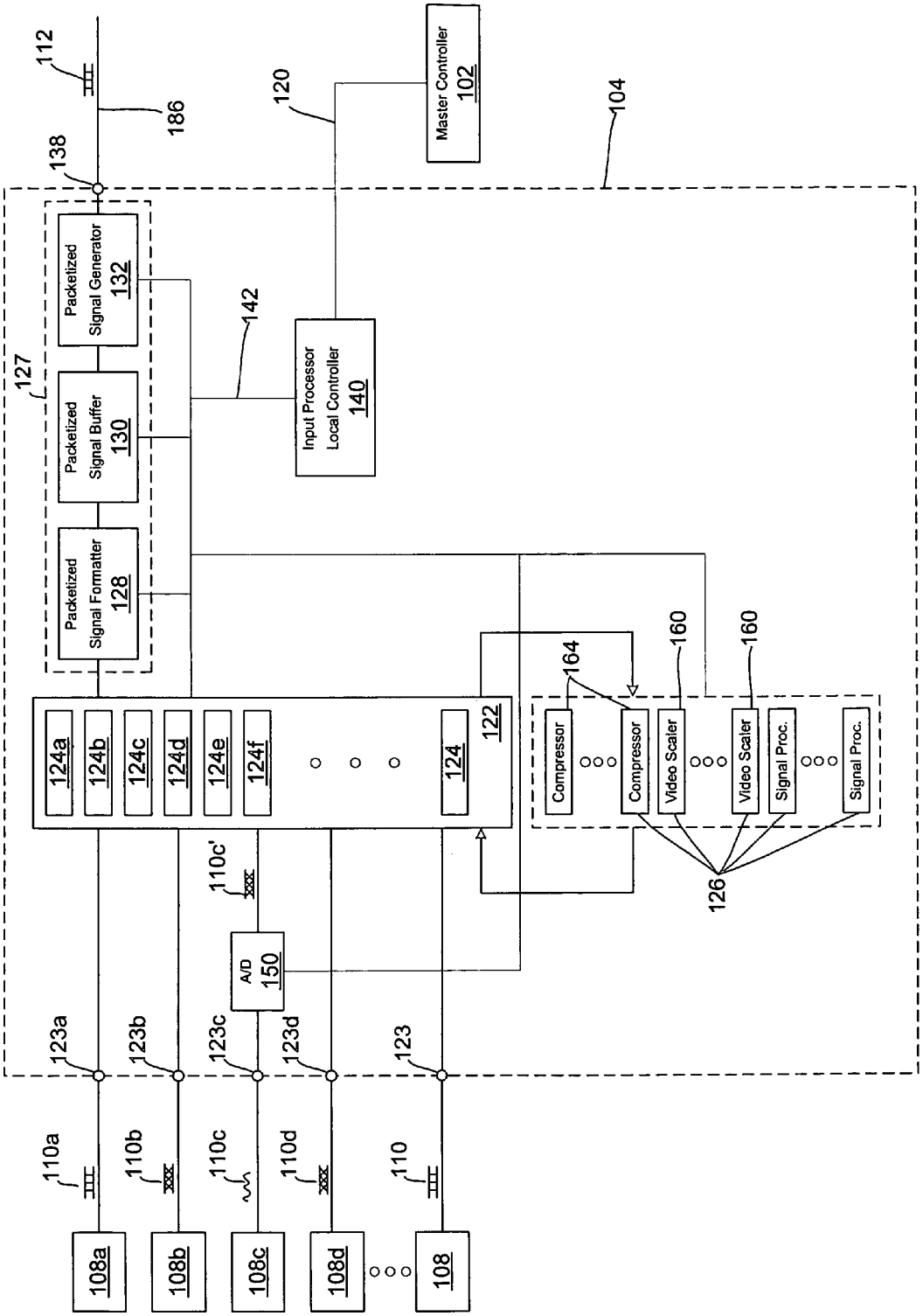

112
186
120
Master Controller 102
104
138
Packetized Signal Generator 132
142
127
Packetized Signal Buffer 130
Input Processor Local Controller 140
Packetized Signal Formatter 128
124a
124b
124c
124d
124e
124f
124
122
164
160
160
Compressor
Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
126
110c'
A/D 150
123a
123b
123c
123d
123
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 2

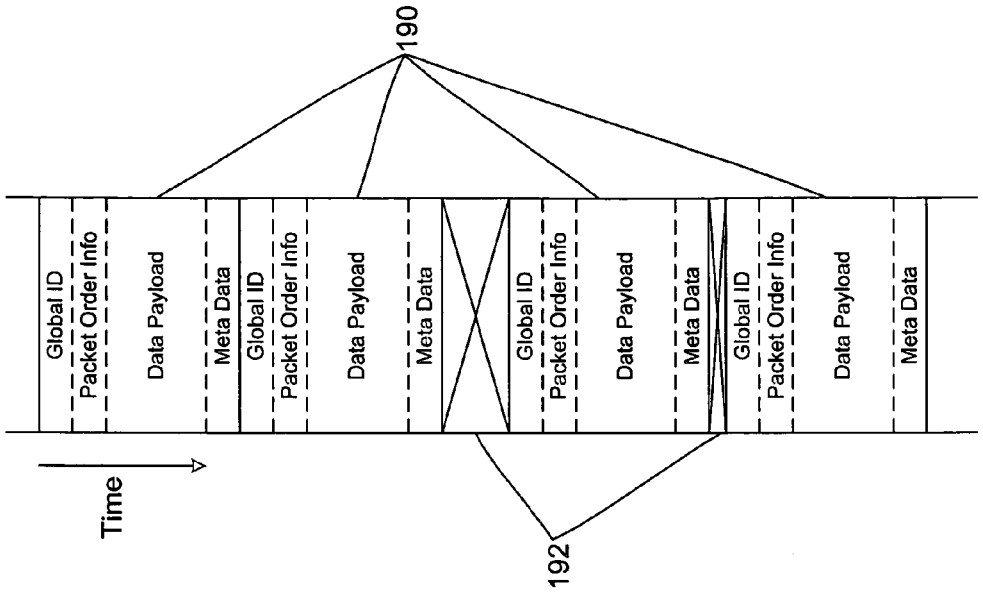


Figure 5

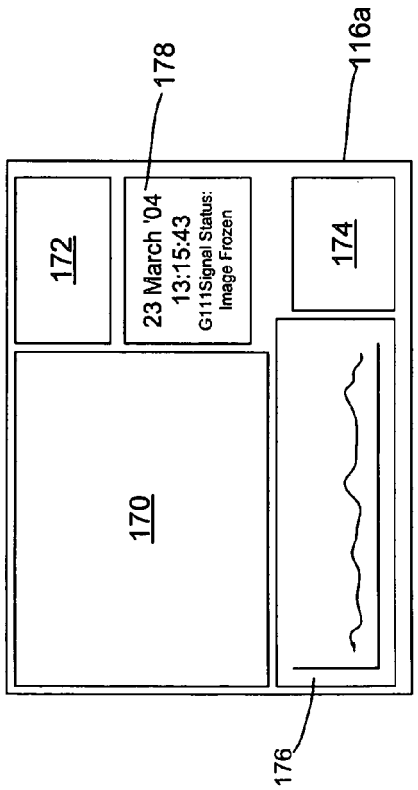


Figure 3

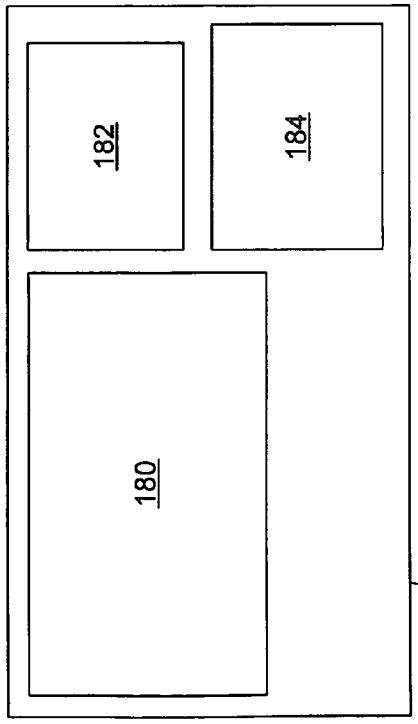


Figure 4

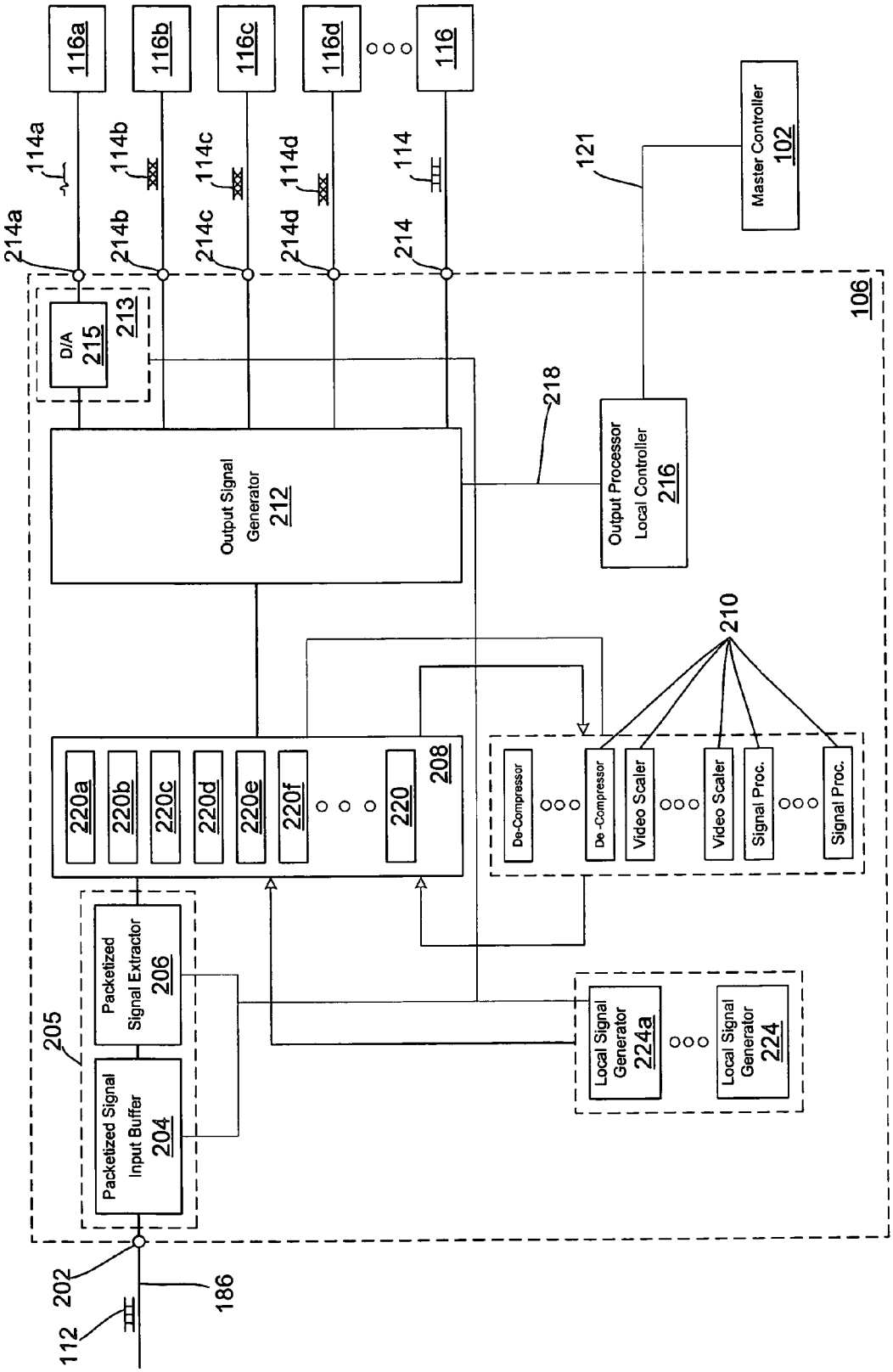

116a
116b
116c
116d
116
114a
114b
114c
114d
114
214a
214b
214c
214d
214
121
Master Controller
102
106
D/A
215
213
218
Output Signal Generator
212
Output Processor Local Controller
216
210
220a
220b
220c
220d
220e
220f
220
208
De-Compressor
De -Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
205
Packetized Signal Extractor
206
Packetized Signal Input Buffer
204
Local Signal Generator
224a
Local Signal Generator
224
202
112
186

Figure 6

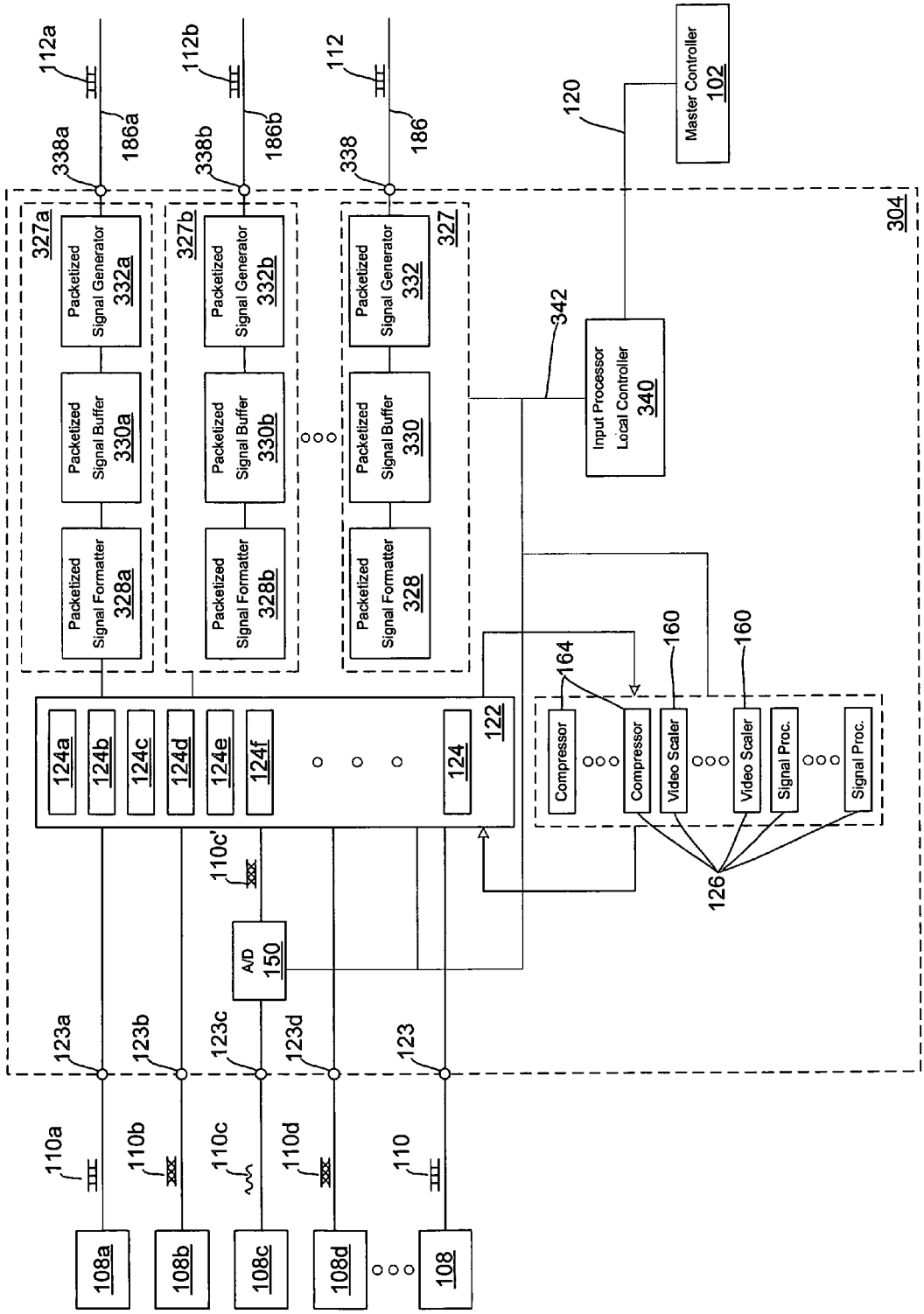

112a
112b
112
186a
186b
186
338a
338b
338
120
Master Controller 102
304
327a
327b
327
Packetized Signal Generator 332a
Packetized Signal Generator 332b
Packetized Signal Generator 332
Packetized Signal Buffer 330a
Packetized Signal Buffer 330b
Packetized Signal Buffer 330
Packetized Signal Formatter 328a
Packetized Signal Formatter 328b
Packetized Signal Formatter 328
342
Input Processor Local Controller 340
124a
124b
124c
124d
124e
124f
124
122
164
160
160
Compressor
Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
126
110c'
A/D 150
123a
123b
123c
123d
123
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 7

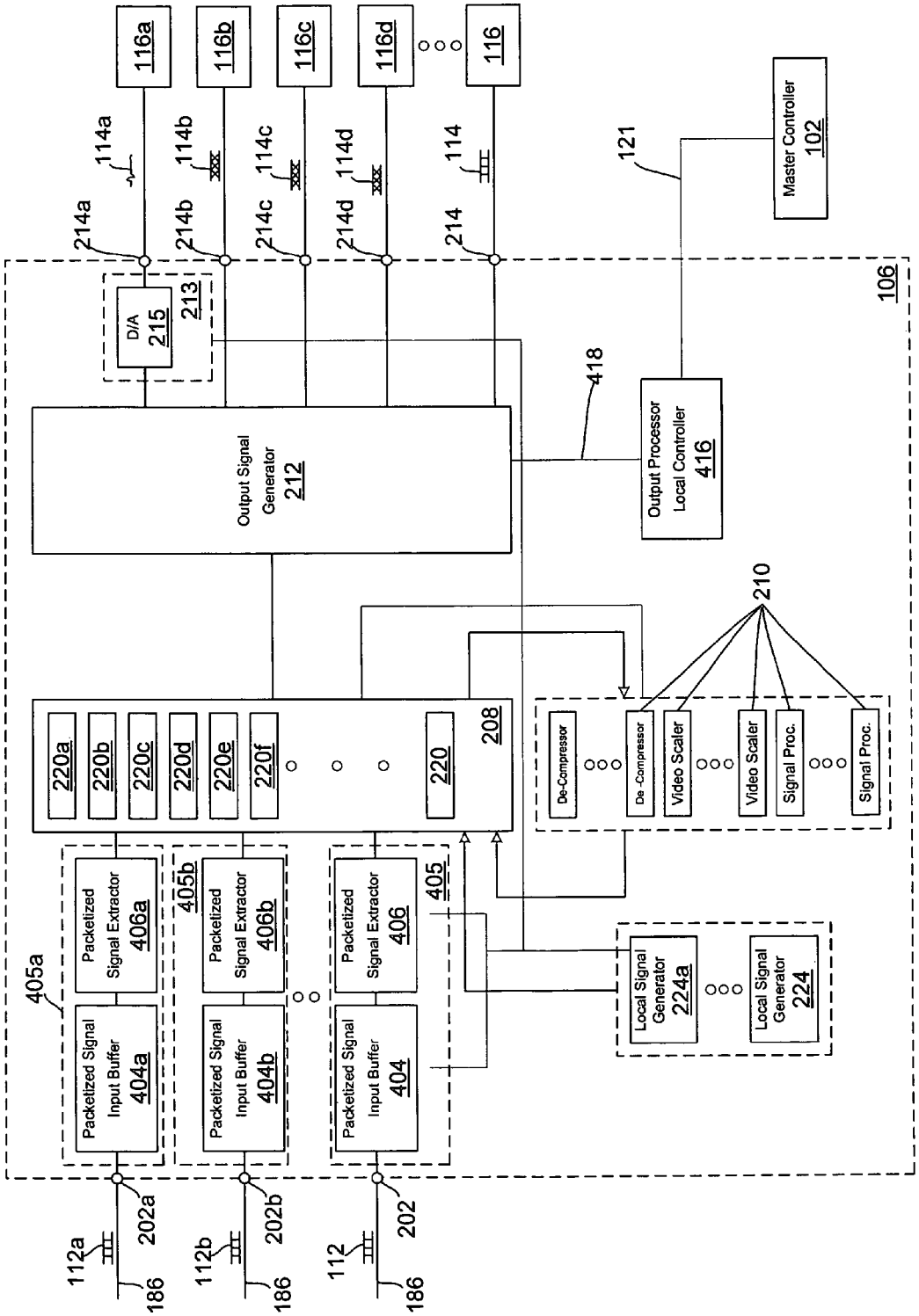

116a
116b
116c
116d
116
114a
114b
114c
114d
114
214a
214b
214c
214d
214
121
Master Controller 102
106
D/A 215
213
418
Output Signal Generator 212
Output Processor Local Controller 416
210
220a
220b
220c
220d
220e
220f
220
208
De-Compressor
De -Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
405a
405b
405
Packetized Signal Extractor 406a
Packetized Signal Extractor 406b
Packetized Signal Extractor 406
Packetized Signal Input Buffer 404a
Packetized Signal Input Buffer 404b
Packetized Signal Input Buffer 404
Local Signal Generator 224a
Local Signal Generator 224
112a
112b
112
202a
202b
202
186
186
186

Figure 8

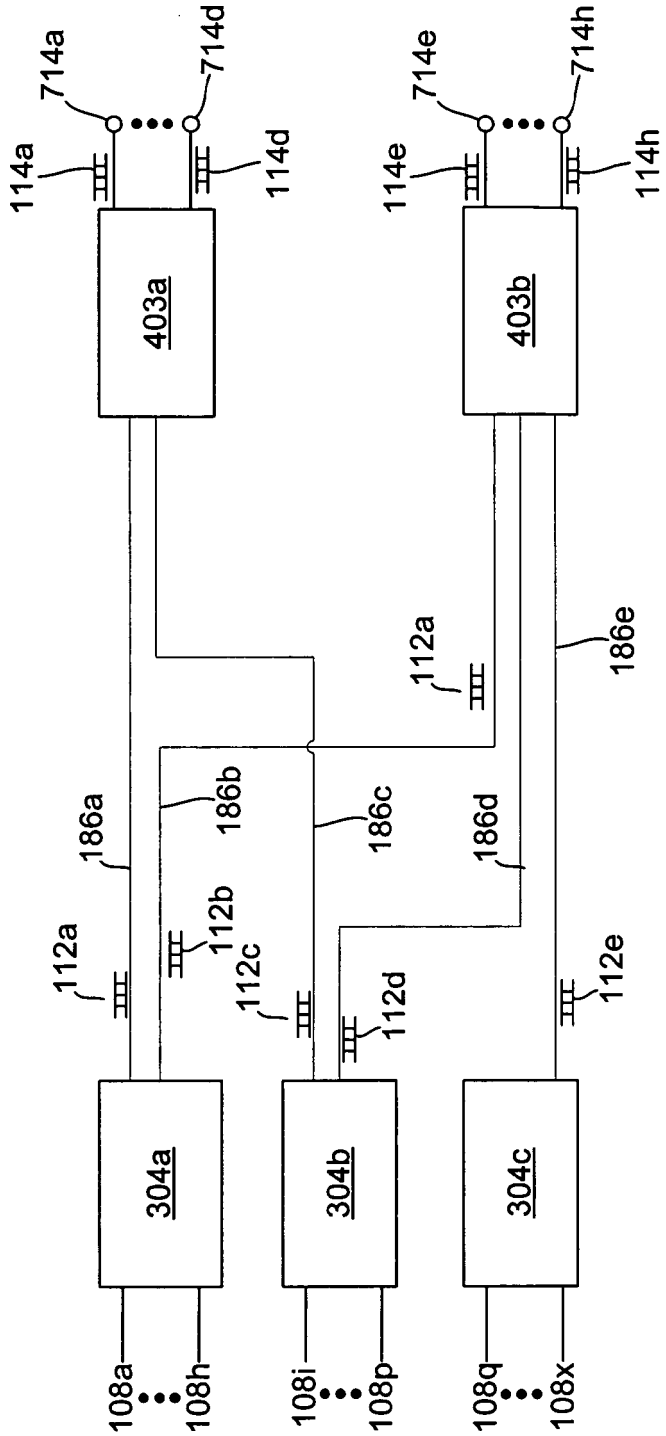

714a
714d
714e
714h
114a
114d
114e
114h
403a
403b
112a
186e
186a
186b
186c
186d
112b
112c
112d
112e
304a
304b
304c
108a
108h
108i
108p
108q
108x

Figure 9

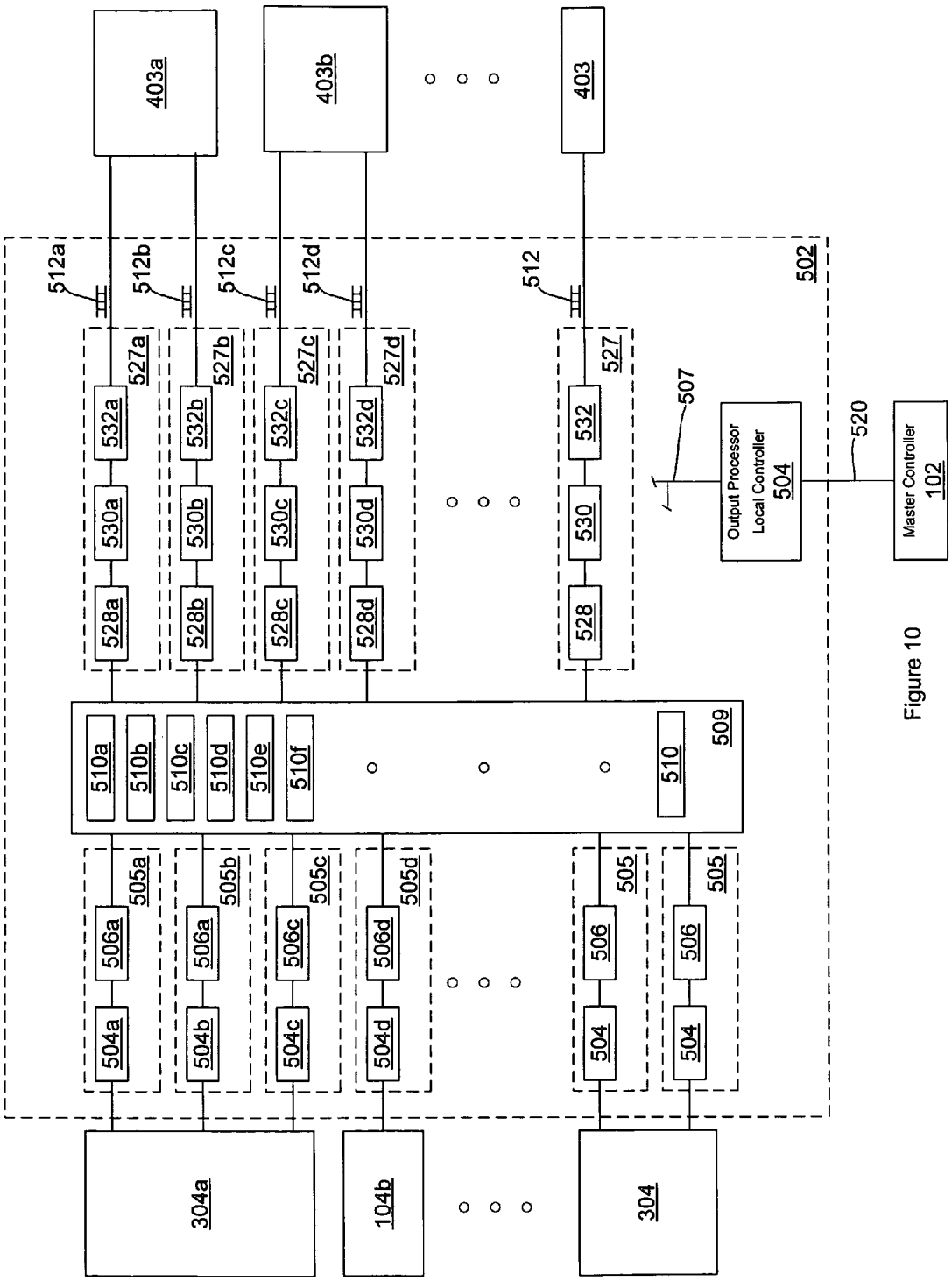

403a
403b
403
512a
512b
512c
512d
512
502
527a
527b
527c
527d
527
532a
532b
532c
532d
532
530a
530b
530c
530d
530
528a
528b
528c
528d
528
507
Output Processor
Local Controller
504
520
Master Controller
102
510a
510b
510c
510d
510e
510f
510
509
505a
505b
505c
505d
505
506a
506b
506c
506d
506
504a
504b
504c
504d
504
304a
104b
304

Figure 10

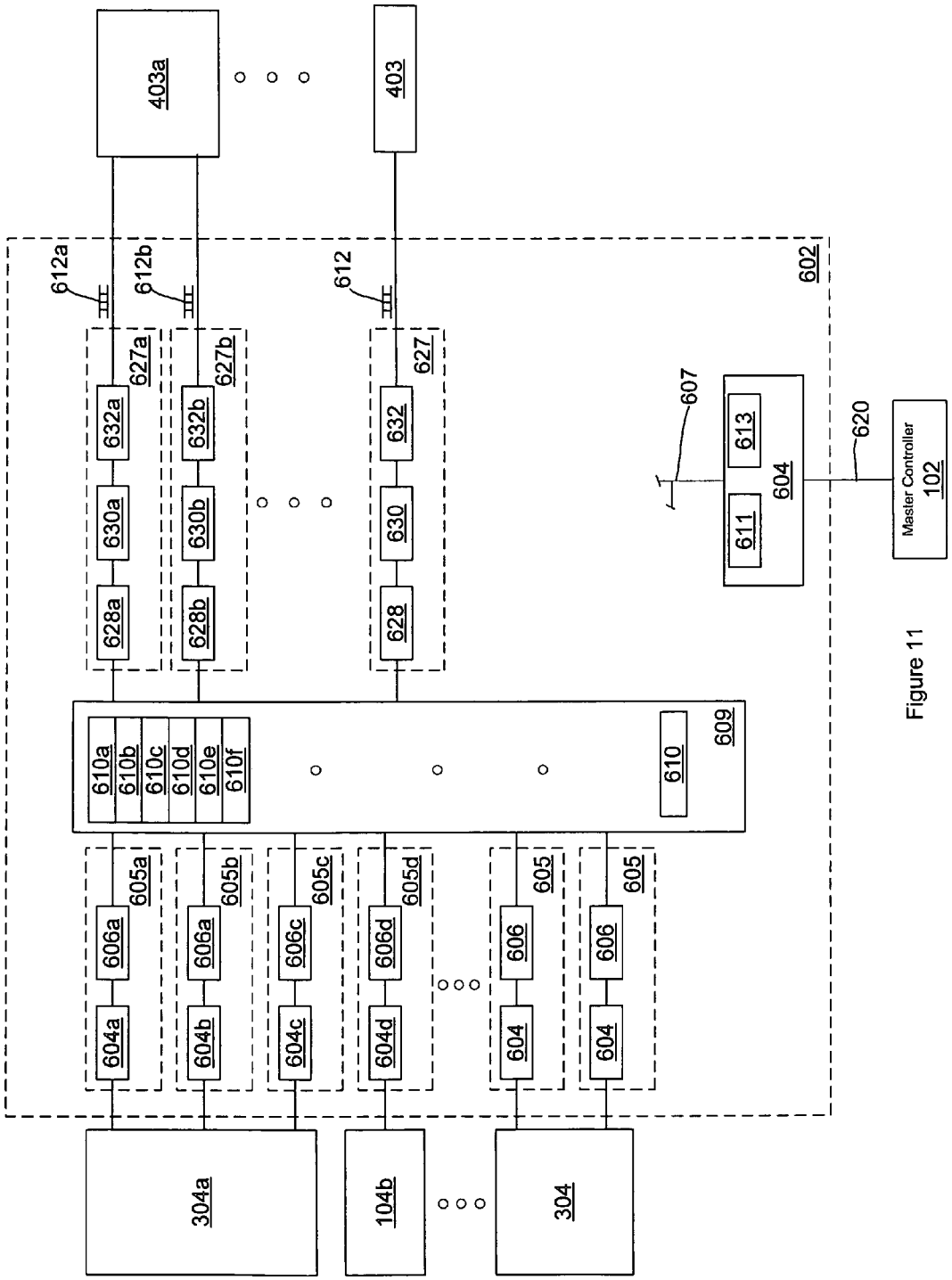

403a
403
612a
612b
612
602
627a
627b
627
632a
632b
632
630a
630b
630
628a
628b
628
607
613
611
604
620
Master Controller
102
610a
610b
610c
610d
610e
610f
610
609
605a
605b
605c
605d
605
606a
606c
606d
606
604a
604b
604c
604d
604
304a
104b
304

Figure 11

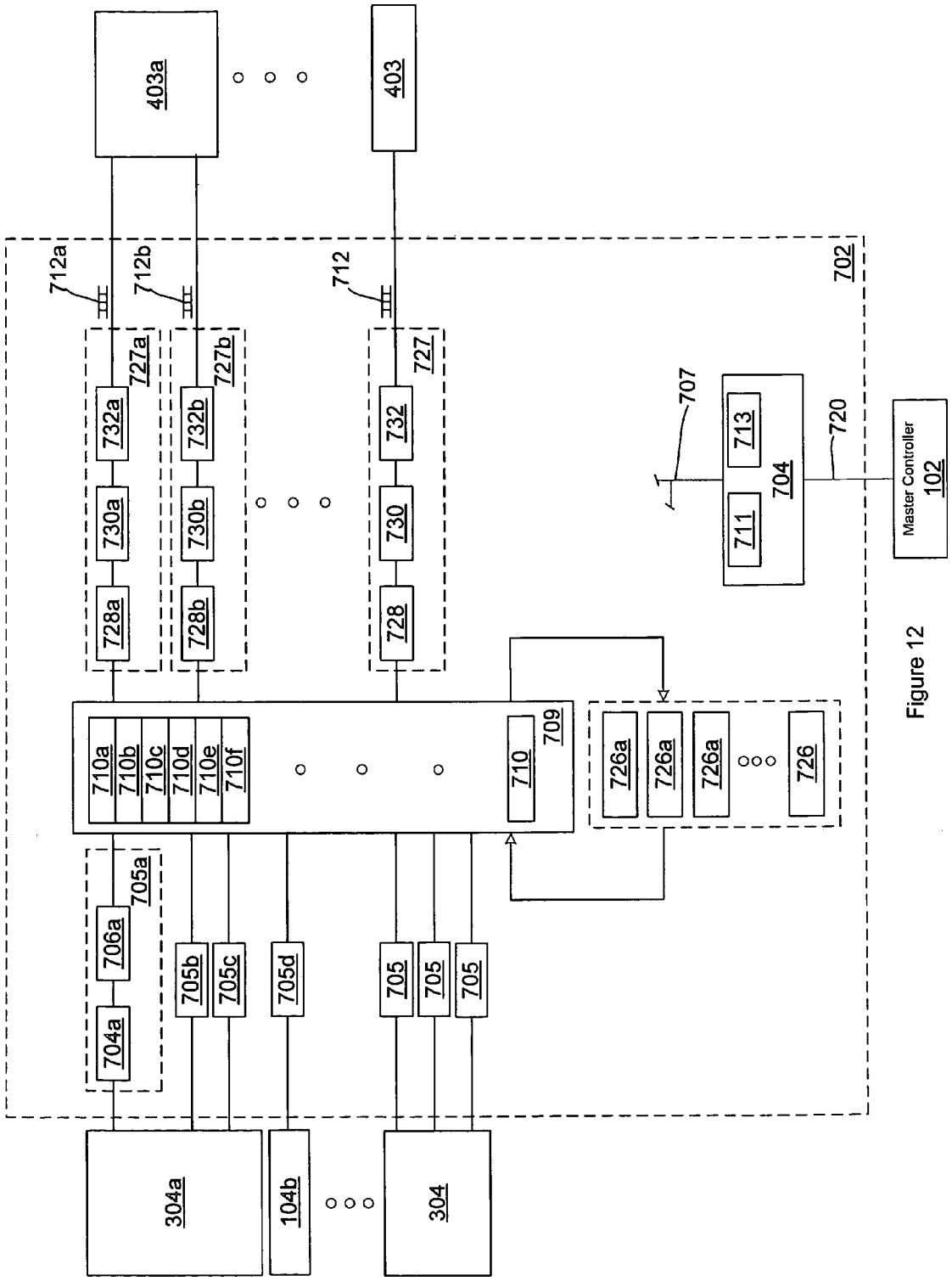

403a
403
712a
712b
712
702
727a
727b
727
732a
732b
732
730a
730b
730
728a
728b
728
707
713
711
704
720
Master Controller
102
710a
710b
710c
710d
710e
710f
710
709
726a
726a
726a
726
705a
706a
704a
705b
705c
705d
705
705
705
304a
104b
304

Figure 12

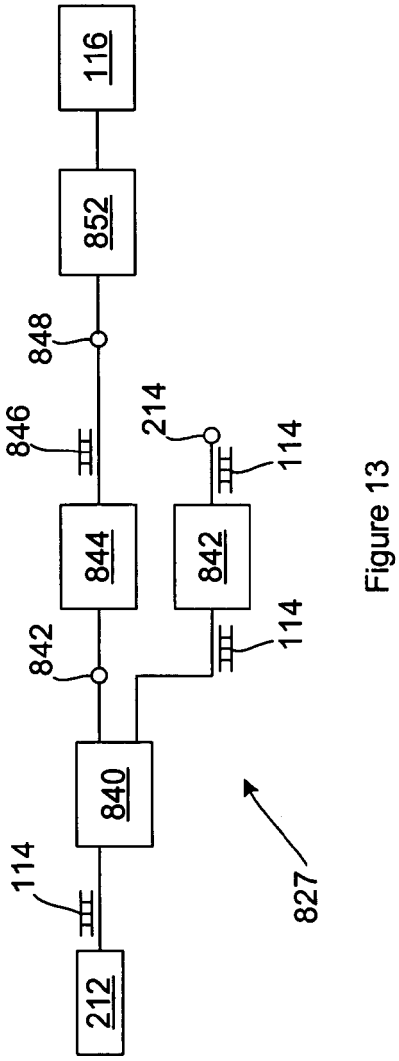


Figure 13

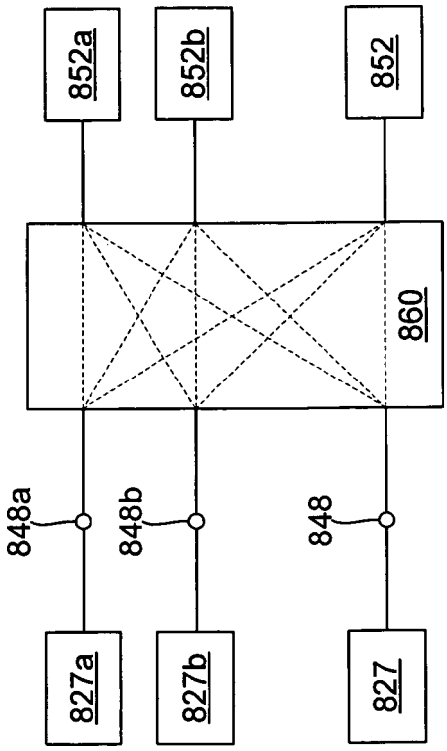


Figure 14

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 9,100,217 B2
APPLICATION NO. : 14/026039
DATED : August 4, 2015
INVENTOR(S) : Rakesh Patel et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the claims,
Claim 9, column 24, line 9, "he packet source signals" should read --the packet source signals--

Signed and Sealed this
Twenty-ninth Day of December, 2015

Michelle K. Lee

Michelle K. Lee
Director of the United States Patent and Trademark Office

EXHIBIT C

US009473322B2

(12) **United States Patent**
Patel et al.

(10) **Patent No.: US 9,473,322 B2**
(45) **Date of Patent: Oct. 18, 2016**

(54) **APPARATUS, SYSTEMS AND METHODS FOR PACKET BASED TRANSMISSION OF MULTIPLE DATA SIGNALS**

(71) Applicant: **Evertz Microsystems Ltd.**, Burlington (CA)

(72) Inventors: **Rakesh Patel**, Mississauga (CA); **Romolo Magarelli**, Kleinburg (CA)

(73) Assignee: **Evertz Microsystems Ltd.**, Burlington (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/753,424**

(22) Filed: **Jun. 29, 2015**

(65) **Prior Publication Data**

US 2015/0304128 A1 Oct. 22, 2015

Related U.S. Application Data

(63) Continuation of application No. 14/026,039, filed on Sep. 13, 2013, now Pat. No. 9,100,217, which is a continuation of application No. 10/816,841, filed on Apr. 5, 2004, now Pat. No. 8,537,838.

(60) Provisional application No. 60/459,964, filed on Apr. 4, 2003.

(51) **Int. Cl.**

H04L 12/28	(2006.01)
H04L 12/54	(2013.01)
H04L 29/06	(2006.01)

(Continued)

(52) **U.S. Cl.**

CPC ***H04L 12/56*** (2013.01); ***H04L 29/06027*** (2013.01); ***H04L 65/601*** (2013.01); ***H04L 65/607*** (2013.01); ***H04N 7/52*** (2013.01); ***H04N 21/226*** (2013.01); ***H04N 21/234*** (2013.01); ***H04N 21/236*** (2013.01)

(58) **Field of Classification Search**

USPC 370/412–418, 422–424, 428, 429; 455/132, 133, 137

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,682,195 A 10/1997 Hendricks et al.
6,430,183 B1 8/2002 Satran et al.

(Continued)

FOREIGN PATENT DOCUMENTS

FR 2655451 A1 6/1991
FR 2726723 A1 5/1996

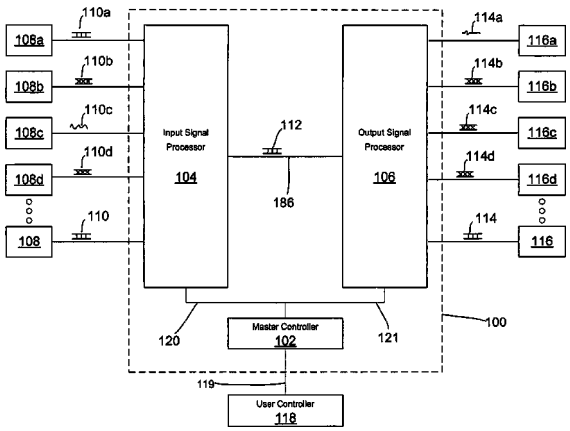
(Continued)

Primary Examiner — Alvin Zhu
(74) *Attorney, Agent, or Firm* — Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.

(57) **ABSTRACT**

Apparatus, systems and methods for receiving one or more input signals and providing output signals in various video, audio, data and mixed formats are described. One or more input processors receive the input signals. Each of the input processors provides one or more packetized signals corresponding to one or more of the input signals received at the input processor. Each output processor can receive one or more packetized signals and generate one or more output signals. The output signals correspond to one or more of the input signals, additional locally generated signals or data relating to the signals or any combination of such signals. Use of a packet router according to the invention allows input signals encoded as one set of packetized signals to be recombined to provide additional packetized signals incorporating the same or different combinations of the packetized signals.

12 Claims, 11 Drawing Sheets



(51) **Int. Cl.**

H04N 7/52	(2011.01)
H04N 21/226	(2011.01)
H04N 21/234	(2011.01)
H04N 21/236	(2011.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,519,223	B1	2/2003	Wager
6,937,291	B1	8/2005	Gryskiewicz
6,950,097	B1	9/2005	Hojabri
7,190,695	B2	3/2007	Schaub et al.
7,269,841	B1	9/2007	Hendricks et al.
2003/0063611	A1	4/2003	Schaub et al.
2003/0156535	A1	8/2003	Lebizay et al.
2003/0198226	A1	10/2003	Westberg

FOREIGN PATENT DOCUMENTS

FR	2793042	A1	11/2000
FR	2793043	A1	11/2000

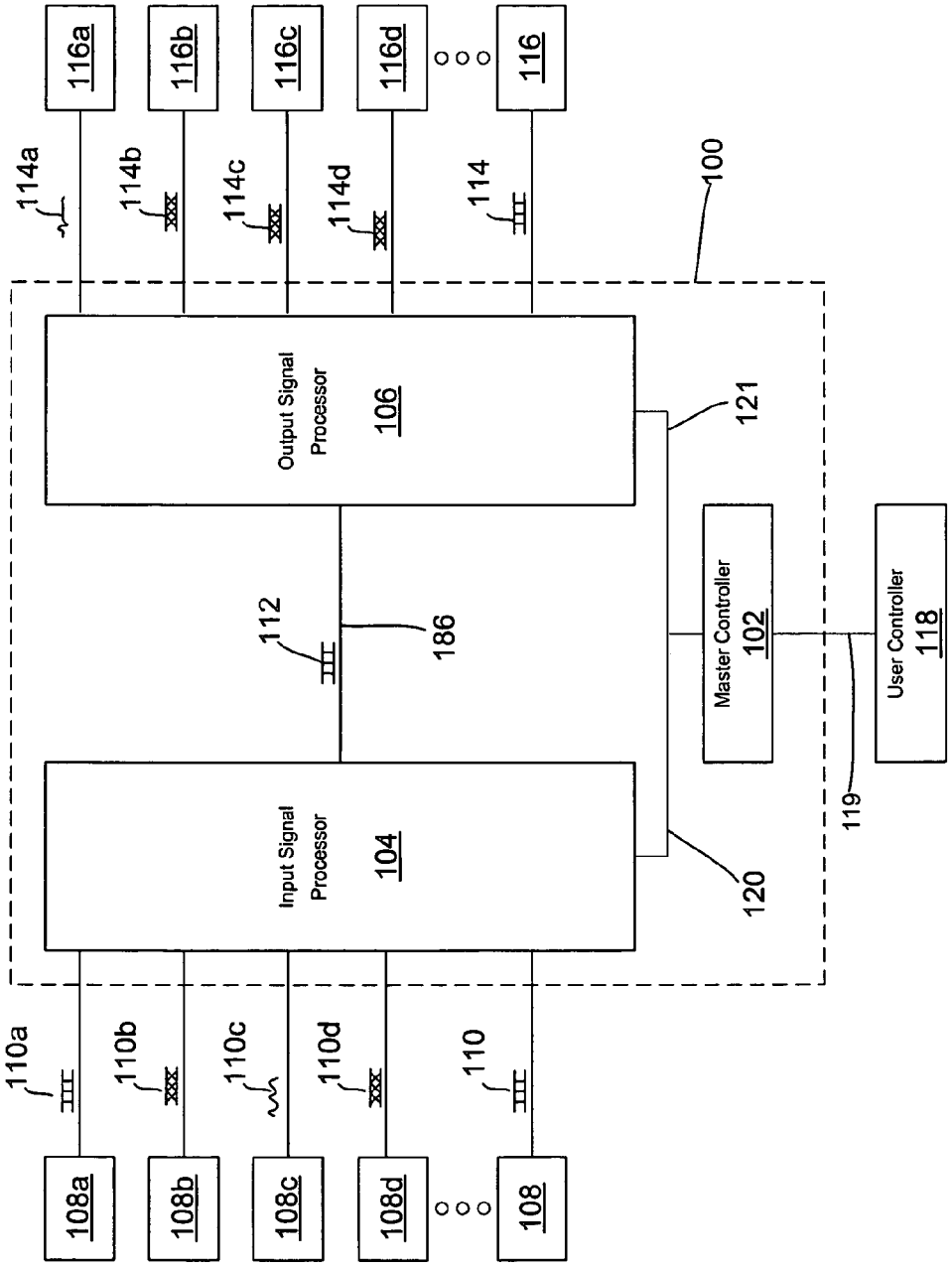

116a
116b
116c
116d
116
114a
114b
114c
114d
114
100
Output Signal Processor
106
121
112
186
Master Controller
102
User Controller
118
119
Input Signal Processor
104
120
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 1

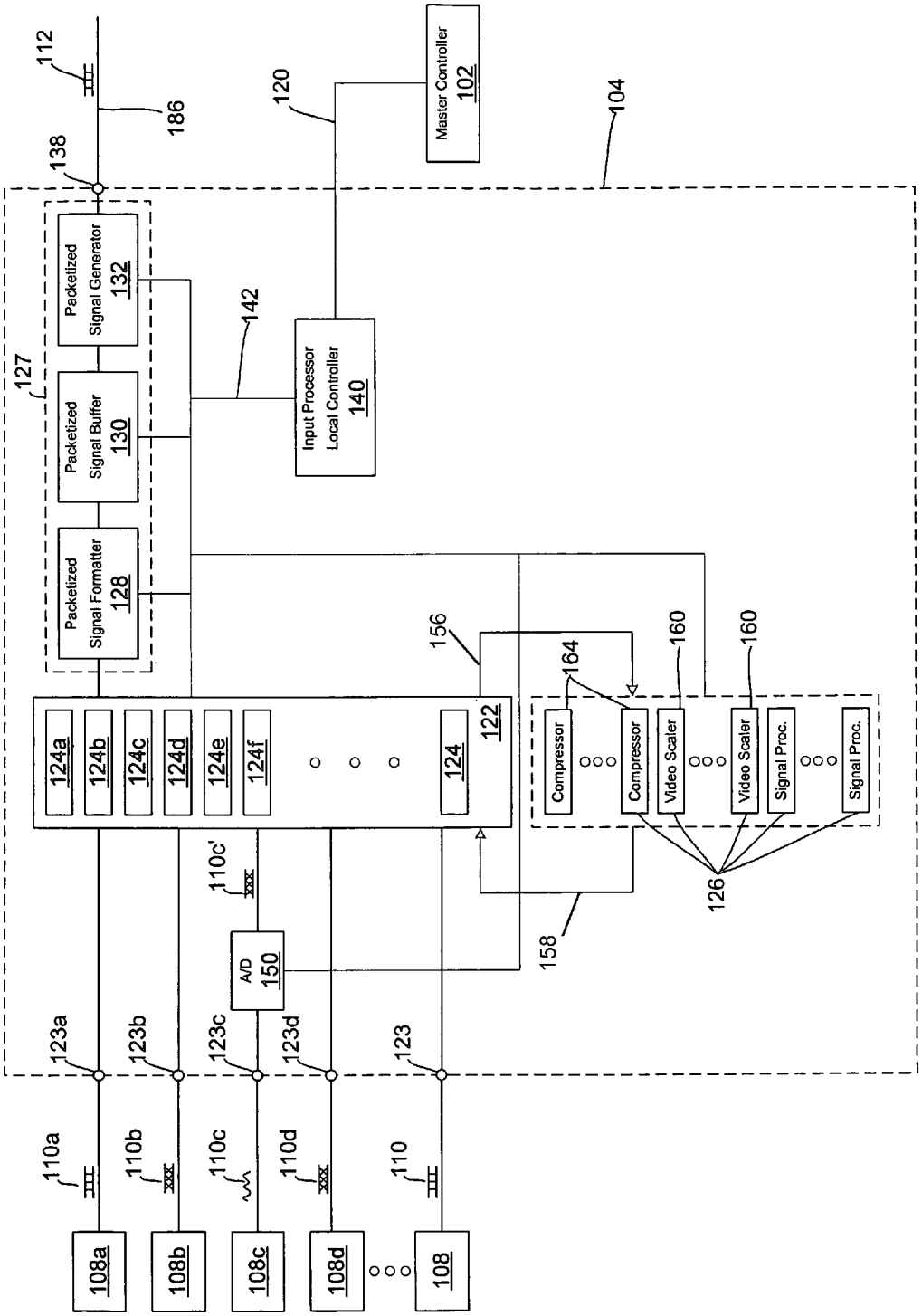

112
186
138
120
Master Controller 102
104
127
Packetized Signal Generator 132
Packetized Signal Buffer 130
Packetized Signal Formatter 128
142
Input Processor Local Controller 140
156
164
160
160
124a
124b
124c
124d
124e
124f
124
122
Compressor
Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
110c'
A/D 150
158
126
123a
123b
123c
123d
123
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 2

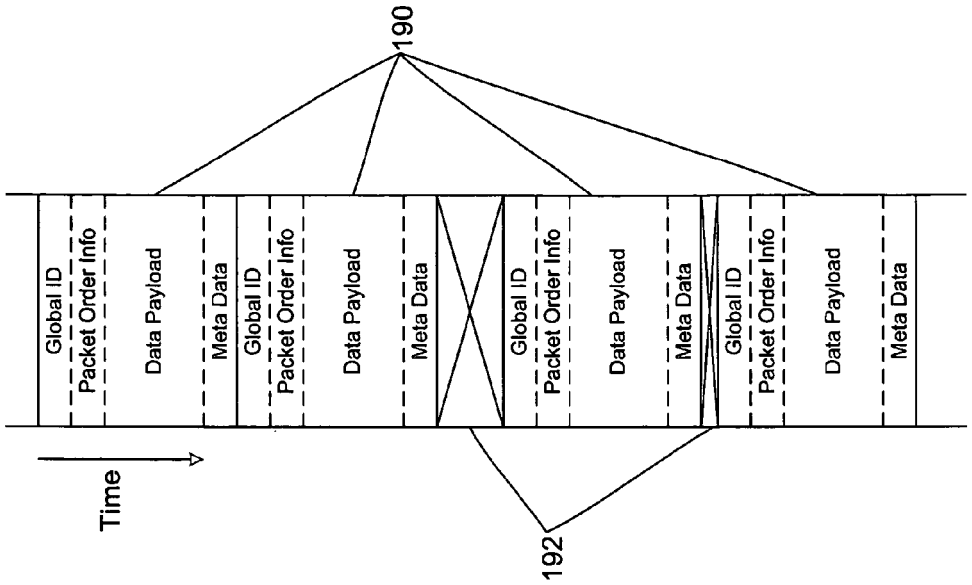


Figure 5

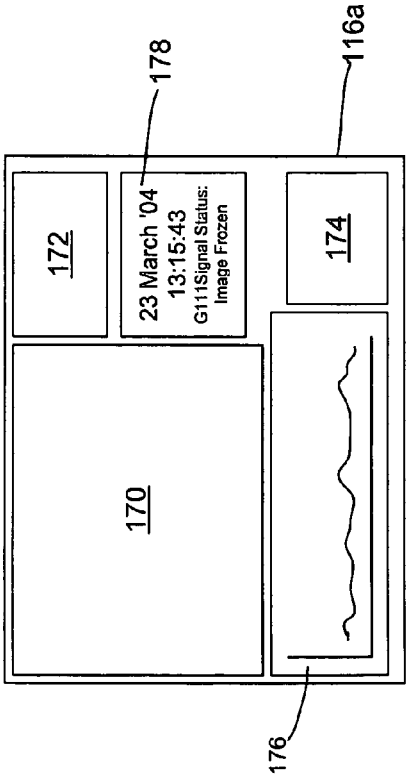


Figure 3

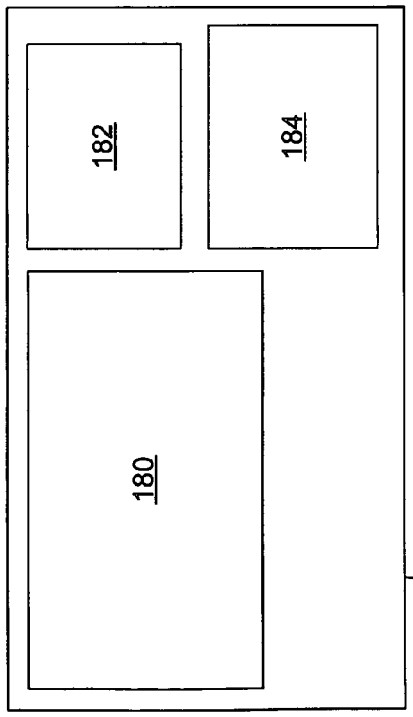


Figure 4

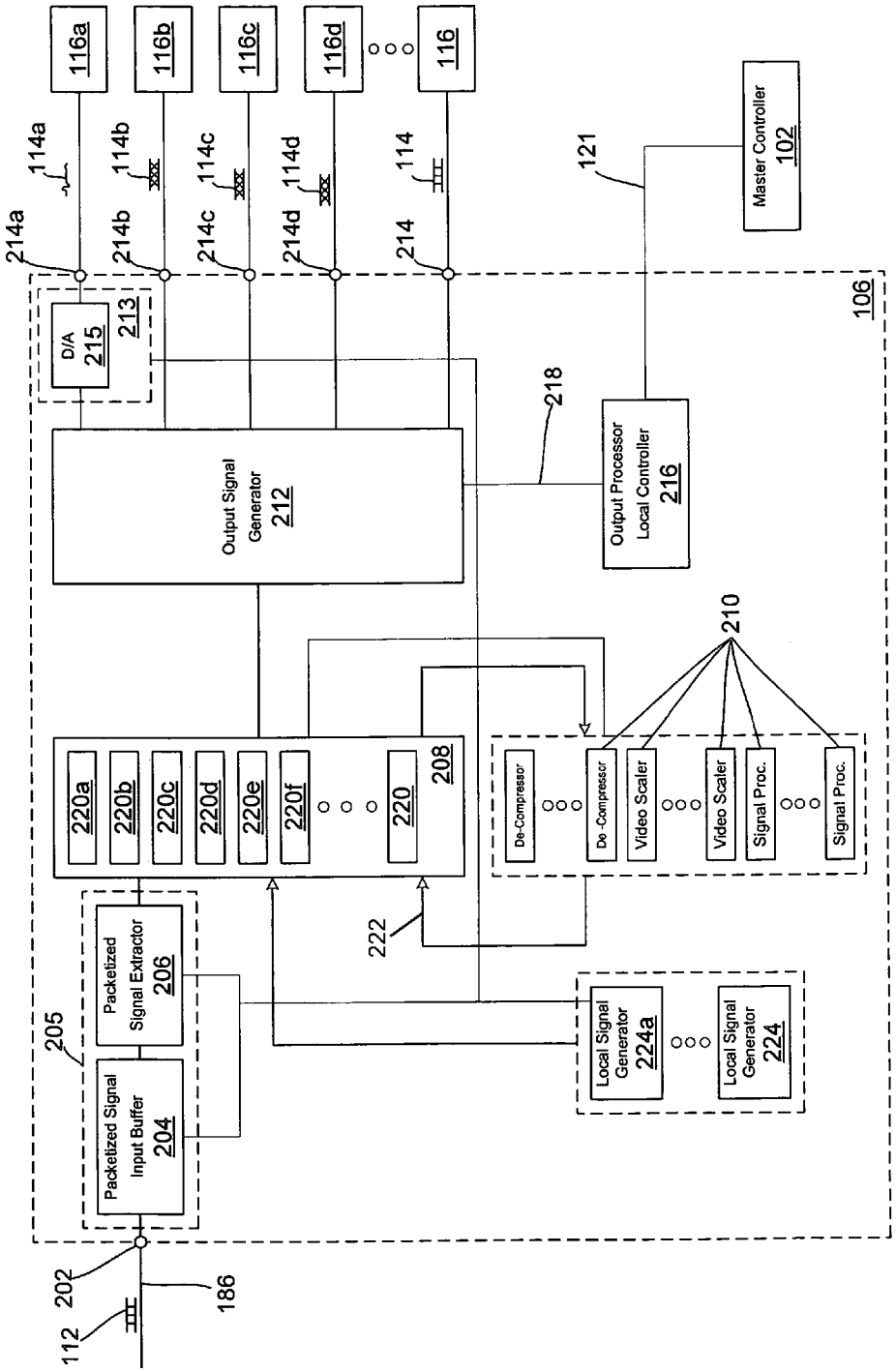

116a
116b
116c
116d
116
114a
114b
114c
114d
114
214a
214b
214c
214d
214
121
Master Controller 102
106
D/A 215
213
218
Output Signal Generator 212
Output Processor Local Controller 216
210
220a
220b
220c
220d
220e
220f
220
208
De-Compressor
De -Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
222
Packetized Signal Extractor 206
205
Packetized Signal Input Buffer 204
Local Signal Generator 224a
Local Signal Generator 224
202
112
186

Figure 6

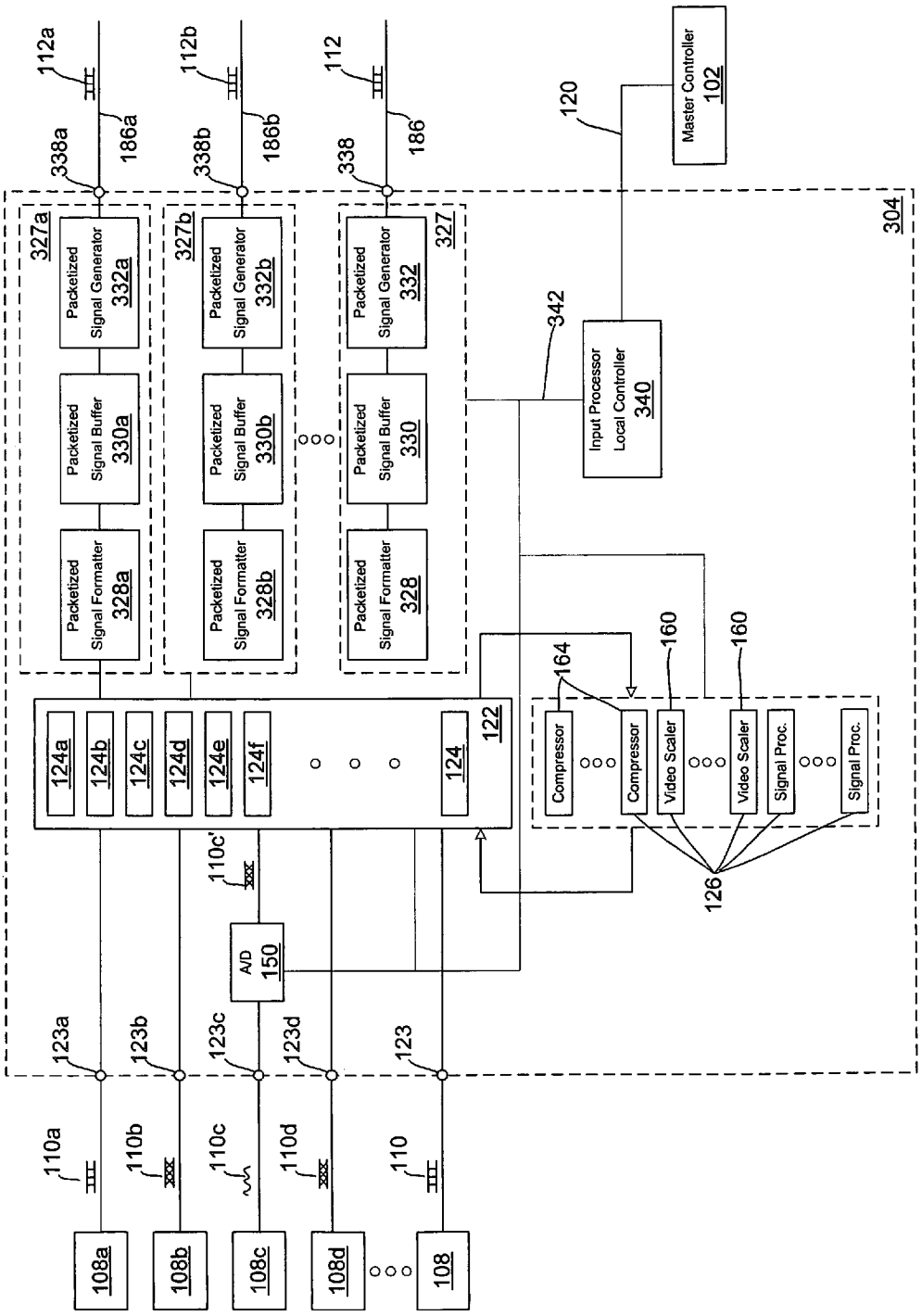

112a
112b
112
186a
186b
186
338a
338b
338
120
Master Controller
102
304
327a
327b
327
Packetized Signal Generator
332a
332b
332
Packetized Signal Buffer
330a
330b
330
Packetized Signal Formatter
328a
328b
328
342
Input Processor Local Controller
340
124a
124b
124c
124d
124e
124f
124
122
164
160
160
Compressor
Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
126
110c'
A/D
150
123a
123b
123c
123d
123
110a
110b
110c
110d
110
108a
108b
108c
108d
108

Figure 7

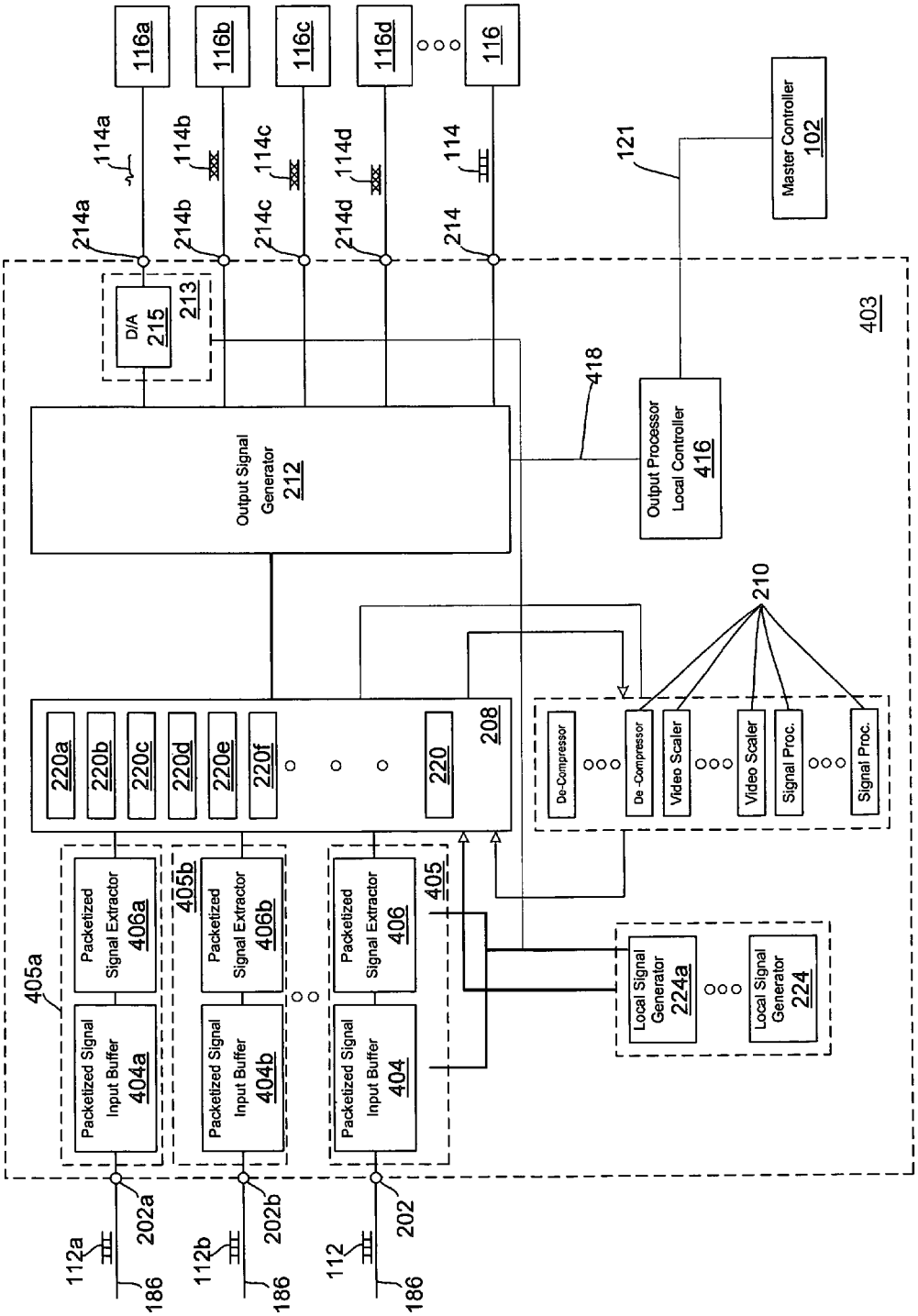

116a
116b
116c
116d
116
114a
114b
114c
114d
114
214a
214b
214c
214d
214
121
Master Controller 102
403
D/A 215
213
418
Output Signal Generator 212
Output Processor Local Controller 416
210
220a
220b
220c
220d
220e
220f
220
208
De-Compressor
De -Compressor
Video Scaler
Video Scaler
Signal Proc.
Signal Proc.
405a
405b
405
Packetized Signal Extractor 406a
Packetized Signal Extractor 406b
Packetized Signal Extractor 406
Packetized Signal Input Buffer 404a
Packetized Signal Input Buffer 404b
Packetized Signal Input Buffer 404
Local Signal Generator 224a
Local Signal Generator 224
112a
112b
112
202a
202b
202
186
186
186

Figure 8

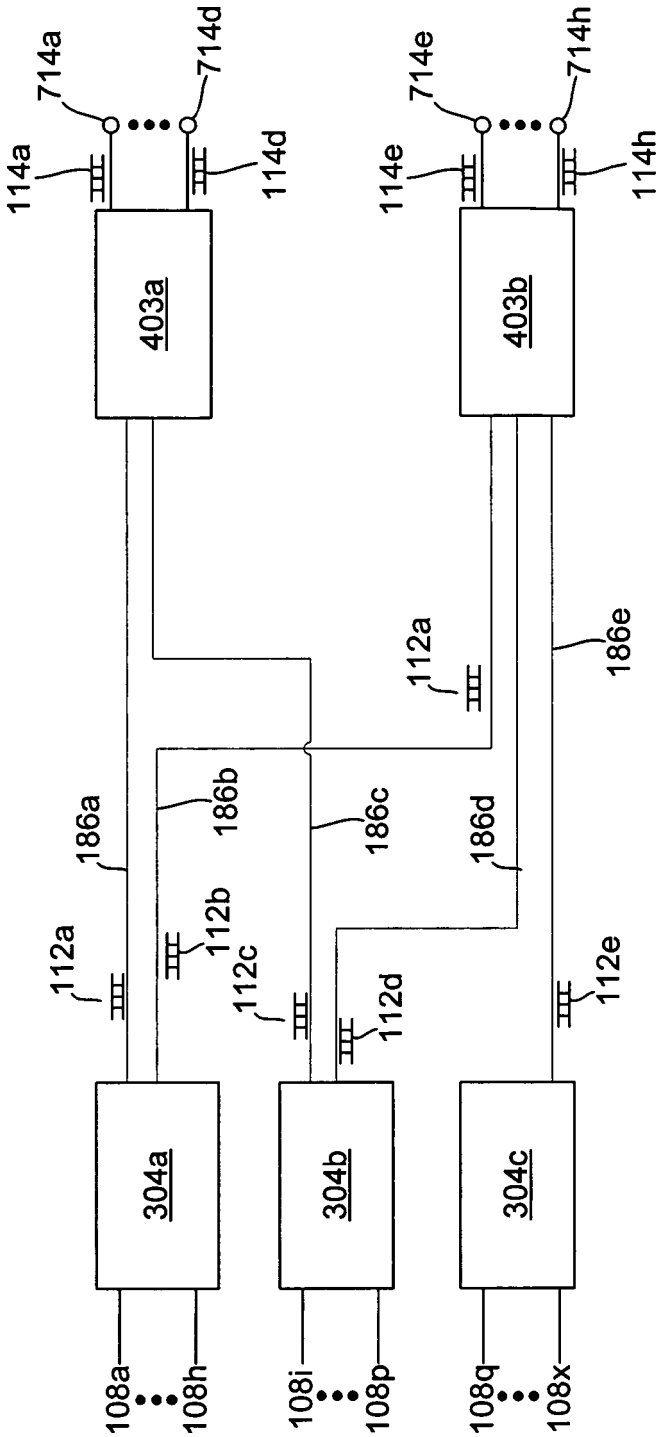

714a
714d
714e
714h
114a
114d
114e
114h
403a
403b
112a
186e
186a
186b
186c
186d
112b
112c
112d
112e
304a
304b
304c
108a
108h
108i
108p
108q
108x

Figure 9

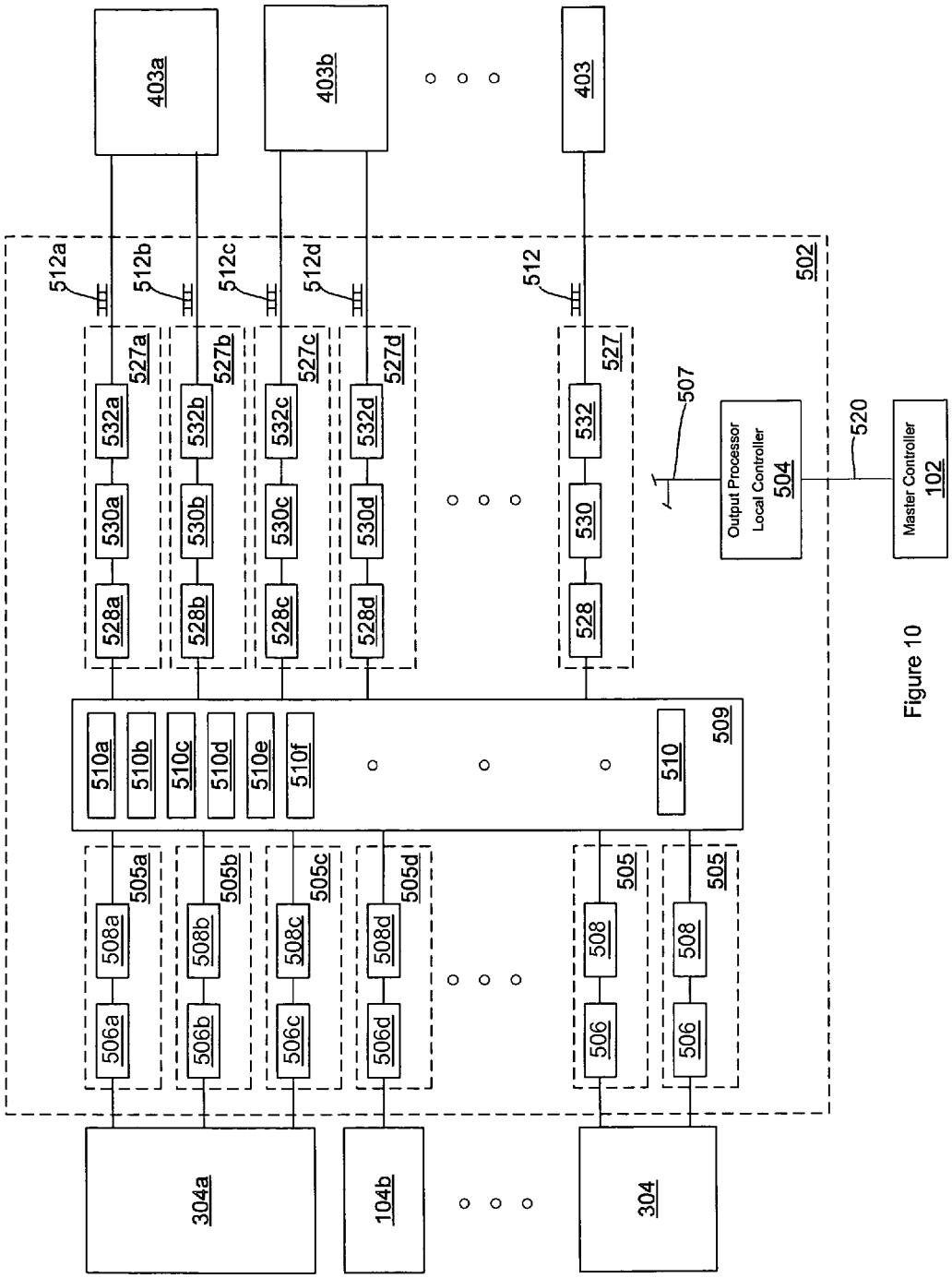

403a
403b
403
512a
512b
512c
512d
512
502
527a
527b
527c
527d
527
532a
532b
532c
532d
532
530a
530b
530c
530d
530
528a
528b
528c
528d
528
507
Output Processor
Local Controller
504
520
Master Controller
102
510a
510b
510c
510d
510e
510f
510
509
505a
505b
505c
505d
505
508a
508b
508c
508d
508
506a
506b
506c
506d
506
304a
104b
304

Figure 10

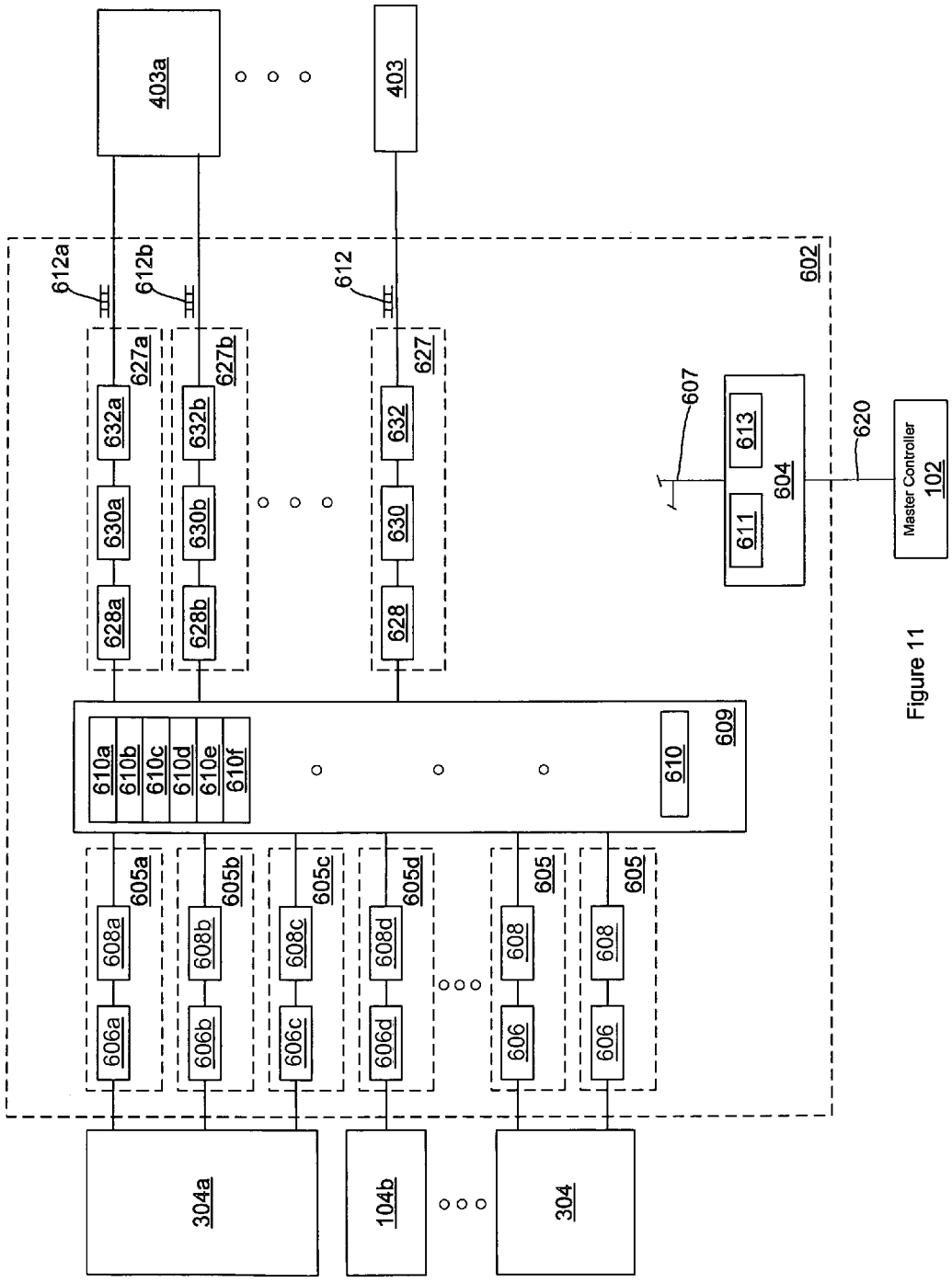

403a
403
612a
612b
612
602
627a
627b
627
632a
632b
632
630a
630b
630
628a
628b
628
607
613
611
604
620
Master Controller
102
610a
610b
610c
610d
610e
610f
610
609
605a
605b
605c
605d
605
608a
608b
608c
608d
608
606a
606b
606c
606d
606
304a
104b
304

Figure 11

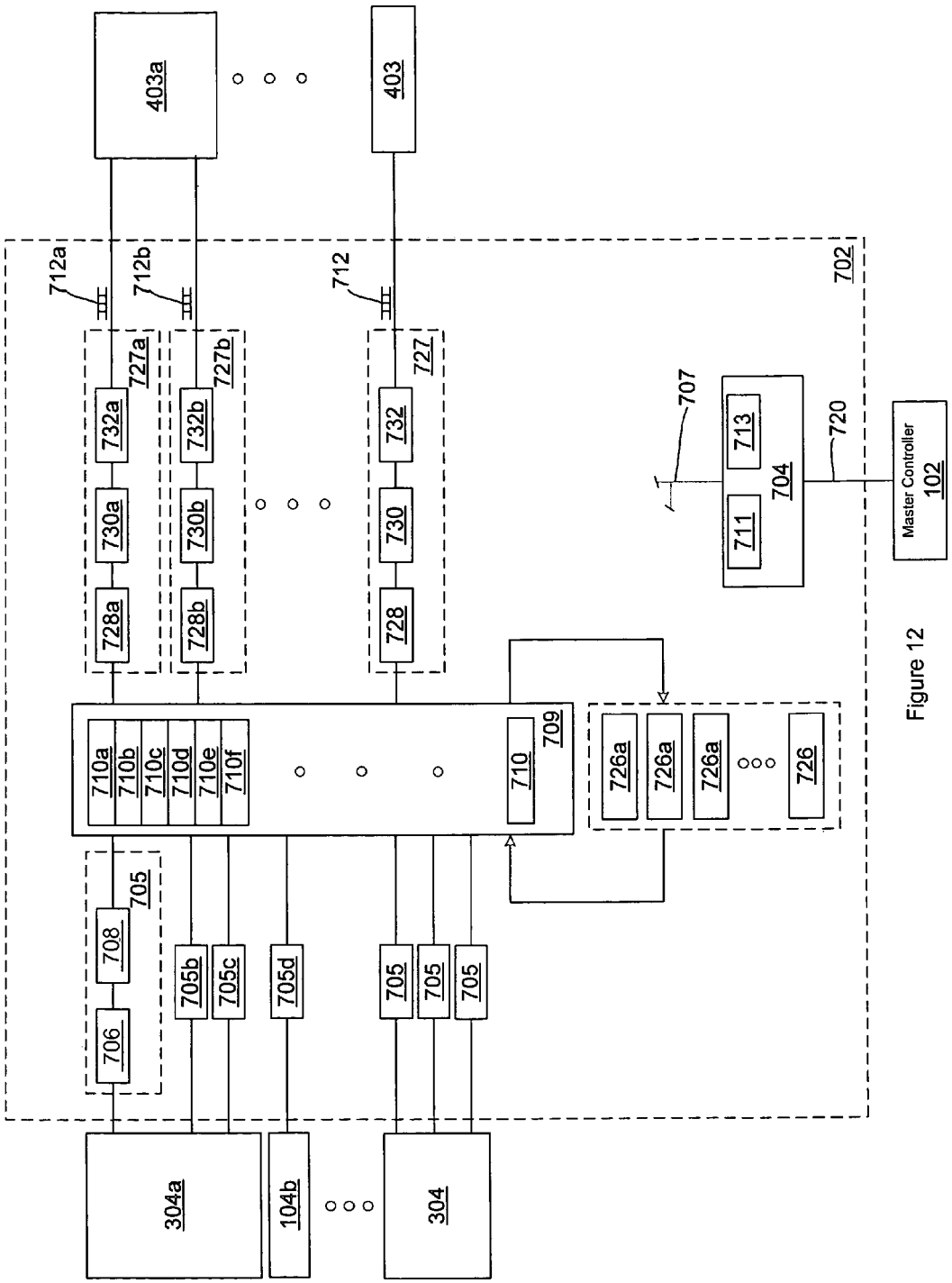

403a
403
712a
712b
712
702
727a
727b
727
732a
732b
732
730a
730b
730
728a
728b
728
707
713
711
704
720
Master Controller
102
710a
710b
710c
710d
710e
710f
710
709
726a
726a
726a
726
705
708
706
705b
705c
705d
705
705
705
304a
104b
304

Figure 12

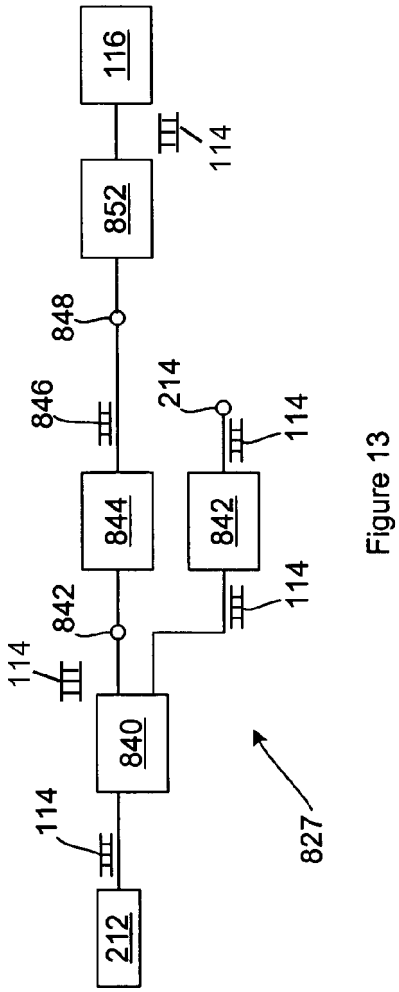


Figure 13

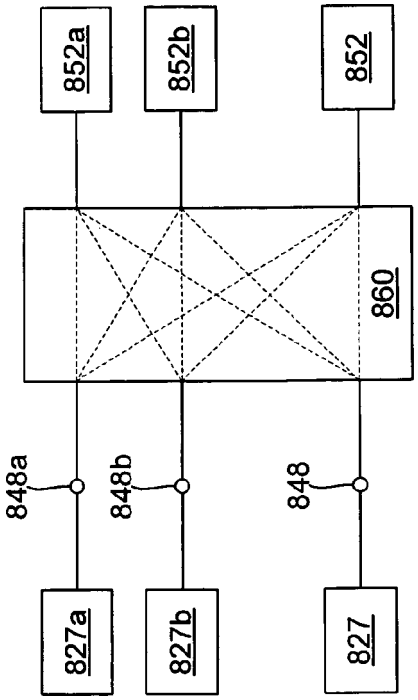


Figure 14

APPARATUS, SYSTEMS AND METHODS FOR PACKET BASED TRANSMISSION OF MULTIPLE DATA SIGNALS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 14/026,039, filed Sep. 13, 2013, which is a continuation of U.S. patent application Ser. No. 10/816,841 filed Apr. 5, 2004, which claims priority to U.S. patent application Ser. No. 60/459,964, filed Apr. 4, 2003. The entire contents of U.S. patent application Ser. No. 14/026,039, U.S. patent application Ser. No. 10/816,841 and U.S. patent application Ser. No. 60/459,964 are hereby incorporated by reference.

FIELD OF THE INVENTION

This invention relates to systems for transmitting and distributing audio data, video data and other types of data.

BACKGROUND OF THE INVENTION

Recent advances in video monitor technology have resulted in the development of large format, high quality video displays that are capable of displaying multiple video signals simultaneously. In television studios and other locations where many different video sources must be monitored, these video displays have begun to displace traditional individual monitors that displayed a single video source to which they were physically coupled.

Modern signal processing equipment allows video and other data to be routed to different display monitors, however, this equipment can still require that for a particular signal to be used in multiple locations on multiple display devices it must be replicated and coupled to equipment in the different locations. This results in excessive cabling requirements, multiple signal regeneration and replication stages, and can result in degraded signals and multiple failure points within the signal path.

There is a need for an improved efficient system for receiving various input signals, including video, audio and data signals, formatting the received signals and routing the formatted signals to various output devices.

SUMMARY OF THE INVENTION

In one embodiment, the present invention provides a system that includes a master controller, one or more input processors, one or more output processors, and one or more user controllers. The system may also include additional master controllers that serve as back-up master controllers.

One or more input devices are coupled, directly or indirectly, to each of the input processors. Each input device provides one or more input signals to the input processors. One or more output devices are coupled to the output processors. Each output processor receives an output signal from an output device. Each of the input processors generates one or more packetized signals. Each packetized signal is transported across a communications link to one or more of the output processors. Each output processor may receive one or more packetized signals.

The master controller receives user control signals from one or more user controllers indicating which input signals are to be routed to which output devices. The user control signals may also indicate the format in which the input signal is to be presented at the output device.

For example, for a video input signal the user control signals may specify the position and dimensions of a window on an output display device in which the input video signal is to be displayed. The user control signals may also define other characteristics for the displayed video signal, such as color balance, white balance, color to black & white conversion or the addition of a border or any other characteristics that a video signal may be processed to have.

For an audio input, the user control signals may specify a particular equalization (ie. jazz, rock, classical), volume balancing to some specified maximum or average volume, left-right signal balance, encoding in or conversion to a particular noise reduction and/or a multi-channel audio standard. For a data signal, the user controls may specify characteristics that are appropriate to the data.

In response to the user control signals, the master controller generates input processor control signals to operate the input processors and output processor control signals to operate the output processors. The input and output processor control signals may be transmitted to the various input and output processors using any type of communications link. The master controller coordinates the operations of the various input and output processors (and other elements of the system) to provide the output signals requested by a user or users who operate the user controllers to generate the user control signals.

If the system includes more than one master controller, one of the master controllers may be designated as a primary master controller and the remaining master controllers designated as backup master controllers. Each of the master controllers is coupled to the user controllers to receive the user control signals and is capable of generating the input processor control signals and the output processor control signals. The primary master controller actively generates the input processor control signals and output processor control signals. If the master controller fails or is disabled for any reason, one of the backup master controllers may be designated as the primary master controller.

Each input processor has an input processor local controller which receives the input processor control signals for that input processor. In response to the input processor control signals, the input processor local controller configures the various components of the input processor to receive and process the input signals coupled to that input processor and to generate one or more packetized signals, as requested by the master controller.

Each input processor includes a plurality of data buffers to store each input signal in a digital form. If an input signal is received in an analog form, an analog-to-digital converter is dynamically coupled between an input port at which the input signal is received and a data buffer to digitize the signal. Some input signals may be received in a processed manner, meaning that the signal has been processed in some manner. If an input signal is received in a processed manner then an un-processor may be dynamically coupled between an input port at which the input signal is received and a data buffer in which the input signal is stored. Additionally each input processor may include one or more data analyzers that can be dynamically coupled to each input signal to provide information about the input signal. The resulting information is also stored in a data buffer. The buffered signals are then read out and processed by signal processors to format them as indicated by the user control signals. The processed signals are also buffered in data buffers. Each input signal may be processed multiple times to create different versions of the input signal, to extract various information regarding

the input signal for use on different output devices, or for use on the same output device in different versions, formats or sizes.

The data buffers in the input processor, the output processor and other components and embodiments of the invention are used to temporarily store data that is received from a source and is subsequently read out by one or more recipient or destination elements or devices. Data may be read out in the order in which it is received, in which case a first-in/first-out buffer may be used to store the data. Alternatively, the data may be read out in any order by the recipient elements. In each case, the data buffer is configured to retain any datum until it has been read by all recipient elements that use the datum.

Each of the buffered signals (including the input signals and the processed signals) is assigned a global identification code. One or more of the buffered signals are converted into a packetized signal by a packetized signal output stage in each input processor. Each packetized signal contains a series of packets. Each packet contains a part of the data from the buffered signal along with the global identification code of the buffered signal. An input processor may have more than one packetized signal output stages to produce more than one packetized signal.

A packetized signal may be converted into and transmitted as a bitstream, or it may be transmitted using any communications protocol.

Each output processor receives one or more packetized signals. Each packetized signal is buffered as it is received. As complete packets corresponding to each global identification code are received, they are assigned a local identification code and are buffered in a separate data buffer. The isolated packets in data buffer correspond to a particular version of an input signal received at one of the input processors. The isolated stream may be processed to reverse any signal processing step or steps applied in the input processor or in an input device or combination of devices that combine to produce an input signal coupled to the input processor, such as a data compression step, or to apply additional signal processing steps. Any such processed signal is buffered again and assigned a new local identification code. One or more these buffered signals is then combined to form each output signal.

In one embodiment of the invention, a packet router is coupled between a plurality of input processors and a plurality of output processors. The packet router receives packetized signals from the input processors and isolates the packets corresponding to each global identification code. The packetized router then assembles new packetized signals corresponding to a combination of the global identification codes. The packetized router operates under the control of the master controller to route packets with the appropriate global identification code to the appropriate output processor. The packet router allows an input signal received at any of the input processors to be formatted and routed to any of the output processors.

In another embodiment of a packet router according to the invention, packets from one or more incoming packetized signals are stored in packet storage locations within a memory system. The packets are then read by one or more packetized signal output stages, each of which produces an outgoing packetized signal corresponding to a selected set of global identification codes. Storage of incoming packets and distribution of the packets to the packetized signal output stages is controlled by a router controller. In another embodiment of a packet router, one or more signal processors, such as video scalers or delay elements, are provided to process the incoming packets to provide processed packets that form a processed signal. Each processed signal is assigned a unique global identification code and may be included in an outgoing packetized signal.

These and other aspects of the invention and its various embodiments are described in greater details below,

BRIEF DESCRIPTION OF THE DRAWINGS

Several exemplary embodiments of the present invention will now be described in detail with reference to the drawings, in which:

FIG. **1** illustrates a system according to the present invention;

FIG. **2** illustrates a first input processor according to the present invention;

FIG. **3** illustrates the display of a first video monitor;

FIG. **4** illustrates the display of a second video monitor;

FIG. **5** illustrates one form of a packetized signal according to the present invention;

FIG. **6** illustrates a first output processor according to the present invention;

FIG. **7** illustrates a second input processor according to the present invention;

FIG. **8** illustrates a second output processor according to the present invention;

FIG. **9** illustrates an exemplary coupling between a plurality of input processors and a plurality of output processors;

FIG. **10** illustrates a first packet router according to the present invention;

FIG. **11** illustrates a second packet router according to the present invention;

FIG. **12** illustrates a third packet router according to the present invention;

FIG. **13** illustrates an alternative output stage for an output processor according to the present invention; and

FIG. **14** illustrates a switch for use with the alternate output stage of FIG. **13**.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Reference is first made to FIG. **1**, which illustrates a signal processing system **100**. The signal processing system includes a master controller **102**, an input processor **104** and an output processor **106** according to the present invention. The input processor **104** receives a plurality of input signals **110** from various signal sources **108** and provides a packetized signal **112**. The packetized signal **112** corresponds to some or all of the input signals **110** or to data or processed signals derived from the input signals. The packetized signal **112** is transported by a communications link **186** to the output processor **106**. The output processor **106** receives the packetized signal **112** and produces one or more output signals **114**, which are processed by output devices **116**. The output signals **114** correspond, at least in part, to one or more of the input signals **110**.

The system may additionally include backup master controllers (not shown).

The input processor **104** and output processor **106** operate under the control of the master controller **102**. The master controller **102** is coupled to one or more user controllers **118**, from which the master controller receives user control signals **119**. The master controller **102** and the user controllers **118** may be combined in a single unit, or may be

assembled in a single assembly, or they may be separate units that are coupled together.

A user or multiple users (not shown) human or some other type of device (for example automated monitoring and control systems) operate the user controllers **118** to generate the user control signals **119**. The user control signals **119** indicate which input signals or signals derived from the input signals **110** the user would like included in the output signal **114** provided to each output device. Each user may have control over one or more output devices **116** in whole or in part. The user control signals **119** may also indicate additional characteristics about the output signal **114** provided to each output device **116**. The master controller **102** translates the users control signals **119** into input processor control signals **120** and output processor control signals **121** to control the operation of the input processor **104** and the output processor **106** respectively so that the output signals **114** are provided in accordance with the user control signals **119**.

The input signals **110** may be base-band, compressed, time division multiplexed audio signals, video signals (which may also include audio information), metadata, or other data signals. Similarly, the output signals **114** may be audio signals, video signals, or data signals. Typically, each output signal **114** will correspond to one or more of the input signals and or information derived from the input signal. A particular output signal may include a combination of audio, video or data input signals or signals produced by input signal analyzers or any combination of these types of signals. The nature of each output signal **114** is appropriate for the output device **116** that receives the output signal **114**. Some of the output devices **116** may be video monitors, such as analog video monitor **116***a* and digital video monitor **116***b*, for displaying output video signals. Some of the output devices **116** may be sound systems, such as sound amplification and broadcast system **116***c*, for further processing or playing output audio signals. Some of the output devices may be data processing systems, such as computer system **116***d*, for further processing or displaying the output data signals. In any particular embodiment of the present invention, the output signals **114** may be of the same or different types, depending on the usage of the embodiment. In an alternative embodiment of the invention, the output processor may provide only a single output signal. The type of any particular signal may change depending on the usage of the signal, under the control of the master controller **102**.

Reference is next made to FIG. **2**, which illustrates the input processor **104** in greater detail. Input processor **104** includes a plurality of input ports **123**, a plurality of input signal analyzers or processors, such as A/D converter **150**, a plurality of data buffers **124**, which are part of a memory system **122**, one or more signal processors **126**, a packetized signal output stage **127**, a packetized signal output port **138** and an input processor local controller **140**. The packetized signal output stage **127** includes a packetized signal formatter **128**, a packetized signal buffer **130** and a packetized signal generator **132**.

Memory system **122** may be a local memory device or memory space within the input processor **104** or it may be located on an attached storage device or other medium. Data buffers **124** will typically comprise memory space allocated within memory system **122**.

The input processor local controller **140** receives the input processor control signals **120**. The input processor local controller **140** controls the operation of the various elements of the input processor **104** through control lines **142** in response to the input processor control signals **120**.

Each input signal **110** is coupled to one of the input ports **123**. Each of the input ports **123** is coupled to the memory system **122**. Each input signal **110** is buffered in a data buffer **124** in memory system **122**. Analog input signals **110** are converted to a digital version and the digital version is buffered in a data buffer **124**. For example, if input signal **110***c* is an analog input signal, then an analog-to-digital (ND) converter **150** is dynamically coupled between input port **110***c* and memory system **122** to convert input signal **110***c* into a corresponding signal **110***c*' comprising a stream of packets according to a digital signal standard. For example, if input signal **110***c* is a standard 1V peak-to-peak audio signal, it may be sampled and converted by A/D converter **150** into a WAV digital audio signal **110***c*', as is commonly done to record music on CD-ROM media. For example, if input signal **110***c* is a compressed video stream a decompressor (one of the signal processors) may be dynamically coupled between input port **110***c* and memory system **122** to convert input signal **110***c* into a corresponding signal **110***c*' comprising a stream of data according to a digital signal standard.

In the present embodiment, the input processor contains a bank of A/D converters, each of which may dynamically be coupled between any of the input ports **123** (or a group of the input ports) and memory system **122**. The input processor local controller **140** controls the coupling of any particular A/D converter between any particular input port **123** and memory space **122**. A particular A/D converter **150** may be shared by two or more input ports **123** under the control of input processor local controller **140**. Alternatively, a dedicated A/D converter **150** may be provided for some or all of the input ports **123**. The A/D converter **150** may be activated by the input process local controller **140** if the input signal **110** received at a port **123** is an analog signal. In another alternative embodiment, some or all of the input ports **123** may be designated as analog input ports and an A/D converter may be permanently coupled between those ports and the memory system **122**.

The input ports may be any type of communication port, such as an Ethernet, BNC, optical, telephone line or any port suitable with any type of communication system. The input signals may be in any communication standard or protocol, including, including TCP/IP. In this case, the coupling between the input device and the processor may be a LAN, WAN, the Internet or another TCP/IP communication system.

The input processor **104** may also contain a bank of input signal analyzers (not shown). The input signal analyzers may be dynamically or statically coupled to an input port in the same manner as A/D converter **150**. For example if the input signal is a video signal, an input signal analyzer may extract performance and signal content metrics from or about the input signal such as blackness of the signal, the amount of motion within the signal, bit or formatting errors in the signal. The metadata produced by the data analyzer is stored in a data buffer and is considered and treated as a processed signal that can be packetized and coupled to the output processor over a communication link.

Each of the input signals **110** may be retrieved from the corresponding data buffer **124** as a buffered signal **156**.

The input processor may also include other input signal processing elements that may be coupled between an input port **123** and memory system **122**. The signal processing elements may include video scalers, video de-interlacers, data compressors, data de-compressors, data format converters or any other type of signal processor, including the signal processing elements described below. For example, if one of

the input signals is an analog NTSC video signal, then a video signal digitizer may be dynamically coupled between an input port at which the signal is received to convert the input signal into a MPEG2 digital video signal. The input processor may contain a bank of input signal processing elements and analyzers, which may be dynamically coupled between any input port and memory system **122**. Input processing elements or analyzers may also (or alternatively) be coupled to only one port for selective use with that port. Input processing elements may also (or alternatively) be permanently coupled to one or more of the input ports.

The signal processors **126** are coupled to memory system **122** to retrieve the buffered signals **156** from memory system **122**, process the buffered data signals to generated processed signals **158**, which are then buffered in data buffers **124** in memory system **122**. A processed signal **158** is stored in a different data buffer **124** than the input signal **110** from which the processed signal is derived. The signal processors **126** are illustrated in a dotted box and the coupling between the memory system **122** and the signal processor extends to the dotted box to indicate that any of the signal processors may be dynamically coupled to any of the data buffers **124** to retrieve a buffered signal and to store a processed signal.

A particular input signal **110** may be processed to generate more than one processed signal **158** and each of the resulting processed signals **158** are stored in different data buffers **124** in memory system **122**. As a result, the original input signal **110** and any versions of the original input signal **110** that are generated as processed signals **158** are available from memory system **122** as buffered signals **156**.

In the present embodiment, the signal processors **126** include video scalers **160**, embedded audio extractors, ancillary data extractors, signal content analysers and data compressors **164**. The signal processors **126** may also include data de-compressors, image rotation devices, special effects processors, image invertors, spatial filters, edge enhancement processors, color space converters, audio sweetening processors, digital audio decompressors, and digital audio processors. A signal processor may be used to process two or more input signals (or processed signals) by time-division-multiplexing the signal processor between the data buffers used to buffer the two or more input signal (or processed signals).

Each video input signal **110** will have height and width dimensions, usually defined in pixels. For example, a video image may comprise a series of frames that are 640 pixels wide by 400 pixels high. A video scaler **160** is capable of rescaling a video signal from its original dimensions to different dimensions. In the present embodiment, the input processor **104** includes a plurality or bank of video scalers **160**. Each video scaler **160** receives control instructions from the input processor local controller **140** to extract a particular video input signal **110** from the appropriate data buffer **124** and rescale the video input signal to specified dimensions and to store the resulting processed signal **158** in another data buffer **124**. A video scaler **160** may be configured to retain or change the aspect ratio of an input data signal or to crop the input data signal in the processed signal and to provide any other function that a conventional scaler is capable of providing. For example, a video scaler may be configured to crop the input data signal to select a portion of it, and then scale the cropped video image to specified dimensions.

A particular video scaler **160** may be instructed to scale a video input signal **110** to more than one set of new dimensions and may generate two or more processed signals **158**, each of which is separately buffered in separate data buffers **124**. In addition, a particular video scaler **160** may be shared (or multiplexed) between two or more video input signals **110** to generate two or more corresponding processed signals **158**, each of which is separately buffered in separate data buffers **124**. One video input signal **110** may also be retrieved by two or more video scalers **160** (or other signal processors **126**) to produce two or more corresponding processed signals **158**, which are similarly buffered in separate data buffers **124**.

Data compressors **164** are used to generate a processed signal **158** that is a compressed version of any signal stored in a data buffer **124**. For example a video input signal **110** in a DVI format may be compressed into an MPEG-2 format to reduce the amount of data required to transmit the signal. The resulting MPEG-2 format video signal is stored in a data buffer **124** as a processed signal **158**. The data compressors **164** may include a plurality of compression elements, which may be hardware or software elements, designed to compress audio, video or data signals into various types of compressed signals. The data compressors may provide a lossy or lossless compression. In each case, the compressed data signal produced by a data compressor **164** is stored as a processed signal **158**. A particular embodiment of an input processor **104** may include any number and type of data compressors **164**.

The data compressors **164** may include horizontal or vertical line filters that produce a processed video data signal comprising a portion of the video data from a video input data signal. For example, a horizontal line filter may be configured to horizontally compress a 640×400 pixel video signal into a 320×400 pixel video signal by discarding every other pixel in each line of the video signal. A vertical line filter may be configured to compress a 640×400 pixel video signal into a 640×200 pixel video signal by discarding every other line in the video signal. A horizontal/vertical line filter may be configured to compress a 640×400 pixel video signal into a 160×100 pixel video signal by discarding three of every four lines of the video signal and discarding three of every four pixels in each line that is retained.

The input signals **110** will typically be asynchronous with respect to one another (unless they are from a synchronized source). The size and timing of packets in each input signal will depend on the signal standard used to encode the signal. For example, input signal **110***a* may be formed of uniformly sized packets that are spaced equally apart in time. Input signal **110***b* may be formed of differently sized packets that do not arrive at equally spaced intervals. As a result of the differences between the input signals **110**, data buffers **124** may be allocated with a large or smaller memory space to properly buffer each input signal **110**.

In addition, the size and timing of packets in processed signals **158** may also vary depending on the data standard used to encode the processed signals **158**. The data buffers **124** used to buffer a processed signal **158** may similarly be dynamically allocated a memory space of a suitable size.

The master controller **102** (FIG. **1**) controls the operation of the input processor **104**. The master controller **102** assigns a unique global identification code to each signal that is buffered in a data buffer **124**. This global identification code is used to identify the signal in both the input processor and the output processor. Each version of a particular input signal **110** that is buffered in a data buffer **124** is assigned a different global identification code. For example, video input signal **110***a* is first buffered in the form in which it is received. The input signal **110***a* may be scaled to new dimensions using a video scaler **160** to produce a scaled signal **110***aa*, which is separately buffered. The input signal

110*a* may also be scaled to a second set of dimensions to produce a second scaled signal **110***ab*, which is also separately buffered. The second scaled signal **110***ab* may then be compressed to produce a scaled and compressed signal **110***ac*, which is also separately buffered. Each of the different versions **110***a*, **110***aa*, **110***ab*, **110***ac* may be separately retrieved from its data buffer and may be identified using its unique global identification code.

To further explain the invention and the present embodiment, an example of the use of this embodiment will be described. In the example, the input processor **104** receives three digital video input signals **110***a*, **110***b* and **110***d* and one analog video input signal **110***e*. Analog video input signal **110***e* is digitized using a A/D converter **150** to produce a digital signal **110***e*' corresponding to analog signal **110***e*. Signals **110***a*, **110***b*, **110***d* and **110***e*' are buffered in separate data buffers **124**.

Reference is made to FIG. **3**, which illustrates output device **116***a*, which is an analog standard definition 4:3 format video monitor capable of displaying images with a resolution of 640×480 pixels. The display of video monitor **116***a* is used to display information in five different parts or windows: video windows **170**, **172**, **174** and **176** and graphics window **178**.

A user configures the video and other information shown on each video monitor using user controller **118**. User controller **118** may provide a graphical or other interface allowing the user to define windows and other elements on a video monitor and assign specific input signals or other information to be displayed in each window or other element. The user has defined the parts of the display on video monitor **116***a* as follows:

Window/Element	Position (relative to top left corner)	Dimensions	Contents
Video window 170	10, 10	400 × 300	A version of video input signal 110a
Video window 172	420, 10	200 × 113	A version of video input signal 110b
Video window 174	460, 340	160 × 120	A version of video input signal 110d
Video window 176	10, 320	440 × 140	Rejected packets data for video signal 110b
Graphics window 178	420, 150	200 × 150	Date/Time/ Metadata Information

Reference is made to FIG. **4**, which illustrates output device **116***b*, which is a digital high definition 16:9 format video monitor with a resolution of 1920×1080 pixels. The display of video monitor **116***b* is used to display information in three different parts or windows: video windows **180**, **182** and **184**. The user has defined the parts of the display on display monitor **116***b* as follows:

Window/Element	Position (relative to top left corner)	Dimensions	Contents
Video window 180	60, 60	1140 × 640	A version of video input signal 110b
Video window 182	1280, 60	560 × 420	A version of video input signal 110e
Video window 184	1280, 540	610 × 460	A version of video input signal 110a

In an alternate example, window positions maybe such that some or all of the windows are overlapping, or arranged in a cascaded manner.

The video windows have been described as containing "a version of" one of the video input signals **110**. The user will typically specify the position and dimension of a window on a video monitor and the input signal **110** that the user would like displayed in each window. An appropriate version of the input signal is prepared by the input processor **104** and provided to the output processor **106** for display on the video monitor. Alternatively, the user may specify certain signal processing steps to be performed on an input signal before it is displayed in a window. For example, if the signal processors **126** (FIG. **2**) include a color/black & white converter, then a user may specify that a color input signal be converted into a black & white signal and that the black & white version of the input signal (or a version of the black & white signal) be displayed in a particular window.

Reference is again made to FIG. **1**. The user controller **118** transmits the user's instructions for each output device **116** to the master controller as user control signals **119**. The user's instructions relating each output device **116** will typically depend on the nature of the output device **116**. For example, if an output device **116** is an audio processing system capable of receiving and switching between multiple audio signals, then the user may specify that one or more audio input signals **110**, or the audio components of video input signals **110**, be directed to the sound output device **116**. If an output device **116** is only capable of receiving a single audio signal and then amplifying and broadcasting the audio signal, the user may specify that a particular input audio signal or the audio component of a particular video input signal **110** be directed to the sound output device **116**. Similarly, a user may specify that any particular output device **116** can receive any combination of information that the output device is capable of receiving.

Referring again to FIG. **3**, the windows **176** and **178** contain information that is not present in any input signal **110**. The user controller is configured to allow the user to select any information that may be generated within system **100** and which is suitable for a particular display device. The rejected packets information displayed in video window **176** may be determined by a signal analyzer (not shown) that analyzes input signal **110***b* to determine the number of defective packets received as part of the input signal **110***b*. The signal analyzer is one of the signal processors **126**. The signal analyzer then generates a video signal illustrating this information in a standard video signal format and stores the video signal in a data buffer **124** as a processed signal **158**.

In response to the user control signals **119**, the master controller transmits input processor control signals **120** to the input processor local controller **140** indicating the final version of each input signal **110** that will be required by the output processor **106** to produce the output signals **114** for the output devices **116**. For each required version, the master controller **102** also indicates the top left pixel at which that version will be displayed.

For the example input signals **110** and output video monitors **116** described above, the master controller instructs the input processor to prepare the following signals:

i. 400×300 pixel scaled version of video input signal **110***a*;

ii. 610×460 pixel scaled version of video input signal **110***a*;

iii. 200×113 pixel scaled version of video input signal **110***b*;

iv. 1140×640 pixel scaled version of video input signal **110***b*;

v. 160×120 pixel scaled version of video input signal **110***d*;

vi. 560×420 pixel scaled version of video input signal **110***e*; and

vii. 440×140 pixel video image illustrating rejected packet information for video signal **110***b*.

The master controller **102** does not instruct the input processor to produce a signal showing the date, time and analyzed information, which is required for graphics window **178** on video monitor **116***a*. This signal is produced in the output processor and is described below.

In response to the input processor control signals **120**, the input processor local controller **140** determines how the required versions of each input signal **110** can be produced and configures and couples the input ports **123**, A/D converters **150**, data buffers **124** and signal processors **126** to produce the required versions of each input signal. As described above, every signal stored in a data buffer **124** is assigned a unique global identification code.

In the present example, the input processor local controller **140** configures the input processor **104** as follows:

i. Store input signal **110***a* in data buffer **124***a*. Assign global identification code G**101** to the stored signal.

ii. Store input signal **110***b* in data buffer **124***b*. Assign global identification code G**102** to the stored signal.

iii. Store input signal **110***d* in data buffer **124***c*. Assign global identification code G**103** to the stored signal.

iv. Couple an A/D converter **150** between input port **123***e* at which input signal **110***e* is received to produce a digital version **110***e*' of input signal **110***e*. Store digital signal **110***e*' in data buffer **124***e*. Assign global identification code G**104** to the stored signal.

v. Couple video scaler **160***a* to memory system **122** to retrieve signal G**101** and produce a scaled version of 400×300 pixel scaled version of signal G**101**. The scaled version is stored in data buffer **124***f* and is assigned global identification code G**105**.

vi. Couple video scaler **160***b* to memory system **122** to retrieve signal G**101** and produce a 610×460 pixel scaled version of signal G**101**. The scaled version is stored in data buffer **124***g* and is assigned global identification code G**106**.

vii. Couple video scaler **160***c* to memory system **122** to retrieve signal G**102** and produce a 200×113 pixel scaled version of signal G**102**. The scaled version is stored in a memory buffer **124***h* and is assigned global identification code G**107**.

viii. Couple video scaler **160***d* to memory system **122** to retrieve signal G**102** and produce an 1140×640 pixel scaled version of signal G**102**. The scaled version is stored in data buffer **124***i* and is assigned global identification code G**108**.

ix. Couple video scaler **160***e* to memory system **122** to retrieve signal G**103** and produce a 160×120 pixel scaled version of signal G**103**. The scaled version is stored in data buffer **124***j* and is assigned global identification code G**109**.

x. Couple video scaler **160***f* to memory system **122** to retrieve signal G**104** and produce a 560×420 pixel scaled version of signal G**104**. The scaled version is stored in data buffer **124***k* and is assigned global identification code G**110**.

xi. Couple a signal analyzer (one of the signal processors **126**, as described above) to the memory system **122** to retrieve and analyze signal G**102**. The signal analyzer produces a video signal with a standard size of 320×200 pixels and metadata. The output of the signal analyzer is stored in data buffer **124***m* and is assigned global identification code G**111**.

xii. Couple a video scaler **160***g* to memory system **122** to retrieve signal G**111** and produce a 440×140 pixel scaled version of signal G**111**. The scaled version is stored in data buffer **124***n* and is assigned global identification code G**112**.

During the operation of input processor **104**, successive packets of each signal stored in a data buffer **124** are stored in the data buffer and previously stored packets are read out and then discarded. Some signals, such as input signal **110***a* are read by more than one device. Input signal **110***a*, identified by its global identification code G**101**, is read out by video scalers **160***a* and **160***b*. The data buffer **124***a* in which input signal **110***a* is buffered is configured to discard each packet in the input signal only after the packet has been read by both of the video scalers.

Signals G**105**-G**110** and G**112** are required to produce the output signals **114** for video monitors **116**. These signals are combined into packetized signal **112** using packetized signal formatter **128**, packetized signal buffer **130** and packetized signal generator **132**. The signals that are used to produce a packetized signal **112** are referred to herein as packet source signals for that packetized signal.

Reference is next made to FIG. **5**, which illustrates the format of the packetized signal **112**. In the present embodiment, the packetized signal **112** comprises a series of packets **190**, each of which contains the following fields:

i. global identification code for the signal from which the data in the packet was obtained;

ii. packet ordering information;

iii. a data payload;

iv. optional error detection and correction information and other metadata.

The packet ordering information, which may comprise a sequential packet number for each packet with the same global identification code, allows packets derived from the same packet source signal to be isolated from other packets, allowing the data in the packet source signal or a version of the data in the packet source signal to be re-assembled in the output processor **106**, as is described below.

In addition to packet ordering information, a video data packet may optionally also include frame ordering information, identifying the particular frame of video signal to which the packet corresponds.

The content and format of the data payload in each packet **190** will depend on the type of the data contained in the packet.

For example, if a packet source signal comprises a stream of data that is not organized as packets of information, then each packet **190** formed from that packet source signal contains a fixed amount of data in the data payload field. For example, if a packet source signal is a continuous stream of video data, then each corresponding packet **190** contains up to 320 bytes of the video data. In other embodiments, the amount of data in a particular packet may be fixed at a different size or may be variable.

If a packet source signal is organized as a series of packets of information, as in case of MPEG-2 encoded video or MP3 encoded audio or AES encoded audio, then the data payload may comprise the entire packet from the packet source signal.

Referring to FIG. **3**, video window **170** is a 400×300 pixel window in which signal G**105** will be displayed. Signal G**105** is created by video scaler **160***a* in a digital video

standard that comprises a stream of video data that is not separated into packets. If each pixel in the 400×300 pixel window **170** requires one byte of video data from signal G**105**, then an entire frame of video information for the window requires 120,000 bytes of data. If the standard according to which the signal is encoded provides that one complete horizontal line of information will be encoded in a single packet, then each frame will be encoded in **300** packets in data buffer **124***f*.

Packet signal formatter **128** retrieves the successive packets in data buffer **124***f* that encode each frame of video signal G**105** and produces a series of packets **190** that correspond to the retrieved packets. In the following discussion, pixel numbers are set out as n,m where n is the number of the pixel in a window in a horizontal line of a window or frame and m is the number of the line in the window or frame. Pixels and lines are numbered starting at 1. The packets **190** corresponding to one frame of the 400×300 pixel window include pixel data for the following ranges of pixels:

Packet	Pixel range
1	1.1-320.1
2	321.1-240.2 (i.e. pixels 321-400 on line 1 and pixels 1-240 on line 2)
3	241.2-160.3
4	161.3-80.4
5	81.4-400.4
6	1.5-320.5
•	•
•	•
•	•
373	241.298-160.299
374	161.299-80.300
375	81.300-400.300

Similarly, the packetized signal formatter reads the successive packets in data buffer **124***h* that encode each frame of video signal G**107** and produces a series of packets **190**. The packets **190** corresponding to one frame of the 200×113 pixel window include pixel data for the following ranges of pixels:

Packet	Pixel range
1	1.1-120.2
2	121.2-40.4
3	41.4-160.5
•	•
•	•
•	•
70	81.111-300.112
71	1.113-200.113

The last packet **190** used to packetize each frame of video signal G**107** contains data for only 200 pixels. The remaining data space is filled with null characters by the packetized signal formatter **128**. Alternatively, the last packet may have a shortened data payload length.

The packetized signal formatter **128** produces packets **190** corresponding to the data in the packet source signals. Depending on the data formats used for the packet source signals, the packetized signal formatter **128** may produce one or more packets **190** that correspond to the data in one packet of a packet source signal. For example, if packet source signal G**105** is encoded using a digital video standard that includes a complete frame of video in a single packet, then the packetized signal formatter **128** will produce **375** packets **190** corresponding to each packet in the packet source signal.

A single packet **190** may correspond to data from more than one packet of a packet source signal. For example, if packet source signal G**107** is encoded using a digital video standard that encodes a single line of a frame in each packet, then the packetized signal formatter will generate packets **190** corresponding to more than one packet in the packet source signal, since each of the packets **190** can contain data for 320 pixels and since each line in packet source signal G**107** is only 200 pixels wide.

The packetized signal formatter **128** proceeds to generate packets **190** for each of the packet source signals for the packetized signal **112**, as packets from the packet source signals are available from the corresponding data buffers **124**. As packetized signal formatter **128** produces packets **190**, it stores them in packetized signal buffer **130**. Packetized signal buffer **130** is a data buffer and may include memory space in memory system **122**.

Packetized signal generator **132** retrieves the packets **190** stored in packetized signal buffer **130** and generates packetized signal **112** at packetized signal output port **138**. Packetized signal **112** may be a synchronous signal. For example, in the present embodiment, the packetized signal is a synchronous signal transmitted at 2.5 Gbits/second. Referring to FIG. **5**, if there are no packets **190** in the packetized signal buffer **130**, the packetized signal generator transmits null characters **192** between packets. In other embodiments, the packetized signal generator may transmit the packetized signal **112** at any bit rate, depending on requirements and capabilities of the system **100**.

Reference is made to FIG. **1**. In the present embodiment, the packetized signal output port **138** will typically be coupled to the output processor (FIG. **1**) through a communication link **186**, which may be a data cable such as an electrical or optical cable. The data rate and other aspects of the data protocol used to transmit the packetized signal **112** correspond to the ability of the communication link **186**.

In other alternative embodiments, the packetized signal generator **132** may transmit the buffered packets **190** as an asynchronous stream of packets to the output processor using any communication protocol, including TCP/IP. In this case, the communication link **186** may be a cable or may be a LAN, WAN, the Internet or another communication system.

Reference is next made to FIG. **6**, which illustrates the output processor **106**. The output processor **106** has a packetized signal input port **202**, a packetized signal input stage **205**, a memory system **208**, a plurality of signal processor **210**, an output signal generator **212**, a bank **213** of digital-to-analog (D/A) converters **215**, a plurality of output ports **214** and one or more local signal generators **224**. Each packetized signal input stage **205** comprises a packetized signal input buffer **204**, a packetized signal extractor **206**. The display devices **116** are coupled to the output ports **214**. The output processor **106** also includes an output processor local controller **216** that receives output processor control signals **121** from the master controller **102** (FIG. **1**). The output processor local controller **216** is coupled to the various components of the output controller **106** through control lines **218** and controls the operation of those components in response to the output processor control signals **121**.

Memory system **208** includes a plurality of data buffers **220**.

The output processor control signals **121** received by the output processor local controller **216** indicate:

i. which signals (by their global identification codes) are required for each output signal **114**; and

ii. the format of each output signal **114** and, if the output signal is a video signal, the layout of the display including the position and dimensions of each window on the display, in accordance with the user control signals (FIG. **1**).

In the present embodiment, the output processor local controller **216** translates each global identification code into a local identification code.

The packetized signal **112** is received at input port **202** and is buffered in packetized stream input buffer **204**. As complete packets **190** are stored in buffer **204**, they are retrieved by packetized signal extractor **206**. The packetized signal extractor **206** determines the global identification code of each packet, translates the global identification code into the corresponding local identification code assigned by output processor local controller and stores packets **190** corresponding to each local identification code in a different data buffer **220**. Through this process, the data from each source signal for the packetized signal is isolated in a different data buffer **220**. Each isolated signal corresponds to one of the packet source signals for the packetized signal. The packet ordering information from each packet **190** is used to organize the packets **190** into their original sequence. Each isolated signal is referred to herein as an output source signal.

The local identification codes are used within the output processor **106** in place of the global identification code to distinguish between the different local source signals encoded in the packetized signal. In alternative embodiments, the global identification code may be used to identify the different local source signals within the output processor **106**.

The signal processors **210** may be used to reverse any compression or other signal processing operation applied in the input processor **104** (FIG. **2**) using the signal processors **126**. Depending on the signal processing operations performed in the input processor **104**, a reversing step may or may not be required. For example, if one of the input signals **110** was compressed using a standard compression format that may be directly used to produce an output signal **114**, then it is not necessary to reverse the compression. However, if the result of the compression step produced data that cannot be directly used to produce an output signal **114**, then a decompressor may be used to reverse the compression step. For example, one of the signal processor described above was a horizontal line filter, which compresses an input video signal **110** by discarding a portion of the video signal. This compression step may be reversed by interpolating the discarded data from the retained data. The resulting processed signal **222** is stored in a data buffer as an output source signal and is assigned a unique local identification code by the output processor local controller **216**.

In addition to reversing signal processing operations applied in the input processor, a signal processor **210** may be used to apply any other signal processing operations to a signal buffered in a data buffer **220** to produce an output source signal.

Reference is made to FIG. **3**. Graphics display window **178** on video monitor **116***a* contains a display of the current date, time and warning messages based on metadata extracted from the input packetized signal received with global identification code G**111**. The date and time information is generated by a local signal generator **224***a*, which operates under the control of the output processor local controller. Each local signal generator **224** produces an output source signal containing information and formatted for the use in an output signal. In this example, the local signal generator **224***a* generates a 200×150 pixel window containing the date and time. The output processor **106** may include other local signal generators **224** that produce other video, audio or data signals for inclusion in the output signals. In each case, the signal produced by the local signal generator is stored in a separate data buffer **220** and is assigned a local identification code. The metadata maybe generated using a signal analyzer among the signal processors **210**. The metadata produced by the signal analyzer is combined with the data and time information by a video signal generator in the output signal generator **212**. Similar metadata could also be generated in the input processor **104** (FIG. **1**) using a signal processor **126**.

The output signal generator **212** can generate a variety of digital output signals that may be used directly, or after conversion through a D/A converter **215**, by output device **116**. The output signal generator **212** may include one or more digital video signal generators, one or more digital audio signal generators or one or more data signal generators or any combination of video, audio and data signal generators. The data signal generators may include TCP/IP signal generators that produce an output signal **114** suitable for transmission using a communications link to a remote computer system, where the output signal may be decoded and used by a video, audio or data system. Similarly, the data signal generator may generate signals in any data format.

The output signal generator **212** extracts the data required for each output signal **114** from the appropriate data buffers **220** and generates the output signal **114**. For example, a video output signal generator receives instructions from the output processor local controller **216** identifying the output source signals (by their local identification code and the data buffer **220** in which they are buffered) required for an output signal, the layout of the output video signal in terms of the position and dimensions of each window, and the output source signal for each window. The video output signal generator extracts the video information for each frame from the corresponding data buffers **220** and generates each frame for the output signal **114**. If the video signal includes audio components, these audio components are similarly retrieved as output source signals and added to the output video signal **114**.

Similarly, the audio and data output signal generators retrieve the output source signals from the appropriate buffers and produce their output signals.

If the device coupled to a particular output port **214** requires an analog output signal, then one of the D/A converters **215** may be dynamically coupled between the output signal generator and the output port **214** to convert the digital output signal into a corresponding analog output signal.

To produce the video signals for the example output video monitors **116***a* (FIG. **3**) and **116***b* (FIG. **4**), the output processor local controller configures the output processor **106** to operate as follows:

i. Packetized signal extractor **206** operates as follows:
   a. Extract signal G**105** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**201** and store it as an output source signal in data buffer **220***a;*
   b. Extract signal G**106** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**202** and store it as an output source signal in data buffer **220***b;*
   c. Extract signal G**107** from the packetized signal **112** stored in packetized stream input buffer **204**, assign

it local identification code B**203** and store it as an output source signal in data buffer **220***c;*

d. Extract signal G**108** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**204** and store it as an output source signal in data buffer **220***d;*

e. Extract signal G**109** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**205** and store it as an output source signal in data buffer **220***e;*

f. Extract signal G**110** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**206** and store it as an output source signal in data buffer **220***f*; and

g. Extract signal G**112** from the packetized signal **112** stored in packetized stream input buffer **204**, assign it local identification code B**207** and store it as an output source signal in data buffer **220***g*.

ii. Local signal generator **224** produces a 200×150 pixel data and time window as described above. Assign local identification code B**208** to this signal and store it as an output source signal in data buffer **220***h*.

iii. Output signal generator **212** generates two output signals as follows:

a. One output video signal generator **212***a* extracts local signals B**201**, B**203**, B**205**, B**207** and B**208** from the corresponding data buffers **220** and produces an output signal **114***a*.

b. A second output video signal generator **212***b* extracts local signals B**202** B**204** and B**206** from the corresponding data buffers **220** and produces an output signal **114***b*.

iv. A D/A converter is coupled between video signal generator **212***a* and output terminal to convert output signal **114***a* into an analog output signal, which is then displayed by video monitor **116***a*.

v. Output signal **114***b* is coupled directly to output port **214***b*. Video monitor **116***b* receives and displays the digital output signal **114***b*.

Referring to FIG. **1**, the input processor **104** receives a plurality of different input signals **110**, which are asynchronous with respect to one another to be received at the input processor. The input signals are processed using signal processor **126** to put them into a format that is required for the output signals **114** and resulting processed signal (the packet source signals) are combined into a single packetized signal **112**. If an input signal **110** does not require any processing to be used as part of an output signal, the input signal **110** may be a packet source signal. The input processor allows a plurality of asynchronous data signals **110**, which may include video, audio and data signals, to be combined into a single packetized signal that may be transmitted using a single communication link **186**.

The output processor **106** receives the packetized signal **112** and isolates the different packet source signals and stores them in buffers **220** as output source signals. Local signal processor **210** in the output processor **106** may be used to reverse any signal processing operation performed in the input processor, if necessary or desired, to produce the output source signals. In addition, local signal generators **224** in the output processor **106** may be used to produce additional output source signals. One or more of the output source signals is used by a set of output signal generators **212** produce output signals **214**. If necessary, a D/A converter may dynamically be coupled between an output signal generator and an output port to convert the corresponding output signal into an analog form.

Together, the input processor **104** and output processor **106** allow a plurality of input signals to be transported from the input ports **108**, combined in a manner controlled by a user through the user controller **118** and then provided in the final combined manner to the output devices **116**. The input processor **104** and output processor **106** are coupled together using a single communication link **186**, eliminating the need to couple each of the input signals separately to the output processor **106**.

In the embodiment of FIGS. **1** to **6**, the input processor **104** includes video scalers **160** to scale video input signals **110** from their original dimension to other dimensions required for the output signals **114**. In some cases, this may required that the input video signal may be expanded to large dimensions, resulting in a packet source signal that requires a larger portion of the packetized signal bandwidth to transmit than the original input signal **110**. To reduce this increased usage of bandwidth, another embodiment of the invention may be configured to ensure that the scalers **160** in the input processor **104** are only used to reduce an input signal **110** to smaller dimensions. Video scalers may be included in the output processor as signal processors **210** to scale any input signal that must be enlarged before it is incorporated into an output signal.

Reference is next made to FIG. **7**, which illustrates a second input processor **304**. Input processor **304** is similar to input processor **104** (FIG. **2**) and similar components are given similar reference numbers. The input processor local controller **340** is coupled to the various components of the input processor **304**. These couplings are not illustrated to simplify the Figure. Input processor **304** has a plurality of packetized signal output stages **327**, each of which comprises a packetized signal formatter **328**, packetized signal buffer **330** and packetized signal generator **332**. Each packetized signal output stage **327** is capable of generating a packetized signal **112**. Each packetized signal **112** may include information of any one or more of the input signals **110**. Input processor **304** may be used to provide packetized signals to different output processors **106** (FIG. **7**). Each output processor can receive a packetized signal containing only information from packet source signals that are required to produce the output signals **114** produced by that specific output processor.

The number of packet source signals (which are generally different versions of input signals **110**) that can be transmitted in a single packetized signal may be limited by the amount of data in each signal and the bandwidth of the packetized signal. Particularly in the case of audio and video signals, which may be required to be received in real time at the output processor **106** in order to be properly displayed on an output device **116**. Input processor **304** allows each input source **108** to be coupled to a single input port on a single input processor and then be combined in different combinations for transmission to different output processors **106**. In one embodiment, an input processor includes four output stages to provide four packetized signals **112**, which may be coupled to four different output processors **106**.

Reference is next made to FIG. **8**, which illustrates a second output processor **403**. Output processor **403** is similar to output processor **106** (FIG. **6**) and similar components are identified with similar reference numbers. Output processor **403** has a plurality of packetized signal input stages **405**, each of which comprises a packetized signal input buffer **404** and a packetized signal extractor **406**. Each input stage **405** receives a packetized signal **112** at a packetized signal input port **202** and stores the data for each source signal for each packetized signal in a separate data buffer in

memory system **208**. This allows output processor **403** to receive a larger number of source signal than could be transmitted in a single packetized signal. Output processor **403** operates in the same manner to further process and generate output signals **114**, which may incorporate data from one or both of the packetized signals.

Reference is next made to FIG. **9**, which illustrates three input processors **304** and two output processors **403**. Input processor **304***a* receives eight input signals from eight sources **108***a*-**108***h* and generates two packetized signals **112***a* and **112***b*. Input processor **304***b* receives eight input signals from eight sources **108***i*-**108***p* and generates two packetized signals **112***c* and **112***d*. Input processor **304***c* receives eight input signals **108***q*-**108***x* and generates one packetized signal **112***e*. Output processor **403***a* receives packetized signals **112***a* and **112***c* and produces four output signals **114***a*-**114***d* at output terminals **714***a*-**714***d*. These output signals may include information from any of the sixteen input signals **108***a*-**108***p*. Output processor **403***b* receives packetized signals **112***b*, **112***d* and **112***e* and produces four output signals **114***e*-**114***h* at terminals **714***d*-**714***h*. The output signals **114***a*-**114***h* may include information from any of the twenty-four input signals **108***a*-**108***x*. In each case, each input source is coupled to only one input processor, but may be combined with the other input sources in the output signals.

A single packetized signal **112** produced by a input processor **104** or **304** may be coupled to more than one output processor by first routing the packetized signal **112** through a signal replicating device. For example, the packetized signal **112** may be replicated using a cable driver with multiple duplicate outputs or other signal replication device and transmitted on multiple communications links to more than one output processor.

Reference is next made to FIG. **10**, which illustrates a first packet router **502** coupled between a plurality of input processors **104** and **304** and a plurality of output processors **403**. Each of the input processors produces one or more packetized signals **112** that are received by the packet router **502**. Packet router **502** includes a packet router controller **504**, a plurality of packetized signal buffers **506**, a plurality of packetized signal extractors **508**, a plurality of data buffers **510** and a plurality of packetized link output stages. Packet router controller **504** controls the operation of packet router **502** through control lines **507**, which couple the packet router controller **504** to the other elements of packet router **502** (connections are not shown to simplify the Figure). Each packetized signal **112** is buffered in a packetized signal buffer **506**. As complete packets **190** of a packetized signal **112** arrive, a packetized signal extractor **508** determines the global identification code of each packet **190** and stores all packets **190** corresponding to the same global identification code in a single data buffer **510**. The packetized signal extractor operates under the control of the packet router controller, which designates the particular data buffer in which the packets **190** having the same global identification code are stored. Through this process, all packets having the same global identification code are isolated in a data buffer. The actual content of the packets **190** is not altered.

Each packetized signal output stage **527** includes a packet selector **528**, a packetized signal buffer **530** and a packetized signal generator **532**. The packet selector **528** operates under the control of the packet router controller **504** to extract packets **190** from one or more of the data buffers **510** and place them in packetized signal buffer **530**. The packetized router controller receives packet router control instructions **520** from the master controller **102** to generate one or more packetized signals containing corresponding to a set of specified global identification codes. For each requested packetized signal, the packet router controller instructs the packet selector **528** in one of the packetized signal output stages **527** to extract packets from the data buffers **510** corresponding to the specified global identification codes for that requested packetized signal. As the packets become available in the data buffers **510**, the packet selector **528** extracts them and stores them in the packetized signal buffer **530**. Packetized signal generator **532** operates in the same manner as packetized signal generator **132** to generate a new packetized signal **512**.

Each packetized signal output stage **527** operates independently of the others. Any number of packetized signals **512** generated by the packetized signal output stages may include packets from the same data buffer **510** (corresponding to a particular global identification code). Each data buffer is operated to ensure that each packet in the data buffer are not discarded until each packet has been read by every packetized signal output stage that requires the packet.

Through this operation, the packet router receives a plurality of packetized signals **112** and generates a new set of packetized signals **512**. The created packetized signals may comprise packets with any combination of global identification codes, allowing input signals received at different input processors to be combined in a single packetized signal **512** for delivery to an output processor **403**. Each output processor may receive multiple packetized signals **512** from a packet router **502** and may also receive one or more packetized signals **112** directly from an input processor.

Inserting packet router **502** between a plurality of input processors **104** and **304** and output processors **106** and **403** allows an input signal **110** received at any one of the input processors to be routed (possibly after being processed in the input processor by signal processor **126**) to any of the output processors for use by any of the output devices **116** coupled to an output processor. Each input signal is received in only one location, but may be used in multiple formats (by creating appropriate versions of the input signal using signal processor **126**) at multiple output devices **116**.

Reference is next made to FIG. **11**, which illustrates a second packet router **602**. Like packet router **502**, packet router **602** receives one or more packetized signals **112** from one or more input processors, and provides one or more packetized signals **612** to one or more output processors. Packet router **602** includes a packet router controller **604**, a plurality of packetized signal buffers **606**, a plurality of packetized signal extractors **608**, a memory system **609** including a plurality of packet storage locations **610** and a plurality of packetized signal output stages **627**. Packet router controller **604** controls the operation of packet router **602** through control lines **607**. Packet router controller **604** uses the packet storage locations **610** to temporarily store packets **190** from the packetized signal **112**.

Each packetized signal **112** is buffered in a packetized signal buffer **606**. As each complete packet **190** of a packetized signal **112** arrives, a packetized signal extractor **608** stores the complete packet **190** in one of the packet storage locations **610**. The packet router controller **604** maintains a storage location table **611** indicating whether each packet storage location **610** is available to store a newly arrived packet. The packet router controller **604** selects an available packet storage location **610** and instructs the packetized signal extractor to store the newly arrived packet **190** in the selected packet storage location **610**. The packet router

controller **604** then updates the storage location table **611** to indicate that the packet storage location **610** is not available to store another packet **190**.

The packet router controller **604** receives router control instructions **620** (similar to the router control instructions **520** received by packet router controller **502** (FIG. **10**)) from master controller **102** instructing the packet router controller to generate the packetized links **612** using packets **190** with specified global identification codes. The packet router controller **604** determines and assigns one of the packetized signal output stages **627** to generate each of the required packetized signals **612** and maintains a global identification code distribution table **613** correlating each global identification code with the packetized signal output stages **627** that require the global identification code. For example, a specified global identification code G**603** may be required for three of the outgoing packetized signals **612**. The three packetized signal output stage **627** used to generate those three packetized signals **612** are listed in the global identification code distribution table **613** in association with global identification code G**603**.

Each packetized signal output stage **627** includes a packet selector **628**, a packetized signal buffer **630** and a packetized signal generator **632**. Packet selector **628** reads packets **190** from the packet storage locations **610** as described below and stores the packets **190** in packetized signal buffer **630**. Packetized signal buffer **630** and packetized signal generator **632** operate in the same manner as packetized signal buffer **130** (FIG. **2**) and packetized signal generator **132** to produce the packets signals **612**.

In the storage location table **611**, the router controller **604** maintains the status of each packet storage location **610** by recording the number of packetized link output stages **627** that must read a packet **190** stored in the packet storage location before the packet **190** may be discarded. In the present embodiment, when a newly received packet **190** is stored in a free packet storage location **610**, the router controller records the number of packetized signal output stages **627** that require the packet **190** to generate a packetized signal **612**. The router controller **604** then instructs each of the packetized signal output stages **627** to read the packet **190** from the packet storage location **610**. The packet selector **628** in each packetized signal output stage reads the packet from the packet storage location **610** and indicates to the router controller **604** that it has done so. The router controller **604** than decrements the number of packetized signal output stages **627** that still require the packet in that packet storage location **610**. When each of the packetized signal output stages **627** that require the packet **190** have indicated that they have read the packet **190**, the packet is no longer required (i.e. the number of packetized signal output stages still requiring the packet **190** is zero), and the router controller **604** treats the packet storage location **610** as free.

For example, each packet **190** with global identification code G**603** may be required by packetized signal output stages **627***a*, **627***c* and **627***d* to produce outgoing packetized signal **612***a*, **612***c* and **612***d*. When a complete packet **190** with global identification code G**603** is received, router controller **604** selects a free packet storage location **610***b* and instructs the appropriate packetized signal extractor **608** to store the packet **190** in packet storage location **610***b*. The router controller **604** then sets the status of packet storage location **610***b* to "3", indicating that the packet **190** must still be read by three packetized signal output stages. The router controller than instructs packetized signal output stages **627***a*, **627***c* and **627***d* to read the packet **190**. Each of packet selectors **628***a*, **628***c* and **628***d* reads the packet **190** and indicates to router controller **604** that it has done so. Router controller **604** decrements the status of the packet storage location **610***b* as it receives each indication and when the status returns to "0", the packet storage location **610***b* is again free to store another packet **190**.

Reference is next made to FIG. **12**, which illustrates a third packet router **702**. Packet router **702** is similar in structure and operation to packet router **602** and similar components are identified with similar reference numbers, increase by one hundred. Packet router **702** has a plurality of signals processors **726** coupled to memory system **709**. Signal processor **726** operate in a similar manner to signal processors **210** (FIG. **6**) under the control of router controller **704**. Router controller **704** receives instructions from master controller **102** to perform one or more signal processing steps on the signal encoded with a particular global identification code. For example, the master controller may indicate that a video signal with global identification code G**734** must be scaled to dimensions of 800×600 pixels and the resulting processed signal is to be assigned global identification code G**783** and must be included in packetized signals **712***b* and **712***c*. Router controller **704** then configures the global identification code distribution table **713** to route packets with global identification code G**734** to a video scaler (not shown) among the signal processors **726**. The router controller **704** may also route the same packet to one or more other signal processors **726** or packetized signal output stages **727**. The video scaler (not shown) is configured to perform the video scaling operation and produces packets **190** identified with global identification code G**783**. These packets are stored in free packet storage locations as designated by the router controller **702**. The packets are then distributed to packetized signal output stages **727***b* and **727***c* using global identification code distribution table **713** and storage location table **711**.

Reference is next made to FIG. **13**, which illustrates an output stage **827** for an output processor. Output stage **827** may be used for video output signals and includes a buffer **840** coupled to output signal generator **212** and an output terminal **214**. The output signal **114** generated by the output signal generator **212** is stored in data buffer **840**. The stored signal is extracted from the data buffer **840** by a local output generator **842** which makes the output signal **114** available at an output terminal **214**. Optionally a D/A converter may be coupled between the local output generator **842** and output terminal **214** to convert the output signal into a corresponding analog output signal for use by an analog device coupled to terminal **214**.

The buffered stream is also extracted from the data buffer **840** by a remote output generator **844**, which packetizes the video output signal **114** into a graphics packet stream **846**. Each packet in the graphics packet signal **846** contains video data for a fixed number of pixels in the output signal **114**. Each packet has the following fields:

i. packet ordering information, such as video positioning information indicating the first pixel at which the video data is to be displayed; and

ii. the video data.

Each packet may also contain additional metadata including error correction and detection information, frame numbering information and other information.

The graphics packet signal **846** is transmitted to a graphics packet signal ports **848**, from which it may be transmitted across a communication link **850** to display interface **852** capable of receiving the graphics packet signal **846**, reconstructing the output signal **114** and displaying the output signal on a display monitor **116**.

Output stage **827** allows an output signal **114** to be replicated on two different display monitors. The output signal **114** may be replicated on any number of display monitors by providing a remote output generator for each such monitor.

Reference is next made to FIG. **14**, which illustrates a switch **860** coupled between a plurality of graphics packet signal ports **848**, which may be part of one or more output stage **827** in one or more output processors, and a plurality of display interfaces **852**. The switch **860** may be implemented as a physical switch, which may be manually operable or automatically operable under the control of the master controller **102** (not shown in FIG. **12**). The switch **860** may be implemented using a field-programmable gate array (FPGA) or with any other switching or packet routing technology. Switch **860** allows any of the graphics packet signal ports **848** to be coupled to any display interface **852**, allows any of the output signals **114** available at any of the graphics packet signal ports **848** to be displayed at any display monitor coupled to a display adapter **852**.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the spirit and scope of the invention, which is limited only by the appended claims.

We claim:

1. A method of producing a packetized signal comprising:

receiving one or more input signals;

automatically assigning each of the one or more input signals a unique global identification code;

deriving one or more derived signals from at least one of the input signals;

determining which of the one or more input signals and derived signals are required to generate the packetized signal;

upon determining the derived signals required to generate the packetized signal, processing at least one input signals to provide a corresponding processed signal, wherein the corresponding processed signal is required to generate the packetized signal, and buffering the corresponding processed signal in the memory system; and

upon determining the one or more input signals required to generate the packetized signal, buffering the one or more input signals required to generate the packetized signal in a memory system.

2. The method of claim **1** further comprising:

designating at least one of the input signals as a packet source signal, wherein each packet source signal comprises a series of packet source signal packets; and

generating at least one packetized signal, wherein each packetized signal includes a series of packetized signal packets and wherein each packetized signal packet is formed by:

retrieving one or more of the packet source signal packets corresponding to a single packet source signal;

extracting data from the retrieved packet source signal packets;

recording the unique global identification code of the single packet source signal and at least a portion of the extracted data in the packetized signal packet.

3. The method of claim **1** wherein each of the packet source signals comprises a series of packet source signal packets, and wherein each of the packetized signal packets is formed by retrieving one or more the packet source signal packets corresponding to a single packet source signal and including the packet source signal packet within the packetized signal packet.

4. The method of claim **1** wherein each of the packetized signal packets includes a global identification code, packet sequencing information and a data payload.

5. The method of claim **4** wherein at least one of the packet source signals is a video signal and wherein the packetized signal corresponding to the packet source signal includes video data and position information indicating how the video data is to be displayed on a video display.

6. The method of claim **5** wherein the position information includes pixel information indicating a position within a window of the video display at which the video data is to be displayed.

7. The method of claim **1** wherein the processed signal is a scaled version of the at least one of the one or more input signals.

8. The method of claim **1** wherein the processed signal is a compressed version of the at least one of the one or more input signals.

9. The method of claim **1**, further comprising:

buffering at least some of the one or more input signals.

10. An input processor comprising:

one or more input ports for receiving one or more input signals;

a memory system for buffering the one or more input signals;

one or more signal processors for retrieving at least one of the one or more input signals from the memory system and for processing the at least one input signal to generate processed signals and for storing the processed signals in the memory system;

an output port;

a packetized signal output stage for retrieving at least some of the one or more input signals and the processed signals from the memory system, designating the at least some input signals and the processed signals as packet source signals, generating a packetized signal containing a series of packetized signal packets, wherein each packetized signal packet contains a unique global identification code of one of the packet source signals and data corresponding to the same packet source signal, and for providing the packetized signal at the output port; and

an input processor local controller further configured to control the operation of the memory system, the one or more signal processors and the packetized signal output stage,

wherein the input processor local controller is configured to determine which of the one or more input signals and the signals derived from the one or more input signals are required to generate the packetized signal.

11. The input processor of claim **10** wherein the one or more signal processors include one or more video scalers for processing a scaled version of the input signal as a processed signal.

12. The method of claim **10** further comprising a step of processing at least one of the one or more packetized signal packets that is required to generate the one or more outgoing packetized signals to provide a set of processed packets, assigning the set of processed packets a unique global identification code and storing the processed packets in the packet storage location.

* * * * *

EXHIBIT D

US008270398B2

(12) **United States Patent**
Magarelli et al.

(10) **Patent No.: US 8,270,398 B2**
(45) **Date of Patent: Sep. 18, 2012**

(54) **SYSTEM AND METHOD FOR SIGNAL PROCESSING**

(75) Inventors: **Romolo Magarelli**, Woodbridge (CA); **Rakesh Thakor Patel**, Mississauga (CA); **Eric Fankhauser**, Burlington (CA); **Daniel G. Turow**, Burlington (CA)

(73) Assignee: **Evertz Microsystems Ltd.**, Burlington (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 273 days.

(21) Appl. No.: **11/738,675**

(22) Filed: **Apr. 23, 2007**

(65) **Prior Publication Data**

US 2007/0248087 A1 Oct. 25, 2007

Related U.S. Application Data

(60) Provisional application No. 60/793,716, filed on Apr. 21, 2006.

(51) **Int. Cl.**
H04L 12/50 (2006.01)
H04L 12/28 (2006.01)

(52) **U.S. Cl.** **370/360**; 370/419; 370/422

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,389,963 A * 2/1995 Lepley et al. 725/144
6,108,308 A * 8/2000 Flavin et al. 370/238
6,154,154 A * 11/2000 Suzuki et al. 341/61

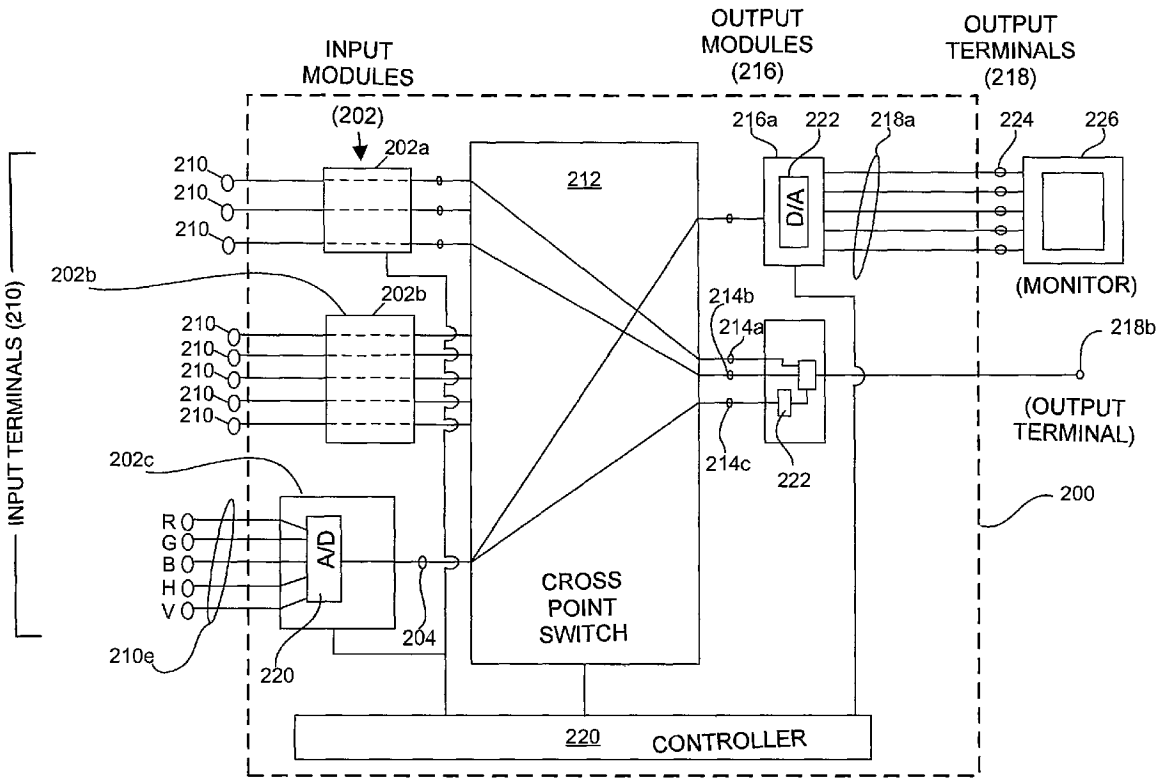
* cited by examiner

Primary Examiner — Nittaya Juntima
(74) *Attorney, Agent, or Firm* — Bereskin and Parr LLP/S.E.N.C.R.L., s.r.l.

(57) **ABSTRACT**

A plurality of systems for routing signals are described. The system includes a cross point switch that can couple any one of a plurality of input terminals to a plurality of output processor terminals. Signals received at the input terminals are coupled to corresponding output terminals and are processed by output modules. The resulting processed signals are provided at output terminals. In some embodiments, one or more input modules are provided to process input signal prior to routing through the cross-point switch.

14 Claims, 3 Drawing Sheets



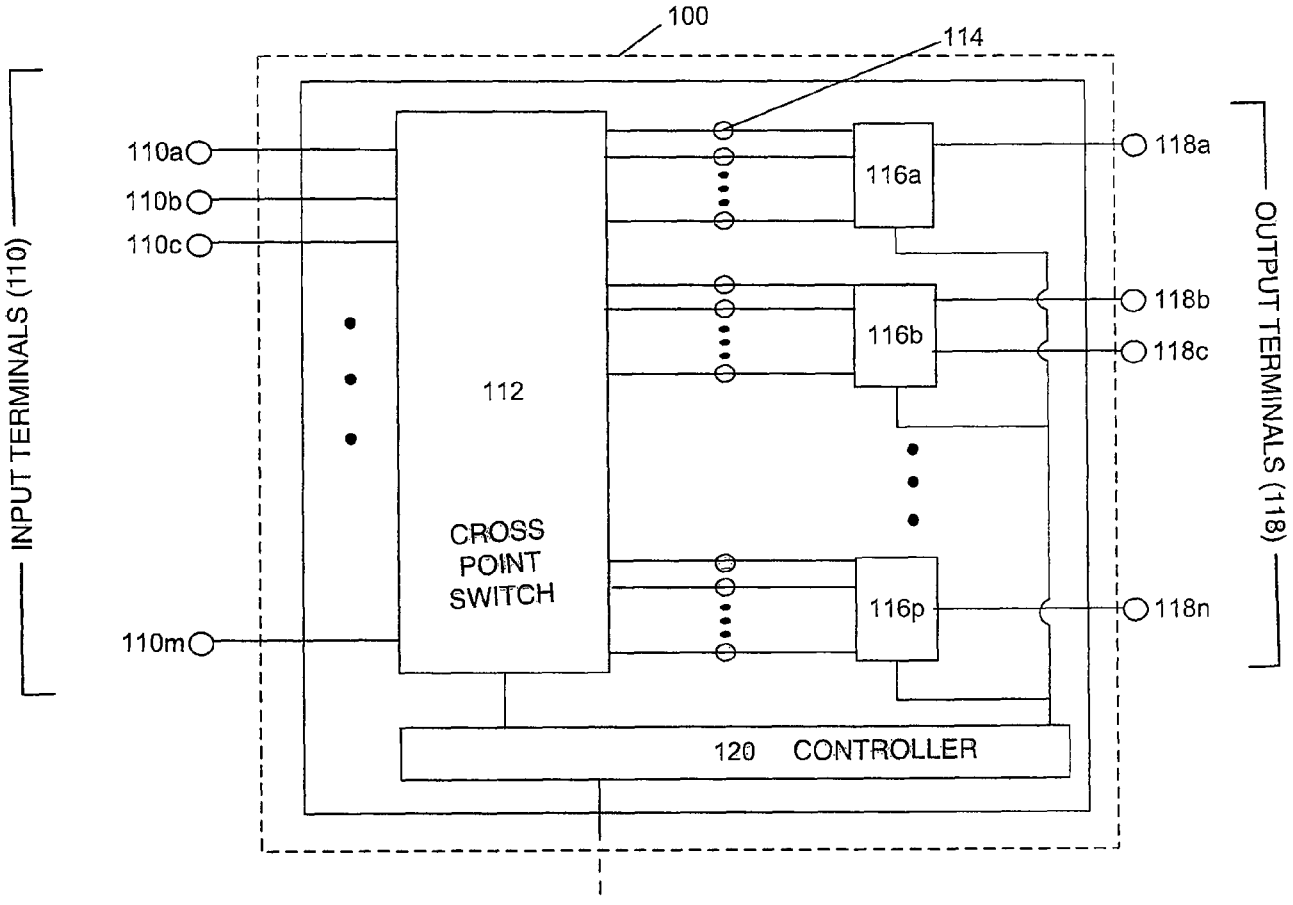

100
114
INPUT TERMINALS (110)
110a
110b
110c
110m
112
CROSS POINT SWITCH
116a
116b
116p
OUTPUT TERMINALS (118)
118a
118b
118c
118n
120 CONTROLLER

FIGURE 1

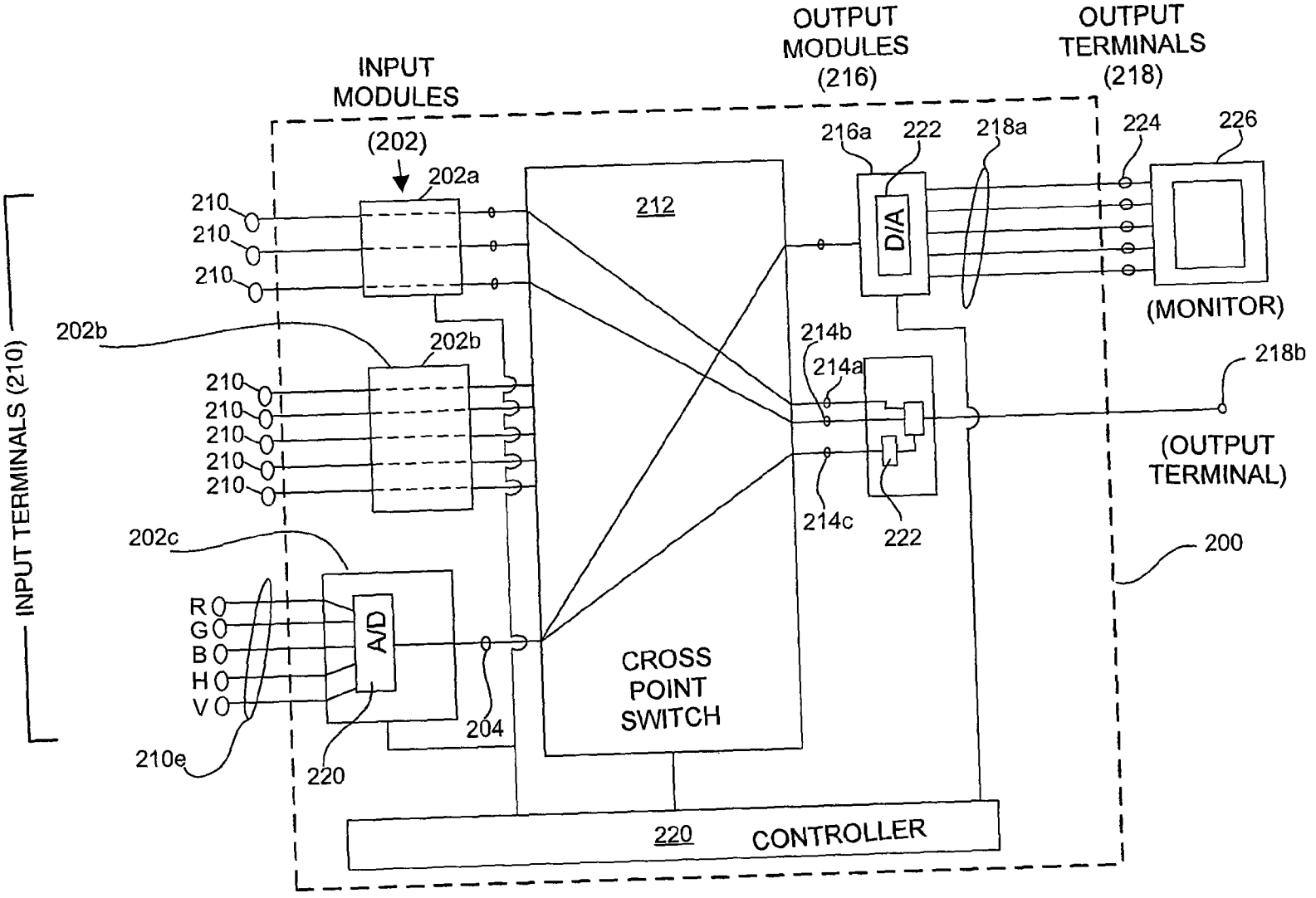

INPUT TERMINALS (210)
INPUT MODULES (202)
OUTPUT MODULES (216)
OUTPUT TERMINALS (218)
202a
202b
202b
202c
210
210
210
210
210
210
210
210
210
210e
R
G
B
H
V
A/D
204
220
212
CROSS POINT SWITCH
216a
222
218a
D/A
224
226
(MONITOR)
214b
214a
214c
222
218b
(OUTPUT TERMINAL)
200
220
CONTROLLER

FIG. 2

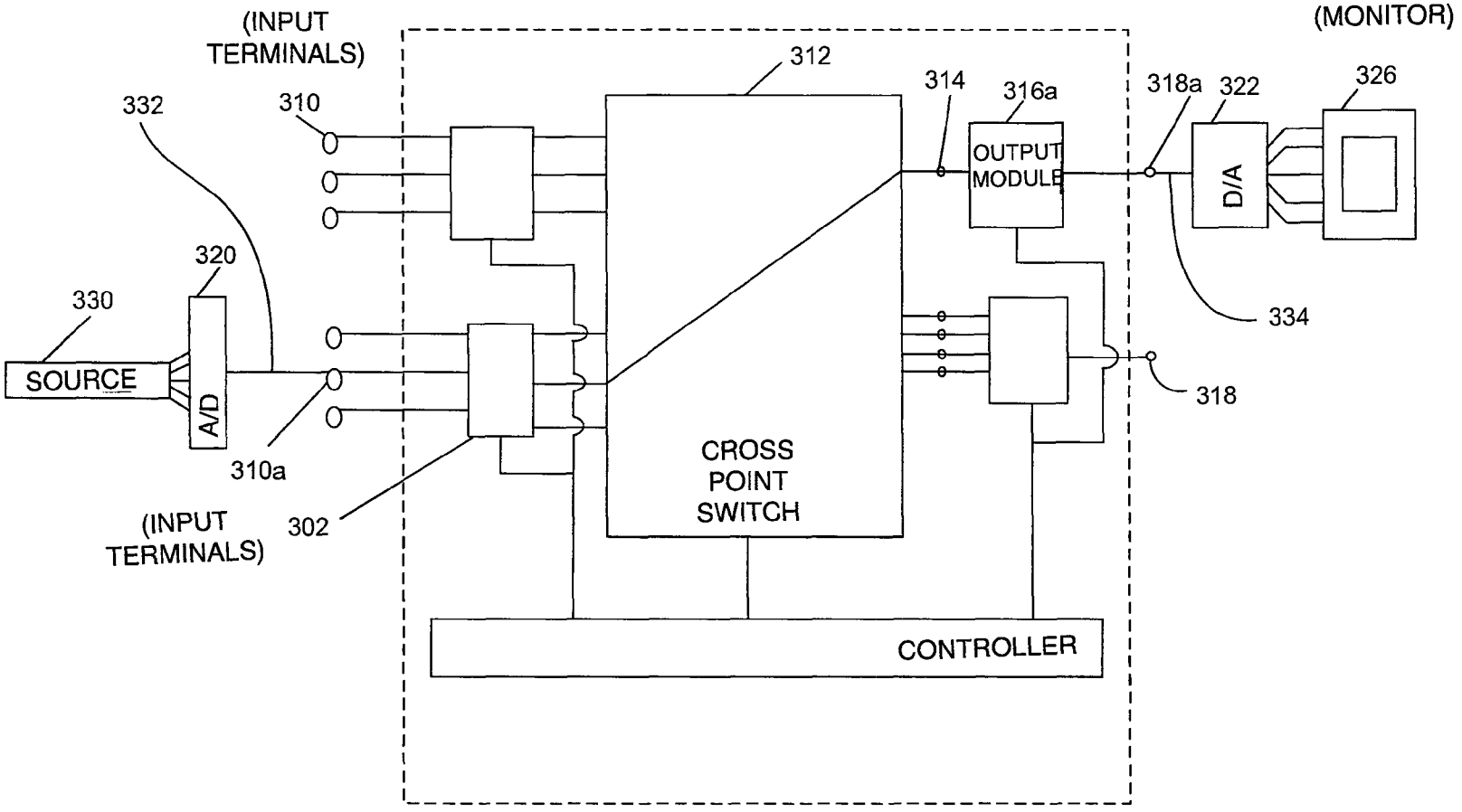

(INPUT TERMINALS)
(MONITOR)
312
314
316a
318a
322
326
332
310
OUTPUT MODULE
D/A
320
330
SOURCE
A/D
334
318
310a
CROSS POINT SWITCH
(INPUT TERMINALS)
302
CONTROLLER

FIG. 3

SYSTEM AND METHOD FOR SIGNAL PROCESSING

This application claims the benefit under 35 USC 119(e) of U.S. Provisional Patent Application Ser. No. 60/793,716, filed Apr. 21, 2006 and incorporated herein, in its entirety, by this reference to it.

FIELD

This invention relates to systems and methods for processing signals.

BACKGROUND

In many signal processing systems a plurality of input signals are received and some or all of these signals are processed to form one or more output signals. The output signals are provided to one or more output devices for further processing or reproduction. In systems where digital domain processing is required it is common to receive the digital signal and/or an analog input signal (which is then converted to a digital signal), pre-process and route and process the digitized signals. As technology advances, data signals contain an increasing amount of information and increasing numbers of signals are processed to form output signals. For example, television signals have increasingly higher resolutions and multiple of signals are often combined on a single display.

As the number and complexity of the input signals increases it can become increasing difficult to successfully route the digitized versions of the input signals to appropriate signal processors. For example, in one existing system, a plurality of input signals are received and are potentially initially digitized and packetized to provide a plurality of packetized signal streams each of which corresponds to a number of the input signals. The packetized streams are provided to a packet router, typically on a common transmission line. The packet router is configured to route a subset of signals to each of a number of output devices. Each output device then process each of the packetized signals that it receives to produce an output signal.

Such systems have bottlenecks associated with the initial pre-processing process in terms of processor speeds, overall memory bandwidth requirements, memory controller speeds, memory sizes and memory interfaces. In addition, such systems have bottlenecks associated with the transmission capacity of the transmission lines carrying the packetized input signals, which may be time division multiplexed communication lines. There are also bottlenecks in the packet router associated with processor speeds, overall memory bandwidth requirements, memory controller speeds, memory sizes and memory interfaces. There are bottlenecks associated with the data capacities of the communication links between the packet router and the output cards which may also be time division multiplexed or otherwise shared communication links. Finally, the output cards have similar bottlenecks as the input cards and the packet processors in terms of processor speeds, overall memory bandwidth requirements, memory controller speeds, memory sizes and memory interfaces.

There is a need for a simple and low cost system that alleviates these system bottlenecks and allows for a plurality of input signals to be selected and routed to one or more output processors each of which processes the input signals selected in relation to that output processor and each of which provides one or more output signals based on the input signals coupled to it.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described by way of example, with reference to the drawings, in which:

FIG. **1** illustrates an example system according to the present invention;

FIG. **2** illustrates another example system according to the present invention; and

FIG. **3** illustrates another example system according to the present invention.

DESCRIPTION

Reference is first made to FIG. **1** which illustrates a first exemplary system **100** according to the present invention. System **100** includes a plurality of frame input terminals **110**, an electrical cross point switch **112**, a plurality of output processors **116**. Each of the output processors **116** is coupled to the electrical cross point switch **112** at a plurality of processor input terminals **114**. Each of the processors **116** also has one or more frame output terminals **118**. The electrical cross point switch **112** is coupled to each of the input terminals **110** and receives an input signal, which may be an electrical, optical or radio frequency (RF) signal at one or more of the input terminals **110**. The electrical cross point switch **112** can be configured to couple every one of the input terminals to any one or more of the processor input terminals **114**. Cross-point switch **112** is a high bandwidth crosspoint switch capable of routing high bandwidth signals to any one or more of the processor input terminals. By utilizing this architecture processing speed constraints, interface speed constraints, routing constraints and overall memory bandwidth/controller speeds/memory sizes/memory interface constraints are mitigated.

Controller **120** may operate under the control of an external master controller or in response to signals received from a user, or both. Controller **120** controls the configuration of cross point switch **112** to switch a desired sub-set of input signals to each of the output processors **116**. Controller **120** may also control and monitor the operation of each of the output processors **116**.

Each of the output processors **116** is configured to receive one or more selected RF, optical or electrical input signals at the corresponding output processor input terminals. Each of the output processors processes the selected input signals coupled to it and produces one or more output signals at frame output terminals **118**.

Signal processing device **100** is a modular system. For example each of the output processors is configured to operate independently of the remaining output processors and provides its output signals at the corresponding frame output terminals **118** independently of the other devices. Input terminals **110** may be built into one or more input modules, each of which is designed to receive a particular kind of signal. For example, an input module may be provided to receive a set of video input signals while another input module may be provided to receive data signals while another input module may be provided to receive data signals and while another input module may be provided to simultaneously receive audio signals. Each of the output modules may require one or more of the input signals in order to produce its corresponding output signal (internal test patterns etc are possible in the

input modules). The cross point array couples each required input signal to each output module.

The signal flow between the frame input terminals **110** and output processor input terminals **114** operates in the electrical domain and each signal is switched through the cross point array in its original format, as received at the frame input terminals **110**. For example, video input signals may be received in a variety of formats. For example the input signals may be received as baseband NTSC, or ATSC signals, as L-Band frequency signals and in many other frequencies and formats. Audio signals may be received as digital audio signals, as 1 volt peak-to-peak signals or in other formats. Data signals may be received according to various standards that specify different voltage levels, frequencies and other signal characteristics.

The cross-point switch **112** does not distinguish between different types of signals and is operable to switch any input signal to any one or more of the signal processor input terminals **114**.

The output modules may have a variety of functions. For example one or more of the output modules may be a multi-image viewer which receives a plurality of video, audio, data signals and then combines the signals into a single video signal (potentially with embedded audio (such as an NTSC or ATSC signal). Controller **120** configures the cross-point switch **112** to route the required audio, video and data signals to the output processor. Controller **120** also configures the output processor to form the output video signal as requested by the master controller or the user. The resulting output signal is reproduced using an audio/video reproduction device.

FIG. **2** illustrates the configuration of another system **200**. Components of system **200** that correspond to components of system **100** are identified by similar reference numerals.

System **200** has a plurality of modular input modules **202**. Each input module **202** includes a plurality of input terminals **210**.

Input module **202***a* is an audio input module that has a plurality of audio input terminals that are coupled to the cross-point switch **212**. The audio input terminals may be any type or any combination of types of audio connectors. For example, these terminals may be RCA, BNC or other types of connectors.

Input module **202***b* is a video input module that is configured to receive video signals. Video input module **202***b* may have a variety of input terminals suitable for receiving video signals. In a professional or commercial video processing environment, the video input terminals **210** may comprise co-axial terminals, which have a single signal conductor. Other video input terminals may have multiple signal conductors. Optical video interfaces are also possible. Each of these conductors is coupled to the cross-point switch and each of them is separately coupled to an output processor that requires the video signal. Some input terminals may have a dozen or more conductors and each of these conductors is separately coupled to the cross-point switch.

Input module **202***c* illustrates an alternative to this arrangement. One system of transmitting video signals includes five analog components labeled R: red, G, green, B: blue, H: horizontal and V: vertical. The acronym RGBHV can be used to refer to this system. Five separate cables are used to couple devices that transmit and receives RGBHV video signals. Input module **102***c* has five input connectors—one each for the R, G, B, H and V components which together form a single video signal input terminal **210***e*. The input module **202***c* converts the five analog components into a combined digital signal using a high speed analog to digital converter **220**. The combined digital signal is coupled to a cross-point switch input terminal **204**. Cross-point switch **212** switches the combined digital signal to any output card that requires the video signal. For example, output module **216***a* includes a high speed digital to analog converter **222** that recreates the RGBHV signal on five output connectors **224** that form one output terminal **218***a*. The output connectors **224** are coupled to a monitor **226** for display.

Input module **202***c* is an example of an input module that pre-processes one or more input signals to produce a modified or combined signal (or signals) which are then routed through the cross-point switch **212**. The use of input modules cards that provide signal processing functions allows efficient use of the input signals and the cross-point switch.

The use of input module **202***c* to form the combined digital signal and output module **216***a* to recreate a RGBHV signal allows the single combined digital signal to be routed through cross-point switch **212**. The combined digital signal may be a digitally sampled and time division multiplexed representation of the RGBHV signal.

Reference is next made to FIG. **3**, which illustrates another system **300**. In FIG. **3** an analog to digital converter **320** receives an RGBHV signal from a signal source **330** and creates a combined digital signal. The combined digital signal is coupled through a single cable to input terminal **310***a*. Cross-point **312** couples the combined digital signal through an output module **316***a* to an output terminal **318***a*. From terminal **318***a*, the combined digital signal is transmitted to a digital to analog converter **322** that recreates the RGBHV video signal. The RGBHV video signal is transmitted to a monitor **326** for display. The configuration of FIG. **3** allows an RGBHV video signal to be efficiently transmitted in the form of a combined digital signal from A/D converter **320** through system **300** to D/A converter **322** on single conductor cables **332** and **334**. Single conductor cables can thus be used to replace the five cables normally required to transmit a RGBHV video signal and system **300**, which is capable of coupling a wide variety of signals, can be used to switch the RGBHV signal (in the form of a combined digital signal) to any one or more output modules and digital to analog converters.

The use of systems **100**, **200** and **300** is not limited to switching and processing any particular type or types of signals. A system according to the invention could be used to switch telecom/datacomm signals by installing appropriate input terminals (or input modules) and output modules. The output modules may be configured to process such telecom/datacomm signal to drop, add or multiplex components of the signals to produce modified output signals corresponding to one or more of the input telecom/datacomm signals. Furthermore, such signals could be combined with other signals coupled to the system to provide output signals.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the spirit and scope of the invention.

We claim:

1. A modular system for processing signals comprising:

a plurality of frame input terminals for receiving a plurality of input signals;

a plurality of input modules coupled to the frame input terminals to receive the input signals;

a cross point switch coupled to the input modules at a plurality of cross point switch input terminals to receive the input signals, wherein the cross point switch is configurable to couple one or more of the input signals to each of a plurality of processor input terminals;

a plurality of output modules, each output module including one or more output processors, wherein each of at least some of the output processors is coupled to one or more of the processor input terminals to receive one or more of the input signals, and wherein each output module has one or more frame output terminals; and

a controller coupled to:

the cross point switch to couple at least some of the frame input terminals to one of the processor input terminals whereby the input signals received at the frame input terminals are provided at the corresponding processor input terminals; and

at least one of the output processors to controllably configure the at least one output processor to processor input signals received at the one or more processor input terminals of the at least one output processor to provide one or more output signals at the one or more frame output terminals of the at least one output processor.

2. The system of claim **1** wherein at least one of the input modules is configured to process one or more input signals coupled to the at least one input modules and to provide a processed version of the input signals to the cross point switch.

3. The system of claim **1** wherein at least some of the frame input terminals are configured to receive radio frequency signals and at least some of the frame input terminals are configured to receive optical signals.

4. The system of claim **1** wherein at least some of the frame input terminals are configured to receive data signals.

5. The system of claim **1** wherein at least some of the frame input terminals are configured to receive video signals.

6. The system of claim **1** wherein at least some of the frame input terminals are configured to receive video signals.

7. The system of claim **1** where the controller is coupled directly to the cross point switch and to at least one of the output processors.

8. The system of claim **7** wherein at least one of the input modules is configured to process one or more input signals coupled to the at least one input module and to provide a processed version of the input signals to the cross point switch.

9. The system of claim **7** wherein at least some of the frame input terminals are configured to receive radio signals and at least some of the frame input terminals are configured to receive optical signals.

10. The system of claim **7** wherein at least some of the frame input terminals are configured to receive data signals.

11. The system of claim **7** wherein at least some of the frame input terminals are configured to receive video signals.

12. The system of claim **7** wherein at least some of the frame input terminals are configured to receive audio/video signals.

13. The system of claim **1** wherein at least one of the input modules is configured to process one or more input signals of a selected signal type and at least one of the output processors is configured to process signals of a corresponding signal type.

14. The system of claim **1** wherein at least one of the input modules is configured to process one or more input signals of a selected signal type and at least one of the output processors is configured to process signals of the selected signal type.

* * * * *

EXHIBIT E

US009654391B2

(12) **United States Patent**
Patel

(10) **Patent No.: US 9,654,391 B2**
(45) **Date of Patent: May 16, 2017**

(54) **VIDEO ROUTER**

(71) Applicant: **Evertz Microsystems Ltd.**, Burlington (CA)

(72) Inventor: **Rakesh Patel**, Mississauga (CA)

(73) Assignee: **Evertz Microsystems Ltd.**, Burlington (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 44 days.

(21) Appl. No.: **14/505,124**

(22) Filed: **Oct. 2, 2014**

(65) **Prior Publication Data**

US 2015/0092771 A1 Apr. 2, 2015

Related U.S. Application Data

(60) Provisional application No. 61/885,588, filed on Oct. 2, 2013.

(51) **Int. Cl.**

H04L 12/713	(2013.01)
H04L 12/931	(2013.01)
H04L 12/933	(2013.01)
H04L 12/773	(2013.01)

(52) **U.S. Cl.**

CPC ***H04L 45/60*** (2013.01); ***H04L 49/101*** (2013.01); ***H04L 49/206*** (2013.01)

(58) **Field of Classification Search**

CPC H04L 29/06027; H04L 65/4084; H04L 45/60; H04L 49/206; H04L 49/101; H04N 21/2318; H04N 21/2326; H04N 21/23406; H04N 21/2347; H04N 21/2362; H04N 21/2365; H04N 21/26606; H04N 21/4347; H04N 21/47202; H04N 21/6405; H04N 21/6587; H04N 19/40; H04N 21/2343; H04N 21/234309

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,976,134	B1 *	12/2005	Lolayekar et al.	711/148
7,194,550	B1 *	3/2007	Chamdani	H04L 69/08 709/227
8,776,161	B2 *	7/2014	Gazier	H04N 21/222 375/240.07
8,891,406	B1 *	11/2014	Shekhar	370/255

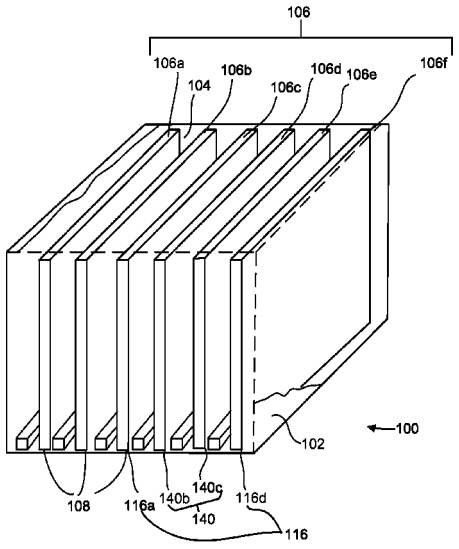
* cited by examiner

Primary Examiner — Ayaz Sheikh
Assistant Examiner — Debebe Asefa
(74) *Attorney, Agent, or Firm* — Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.

(57) **ABSTRACT**

The embodiments described herein provide a video router with integrated control layers and a method of operating the same. The video router includes line cards and fabric cards coupled to a controller communication network. The line cards and fabric cards include crosspoint switches and card controllers. Each card controller controls the operation of the corresponding crosspoint switches. Each crosspoint switch includes a plurality of input switch terminals and output switch terminals coupled to a backplane, providing signal communication paths between the line and fabric cards. The configuration of at least some of the crosspoint switches may be controlled by the controller on the same card or on other cards. The video router may include a switch configuration table to track the coupling of input and output terminals through each of the cross-point switches.

20 Claims, 7 Drawing Sheets



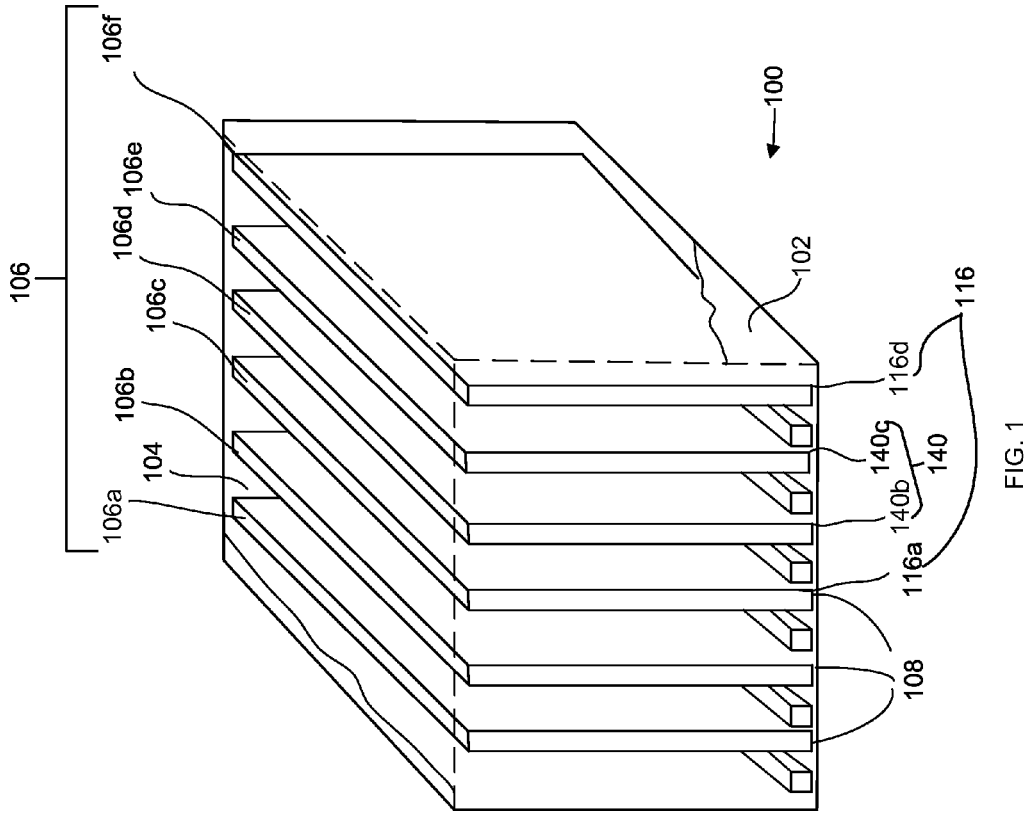

106
106f
106e
106d
106c
106b
104
106a
100
102
116
116d
140c
140
140b
116a
108

FIG. 1

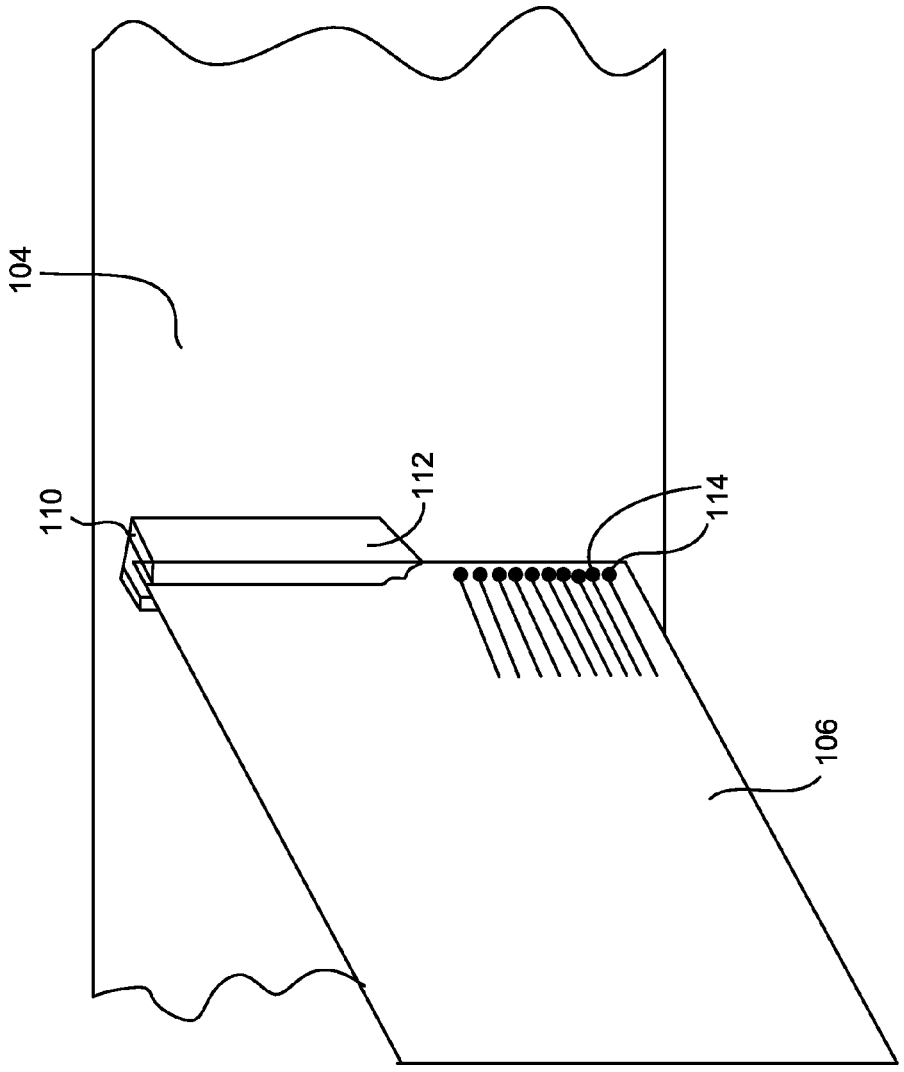


FIG. 2

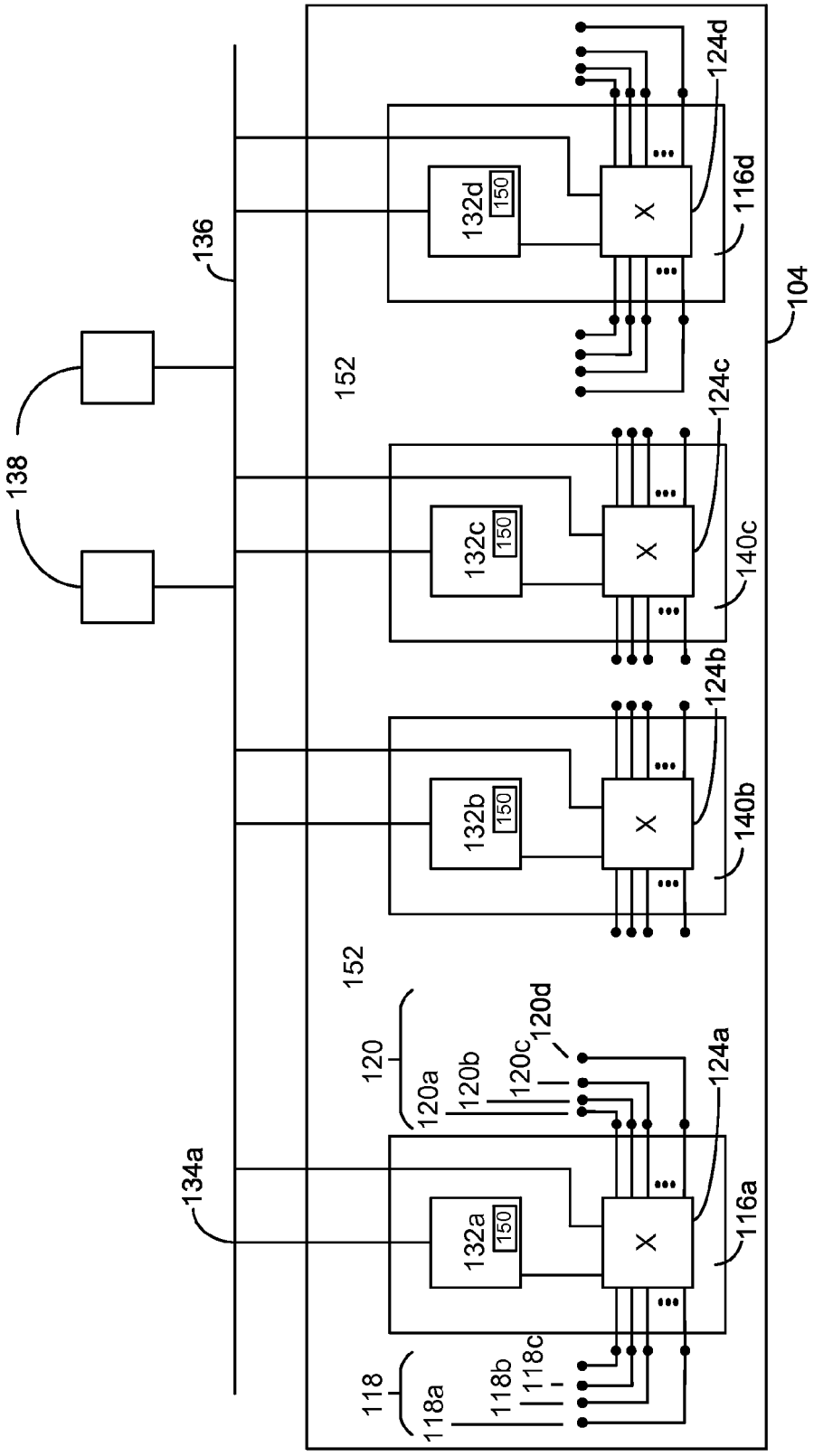

104
136
138
152
152
120
120a
120b
120c
120d
118
118a
118b
118c
134a
132a
132b
132c
132d
150
X
116a
116d
124a
124b
124c
124d
140b
140c

FIG. 3

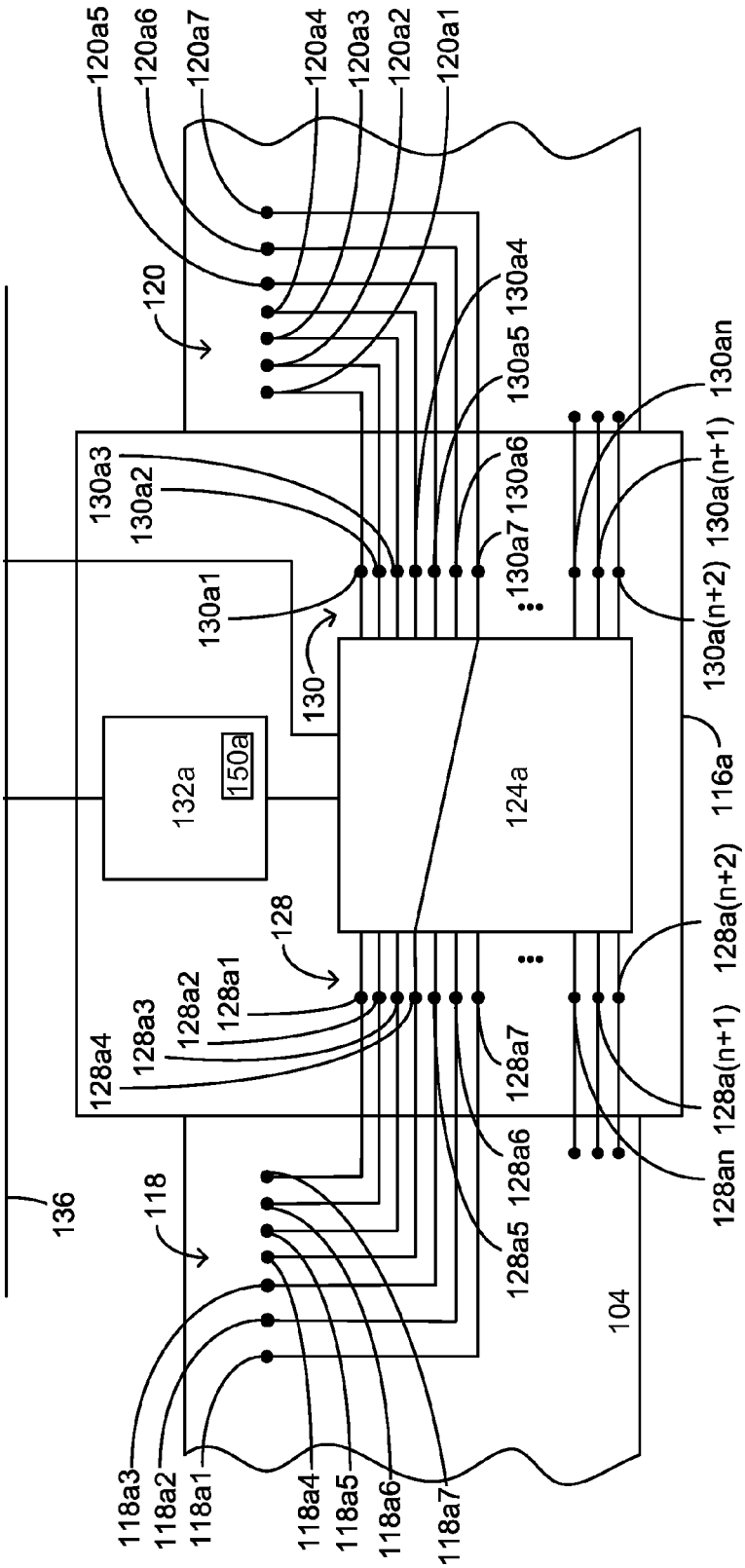

120a5
120a6
120a7
120a4
120a3
120a2
120a1
120
130a4
130a5
130an
130a3
130a2
130a6
130a7
130a(n+1)
130a1
130a(n+2)
130
132a
150a
124a
116a
128
128a(n+2)
128a3
128a2
128a1
128a4
128a7
128a(n+1)
128a6
128an
128a5
136
118
104
118a3
118a2
118a1
118a4
118a5
118a6
118a7

FIG. 4

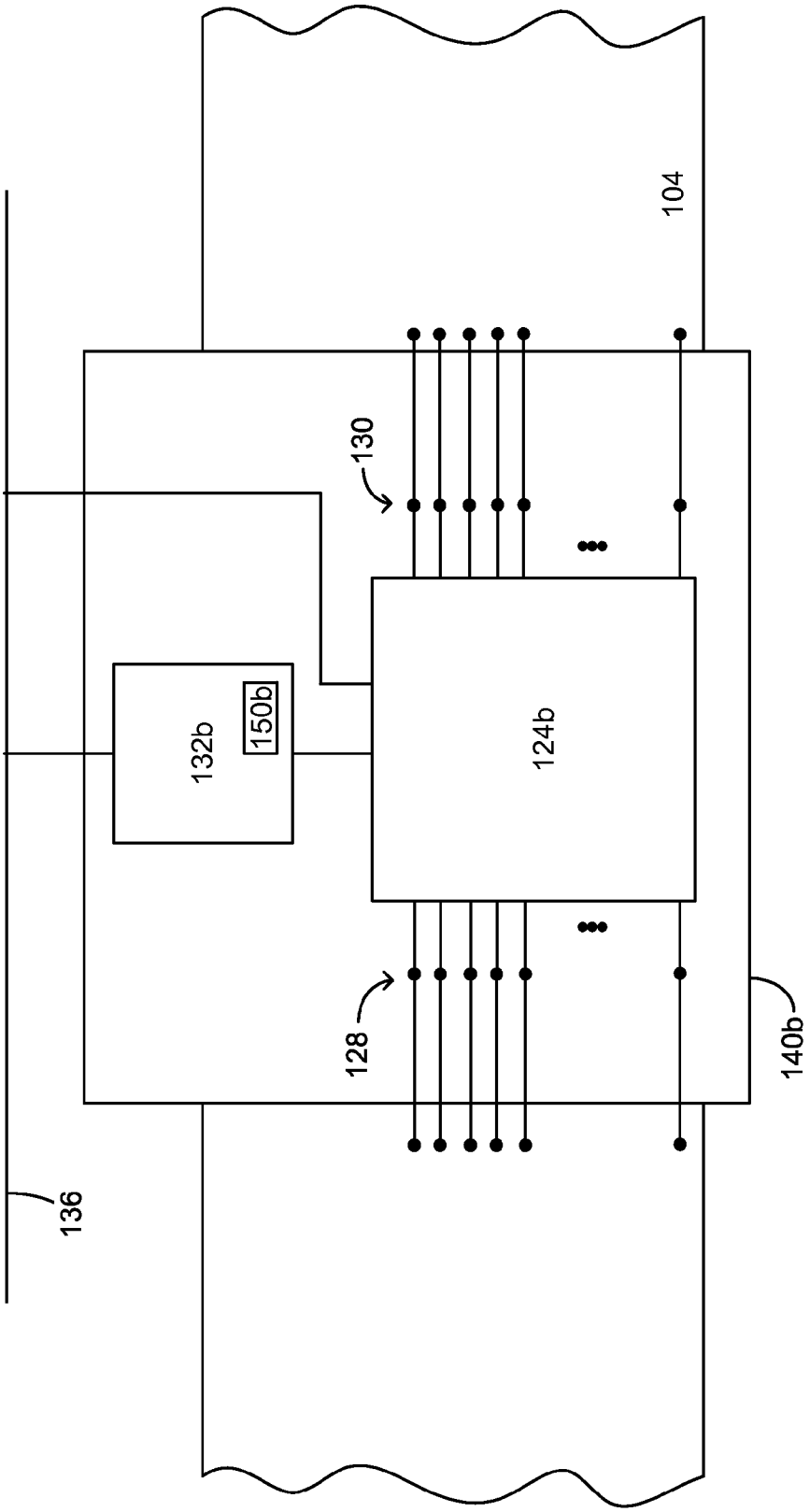


FIG. 5

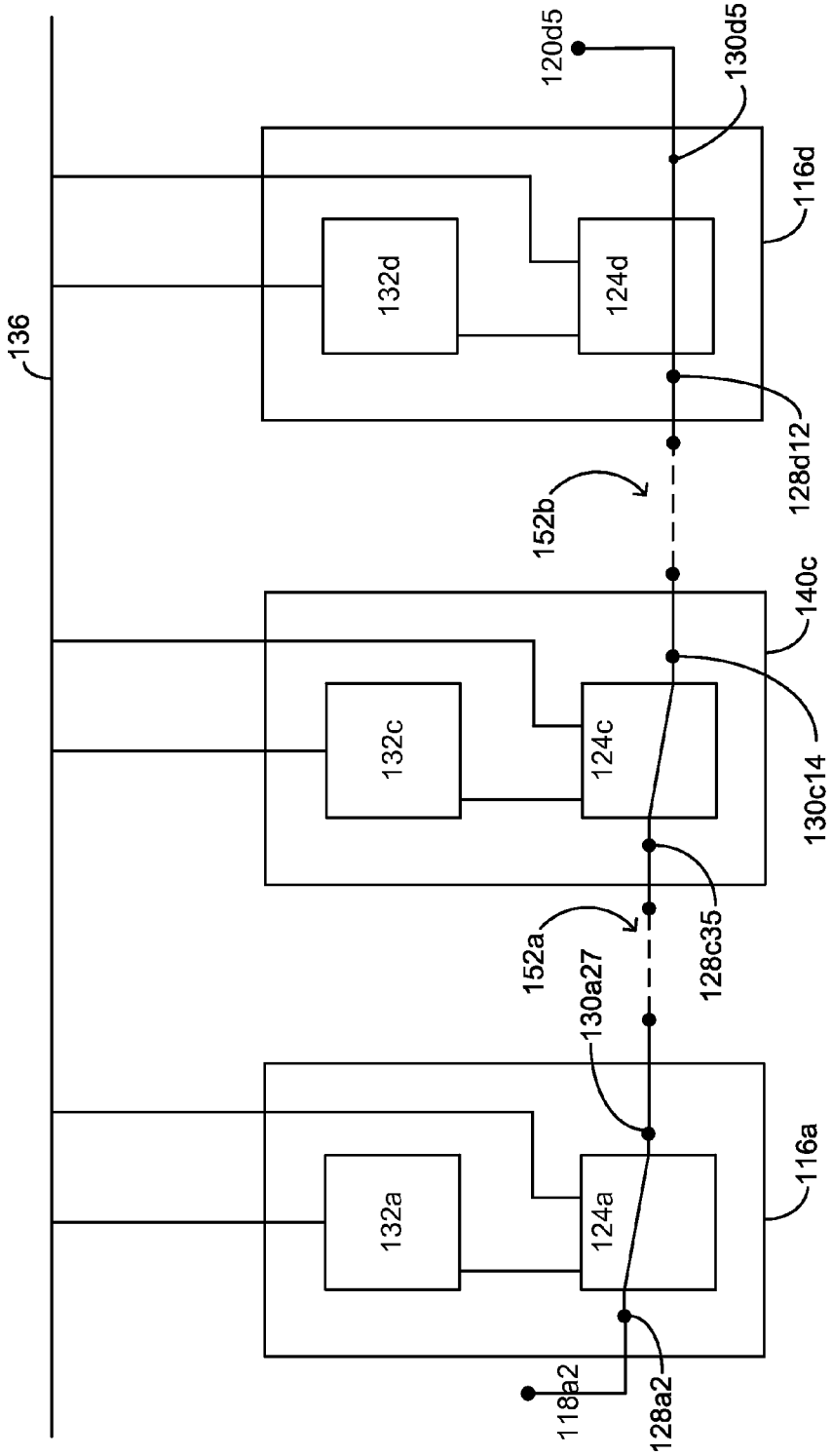

120d5
130d5
116d
132d
124d
136
128d12
152b
140c
132c
124c
130c14
128c35
152a
130a27
116a
132a
124a
118a2
128a2

FIG. 6

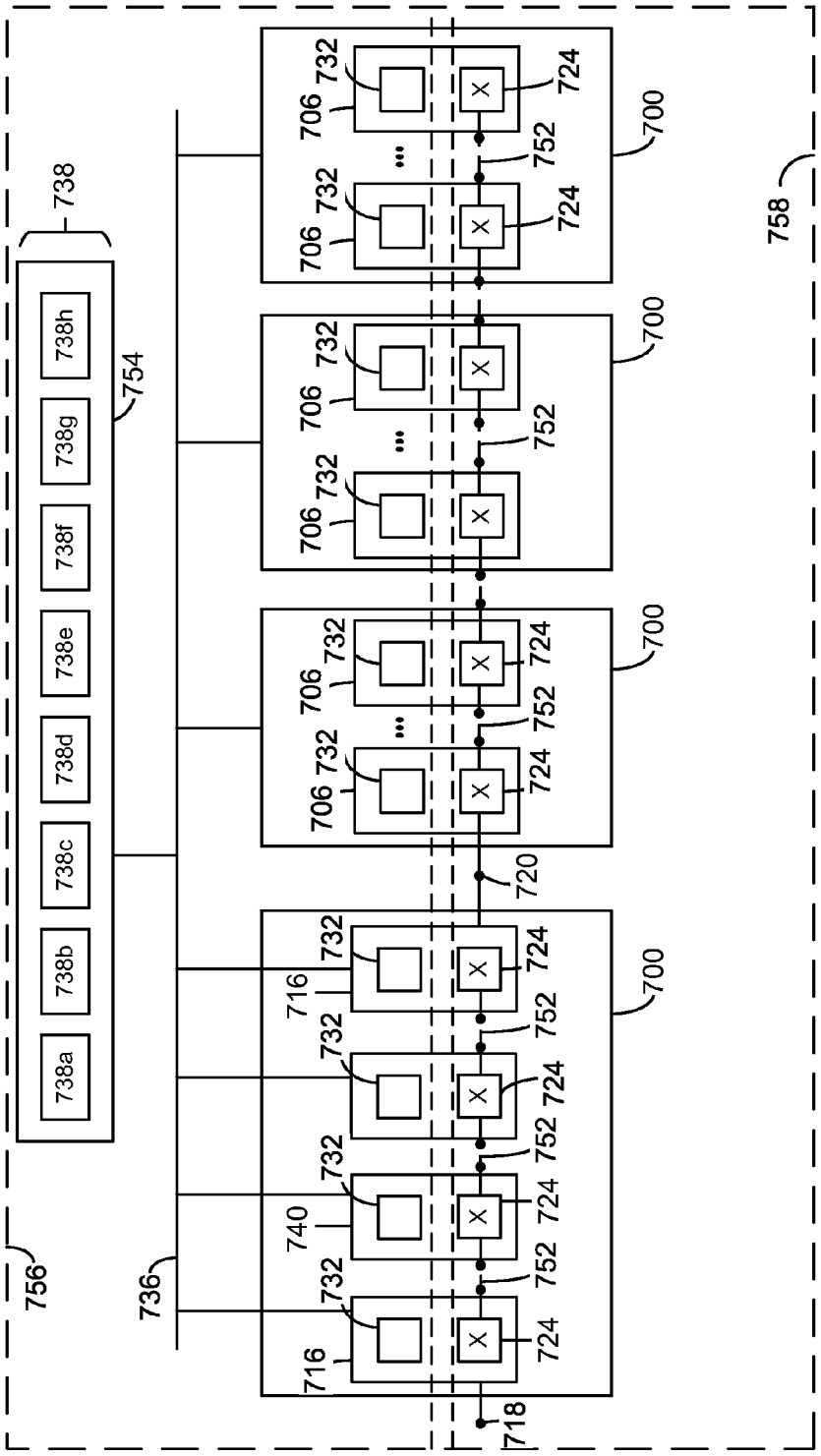

738
738a
738b
738c
738d
738e
738f
738g
738h
754
756
758
736
700
706
732
724
752
740
716
720
718

FIG. 7

VIDEO ROUTER

FIELD

The described embodiments relate to routers for video signals and other data streams.

BACKGROUND

The number of devices coupled to data communications networks is increasing rapidly. The routing of data streams from and to such devices is increasingly more complex and difficulty in allocating efficient routes, or even any route at all, in various components in a communication network can affect the quality of service delivery to a user of a device. For example, communication networks typically contain routers that couple an input data stream received at an input port to an output port at which the data stream is available to a downstream device. As the size of routers increases (i.e. as the number of input and output ports on a router increases, then complexity of creating efficient routing within the router and between network devices increase non-linearly.

It is desirable to provide an efficient system and methods that allows a network device to efficiently configure routes for data streams.

SUMMARY

Various embodiments relate to video routers having integrated control layers.

The video routers include line cards and fabric cards coupled to a controller communication network. The line cards may include input ports or output ports or both for receiving and providing external video signals and other signals. The line cards and fabric cards include crosspoint switches and card controllers. Each card controller controls the operation of the corresponding crosspoint switches. Each crosspoint switch includes a plurality of input switch terminals and output switch terminals coupled to a backplane, providing signal communication paths between the line and fabric cards. The configuration of at least some of the crosspoint switches may be controlled by the controller on the same card and on other cards. The video router may include a switch configuration table to track the coupling of input and output terminals through each of the cross-point switches. Each of the controllers can access the configuration table. The configuration table may be recorded in a central location or may be distributed across multiple locations or both. Some of the configuration table locations may be in some of the controllers or in other data storage elements.

The backplane includes a plurality of static point-to-point backplane connections that couple output switch terminals on one card to input switch terminals on another card. In various embodiments, the backplane may include as many or as few backplane connections between output switch terminals and input switch terminals.

The controllers configure the crosspoint switches, including crosspoint switches on other cards, to provide connections between between input switch terminals and output switch terminals and to provide connections for data streams between input ports and output ports. Each such path or connection may be routed through one or more line cards, fabric cards and one or more backplane connections or any combination of such elements by appropriate configuration of the cross-point switches.

In some embodiments, various data streams may be assigned various priority levels. In some cases, paths required to couple different data streams may conflict such that it may not be possible to maintain all such paths simultaneously. The controllers may be configured to maintain some of the paths in preference to other paths based on the relative priorities of the data streams.

In some embodiments, a controller may establish multiple paths for particular data streams to provide redundancy.

In some embodiments, a controller may not be able to organize a path for a particular data stream. The controller may be able to send a routing request to a supervisor, which may be an external device to the router. The supervisor may monitor and control the configuration of the crosspoint switches in the router and optionally in other routers. In some embodiments, multiple or duplicate supervisors may be provided.

In response to a routing request, a supervisor may modify the configuration of one or more crosspoint switches on one or more cards to provide a desired route. In other embodiments, the supervisor may change provide a response to a controller which may then configure a route based on the response. Subsequently, a controller may use a response to a prior request or a route created by a supervisor in response to a prior request to provide the same route without sending a routing request to a supervisor.

These and other aspects of the invention are further described in the description of example, embodiments set out below.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the applicant's teachings described herein, reference will now be made, by way of example only, to the accompanying drawings which show at least one exemplary embodiment, and in which:

FIG. **1** is a cross-section of a video router according to an example embodiment;

FIG. **2** is a cross-section of a video router according to another example embodiment;

FIG. **3** is a block diagram of a video router according to an example embodiment;

FIG. **4** is a block diagram of a video router according to another example embodiment;

FIG. **5** is a block diagram of a video router according to another example embodiment;

FIG. **6** is a block diagram of a video router according to another example embodiment;

FIG. **7** is a block diagram of a control hierarchy of a video router according to an example embodiment.

For simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements.

DESCRIPTION OF EXEMPLARY EMBODIMENTS

Reference is first made to FIGS. **1** and **2**, which illustrates a first video router **100** with an integrated control layer. Router **100** includes a frame or housing **102**, a backplane **104** and a plurality of cards **106**, such as a first card **106***a*, a second card **106***b*, a third card **106***c*, a fourth card **106***d*, a fifth card **106***e* and a sixth card **106***f*. The frame **102** includes

a plurality of frame slots **108** in which cards may be received and held in place. The backplane **104** includes backplane connector **110** corresponding to each slot **108** and each card **106**. Each backplane connector includes a plurality of backplane pins or contacts **112**. Each card **106** includes a plurality of card pins or contacts **114**, each of which corresponds to a backplane pin **112** of the corresponding backplane connector. When a card **106** is installed in frame **102**, the card pins **114** couple with corresponding backplane pins **112** making an electrical connection through which a data signal may be transmitted.

Cards **106** may include various types of cards. For example, some of the cards may be line cards **116**, such as a first line card **116***a* and a second line card **116***d*, which include input ports or output ports for respectively receiving and transmitting data signals, or both input and output ports. Other cards **106** may be fabric cards **140**, such as a first fabric card **140***b* and a second fabric card **140***c*, which facilitate switching of signals between various input and port ports.

Reference is made to FIGS. **3**, **4** and **5**, which schematically illustrate components of router **100**. In the present example embodiment, each input port **118** or output port **120** on a line card **116***a* is coupled to an external signal through the backplane **104**. In the illustrated embodiment of FIG. **3**, input port **118** comprises a first input port **118***a*, a second input port **118***b* and a third input port **118***c*, and output port **120** comprises a first output port **120***a*, a second output port **120***b*, a third output port **120***c* and a fourth output port **120***d*. The backplane may, for example, include a pass-through connector to which a line card port **118**, **120** may be coupled within frame **102** and to which a cable (not shown) may be coupled on the rear of the backplane. The line card port **118**, **120** is electrically coupled to the cable (not shown), allowing the line card to receive or transmit a data signal on the cable. In other embodiments, line card ports may be directly coupled to a cable or may be coupled to a cable through the backplane using a coupling other than a pass-through connector.

Line card **116***a* includes a line card crosspoint switch **124***a* with a plurality of switch terminals. In this example, crosspoint switch **124***a* has a plurality of input switch terminals **128** and a plurality of output switch terminals **130**. Each input port **118** is coupled to at least one input switch terminal **128** and each output port **120** is coupled to at least one output switch terminal **130**. In addition, a plurality of input switch terminals **128** are coupled to the backplane **104** through the corresponding backplane connector **110**. A plurality of output switch terminals **130** are coupled to the backplane **104** through the corresponding backplane connector **110**.

Line card **116**a also includes a line card controller **132***a* that is coupled to crosspoint switch **124***a* and which provides control signals to couple or decouple particular input switch terminals **128** to particular output switch terminals **130**. Card controller **132***a* is coupled to a controller communication network **136** at a control system terminal **134***a* through which the card controller **132***a* may communicate with other cards **106** and with external control devices such as an external supervisor **138**. In some embodiments, a line card controller **132***a* may be coupled to controller communication network **136** through the backplane or through another communication bus in frame **102** to which the line card is couple when installed in the frame.

Each fabric card, such as fabric card **140***b* includes a card controller **132***b* and a crosspoint switch **124***b*, which are coupled together and operate in a manner similar to the card controller **132***a* and crosspoint switch **124***a* of line card **116***a*. Crosspoint switch **124***b* includes a plurality of input switch terminals **128** and output switch terminals **130** that are coupled to the backplane **104**. The crosspoint switch **124***b* may be configured to couple any of the input switch terminals **128** to any of the output switch terminals **130** under the control of card controller **132***b*. As illustrated in FIG. **3**, router **100** further includes a fabric card **140***c*, which includes a card controller **132***c* and a crosspoint switch **124***c*, and a line card **116***d*, which includes a card controller **132***d* and a crosspoint switch **124***d*.

Each crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* in router **100** is coupled to the controller communication network **136** through which the configuration of the crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* may be changed by card controller **132**a, **132***b*, **132***c*, **132***d* on other cards **106**.

Router **100** also includes a switch configuration table or database **150**. Database **150** records the current setting for every input switch terminal and output switch terminal in all cross-point switches **124***a*, **124***b*, **124***c*, **124***d* in the router **100**. For example, part of the contents of database **150** may be:

Switch	Terminal	Setting
124a	128a1	Coupled to 130c3
124a	128a2	Coupled to 130a27
124a	128a3	Open
124a	128a4	Coupled to 130a8
. . .	. . .	. . .
124a	130a3	Coupled to 128a1
124a	130a4	Open
124a	130a5	Coupled to 128a2
124a	130a6	Open
124a	130a7	Coupled to 128a2
124b	130a8	Coupled to 128a4
. . .	. . .	. . .
124b	128b1	Open
. . .	. . .	. . .
124c	128c35	Coupled to 130c14
. . .	. . .	. . .
124c	130c14	Coupled to 128c35
. . .	. . .	. . .
124d	128d12	Coupled to 130d5
124d	130d5	Coupled to 128d12
. . .	. . .	. . .

where router **100** of FIG. **4** comprises a first input port **118***a***1**, a second input port **118***a***2**, a third input port **118***a***3**, a fourth input port **118***a***4**, a fifth input port **118***a***5**, a sixth input port **118***a***6**, a seventh input port **118***a***7**, a first output port **120***a***1**, a second output port **120***a***2**, a third output port **120***a***3**, a fourth output port **120***a***4**, a fifth output port **120***a***5**, a sixth output port **120***a***6**, a seventh output port **120***a***7**, a first input switch terminal **128***a***1**, a second input switch terminal **128***a***2**, a third input switch terminal **128***a***3**, a fourth input switch terminal **128***a***4**, a fifth input switch terminal **128***a***5**, a sixth input switch terminal **128***a***6**, a seventh input switch terminal **128***a***7**, an nth input switch terminal **128***a*n, a (n+1)th input switch terminal **128***a*(n+1), a (n+2)th input switch terminal **128***a*(n+2), a first output switch terminal **130***a***1**, a second output switch terminal **130***a***2**, a third output switch terminal **130***a***3**, a fourth output switch terminal **130***a***4**, a fifth output switch terminal **130***a***5**, a sixth output switch terminal **130***a***6**, a seventh output switch terminal **130***a***7**, an nth output switch terminal **130***a*n, a (n+1)th output switch terminal **130***a*(n+1) and a (n+2)th output switch terminal **130***a*(n+2).

Database **150** is accessible to each of the controllers **132***a*, **132***b*, **132***c*, **132***d*. In some embodiments, the database **150**

may be recorded in a central location, for example, in one of the controllers **132***a*, **132***b*, **132***c*, **132***d* where the local controller **132***a*, **132***b*, **132***c*, **132***d* may access the database directly and each of the other controllers may access the database through the controller communication network **136**.

In other embodiments, the database may be a distributed database with components that are located in multiple locations within router **100**. For example, components of database **150***a*, **150***b* may be located in each of the controllers **132***a*, **132***b*, as is illustrated in FIGS. **4** and **5**. Each controller **132***a*, **132***b* may contain the status of the cross-point switch **124***a*, **124***b* in the same card **106**. Controllers **132***a*, **132***b*, **132***c*, **132***d* on other cards **106** may access the status of non-local cross-points switches through the controller communication network **136**.

In other embodiments, the database may be recorded in a data storage device or system that is external to router **100**, but which is accessible to the controller **132***a*, **132***b*, **132***c*, **132***d*.

In still other embodiments, a copy of the entire database **150** may be maintained at each controller **132***a*, **132***b*, **132***c*, **132***d*. A synchronization system that locks some or all of each copy of the database may be used to ensure that all copies of the database **150** are maintained in synchronization. In such embodiments, each controller **132***a*, **132***b*, **132***c*, **132***d* may use only its local copy of the entire database **150**.

In various embodiments, a combination of these techniques may be used to maintain database **150**.

As illustrated in FIG. **5**, the backplane **104** includes a plurality of static point-to-point backplane connections **152** that couple output switch terminals on one card **106** to input switch terminals on another card **106**. For example, backplane connections may couple output switch terminal **130***a***27** on line card **116***a* to input switch terminal **128***c***35** on fabric card **140***c*. Various embodiments may include as many or as few backplane connections between output switch terminals to input switch terminals.

In any particular embodiment, the sizes of the various crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* and the number of backplane connections can be selected to provide a desired level of functionality in the router. For example, in a router designed for a specific purpose in which only a limited number of couplings between input ports **118** and output ports **120** may be required may have a correspondingly limited number of backplane connections **152**. Fabric cards are typically useful to increase the flexibility with which a particular input port can be coupled to a particular output port. In some embodiments, all cards **106** may be line cards with no fabric cards.

By selectively configuring one or more crosspoint switches **124***a*, **124***b*, **124***c*, **124***d*, a particular input port **118** on one line card **116** may be coupled to a particular output port **120** on the same or another line card.

Reference is made to FIG. **4**. For example, if:

- input port **118***a***4** is fixedly coupled to input switch terminal **128***a***4**;
- switch crosspoint switch **124***a* couples input switch terminal **128***a***4** to output switch terminal **130***a***7**; and
- output switch terminal **130***a***7** is fixedly coupled to output port **120***a***7**, then an input data signal received input port **118***a***4** on line card **116***a* will be coupled to output port **120***a***7**.

Reference is made to FIG. **6**. If:

- input port **118***a***2** is fixedly coupled to switch input terminal **128***a***2**;
- switch input terminal **128***a***2** is coupled to output switch terminal **130***a***27** in crosspoint switch **124***a*;
- output switch terminal **130***a***27** is coupled to input switch terminal **128***c***35** in fabric card **140***c* through backplane connection **152***a*;
- switch input terminal **128***c***35** is coupled to output switch terminal **130***c***14** in crosspoint switch **124***c*;
- output switch terminal **130***c***14** is coupled to input switch terminal **128***d***12** through the backplane connection **152***b*;
- switch input terminal **128***d***12** is coupled to output switch terminal **130***d***5** in crosspoint switch **124***d*; and
- output switch terminal **130***d***5** is fixedly coupled to output port **120***d***5**, then an input data signal received at input port **118***a***2** on line card **116***a* will be coupled to output port **120***d***5** on line card **116***d*.

In router **100**, each controller **132***a*, **132***b*, **132***c*, **132***d* is coupled to each crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* in the router and may instruct any crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* to couple specific input switch terminals and output switch terminals within the crosspoint switch **124***a*, **124***b*, **124***c*, **124***d*. Through one or more steps through crosspoint switches and through backplane connection **152**, an input signal received at an input port **118** may be coupled to an output port **120** on the same or a different line card. In some embodiments, the crosspoint switches and the number of pairs of output switch terminals and input switch terminals coupled by backplane connections **152** may be sufficient to allow any input port **118** to be coupled to any output port **120**, possibly through a variety of different routes.

A particular controller **132***a*, **132***d* in a line card **116***a*, **116***d* may be configured to ensure that a data signal or data stream received at the line card is routed through to an appropriate destination for the data stream. For example, when a data stream is initially received at an input port **118**, the controller examines the packets in the data stream, which will identify a destination for the data stream. The controller then determines which output port **120** in the router (which may be on the same line card as the controller or on another line card) is coupled to the destination. The controller then determines a path through the router and configures one or more crosspoint switches to provide the path between the input port **118** and the output port **120**. The controller will typically select a route based on router configuration data that is previously recorded in the controller. The router configuration data includes information about the availability of backplane connections between different cards and may include additional information about the router structure or configuration. The controller will also typically consider the contents of the database **150**. Typically a controller will not change the configuration of an input switch terminal or an output switch terminal that is already in use (i.e. coupled to a corresponding switch terminal). In some embodiments, a priority level for some or all of the couplings between different pairs of input switch terminal and output switch terminal may be maintained in database **150**. A controller may determine a priority level for a data stream that the controller is routing through router **100**. If an input switch or an output switch terminal is in use, but the stored priority level for the stream being routed through the switch is lower than the priority of the stream that the controller is attempting to route, then the control may change the configuration of the switch to use it for the higher priority data stream. In some cases, the router may have multiple paths through which a data stream can be routed from a particular input port **118** to a particular output port **120** and it may be possible to provide a needed routing for

a high priority data stream without disrupting a lower priority data stream. Each controller may be configured to identify multiple routings to reduce disruption to existing routes set up within the router.

In some conditions, a controller may not be able to determine a route by which a data stream can be delivered to a particular output port **120**. In such conditions, the controller **132** may send a routing request to a supervisor **138** through the controller communication network **136**. A supervisor will typically be an external device that can monitor and control the configuration of crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* in the router **100** and possibly in other routers. In some embodiments, a supervisor **138** may be built into a router. In some embodiments, duplicate or multiple supervisors may be provided to provide redundancy or improved responsiveness when a request is sent to a supervisor or a group of supervisors.

Each time a controller **132***a*, **132***b*, **132***c*, **132***d* changes the configuration of a switch **124***a*, **124***b*, **124***c*, **124***d*, the changes are recorded in the database **150**.

A supervisor may receive various types of requests. For example, a controller may ask a supervisor to provide a route from a particular input port to a particular output port. A controller may ask a supervisor to examine a packet to determine the output port to which the packet (and the corresponding data stream) should be coupled, and possibly also to provide a routing between the input port on which the data stream is received and the output port.

In some embodiments, a supervisor may directly change the configuration of crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* and update database **150** and advise the requesting controller that the request has been satisfied and optionally provide details of configuration changes made in the router. In other embodiments, a supervisor may provide a response to a controller making a request and the controller may then implement the details of the response.

In some embodiments, each controller **132***a*, **132***b*, **132***c*, **132***d* may record some or all of the routes that are used by the controller, including some or all of the requests provided by a supervisor. The controller **132***a*, **132***b*, **132***c*, **132***d* may subsequently refer to the recorded requests to select routes for data streams between input ports **118** and output ports **120** based on the previously recorded routes. In some embodiments, the controller may track performance information such as the frequency with which transmission failures occur in particular routes and may select more reliable routes. Over time, the recorded route may become a library allowing a controller **132***a*, **132***b*, **132***c*, **132***d* to resolve an increasing number of routing requirement without sending a request to a supervisor. In addition, some or all of the controllers may be configured to find routes without reference to previously recorded route or making a request to a supervisor.

In this manner, the controllers **132***a*, **132***b*, **132***c*, **132***d* in each card **106** are able to provide routes for data streams through the router **100**. Some of the routes may traverse only the line card on which a data stream is received while other routes may traverse various line cards, fabric cards and backplane connections. In doing so, the controller can reduce the number of requests transmitted to the supervisor, increasing the rate at which data streams can be coupled through a router, particularly when a router receives, routes and transmits a large number of data streams.

Router **100** has been described as a video router. A video router will typically receive audio/video data streams (which may be referred to as transport streams). In some embodiments, the data streams may also include non-video streams or may not include any video streams at all.

Reference is next made to FIG. **7**, which illustrates a control hierarchy between the card controllers, supervisors, other routers and other devices in a system. In some situations, a plurality of routers **700** may be coupled to provide a data transmission system. For example, routers **700** may be installed in a video processing facility such as a television studio or broadcast facility. Some or all of the routers may receive and transmit a plurality of input and output data streams. In some facilities, hundreds, thousands or even millions of data streams may be received and transmitted. The group of routers will typically be interconnected with a variety of other equipment including signal processors, analytic devices and other devices that generate or require data streams that are switched through one or more routers.

Each router **700** is coupled to a supervisor system **754**, which may include a plurality of supervisors **738**, such as a first supervisor sub-system **738***a*, a second supervisor sub-system **738***b*, a third supervisor sub-system **738***c*, a fourth supervisor sub-system **738***d*, a fifth supervisor sub-system **738***e*, a sixth supervisor sub-system **738***f*, a seventh supervisor sub-system **738***g* and an eighth supervisor sub-system **738***h*. The supervisor system **754** forms a hierarchy in conjunction with the controllers **732** in each router. As described above, a card controller **732** on a card **706**, including a line card **716** and a fabric card **740**, in a router **700** may control the configuration of the corresponding switch **724** on its card **706** and may also be authorized to control the configuration of switches **724** on other cards within the same router. The controller may send requests to a corresponding supervisor sub-system **738***a* when the controller is unable to determine a route for a data stream, for example, when the controller is unable to allocate switches or connections to set up a required route, or when a route may require coordination between routers or under other conditions, which may include instructions from a supervisor to always make a request to the supervisor when certain types of data streams are received or after a particular time or other conditions. In some cases, two or more supervisors may be assigned to each router and may act as primary and backup routers, may operate in parallel, or may operate in a distributed manner to manage the flow and latency or requests made to the supervisory system **754**.

The supervisory system **754** may itself be coupled to other devices in a facility via a controller communication network **736** to receive and provide control and status information about the routers **700**. Such control and status information may be used to control the routing of data streams within and between routers **700**. For example, the other devices in the facility may identify high priority data streams that are to be switched through one or more routers **700** to reach a particular destination. Supervisor system **754** may instruct one or more of the routers to configure an appropriate route between a port on which a high priority data stream is to be received and its destination. In such a situation, a supervisor sub-system **738** may instruct the routers to configure a route directly, without previously having received a request from a controller **732**.

FIG. **7** illustrates a control hierarchy in which the supervisory system **754** communicates with other devices, which may be at the same or a different facility as the supervisory system. Supervisors **738** in the supervisor system **754** control the routing of data streams within and between the routers and between the routers and other devices. Controllers **732** in the routers can control routes directly within the router and may request control instructions from supervisors

to generate requests. The supervisory system **754** and the controllers **732** are part of a control layer **756** that provides routes for data streams.

FIG. **7** also illustrates a data layer **758** in which the data streams are transmitted. The data layer **758** includes input ports **718**, switches **724**, backplane connections **752** and output ports **720**. The control layer **756** configures the data layer so that data streams are able to traverse the data layer between input ports and output ports.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the scope of the invention, which is limited only by the appended claims.

The invention claimed is:

1. A video router comprising:

a backplane including a plurality of static point-to-point backplane connections;

a plurality of line cards, each line card including:

a plurality of input ports and output ports, each input port and output port is coupled to a respective external signal through the backplane;

a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals such that a first plurality of input and output switch terminals are coupled to the respective plurality of input and output ports and a second plurality of input and output switch terminals are coupled to the plurality of backplane connections;

a line card controller coupled to the line card cross-point switch to selectively couple some of the input switch terminals to the output switch terminals;

one or more fabric cards, each fabric card including:

a fabric card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals coupled to the plurality of backplane connections;

a fabric card controller coupled to the fabric card cross-point switch to selectively couple some of the input switch terminals to the output switch terminals; and

a controller communication network coupled to each of the line cards and fabric cards to control the operation of the fabric card controllers and the line card controllers.

2. The video router of claim **1**, further comprising a switch configuration database coupled to the controller communication network and configured to store coupling of at least one of the input switch terminals of the line card cross-point switch and the fabric card cross-point switch to a corresponding output switch terminal.

3. The video router of claim **2**, wherein the switch configuration database is stored in the fabric card controllers and the line card controllers.

4. The video router of claim **2**, wherein the switch configuration database is stored in an external data storage device and is accessible by the fabric card controllers and the line card controllers.

5. The video router of claim **2**, wherein one or more controllers configure one or more corresponding cross-point switches to route a data stream from an input port to an output port, wherein each of the one or more controllers being a fabric card controller or a line card controller, and the corresponding cross-point switches being a corresponding fabric card cross-point switch or a line card cross-point switch.

6. The video router of claim **5**, wherein when the data stream is received at the input port, a controller corresponding to at least one of the one or more line cards and the one or more fabric cards processes the data stream and generates an output destination identifying the output port for the data stream.

7. The video router of claim **5**, wherein if the data stream is designated a priority stream, at least one of the one or more controllers reconfigures the corresponding cross-point switch to route the priority stream.

8. The video router of claim **1**, wherein the backplane comprises a plurality of backplane connectors for receiving the one or more line cards and the one or more fabric cards.

9. The video router of claim **8**, wherein each backplane connector comprises a plurality of backplane contacts, wherein each of the one or more line cards and the one or more fabric cards comprises a plurality of card pins, and wherein the plurality of backplane contacts and the plurality of card pins provide an electrical connection when coupled.

10. The video router of claim **8**, wherein the backplane further comprises a plurality of point-to-point traces, each point-to-point trace being configured to couple an output switch terminal of a first card to an input switch terminal of a second card, each of the first card and the second card being a line card or a fabric card.

11. The video router of claim **10**, wherein the plurality of point-to-point traces are static traces.

12. The video router of claim **5**, wherein the data stream comprises data selected from the group consisting of audio data, video data, and a combination of audio and video data.

13. A method of routing video signals from a plurality of input ports to a plurality of output ports using at least one video router of claim **1**, the method comprising:

receiving a data stream at an input port of a first card, the first card being a line card;

processing the data stream by a first controller of the first card to generate an output destination identifying an output port, the first controller being a line card controller; and

configuring one or more cross-point switches by corresponding controllers to route the data stream between the input port and the output destination, and wherein at least one of the one or more cross-point switches correspond to a first cross-point switch of the first card.

14. The method of claim **13**, further comprising:

configuring at least one additional cross-point switch other than the first cross-point switch, wherein the one additional cross-point switch is included in a second card coupled to the first card, the second card being a fabric card or a line card.

15. The method of claim **13**, further comprising:

transmitting a routing request to an external device;

receiving router configuration signal from the external device, the router configuration signal comprising instructions to selectively couple input switch terminals to output switch terminals of the one or more cross-point switches; and

configuring the one or more cross-point switches based on the router configuration signal.

16. The method of claim **13**, wherein each controller corresponding to the one or more cross-point switches comprises a configuration database configured to store coupling instructions between input switch terminals to the output switch terminal of the corresponding cross-point switch.

17. The method of claim **13**, wherein each controller corresponding to the one or more cross-point switches is

coupled to a configuration database using a controller communication network, wherein the configuration database is configured to store coupling instructions between input switch terminals to the output switch terminal of the corresponding cross-point switch.

18. The method of claim **13**, wherein if the data stream is designated a priority stream, the method comprises:

determining if reconfiguration of the one or more cross-point switches is required to route the priority stream before a second data stream, the second data stream having a lower priority than the priority stream; and

if reconfiguration is required, reconfiguring the one or more cross-point switches.

19. The method of claim **13**, further comprising:

monitoring one or more performance characteristics of the at least one video router to generate monitored performance data; and

reconfiguring the one or more cross-point switches based on the monitored performance data.

20. The method of claim **19**, wherein at least one performance characteristic comprises frequency of transmission failure of the data stream between an input port and an output port.

* * * * *

EXHIBIT F

US009942139B2

(12) **United States Patent**
Patel

(10) **Patent No.: US 9,942,139 B2**
(45) **Date of Patent: *Apr. 10, 2018**

(54) **VIDEO ROUTER**

(71) Applicant: **Evertz Microsystems Ltd.**, Burlington (CA)

(72) Inventor: **Rakesh Patel**, Mississauga (CA)

(73) Assignee: **Evertz Microsystems Ltd.**, Burlington (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **15/484,852**

(22) Filed: **Apr. 11, 2017**

(65) **Prior Publication Data**

US 2017/0279711 A1 Sep. 28, 2017

Related U.S. Application Data

(63) Continuation of application No. 14/505,124, filed on Oct. 2, 2014, now Pat. No. 9,654,391.
(Continued)

(51) **Int. Cl.**
H04L 12/713 (2013.01)
H04L 12/931 (2013.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04L 45/60*** (2013.01); ***H04L 49/101*** (2013.01); ***H04L 49/206*** (2013.01)

(58) **Field of Classification Search**
CPC H04N 19/40; H04N 21/2343; H04N 21/234309; H04L 45/60; H04L 49/206; H04L 49/101
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,976,134 B1 12/2005 Lolayekar et al.
7,194,550 B1 * 3/2007 Chamdani H04L 67/1097 709/227

(Continued)

FOREIGN PATENT DOCUMENTS

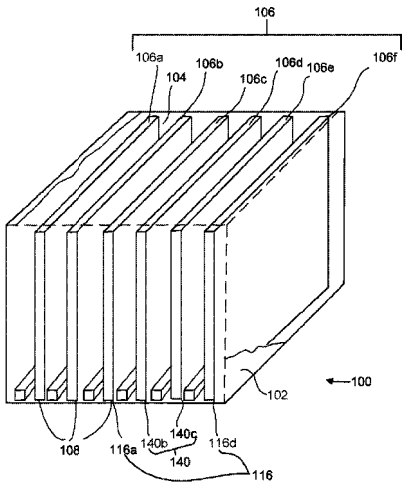
CA 2866073 A1 4/2015

Primary Examiner — Ayaz Sheikh
Assistant Examiner — Debebe Asefa
(74) *Attorney, Agent, or Firm* — Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.

(57) **ABSTRACT**

The embodiments described herein provide a data transmission system comprising a plurality of video routers, a supervisory system for transmitting one or more router configuration signals to one or more video routers, and a control communication network for coupling the plurality of video routers and the supervisory system. Each router in the system comprises a backplane including a plurality of backplane connections, at least one line card and at least one fabric card. Each line card comprises a plurality of input ports and output ports where each input and output port is coupled to a respective external signal through the backplane. Each line card further comprises a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals. Each fabric card comprises a fabric card cross-point switch having a plurality of input switch terminal and a plurality of output switch terminals. Furthermore, each line card and each fabric card comprises a card controller where the card controller selectively couples one or more input switch terminals of a cross-point switch to the output switch terminals of that cross-point switch. The cross-point switches being manipulated by the card controller may belong to one or more different cards within the same video router.

20 Claims, 7 Drawing Sheets



Related U.S. Application Data

(60) Provisional application No. 61/885,588, filed on Oct. 2, 2013.

(51) **Int. Cl.**
H04L 12/933 (2013.01)
H04L 12/773 (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,776,161	B2 *	7/2014	Gazier	H04N 21/222 375/240.07
8,891,406	B1	11/2014	Shekhar	

* cited by examiner

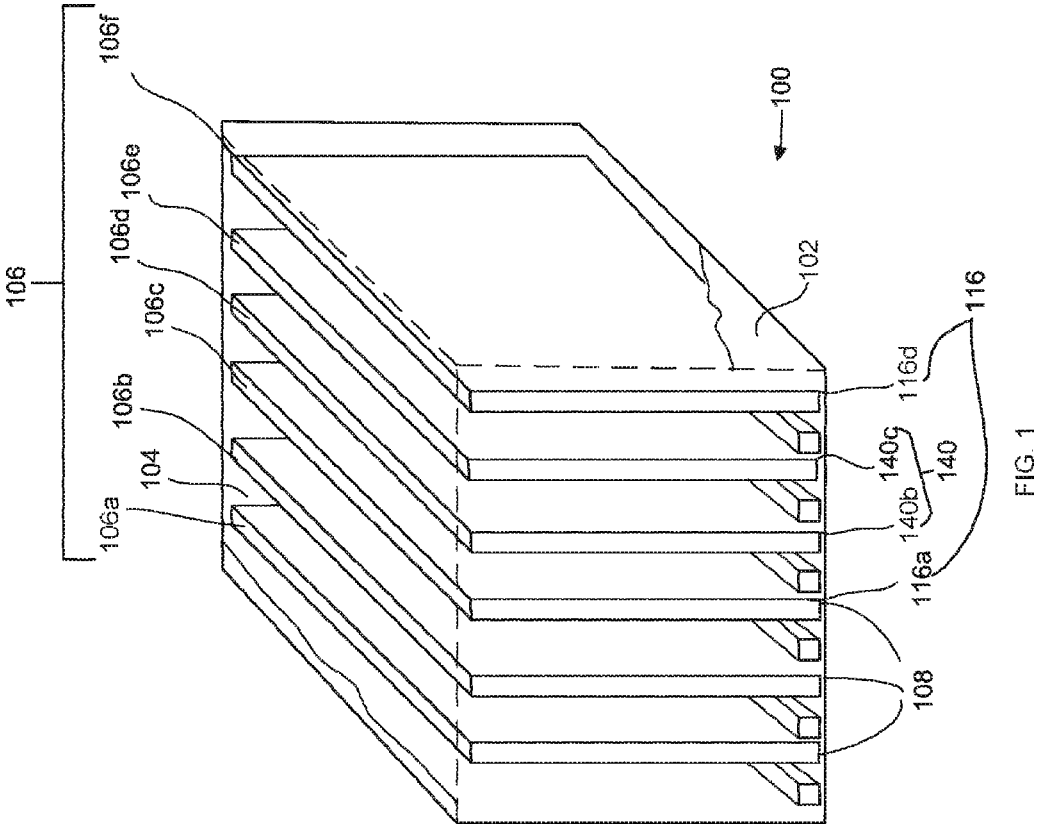

106f
106e
106d
106c
106b
104
106a
106
100
102
116
116d
140c
140
140b
116a
108

FIG 1

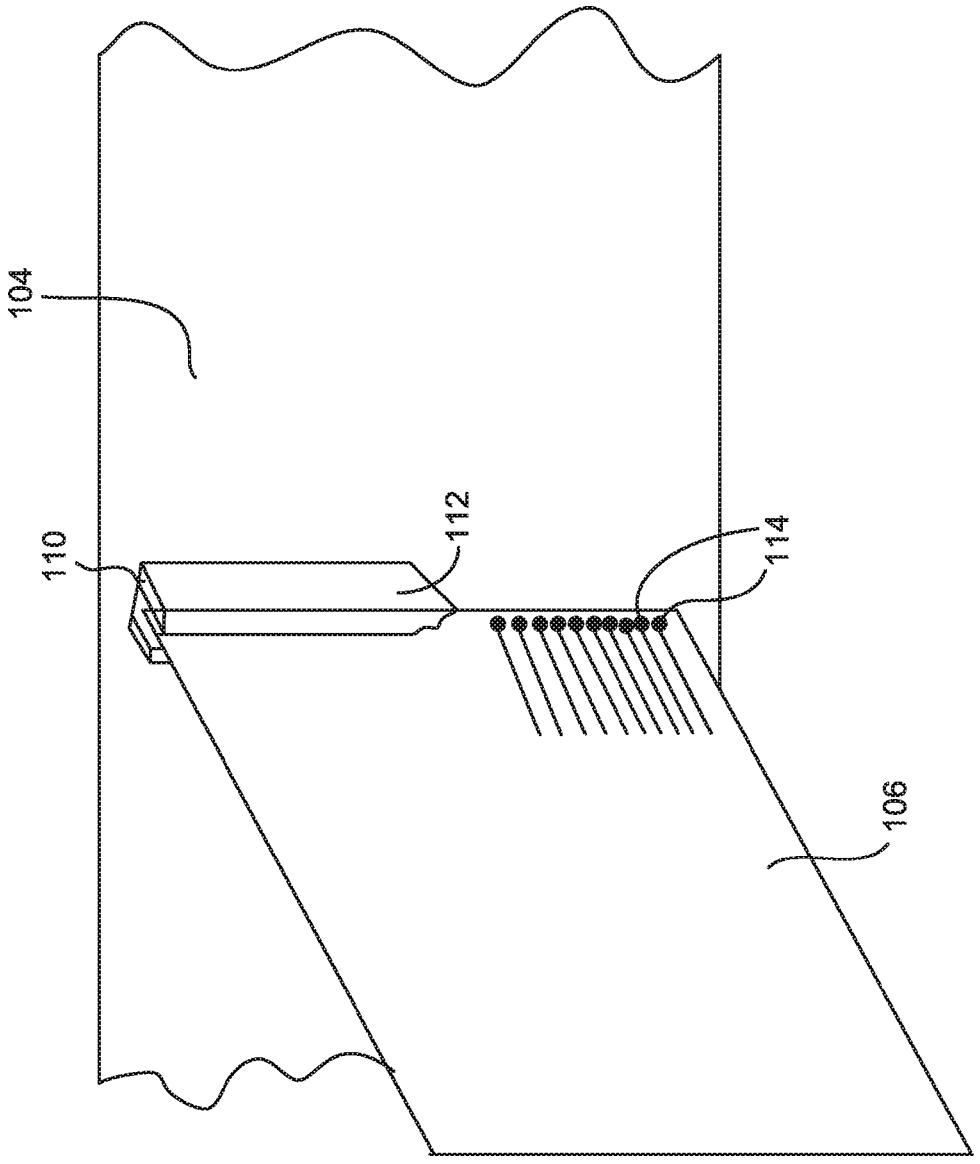


FIG. 2

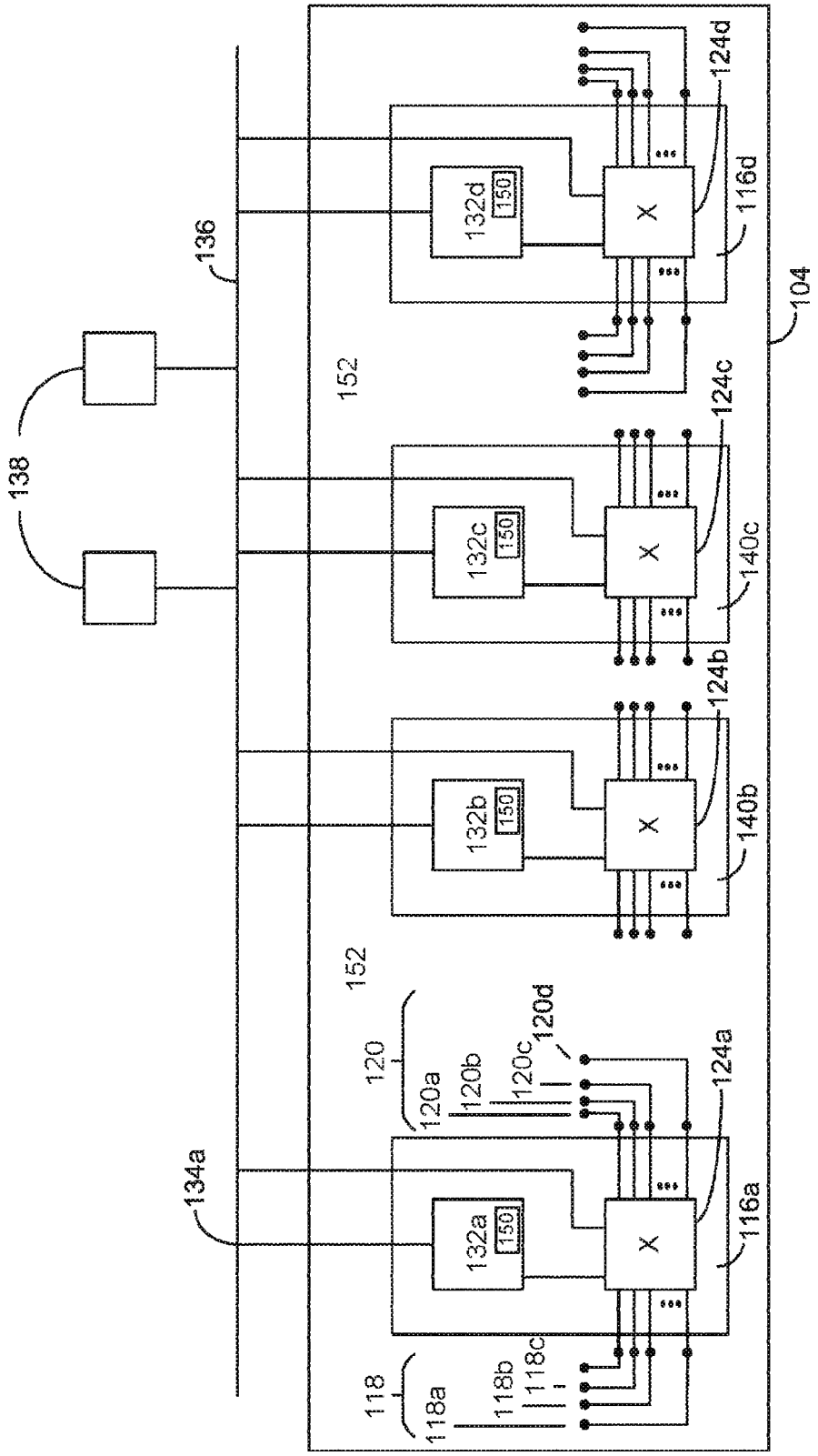

104
136
138
152
152
134a
132a
132b
132c
132d
150
X
116a
116d
118
118a
118b
118c
120
120a
120b
120c
120d
124a
124b
124c
124d
140b
140c

FIG. 3

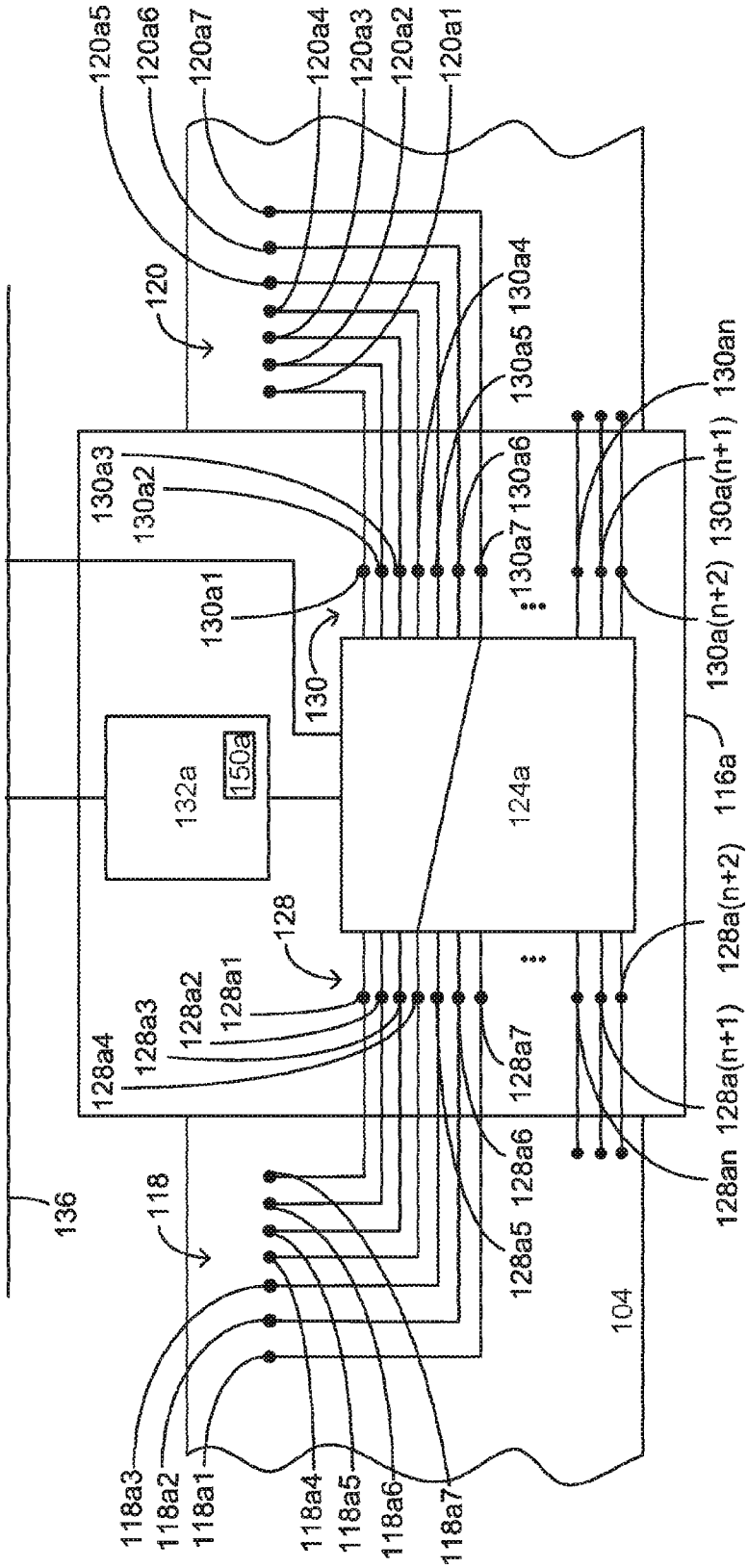

120a5
120a6
120a7
120a4
120a3
120a2
120a1
120
130a4
130a5
130a6
130a7
130an
130a3
130a2
130a1
130
130a(n+1)
130a(n+2)
132a
150a
124a
116a
128
128a4
128a3
128a2
128a1
128a7
128a(n+2)
128a(n+1)
128an
128a6
128a5
136
118
104
118a3
118a2
118a1
118a4
118a5
118a6
118a7

FIG. 4

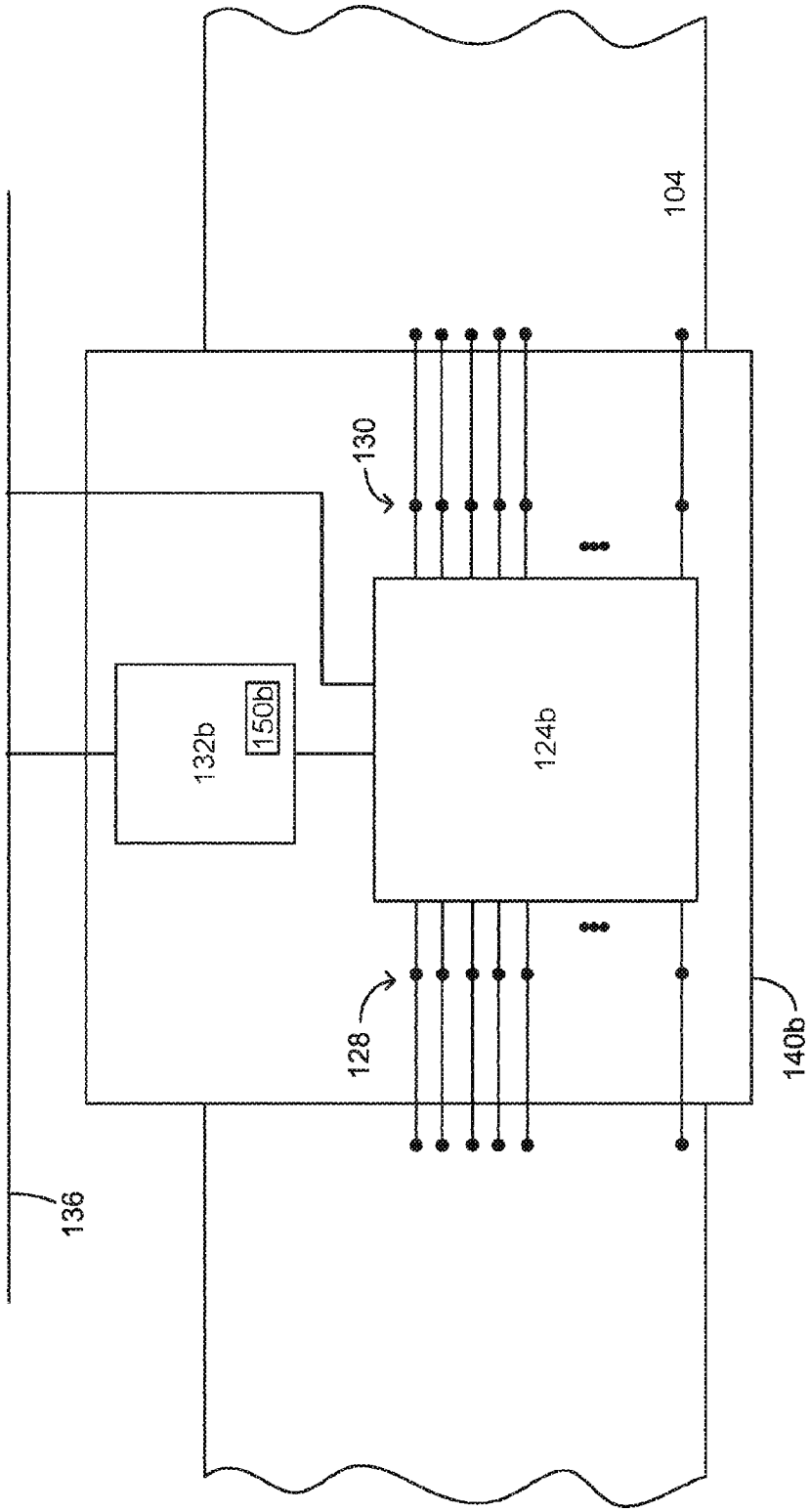


FIG. 5

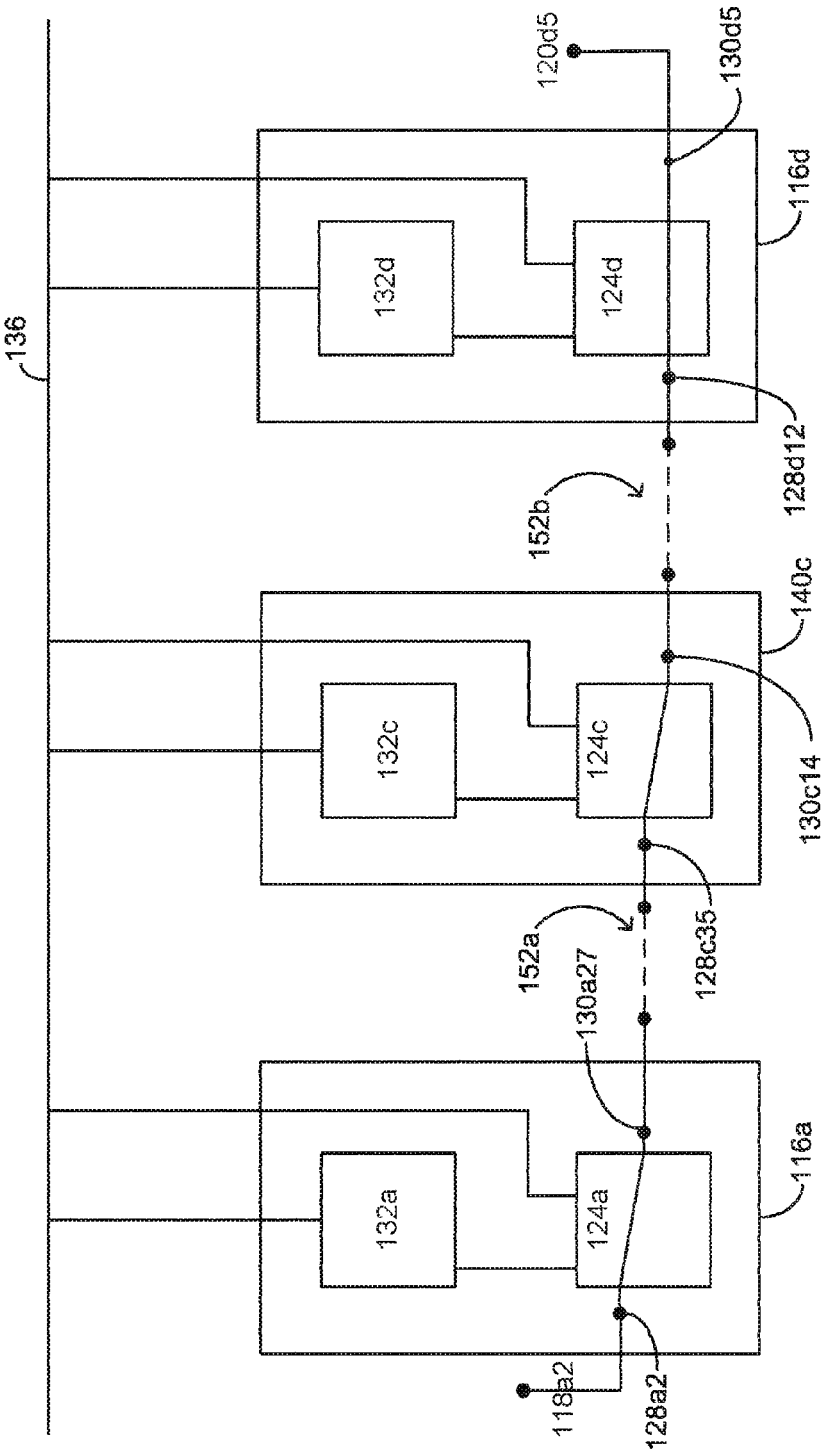

136
120d5
130d5
116d
132d
124d
128d12
152b
140c
132c
124c
130c14
128c35
152a
130a27
116a
132a
124a
118a2
128a2

FIG. 6

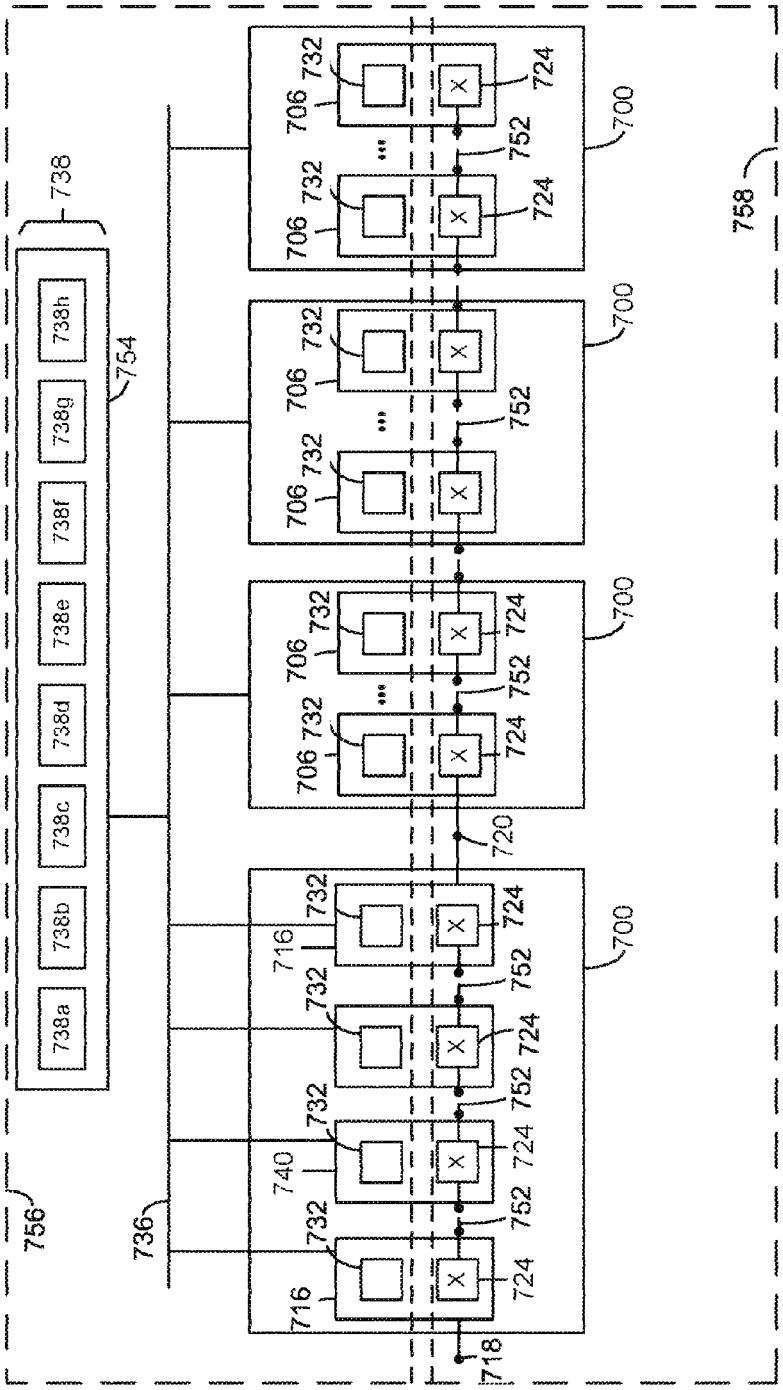

738
738a
738b
738c
738d
738e
738f
738g
738h
754
756
758
736
700
706
732
724
752
716
740
720
718

FIG. 7

VIDEO ROUTER

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 14/505,124, filed on Oct. 2, 2014, which claims the benefit of U.S. Provisional Patent Application No. 61/885,588, filed Oct. 2, 2013. The entire contents of the applications are hereby incorporated by reference.

FIELD

The described embodiments relate to routers for video signals and other data streams.

BACKGROUND

The number of devices coupled to data communications networks is increasing rapidly. The routing of data streams from and to such devices is increasingly more complex and difficulty in allocating efficient routes, or even any route at all, in various components in a communication network can affect the quality of service delivery to a user of a device. For example, communication networks typically contain routers that couple an input data stream received at an input port to an output port at which the data stream is available to a downstream device. As the size of routers increases (i.e. as the number of input and output ports on a router increases, then complexity of creating efficient routing within the router and between network devices increase non-linearly.

It is desirable to provide an efficient system and methods that allows a network device to efficiently configure routes for data streams.

SUMMARY

In one aspect, in at least one embodiment described herein, there is provided a data transmission system comprising a plurality of video routers, a supervisory system for transmitting one or more router configuration signals to one or more video routers, and a control communication network for coupling the plurality of video routers and the supervisory system. Each router in the system comprises a backplane including a plurality of backplane connections, at least one line card and at least one fabric card. Each line card comprises a plurality of input ports and output ports where each input and output port is coupled to a respective external signal through the backplane. Each line card further comprises a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals. Each fabric card comprises a fabric card cross-point switch having a plurality of input switch terminal and a plurality of output switch terminals. Furthermore, each line card and each fabric card comprises a card controller where the card controller selectively couples one or more input switch terminals of a cross-point switch to the output switch terminals of that cross-point switch. The cross-point switches being manipulated by the card controller may belong to one or more different cards within the same video router.

In some cases, the card controller of a first card in a first video router configures a corresponding cross-point switch of the first card to route a data stream from an input port to an output port, where the first card and the second card are a line card or a fabric card.

In some other cases, the card controller of a first card in a first video router configures a cross-point switch of a second card in the first video router to route a data stream from an input port to an output port, where the first card and the second card are a line card or a fabric card.

In various cases, where when a data stream is received at an input port of a first card, a first card controller corresponding to the first card is configured to transmit a data request to the supervisory system, where the supervisory system is configured to: determine an output destination identifying an output port, and generate one or more router configuration signals for one or more card controllers based on the output destination, wherein the one or more card controllers configure one or more cross-point switches to route the data stream to the output port.

In various cases, where when a data stream is received at an input port of a first card, a first card controller corresponding to the first card is configured to: determine an output destination identifying an output port, and transmit a data request to the supervisory system, where the supervisory system is configured to: generate one or more router configuration signals for one or more card controllers based on the output destination, wherein the one or more card controllers configure one or more cross-point switches to route the data stream to the output port.

In various cases, where if the data stream is designated a priority stream, at least one of the one or more card controllers reconfigures the corresponding cross-point switch to route the priority stream.

In various cases, where the backplane comprises a plurality of backplane connectors for receiving the at least one line card and the at least one fabric card.

In various cases, where each backplane connector comprises a plurality of backplane contacts, wherein each line card and each fabric card comprises a plurality of card pins, and wherein the plurality of backplane contacts and the plurality of card pins provide an electrical connection when coupled.

In various cases, the system further comprises a switch configuration database coupled to the controller communication network and configured to store coupling of the input switch terminals of at least one line card cross-point switch and the fabric card cross-point switch to corresponding output switch terminals.

In various cases, the switch configuration database is provided within the card controllers.

In another aspect, in at least one embodiment described herein, there is provided a method of routing video signals from a plurality of input ports to a plurality of output ports using at least one video router of a data transmission system disclosed herein. The method comprises receiving a data stream at an input port of a card, the card being a line card, receiving one or more router configuration signals by one or more card controllers, at least one card controller being a line card controller of the line card, and configuring one or more cross-point switches by card controllers based on the one or more router configuration signals to route the data stream between the input port and an output destination, wherein at least one of the one or more cross-point switches correspond to a cross-point switch of the line card.

In various embodiments, the method of routing video signals is configured to operate in accordance with the devices defined above or in accordance with the teachings herein.

In another aspect, in at least one embodiment described herein, there is provided a data transmission system comprising a control layer, a data layer and a controller com-

munication network for coupling the control layer and the data layer. The control layer comprises a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising instructions to selectively configure the one or more routers, and one or more card controllers provided in the one or more video routers, each card controller configured to selectively couple input switch terminals of one or more cross-point switches to output switch terminals of the corresponding one or more cross-point switches. The data layer comprises one or more cross-point switches, the one or more cross-point switches provided in the one or more video routers, each cross-point switch comprising a plurality of input switch terminals and a plurality of output switch terminals, a backplane including a plurality of backplane connections, wherein a subset of the plurality of input switch terminals and the output switch terminals are coupled to a respective plurality of backplane connections, a plurality of input ports and a plurality of output ports corresponding to each video router, where the supervisory system is configured to: receive a request signal from a card controller, and transmit a router configuration signal to one or more card controllers, the router configuration signal comprising instructions to selectively couple input switch terminals to output switch terminals of the one or more cross-point switches coupled to the one or more card controllers.

In various embodiments, the data transmission system is configured to operate in accordance with the devices and methods defined above or in accordance with the teachings herein.

Other features and advantages of the present application will become apparent from the following detailed description taken together with the accompanying drawings. It should be understood, however, that the detailed description and the specific examples, while indicating preferred embodiments of the application, are given by way of illustration only, since various changes and modifications within the spirit and scope of the application will become apparent to those skilled in the art from this detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the applicant's teachings described herein, reference will now be made, by way of example only, to the accompanying drawings which show at least one exemplary embodiment, and in which:

FIG. **1** is a cross-section of a video router according to an example embodiment;

FIG. **2** is a cross-section of a video router according to another example embodiment;

FIG. **3** is a block diagram of a video router according to an example embodiment;

FIG. **4** is a block diagram of a video router according to another example embodiment;

FIG. **5** is a block diagram of a video router according to another example embodiment;

FIG. **6** is a block diagram of a video router according to another example embodiment;

FIG. **7** is a block diagram of a control hierarchy of a video router according to an example embodiment.

For simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements.

DESCRIPTION OF EXEMPLARY EMBODIMENTS

Reference is first made to FIGS. **1** and **2**, which illustrates a first video router **100** with an integrated control layer. Router **100** includes a frame or housing **102**, a backplane **104** and a plurality of cards **106**, such as a first card **106***a*, a second card **106***b*, a third card **106***c*, a fourth card **106***d*, a fifth card **106***e* and a sixth card **106***f*. The frame **102** includes a plurality of frame slots **108** in which cards may be received and held in place. The backplane **104** includes backplane connector **110** corresponding to each slot **108** and each card **106**. Each backplane connector includes a plurality of backplane pins or contacts **112**. Each card **106** includes a plurality of card pins or contacts **114**, each of which corresponds to a backplane pin **112** of the corresponding backplane connector. When a card **106** is installed in frame **102**, the card pins **114** couple with corresponding backplane pins **112** making an electrical connection through which a data signal may be transmitted.

Cards **106** may include various types of cards. For example, some of the cards may be line cards **116**, such as a first line card **116***a* and a second line card **116***d*, which include input ports or output ports for respectively receiving and transmitting data signals, or both input and output ports. Other cards **106** may be fabric cards **140**, such as a first fabric card **140***b* and a second fabric card **140***c*, which facilitate switching of signals between various input and port ports.

Reference is made to FIGS. **3**, **4** and **5**, which schematically illustrate components of router **100**. In the present example embodiment, each input port **118** or output port **120** on a line card **116***a* is coupled to an external signal through the backplane **104**. In the illustrated embodiment of FIG. **3**, input port **118** comprises a first input port **118***a*, a second input port **118***b* and a third input port **118***c*, and output port **120** comprises a first output port **120***a*, a second output port **120***b*, a third output port **120***c* and a fourth output port **120***d*. The backplane may, for example, include a pass-through connector to which a line card port **118**, **120** may be coupled within frame **102** and to which a cable (not shown) may be coupled on the rear of the backplane. The line card port **118**, **120** is electrically coupled to the cable (not shown), allowing the line card to receive or transmit a data signal on the cable. In other embodiments, line card ports may be directly coupled to a cable or may be coupled to a cable through the backplane using a coupling other than a pass-through connector.

Line card **116***a* includes a line card crosspoint switch **124***a* with a plurality of switch terminals. In this example, crosspoint switch **124***a* has a plurality of input switch terminals **128** and a plurality of output switch terminals **130**. Each input port **118** is coupled to at least one input switch terminal **128** and each output port **120** is coupled to at least one output switch terminal **130**. In addition, a plurality of input switch terminals **128** are coupled to the backplane **104** through the corresponding backplane connector **110**. A plurality of output switch terminals **130** are coupled to the backplane **104** through the corresponding backplane connector **110**.

Line card **116***a* also includes a line card controller **132***a* that is coupled to crosspoint switch **124***a* and which provides control signals to couple or decouple particular input switch

terminals **128** to particular output switch terminals **130**. Card controller **132***a* is coupled to a controller communication network **136** at a control system terminal **134***a* through which the card controller **132***a* may communicate with other cards **106** and with external control devices such as an external supervisor **138**. In some embodiments, a line card controller **132***a* may be coupled to controller communication network **136** through the backplane or through another communication bus in frame **102** to which the line card is couple when installed in the frame.

Each fabric card, such as fabric card **140***b* includes a card controller **132***b* and a crosspoint switch **124***b*, which are coupled together and operate in a manner similar to the card controller **132***a* and crosspoint switch **124***a* of line card **116***a*. Crosspoint switch **124***b* includes a plurality of input switch terminals **128** and output switch terminals **130** that are coupled to the backplane **104**. The crosspoint switch **124***b* may be configured to couple any of the input switch terminals **128** to any of the output switch terminals **130** under the control of card controller **132***b*. As illustrated in FIG. **3**, router **100** further includes a fabric card **140***c*, which includes a card controller **132***c* and a crosspoint switch **124***c*, and a line card **116***d*, which includes a card controller **132***d* and a crosspoint switch **124***d*.

Each crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* in router **100** is coupled to the controller communication network **136** through which the configuration of the crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* may be changed by card controller **132***a*, **132***b*, **132***c*, **132***d* on other cards **106**.

Router **100** also includes a switch configuration table or database **150**. Database **150** records the current setting for every input switch terminal and output switch terminal in all cross-point switches **124***a*, **124***b*, **124***c*, **124***d* in the router **100**. For example, part of the contents of database **150** may be:

Switch	Terminal	Setting
124a	128a1	Coupled to 130c3
124a	128a2	Coupled to 130a27
124a	128a3	Open
124a	128a4	Coupled to 130a8
. . .	. . .	. . .
124a	130a3	Coupled to 128a1
124a	130a4	Open
124a	130a5	Coupled to 128a2
124a	130a6	Open
124a	130a7	Coupled to 128a2
124b	130a8	Coupled to 128a4
. . .	. . .	. . .
124b	128b1	Open
. . .	. . .	. . .
124c	128c35	Coupled to 130c14
. . .	. . .	. . .
124c	130c14	Coupled to 128c35
. . .	. . .	. . .
124d	128d12	Coupled to 130d5
124d	130d5	Coupled to 128d12
. . .	. . .	. . .

where router **100** of FIG. **4** comprises a first input port **118***a***1**, a second input port **118***a***2**, a third input port **118***a***3**, a fourth input port **118***a***4**, a fifth input port **118***a***5**, a sixth input port **118***a***6**, a seventh input port **118***a***7**, a first output port **120***a***1**, a second output port **120***a***2**, a third output port **120***a***3**, a fourth output port **120***a***4**, a fifth output port **120***a***5**, a sixth output port **120***a***6**, a seventh output port **120***a***7**, a first input switch terminal **128***a***1**, a second input switch terminal **128***a***2**, a third input switch terminal **128***a***3**, a fourth input switch terminal **128***a***4**, a fifth input switch terminal **128***a***5**, a sixth input switch terminal **128***a***6**, a seventh input switch terminal **128***a***7**, an nth input switch terminal **128***an*, a (n+1)th input switch terminal **128***a*(n+1), a (n+2)th input switch terminal **128***a*(n+2), a first output switch terminal **130***a***1**, a second output switch terminal **130***a***2**, a third output switch terminal **130***a***3**, a fourth output switch terminal **130***a***4**, a fifth output switch terminal **130***a***5**, a sixth output switch terminal **130***a***6**, a seventh output switch terminal **130***a***7**, an nth output switch terminal **130***an*, a (n+1)th output switch terminal **130***a*(n+1) and a (n+2)th output switch terminal **130***a*(n+2).

Database **150** is accessible to each of the controllers **132***a*, **132***b*, **132***c*, **132***d*. In some embodiments, the database **150** may be recorded in a central location, for example, in one of the controllers **132***a*, **132***b*, **132***c*, **132***d* where the local controller **132***a*, **132***b*, **132***c*, **132***d* may access the database directly and each of the other controllers may access the database through the controller communication network **136**.

In other embodiments, the database may be a distributed database with components that are located in multiple locations within router **100**. For example, components of database **150***a*, **150***b* may be located in each of the controllers **132***a*, **132***b*, as is illustrated in FIGS. **4** and **5**. Each controller **132***a*, **132***b* may contain the status of the cross-point switch **124***a*, **124***b* in the same card **106**. Controllers **132***a*, **132***b*, **132***c*, **132***d* on other cards **106** may access the status of non-local cross-points switches through the controller communication network **136**.

In other embodiments, the database may be recorded in a data storage device or system that is external to router **100**, but which is accessible to the controller **132***a*, **132***b*, **132***c*, **132***d*.

In still other embodiments, a copy of the entire database **150** may be maintained at each controller **132***a*, **132***b*, **132***c*, **132***d*. A synchronization system that locks some or all of each copy of the database may be used to ensure that all copies of the database **150** are maintained in synchronization. In such embodiments, each controller **132***a*, **132***b*, **132***c*, **132***d* may use only its local copy of the entire database **150**.

In various embodiments, a combination of these techniques may be used to maintain database **150**.

As illustrated in FIG. **5**, the backplane **104** includes a plurality of static point-to-point backplane connections **152** that couple output switch terminals on one card **106** to input switch terminals on another card **106**. For example, backplane connections may couple output switch terminal **130***a***27** on line card **116***a* to input switch terminal **128***c***35** on fabric card **140***c*. Various embodiments may include as many or as few backplane connections between output switch terminals to input switch terminals.

In any particular embodiment, the sizes of the various crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* and the number of backplane connections can be selected to provide a desired level of functionality in the router. For example, in a router designed for a specific purpose, for example, in which only a limited number of couplings between input ports **118** and output ports **120** may be required, may have a correspondingly limited number of backplane connections **152**. Fabric cards are typically useful to increase the flexibility with which a particular input port can be coupled to a particular output port. In some embodiments, all cards **106** may be line cards with no fabric cards.

By selectively configuring one or more crosspoint switches **124***a*, **124***b*, **124***c*, **124***d*, a particular input port **118**

on one line card **116** may be coupled to a particular output port **120** on the same or another line card.

Reference is made to FIG. **4**. For example, if:

input port **118***a***4** is fixedly coupled to input switch terminal **128***a***4**;

switch crosspoint switch **124***a* couples input switch terminal **128***a***4** to output switch terminal **130***a***7**; and

output switch terminal **130***a***7** is fixedly coupled to output port **120***a***7**, then an input data signal received input port **118***a***4** on line card **116***a* will be coupled to output port **120***a***7**.

Reference is made to FIG. **6**. If:

input port **118***a***2** is fixedly coupled to switch input terminal **128***a***2**;

switch input terminal **128***a***2** is coupled to output switch terminal **130***a***27** in crosspoint switch **124***a;*

output switch terminal **130***a***27** is coupled to input switch terminal **128***c***35** in fabric card **140***c* through backplane connection **152***a;*

switch input terminal **128***c***35** is coupled to output switch terminal **130***c***14** in crosspoint switch **124***c;*

output switch terminal **130***c***14** is coupled to input switch terminal **128***d***12** through the backplane connection **152***b;*

switch input terminal **128***d***12** is coupled to output switch terminal **130***d***5** in crosspoint switch **124***d*; and

output switch terminal **130***d***5** is fixedly coupled to output port **120***d***5**, then an input data signal received at input port **118***a***2** on line card **116***a* will be coupled to output port **120***d***5** on line card **116***d*.

In router **100**, each controller **132***a*, **132***b*, **132***c*, **132***d* is coupled to each crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* in the router and may instruct any crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* to couple specific input switch terminals and output switch terminals within the crosspoint switch **124***a*, **124***b*, **124***c*, **124***d*. Through one or more steps through crosspoint switches and through backplane connection **152**, an input signal received at an input port **118** may be coupled to an output port **120** on the same or a different line card. In some embodiments, the crosspoint switches and the number of pairs of output switch terminals and input switch terminals coupled by backplane connections **152** may be sufficient to allow any input port **118** to be coupled to any output port **120**, possibly through a variety of different routes.

A particular controller **132***a*, **132***d* in a line card **116***a*, **116***d* may be configured to ensure that a data signal or data stream received at the line card is routed through to an appropriate destination for the data stream. For example, when a data stream is initially received at an input port **118**, the controller examines the packets in the data stream, which will identify a destination for the data stream. The controller then determines which output port **120** in the router (which may be on the same line card as the controller or on another line card) is coupled to the destination. The controller then determines a path through the router and configures one or more crosspoint switches to provide the path between the input port **118** and the output port **120**. The controller will typically select a route based on router configuration data that is previously recorded in the controller. The router configuration data includes information about the availability of backplane connections between different cards and may include additional information about the router structure or configuration. The controller will also typically consider the contents of the database **150**. Typically a controller will not change the configuration of an input switch terminal or an output switch terminal that is already in use (i.e. coupled to a corresponding switch terminal). In some embodiments, a priority level for some or all of the couplings between different pairs of input switch terminal and output switch terminal may be maintained in database **150**. A controller may determine a priority level for a data stream that the controller is routing through router **100**. If an input switch or an output switch terminal is in use, but the stored priority level for the stream being routed through the switch is lower than the priority of the stream that the controller is attempting to route, then the control may change the configuration of the switch to use it for the higher priority data stream. In some cases, the router may have multiple paths through which a data stream can be routed from a particular input port **118** to a particular output port **120** and it may be possible to provide a needed routing for a high priority data stream without disrupting a lower priority data stream. Each controller may be configured to identify multiple routings to reduce disruption to existing routes set up within the router.

In some conditions, a controller may not be able to determine a route by which a data stream can be delivered to a particular output port **120**. In such conditions, the controller **132** may send a routing request to a supervisor **138** through the controller communication network **136**. A supervisor will typically be an external device that can monitor and control the configuration of crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* in the router **100** and possibly in other routers. In some embodiments, a supervisor **138** may be built into a router. In some embodiments, duplicate or multiple supervisors may be provided to provide redundancy or improved responsiveness when a request is sent to a supervisor or a group of supervisors.

Each time a controller **132***a*, **132***b*, **132***c*, **132***d* changes the configuration of a switch **124***a*, **124***b*, **124***c*, **124***d*, the changes are recorded in the database **150**.

A supervisor may receive various types of requests. For example, a controller may ask a supervisor to provide a route from a particular input port to a particular output port. A controller may ask a supervisor to examine a packet to determine the output port to which the packet (and the corresponding data stream) should be coupled, and possibly also to provide a routing between the input port on which the data stream is received and the output port.

In some embodiments, a supervisor may directly change the configuration of crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* and update database **150** and advise the requesting controller that the request has been satisfied and optionally provide details of configuration changes made in the router. In other embodiments, a supervisor may provide a response to a controller making a request and the controller may then implement the details of the response.

In some embodiments, each controller **132***a*, **132***b*, **132***c*, **132***d* may record some or all of the routes that are used by the controller, including some or all of the requests provided by a supervisor. The controller **132***a*, **132***b*, **132***c*, **132***d* may subsequently refer to the recorded requests to select routes for data streams between input ports **118** and output ports **120** based on the previously recorded routes. In some embodiments, the controller may track performance information such as the frequency with which transmission failures occur in particular routes and may select more reliable routes. Over time, the recorded route may become a library allowing a controller **132***a*, **132***b*, **132***c*, **132***d* to resolve an increasing number of routing requirement without sending a request to a supervisor. In addition, some or all of the controllers may be configured to find routes without reference to previously recorded route or making a request to a supervisor.

In this manner, the controllers **132***a*, **132***b*, **132***c*, **132***d* in each card **106** are able to provide routes for data streams through the router **100**. Some of the routes may traverse only the line card on which a data stream is received while other routes may traverse various line cards, fabric cards and backplane connections. In doing so, the controller can reduce the number of requests transmitted to the supervisor, increasing the rate at which data streams can be coupled through a router, particularly when a router receives, routes and transmits a large number of data streams.

Router **100** has been described as a video router. A video router will typically receive audio/video data streams (which may be referred to as transport streams). In some embodiments, the data streams may also include non-video streams or may not include any video streams at all.

Reference is next made to FIG. **7**, which illustrates a control hierarchy between the card controllers, supervisors, other routers and other devices in a system. In some situations, a plurality of routers **700** may be coupled to provide a data transmission system. For example, routers **700** may be installed in a video processing facility such as a television studio or broadcast facility. Some or all of the routers may receive and transmit a plurality of input and output data streams. In some facilities, hundreds, thousands or even millions of data streams may be received and transmitted. The group of routers will typically be interconnected with a variety of other equipment including signal processors, analytic devices and other devices that generate or require data streams that are switched through one or more routers.

Each router **700** is coupled to a supervisor system **754**, which may include a plurality of supervisors **738**, such as a first supervisor sub-system **738***a*, a second supervisor sub-system **738***b*, a third supervisor sub-system **738***c*, a fourth supervisor sub-system **738***d*, a fifth supervisor sub-system **738***e*, a sixth supervisor sub-system **738***f*, a seventh supervisor sub-system **738***g* and an eighth supervisor sub-system **738***h*. The supervisor system **754** forms a hierarchy in conjunction with the controllers **732** in each router. As described above, a card controller **732** on a card **706**, including a line card **716** and a fabric card **740**, in a router **700** may control the configuration of the corresponding switch **724** on its card **706** and may also be authorized to control the configuration of switches **724** on other cards within the same router. The controller may send requests to a corresponding supervisor sub-system **738***a* when the controller is unable to determine a route for a data stream, for example, when the controller is unable to allocate switches or connections to set up a required route, or when a route may require coordination between routers or under other conditions, which may include instructions from a supervisor to always make a request to the supervisor when certain types of data streams are received or after a particular time or other conditions. In some cases, two or more supervisors may be assigned to each router and may act as primary and backup routers, may operate in parallel, or may operate in a distributed manner to manage the flow and latency or requests made to the supervisory system **754**.

The supervisory system **754** may itself be coupled to other devices in a facility via a controller communication network **736** to receive and provide control and status information about the routers **700**. Such control and status information may be used to control the routing of data streams within and between routers **700**. For example, the other devices in the facility may identify high priority data streams that are to be switched through one or more routers **700** to reach a particular destination. Supervisor system **754** may instruct one or more of the routers to configure an appropriate route between a port on which a high priority data stream is to be received and its destination. In such a situation, a supervisor sub-system **738** may instruct the routers to configure a route directly, without previously having received a request from a controller **732**.

FIG. **7** illustrates a control hierarchy in which the supervisory system **754** communicates with other devices, which may be at the same or a different facility as the supervisory system. Supervisors **738** in the supervisor system **754** control the routing of data streams within and between the routers and between the routers and other devices. Controllers **732** in the routers can control routes directly within the router and may request control instructions from supervisors to generate requests. The supervisory system **754** and the controllers **732** are part of a control layer **756** that provides routes for data streams.

FIG. **7** also illustrates a data layer **758** in which the data streams are transmitted. The data layer **758** includes input ports **718**, switches **724**, backplane connections **752** and output ports **720**. The control layer **756** configures the data layer so that data streams are able to traverse the data layer between input ports and output ports.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the scope of the invention, which is limited only by the appended claims.

The invention claimed is:

1. A data transmission system, comprising:

a plurality of video routers;

a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising instructions to selectively configure the one or more routers; and

a controller communication network for coupling the plurality of video routers and the supervisory system,

wherein, each video router comprises:

a backplane including a plurality of backplane connections,

at least one line card, the line card comprising:

a plurality of input ports and output ports, each input port and output port being coupled to a respective external signal through the backplane, and

a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein a first plurality of input and output switch terminals are coupled to a respective plurality of input and output ports and a second plurality of input and output switch terminals are coupled to a respective plurality of backplane connections, and

at least one fabric card, each fabric card comprising:

a fabric card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein the plurality of input and output switch terminals are coupled to a respective plurality of backplane connections, and

wherein, each line card and fabric card comprises a card controller, the card controller being coupled to one or more cross-point switches and configured to selectively couple one or more input switch terminals of a cross-point switch to one or more output switch terminals of that cross-point switch, the cross-point switch being a fabric card cross-point switch or a line card cross-point switch,

wherein, the controller communication network is communicably coupled to each card controller to control the operation of each line card and fabric card.

2. The data transmission system of claim **1**, wherein a card controller of a first card in a first video router configures a corresponding cross-point switch of the first card to route a data stream from an input port to an output port, the first card and the second card being a line card or a fabric card.

3. The data transmission system of claim **1**, wherein a card controller of a first card in a first video router configures a cross-point switch of a second card in the first video router to route a data stream from an input port to an output port, the first card and the second card being a line card or a fabric card.

4. The data transmission system of claim **1**, wherein when a data stream is received at an input port of a first card, a first card controller corresponding to the first card is configured to transmit a data request to the supervisory system, wherein the supervisory system is configured to:

determine an output destination identifying an output port, and

generate one or more router configuration signals for one or more card controllers based on the output destination, wherein the one or more card controllers configure one or more cross-point switches to route the data stream to the output port.

5. The data transmission system of claim **1**, wherein when a data stream is received at an input port of a first card, a first card controller corresponding to the first card is configured to:

determine an output destination identifying an output port, and

transmit a data request to the supervisory system,

wherein the supervisory system is configured to:

generate one or more router configuration signals for one or more card controllers based on the output destination, wherein the one or more card controllers configure one or more cross-point switches to route the data stream to the output port.

6. The data transmission system of claim **4**, wherein if the data stream is designated a priority stream, at least one of the one or more card controllers reconfigures the corresponding cross-point switch to route the priority stream.

7. The data transmission system of claim **1**, wherein the backplane comprises a plurality of backplane connectors for receiving the at least one line card and the at least one fabric card.

8. The data transmission system of claim **7**, wherein each backplane connector comprises a plurality of backplane contacts, wherein each line card and each fabric card comprises a plurality of card pins, and wherein the plurality of backplane contacts and the plurality of card pins provide an electrical connection when coupled.

9. The data transmission system of claim **1**, further comprising a switch configuration database coupled to the controller communication network and configured to store coupling of the input switch terminals of at least one line card cross-point switch and the fabric card cross-point switch to corresponding output switch terminals.

10. The data transmission system of claim **1**, wherein the switch configuration database is provided within the card controllers.

11. A method of routing video signals from a plurality of input ports to a plurality of output ports using at least one video router of the data transmission system of claim **1**, the method comprising:

receiving a data stream at an input port of a card, the card being a line card;

receiving one or more router configuration signals by one or more card controllers, at least one card controller being a line card controller of the line card; and

configuring one or more cross-point switches by card controllers based on the one or more router configuration signals to route the data stream between the input port and an output destination, wherein at least one of the one or more cross-point switches correspond to a cross-point switch of the line card.

12. The method of claim **11**, further comprising:

processing the data stream by a first card controller of a first card to generate an output destination identifying an output port, the first card controller being a line card controller; and

transmitting a routing request to the supervisory system, the routing request comprising the output destination.

13. The method of claim **11**, further comprising:

transmitting a routing request to the supervisory system;

determining, at the supervisory system, an output destination identifying an output port, and

generating, at the supervisory system, the one or more router configuration signals based on the output destination.

14. The method of claim **11**, wherein a card controller of a first card in a first video router configures a cross-point switch of a second card in a second video router to route a data stream from an input port to an output port, the first card and the second card being a line card or a fabric card.

15. The method of claim **11**, wherein a card controller of a first card in a first video router configures a cross-point switch of a second card in the first video router to route a data stream from an input port to an output port, the first card and the second card being a line card or a fabric card.

16. A data transmission system, comprising:

a control layer comprising:

a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising instructions to selectively configure the one or more routers, and

one or more card controllers provided in the one or more video routers, each card controller configured to selectively couple input switch terminals of one or more cross-point switches to output switch terminals of the corresponding one or more cross-point switches, and

a data layer comprising:

one or more cross-point switches, the one or more cross-point switches provided in the one or more video routers, each cross-point switch comprising a plurality of input switch terminals and a plurality of output switch terminals,

a backplane including a plurality of backplane connections, wherein a subset of the plurality of input switch terminals and the output switch terminals are coupled to a respective plurality of backplane connections,

a plurality of input ports and a plurality of output ports corresponding to each video router, and

a controller communication network for coupling the control layer and the data layer,

wherein, the supervisory system is configured to:

receive a request signal from a card controller, and

transmit a router configuration signal to one or more card controllers, the router configuration signal com-

prising instructions to selectively couple input switch terminals to output switch terminals of the one or more cross-point switches coupled to the one or more card controllers.

17. The data transmission system of claim **16**, wherein of a first card in a first video router configures a corresponding cross-point switch of the first card to route a data stream from an input port to an output port, the first card and the second card being a line card or a fabric card.

18. The data transmission system of claim **16**, wherein a card controller of a first card in a first video router configures a cross-point switch of a second card in the first video router to route a data stream from an input port to an output port, the first card and the second card being a line card or a fabric card.

19. The data transmission system of claim **16**, wherein the supervisory system is configured to receive status information from the one or more video routers, wherein the one or more router configuration signals are generated based on the status information.

20. The data transmission system of claim **16**, wherein if a data stream is designated a priority stream, the card controllers reconfigure the cross-point switches to route the priority stream.

* * * * *

EXHIBIT G

US010164877B2

(12) **United States Patent**
Patel

(10) **Patent No.: US 10,164,877 B2**
(45) **Date of Patent: *Dec. 25, 2018**

(54) **VIDEO ROUTER**

(71) Applicant: **Evertz Microsystems Ltd.**, Burlington (CA)

(72) Inventor: **Rakesh Patel**, Mississauga (CA)

(73) Assignee: **Evertz Microsystems Ltd.**, Burlington (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **15/910,099**

(22) Filed: **Mar. 2, 2018**

(65) **Prior Publication Data**

US 2018/0191608 A1 Jul. 5, 2018

Related U.S. Application Data

(63) Continuation of application No. 15/484,852, filed on Apr. 11, 2017, now Pat. No. 9,942,139, which is a (Continued)

(51) **Int. Cl.**
H04L 12/773 (2013.01)
H04L 12/933 (2013.01)
H04L 12/931 (2013.01)

(52) **U.S. Cl.**
CPC ***H04L 45/60*** (2013.01); ***H04L 49/101*** (2013.01); ***H04L 49/206*** (2013.01)

(58) **Field of Classification Search**
CPC H04L 45/60; H04L 49/101; H04L 49/206
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,976,134 B1 12/2005 Lolayekar et al.
7,194,550 B1 * 3/2007 Chamdani H04L 67/1097 709/227

(Continued)

FOREIGN PATENT DOCUMENTS

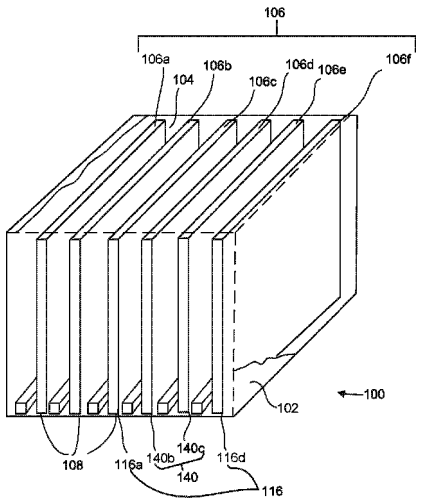
CA 2866073 A1 4/2015

Primary Examiner — Ayaz R Sheikh
Assistant Examiner — Debebe A Asefa
(74) *Attorney, Agent, or Firm* — Bereskin & Parr LLP/S.E.N.C.R.L., s.r.l.

(57) **ABSTRACT**

The embodiments described herein provide a data transmission system comprising a plurality of video routers, a supervisory system for transmitting one or more router configuration signals to one or more video routers, and a control communication network for coupling the plurality of video routers and the supervisory system. Each router in the system comprises a backplane including a plurality of backplane connections, at least one line card and at least one fabric card. Each line card comprises a plurality of input ports and output ports where each input and output port is coupled to a respective external signal through the backplane. Each line card further comprises a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals. Each fabric card comprises a fabric card cross-point switch having a plurality of input switch terminal and a plurality of output switch terminals. Furthermore, each line card and each fabric card comprises a card controller where the card controller selectively couples one or more input switch terminals of a cross-point switch to the output switch terminals of that cross-point switch. The cross-point switches being manipulated by the card controller may belong to one or more different cards within the same video router.

20 Claims, 7 Drawing Sheets



Related U.S. Application Data

continuation of application No. 14/505,124, filed on Oct. 2, 2014, now Pat. No. 9,654,391.

(60) Provisional application No. 61/885,588, filed on Oct. 2, 2013.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,776,161	B2 *	7/2014	Gazier	H04N 21/222 375/240.07
8,891,406	B1	11/2014	Shekhar	
9,654,391	B2	5/2017	Patel	
2017/0279711	A1	9/2017	Patel	

* cited by examiner

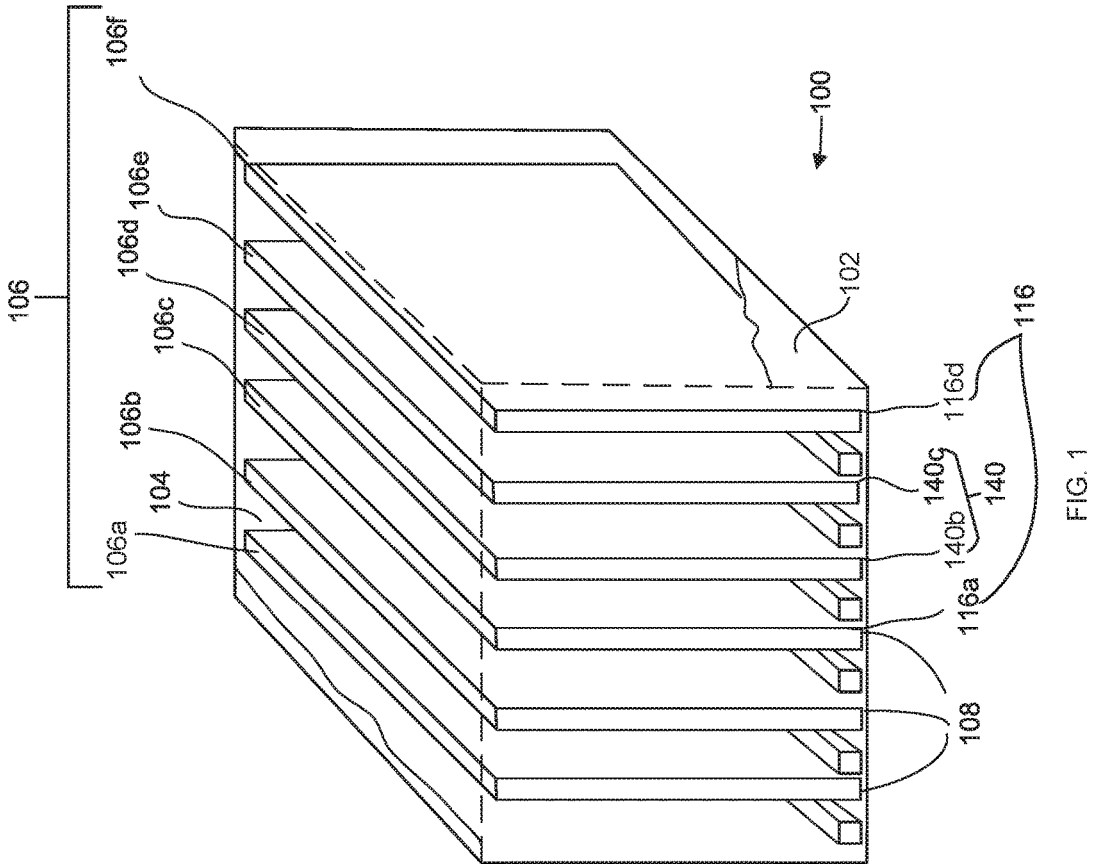

106
106f
106e
106d
106c
106b
104
106a
100
102
116
116d
140c
140
140b
116a
108

FIG. 1

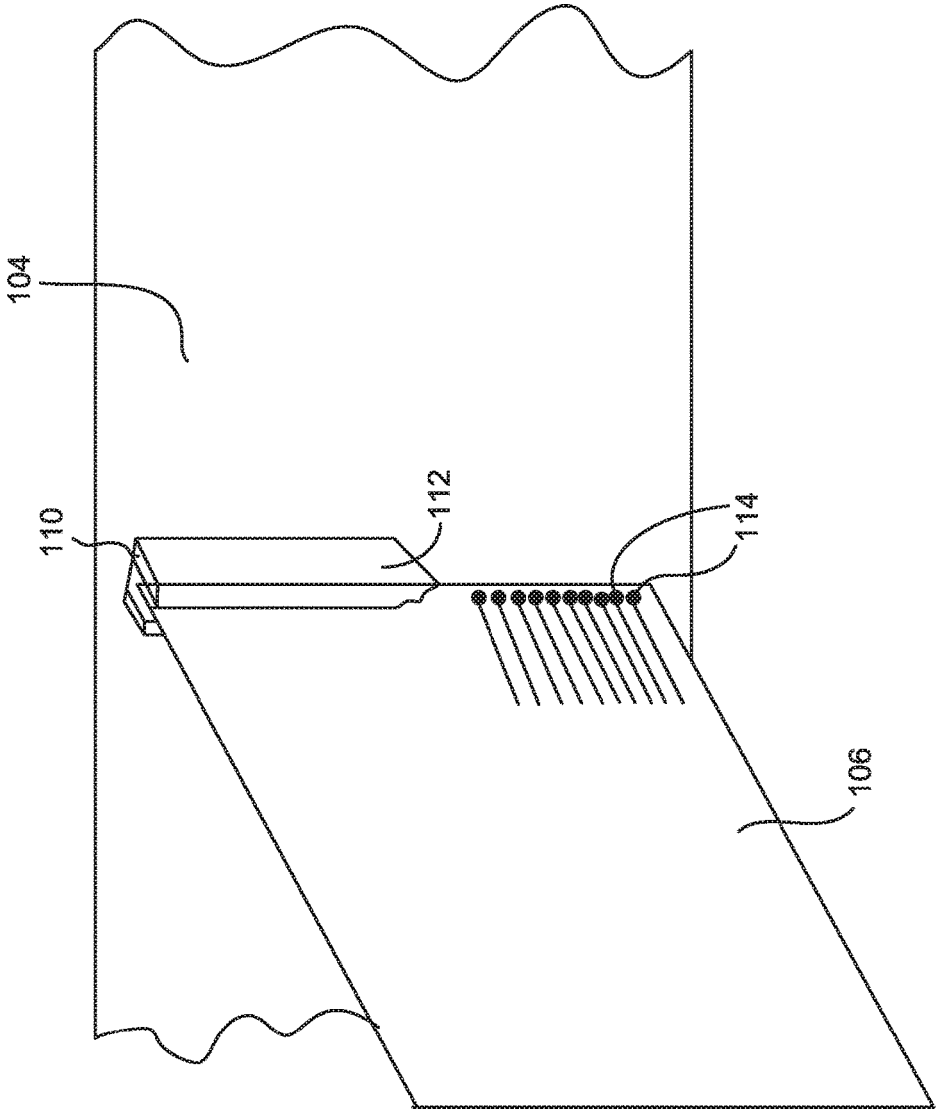


FIG. 2

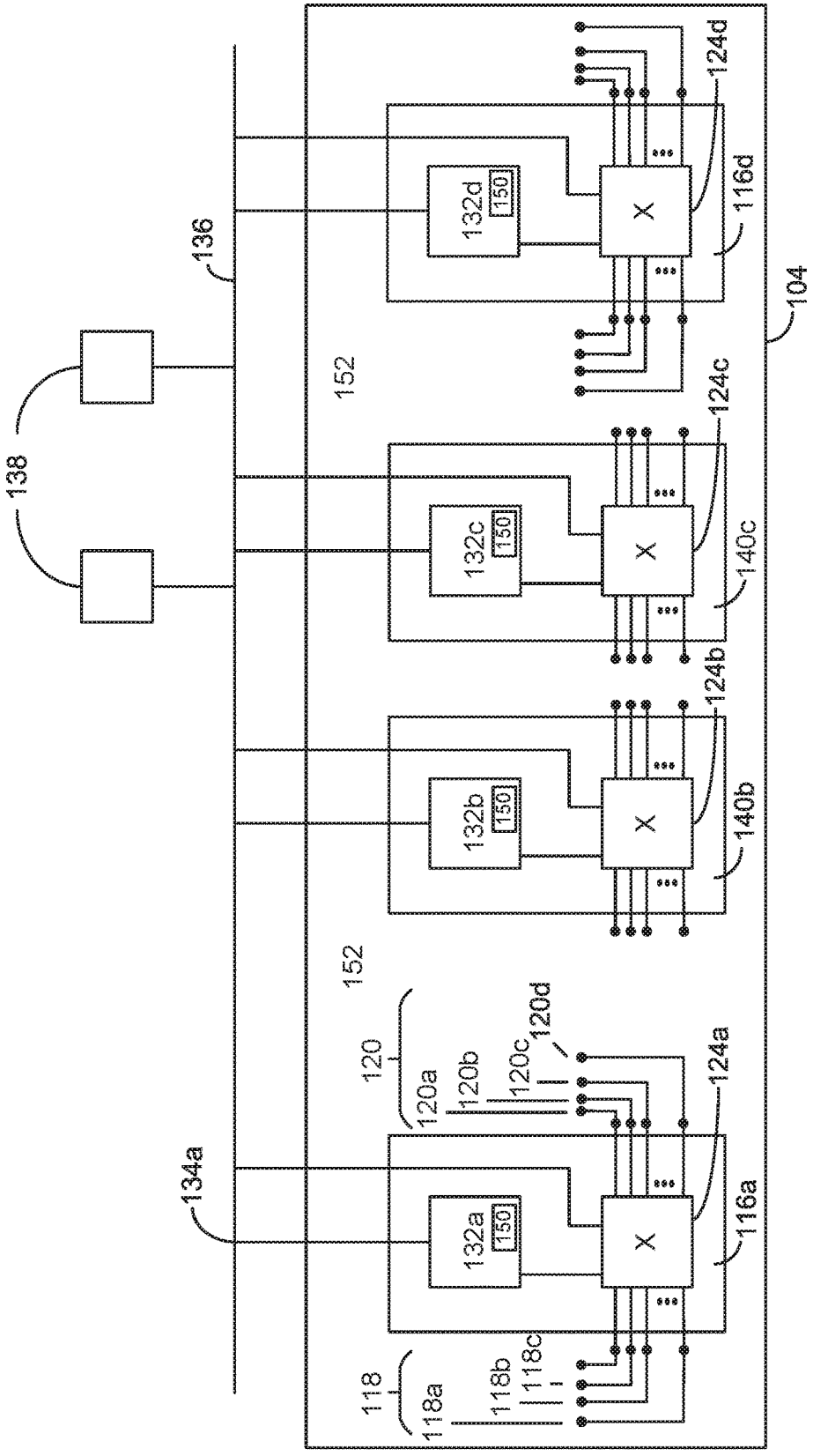

104
136
138
152
152
120
120a
120b
120c
120d
118
118a
118b
118c
134a
124a
124b
124c
124d
116a
116d
140b
140c
132a
132b
132c
132d
150
X

FIG. 3

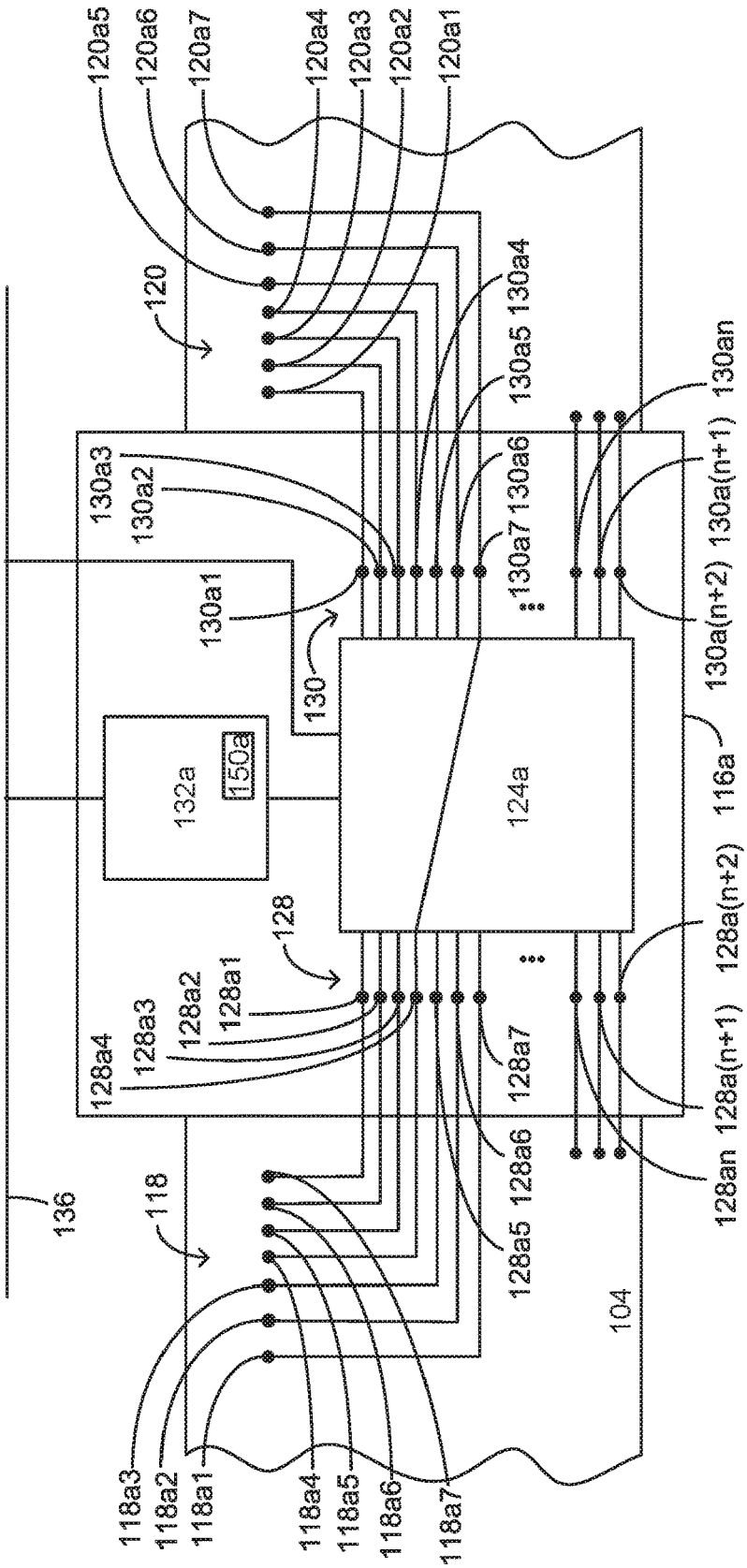

120a5
120a6
120a7
120a4
120a3
120a2
120a1
120
130a4
130a5
130an
130a3
130a2
130a6
130a(n+1)
130a1
130a7
130a(n+2)
130
132a
150a
124a
116a
128
128a4
128a3
128a2
128a1
128a7
128a(n+2)
128a(n+1)
128an
128a6
128a5
118
136
104
118a3
118a2
118a1
118a4
118a5
118a6
118a7

FIG. 4

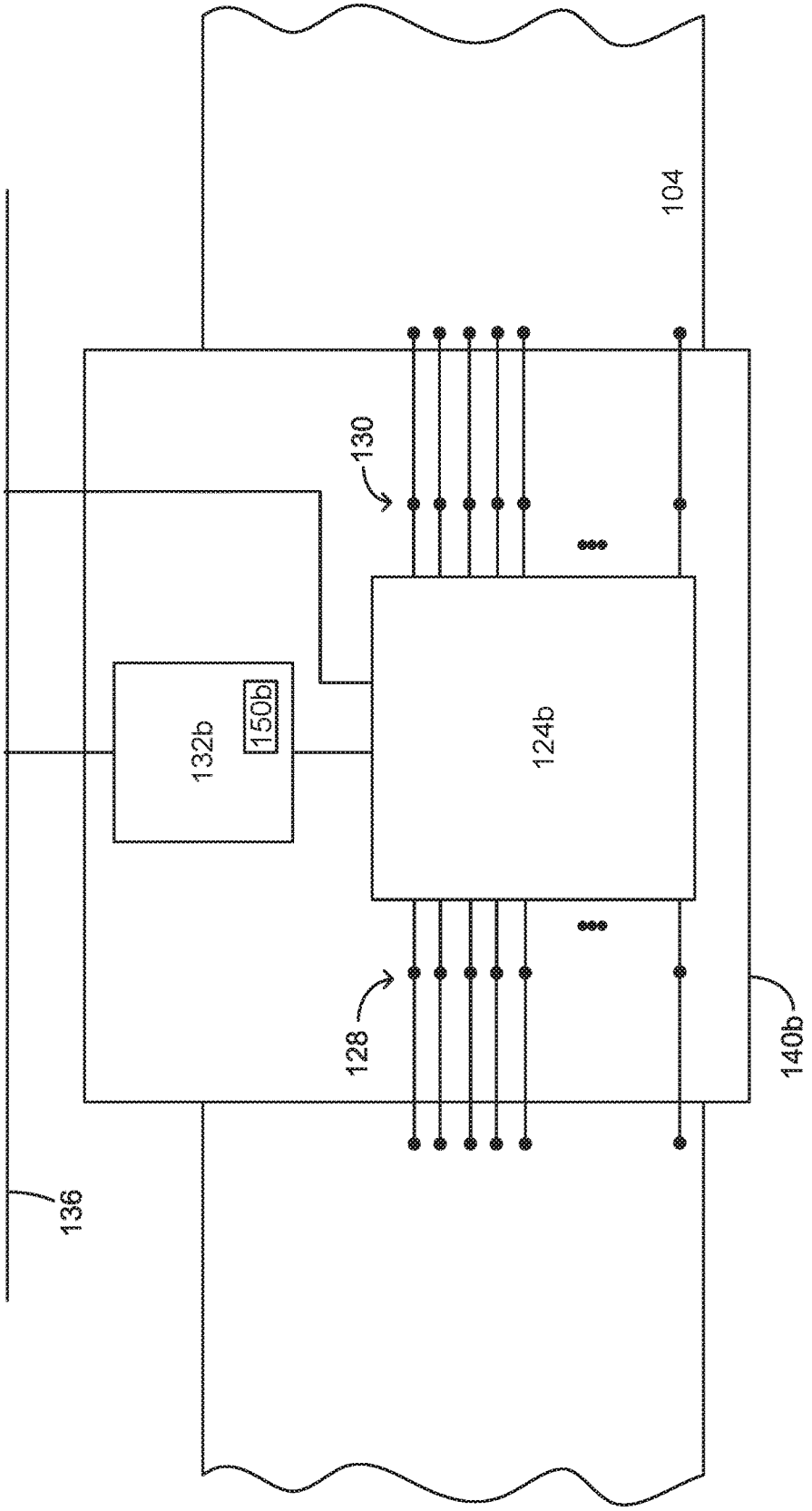


FIG. 5

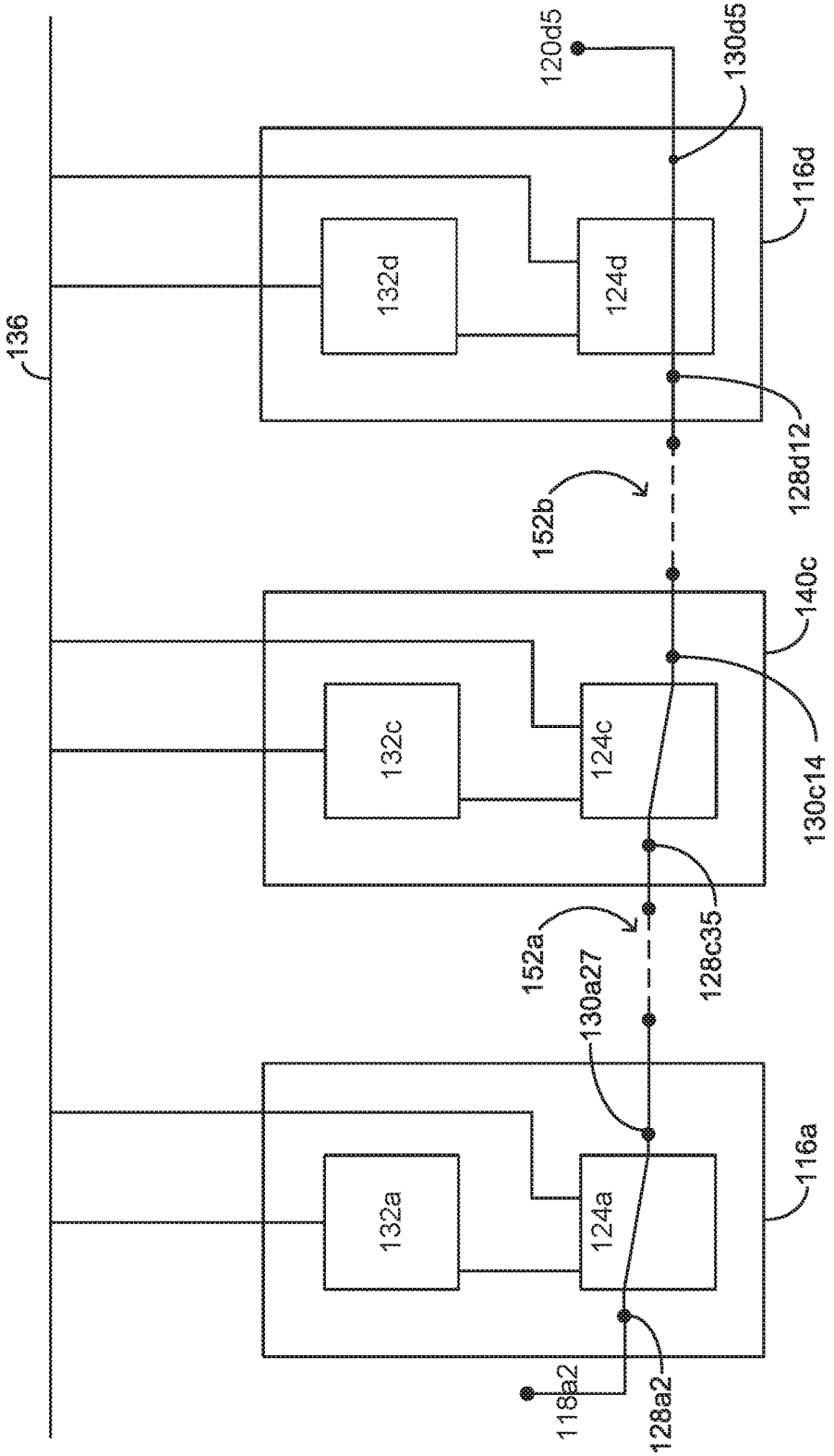

136
120d5
130d5
116d
132d
124d
128d12
152b
140c
132c
124c
130c14
128c35
152a
130a27
116a
132a
124a
118a2
128a2

FIG. 6

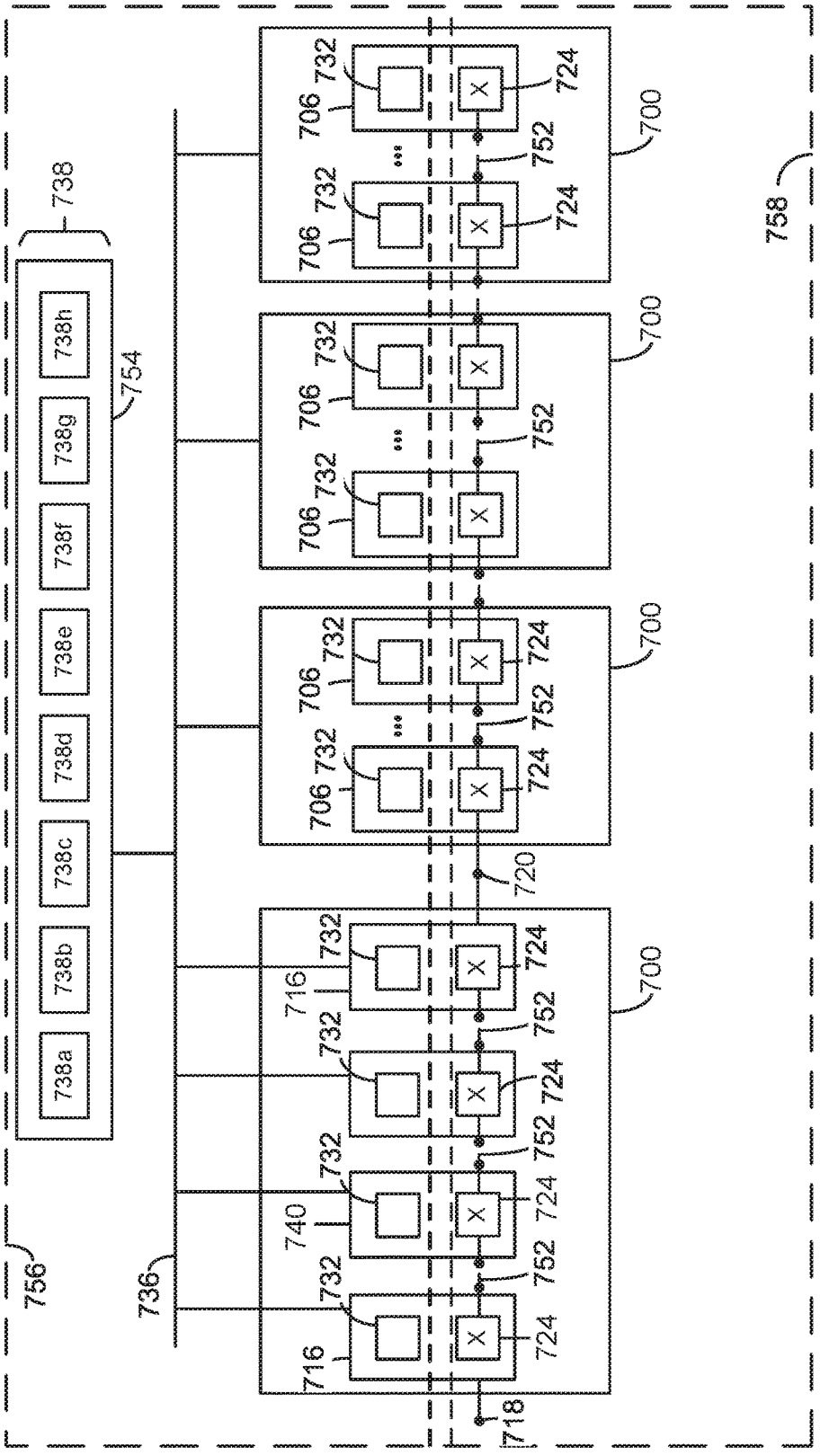

738
738a
738b
738c
738d
738e
738f
738g
738h
754
756
758
736
700
706
732
724
752
716
740
720
718

FIG. 7

VIDEO ROUTER

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 15/484,852, filed on Apr. 11, 2017, which is a continuation of U.S. patent application Ser. No. 14/505,124, filed on Oct. 2, 2014, (now issued as U.S. Pat. No. 9,654,391), which claims the benefit of U.S. Provisional Patent Application No. 61/885,588, filed Oct. 2, 2013. The entire contents of the applications are hereby incorporated by reference.

FIELD

The described embodiments relate to routers for video signals and other data streams.

BACKGROUND

The number of devices coupled to data communications networks is increasing rapidly. The routing of data streams from and to such devices is increasingly more complex and difficulty in allocating efficient routes, or even any route at all, in various components in a communication network can affect the quality of service delivery to a user of a device. For example, communication networks typically contain routers that couple an input data stream received at an input port to an output port at which the data stream is available to a downstream device. As the size of routers increases (i.e. as the number of input and output ports on a router increases, then complexity of creating efficient routing within the router and between network devices increase non-linearly.

It is desirable to provide an efficient system and methods that allows a network device to efficiently configure routes for data streams.

SUMMARY

In one aspect, in at least one embodiment described herein, there is provided a data transmission system comprising a plurality of video routers, a supervisory system for transmitting one or more router configuration signals to one or more video routers, and a control communication network for coupling the plurality of video routers and the supervisory system. Each router in the system comprises a backplane including a plurality of backplane connections, at least one line card and at least one fabric card. Each line card comprises a plurality of input ports and output ports where each input and output port is coupled to a respective external signal through the backplane. Each line card further comprises a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals. Each fabric card comprises a fabric card cross-point switch having a plurality of input switch terminal and a plurality of output switch terminals. Furthermore, each line card and each fabric card comprises a card controller where the card controller selectively couples one or more input switch terminals of a cross-point switch to the output switch terminals of that cross-point switch. The cross-point switches being manipulated by the card controller may belong to one or more different cards within the same video router.

In some cases, the card controller of a first card in a first video router configures a corresponding cross-point switch of the first card to route a data stream from an input port to an output port, where the first card and the second card are a line card or a fabric card.

In some other cases, the card controller of a first card in a first video router configures a cross-point switch of a second card in the first video router to route a data stream from an input port to an output port, where the first card and the second card are a line card or a fabric card.

In various cases, where when a data stream is received at an input port of a first card, a first card controller corresponding to the first card is configured to transmit a data request to the supervisory system, where the supervisory system is configured to: determine an output destination identifying an output port, and generate one or more router configuration signals for one or more card controllers based on the output destination, wherein the one or more card controllers configure one or more cross-point switches to route the data stream to the output port.

In various cases, where when a data stream is received at an input port of a first card, a first card controller corresponding to the first card is configured to: determine an output destination identifying an output port, and transmit a data request to the supervisory system, where the supervisory system is configured to: generate one or more router configuration signals for one or more card controllers based on the output destination, wherein the one or more card controllers configure one or more cross-point switches to route the data stream to the output port.

In various cases, where if the data stream is designated a priority stream, at least one of the one or more card controllers reconfigures the corresponding cross-point switch to route the priority stream.

In various cases, where the backplane comprises a plurality of backplane connectors for receiving the at least one line card and the at least one fabric card.

In various cases, where each backplane connector comprises a plurality of backplane contacts, wherein each line card and each fabric card comprises a plurality of card pins, and wherein the plurality of backplane contacts and the plurality of card pins provide an electrical connection when coupled.

In various cases, the system further comprises a switch configuration database coupled to the controller communication network and configured to store coupling of the input switch terminals of at least one line card cross-point switch and the fabric card cross-point switch to corresponding output switch terminals.

In various cases, the switch configuration database is provided within the card controllers.

In another aspect, in at least one embodiment described herein, there is provided a method of routing video signals from a plurality of input ports to a plurality of output ports using at least one video router of a data transmission system disclosed herein. The method comprises receiving a data stream at an input port of a card, the card being a line card, receiving one or more router configuration signals by one or more card controllers, at least one card controller being a line card controller of the line card, and configuring one or more cross-point switches by card controllers based on the one or more router configuration signals to route the data stream between the input port and an output destination, wherein at least one of the one or more cross-point switches correspond to a cross-point switch of the line card.

In various embodiments, the method of routing video signals is configured to operate in accordance with the devices defined above or in accordance with the teachings herein.

In another aspect, in at least one embodiment described herein, there is provided a data transmission system comprising a control layer, a data layer and a controller communication network for coupling the control layer and the data layer. The control layer comprises a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising instructions to selectively configure the one or more routers, and one or more card controllers provided in the one or more video routers, each card controller configured to selectively couple input switch terminals of one or more cross-point switches to output switch terminals of the corresponding one or more cross-point switches. The data layer comprises one or more cross-point switches, the one or more cross-point switches provided in the one or more video routers, each cross-point switch comprising a plurality of input switch terminals and a plurality of output switch terminals, a backplane including a plurality of backplane connections, wherein a subset of the plurality of input switch terminals and the output switch terminals are coupled to a respective plurality of backplane connections, a plurality of input ports and a plurality of output ports corresponding to each video router, where the supervisory system is configured to: receive a request signal from a card controller, and transmit a router configuration signal to one or more card controllers, the router configuration signal comprising instructions to selectively couple input switch terminals to output switch terminals of the one or more cross-point switches coupled to the one or more card controllers.

In various embodiments, the data transmission system is configured to operate in accordance with the devices and methods defined above or in accordance with the teachings herein.

Other features and advantages of the present application will become apparent from the following detailed description taken together with the accompanying drawings. It should be understood, however, that the detailed description and the specific examples, while indicating preferred embodiments of the application, are given by way of illustration only, since various changes and modifications within the spirit and scope of the application will become apparent to those skilled in the art from this detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the applicant's teachings described herein, reference will now be made, by way of example only, to the accompanying drawings which show at least one exemplary embodiment, and in which:

FIG. **1** is a cross-section of a video router according to an example embodiment;

FIG. **2** is a cross-section of a video router according to another example embodiment;

FIG. **3** is a block diagram of a video router according to an example embodiment;

FIG. **4** is a block diagram of a video router according to another example embodiment;

FIG. **5** is a block diagram of a video router according to another example embodiment;

FIG. **6** is a block diagram of a video router according to another example embodiment;

FIG. **7** is a block diagram of a control hierarchy of a video router according to an example embodiment.

For simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements.

DESCRIPTION OF EXEMPLARY EMBODIMENTS

Reference is first made to FIGS. **1** and **2**, which illustrates a first video router **100** with an integrated control layer. Router **100** includes a frame or housing **102**, a backplane **104** and a plurality of cards **106**, such as a first card **106***a*, a second card **106***b*, a third card **106***c*, a fourth card **106***d*, a fifth card **106***e* and a sixth card **106***f*. The frame **102** includes a plurality of frame slots **108** in which cards may be received and held in place. The backplane **104** includes backplane connector **110** corresponding to each slot **108** and each card **106**. Each backplane connector includes a plurality of backplane pins or contacts **112**. Each card **106** includes a plurality of card pins or contacts **114**, each of which corresponds to a backplane pin **112** of the corresponding backplane connector. When a card **106** is installed in frame **102**, the card pins **114** couple with corresponding backplane pins **112** making an electrical connection through which a data signal may be transmitted.

Cards **106** may include various types of cards. For example, some of the cards may be line cards **116**, such as a first line card **116***a* and a second line card **116***d*, which include input ports or output ports for respectively receiving and transmitting data signals, or both input and output ports. Other cards **106** may be fabric cards **140**, such as a first fabric card **140***b* and a second fabric card **140***c*, which facilitate switching of signals between various input and port ports.

Reference is made to FIGS. **3**, **4** and **5**, which schematically illustrate components of router **100**. In the present example embodiment, each input port **118** or output port **120** on a line card **116***a* is coupled to an external signal through the backplane **104**. In the illustrated embodiment of FIG. **3**, input port **118** comprises a first input port **118***a*, a second input port **118***b* and a third input port **118***c*, and output port **120** comprises a first output port **120***a*, a second output port **120***b*, a third output port **120***c* and a fourth output port **120***d*. The backplane may, for example, include a pass-through connector to which a line card port **118**, **120** may be coupled within frame **102** and to which a cable (not shown) may be coupled on the rear of the backplane. The line card port **118**, **120** is electrically coupled to the cable (not shown), allowing the line card to receive or transmit a data signal on the cable. In other embodiments, line card ports may be directly coupled to a cable or may be coupled to a cable through the backplane using a coupling other than a pass-through connector.

Line card **116***a* includes a line card crosspoint switch **124***a* with a plurality of switch terminals. In this example, crosspoint switch **124***a* has a plurality of input switch terminals **128** and a plurality of output switch terminals **130**. Each input port **118** is coupled to at least one input switch terminal **128** and each output port **120** is coupled to at least one output switch terminal **130**. In addition, a plurality of input switch terminals **128** are coupled to the backplane **104** through the corresponding backplane connector **110**. A plurality of output switch terminals **130** are coupled to the backplane **104** through the corresponding backplane connector **110**.

Line card **116***a* also includes a line card controller **132***a* that is coupled to crosspoint switch **124***a* and which provides

control signals to couple or decouple particular input switch terminals **128** to particular output switch terminals **130**. Card controller **132***a* is coupled to a controller communication network **136** at a control system terminal **134***a* through which the card controller **132***a* may communicate with other cards **106** and with external control devices such as an external supervisor **138**. In some embodiments, a line card controller **132***a* may be coupled to controller communication network **136** through the backplane or through another communication bus in frame **102** to which the line card is couple when installed in the frame.

Each fabric card, such as fabric card **140***b* includes a card controller **132***b* and a crosspoint switch **124***b*, which are coupled together and operate in a manner similar to the card controller **132***a* and crosspoint switch **124***a* of line card **116***a*. Crosspoint switch **124***b* includes a plurality of input switch terminals **128** and output switch terminals **130** that are coupled to the backplane **104**. The crosspoint switch **124***b* may be configured to couple any of the input switch terminals **128** to any of the output switch terminals **130** under the control of card controller **132***b*. As illustrated in FIG. **3**, router **100** further includes a fabric card **140***c*, which includes a card controller **132***c* and a crosspoint switch **124***c*, and a line card **116***d*, which includes a card controller **132***d* and a crosspoint switch **124***d*.

Each crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* in router **100** is coupled to the controller communication network **136** through which the configuration of the crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* may be changed by card controller **132***a*, **132***b*, **132***c*, **132***d* on other cards **106**.

Router **100** also includes a switch configuration table or database **150**. Database **150** records the current setting for every input switch terminal and output switch terminal in all cross-point switches **124***a*, **124***b*, **124***c*, **124***d* in the router **100**. For example, part of the contents of database **150** may be:

Switch	Terminal	Setting
124a	128a1	Coupled to 130c3
124a	128a2	Coupled to 130a27
124a	128a3	Open
124a	128a4	Coupled to 130a8
. . .	. . .	. . .
124a	130a3	Coupled to 128a1
124a	130a4	Open
124a	130a5	Coupled to 128a2
124a	130a6	Open
124a	130a7	Coupled to 128a2
124b	130a8	Coupled to 128a4
. . .	. . .	. . .
124b	128b1	Open
. . .	. . .	. . .
124c	128c35	Coupled to 130c14
. . .	. . .	. . .
124c	130c14	Coupled to 128c35
. . .	. . .	. . .
124d	128d12	Coupled to 130d5
124d	130d5	Coupled to 128d12
. . .	. . .	. . .

where router **100** of FIG. **4** comprises a first input port **118***a***1**, a second input port **118***a***2**, a third input port **118***a***3**, a fourth input port **118***a***4**, a fifth input port **118***a***5**, a sixth input port **118***a***6**, a seventh input port **118***a***7**, a first output port **120***a***1**, a second output port **120***a***2**, a third output port **120***a***3**, a fourth output port **120***a***4**, a fifth output port **120***a***5**, a sixth output port **120***a***6**, a seventh output port **120***a***7**, a first input switch terminal **128***a***1**, a second input switch terminal **128***a***2**, a third input switch terminal **128***a***3**, a fourth input switch terminal **128***a***4**, a fifth input switch terminal **128***a***5**, a sixth input switch terminal **128***a***6**, a seventh input switch terminal **128***a***7**, an nth input switch terminal **128***an*, a (n+1)th input switch terminal **128***a*(n+1), a (n+2)th input switch terminal **128***a*(n+2), a first output switch terminal **130***a***1**, a second output switch terminal **130***a***2**, a third output switch terminal **130***a***3**, a fourth output switch terminal **130***a***4**, a fifth output switch terminal **130***a***5**, a sixth output switch terminal **130***a***6**, a seventh output switch terminal **130***a***7**, an nth output switch terminal **130***an*, a (n+1)th output switch terminal **130***a*(n+1) and a (n+2)th output switch terminal **130***a*(n+2).

Database **150** is accessible to each of the controllers **132***a*, **132***b*, **132***c*, **132***d*. In some embodiments, the database **150** may be recorded in a central location, for example, in one of the controllers **132***a*, **132***b*, **132***c*, **132***d* where the local controller **132***a*, **132***b*, **132***c*, **132***d* may access the database directly and each of the other controllers may access the database through the controller communication network **136**.

In other embodiments, the database may be a distributed database with components that are located in multiple locations within router **100**. For example, components of database **150***a*, **150***b* may be located in each of the controllers **132***a*, **132***b*, as is illustrated in FIGS. **4** and **5**. Each controller **132***a*, **132***b* may contain the status of the cross-point switch **124***a*, **124***b* in the same card **106**. Controllers **132***a*, **132***b*, **132***c*, **132***d* on other cards **106** may access the status of non-local cross-points switches through the controller communication network **136**.

In other embodiments, the database may be recorded in a data storage device or system that is external to router **100**, but which is accessible to the controller **132***a*, **132***b*, **132***c*, **132***d*.

In still other embodiments, a copy of the entire database **150** may be maintained at each controller **132***a*, **132***b*, **132***c*, **132***d*. A synchronization system that locks some or all of each copy of the database may be used to ensure that all copies of the database **150** are maintained in synchronization. In such embodiments, each controller **132***a*, **132***b*, **132***c*, **132***d* may use only its local copy of the entire database **150**.

In various embodiments, a combination of these techniques may be used to maintain database **150**.

As illustrated in FIG. **5**, the backplane **104** includes a plurality of static point-to-point backplane connections **152** that couple output switch terminals on one card **106** to input switch terminals on another card **106**. For example, backplane connections may couple output switch terminal **130***a***27** on line card **116***a* to input switch terminal **128***c***35** on fabric card **140***c*. Various embodiments may include as many or as few backplane connections between output switch terminals to input switch terminals.

In any particular embodiment, the sizes of the various crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* and the number of backplane connections can be selected to provide a desired level of functionality in the router. For example, in a router designed for a specific purpose, for example, in which only a limited number of couplings between input ports **118** and output ports **120** may be required, may have a correspondingly limited number of backplane connections **152**. Fabric cards are typically useful to increase the flexibility with which a particular input port can be coupled to a particular output port. In some embodiments, all cards **106** may be line cards with no fabric cards.

By selectively configuring one or more crosspoint switches **124***a*, **124***b*, **124***c*, **124***d*, a particular input port **118**

on one line card **116** may be coupled to a particular output port **120** on the same or another line card.

Reference is made to FIG. **4**. For example, if:

input port **118***a***4** is fixedly coupled to input switch terminal **128***a***4**;

switch crosspoint switch **124***a* couples input switch terminal **128***a***4** to output switch terminal **130***a***7**; and

output switch terminal **130***a***7** is fixedly coupled to output port **120***a***7**, then an input data signal received input port **118***a***4** on line card **116***a* will be coupled to output port **120***a***7**.

Reference is made to FIG. **6**. If:

input port **118***a***2** is fixedly coupled to switch input terminal **128***a***2**;

switch input terminal **128***a***2** is coupled to output switch terminal **130***a***27** in crosspoint switch **124***a;*

output switch terminal **130***a***27** is coupled to input switch terminal **128***c***35** in fabric card **140***c* through backplane connection **152***a;*

switch input terminal **128***c***35** is coupled to output switch terminal **130***c***14** in crosspoint switch **124***c;*

output switch terminal **130***c***14** is coupled to input switch terminal **128***d***12** through the backplane connection **152***b;*

switch input terminal **128***d***12** is coupled to output switch terminal **130***d***5** in crosspoint switch **124***d*; and

output switch terminal **130***d***5** is fixedly coupled to output port **120***d***5**, then an input data signal received at input port **118***a***2** on line card **116***a* will be coupled to output port **120***d***5** on line card **116***d*.

In router **100**, each controller **132***a*, **132***b*, **132***c*, **132***d* is coupled to each crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* in the router and may instruct any crosspoint switch **124***a*, **124***b*, **124***c*, **124***d* to couple specific input switch terminals and output switch terminals within the crosspoint switch **124***a*, **124***b*, **124***c*, **124***d*. Through one or more steps through crosspoint switches and through backplane connection **152**, an input signal received at an input port **118** may be coupled to an output port **120** on the same or a different line card. In some embodiments, the crosspoint switches and the number of pairs of output switch terminals and input switch terminals coupled by backplane connections **152** may be sufficient to allow any input port **118** to be coupled to any output port **120**, possibly through a variety of different routes.

A particular controller **132***a*, **132***d* in a line card **116***a*, **116***d* may be configured to ensure that a data signal or data stream received at the line card is routed through to an appropriate destination for the data stream. For example, when a data stream is initially received at an input port **118**, the controller examines the packets in the data stream, which will identify a destination for the data stream. The controller then determines which output port **120** in the router (which may be on the same line card as the controller or on another line card) is coupled to the destination. The controller then determines a path through the router and configures one or more crosspoint switches to provide the path between the input port **118** and the output port **120**. The controller will typically select a route based on router configuration data that is previously recorded in the controller. The router configuration data includes information about the availability of backplane connections between different cards and may include additional information about the router structure or configuration. The controller will also typically consider the contents of the database **150**. Typically a controller will not change the configuration of an input switch terminal or an output switch terminal that is already in use (i.e. coupled to a corresponding switch terminal). In some embodiments, a priority level for some or all of the couplings between different pairs of input switch terminal and output switch terminal may be maintained in database **150**. A controller may determine a priority level for a data stream that the controller is routing through router **100**. If an input switch or an output switch terminal is in use, but the stored priority level for the stream being routed through the switch is lower than the priority of the stream that the controller is attempting to route, then the control may change the configuration of the switch to use it for the higher priority data stream. In some cases, the router may have multiple paths through which a data stream can be routed from a particular input port **118** to a particular output port **120** and it may be possible to provide a needed routing for a high priority data stream without disrupting a lower priority data stream. Each controller may be configured to identify multiple routings to reduce disruption to existing routes set up within the router.

In some conditions, a controller may not be able to determine a route by which a data stream can be delivered to a particular output port **120**. In such conditions, the controller **132** may send a routing request to a supervisor **138** through the controller communication network **136**. A supervisor will typically be an external device that can monitor and control the configuration of crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* in the router **100** and possibly in other routers. In some embodiments, a supervisor **138** may be built into a router. In some embodiments, duplicate or multiple supervisors may be provided to provide redundancy or improved responsiveness when a request is sent to a supervisor or a group of supervisors.

Each time a controller **132***a*, **132***b*, **132***c*, **132***d* changes the configuration of a switch **124***a*, **124***b*, **124***c*, **124***d*, the changes are recorded in the database **150**.

A supervisor may receive various types of requests. For example, a controller may ask a supervisor to provide a route from a particular input port to a particular output port. A controller may ask a supervisor to examine a packet to determine the output port to which the packet (and the corresponding data stream) should be coupled, and possibly also to provide a routing between the input port on which the data stream is received and the output port.

In some embodiments, a supervisor may directly change the configuration of crosspoint switches **124***a*, **124***b*, **124***c*, **124***d* and update database **150** and advise the requesting controller that the request has been satisfied and optionally provide details of configuration changes made in the router. In other embodiments, a supervisor may provide a response to a controller making a request and the controller may then implement the details of the response.

In some embodiments, each controller **132***a*, **132***b*, **132***c*, **132***d* may record some or all of the routes that are used by the controller, including some or all of the requests provided by a supervisor. The controller **132***a*, **132***b*, **132***c*, **132***d* may subsequently refer to the recorded requests to select routes for data streams between input ports **118** and output ports **120** based on the previously recorded routes. In some embodiments, the controller may track performance information such as the frequency with which transmission failures occur in particular routes and may select more reliable routes. Over time, the recorded route may become a library allowing a controller **132***a*, **132***b*, **132***c*, **132***d* to resolve an increasing number of routing requirement without sending a request to a supervisor. In addition, some or all of the controllers may be configured to find routes without reference to previously recorded route or making a request to a supervisor.

In this manner, the controllers **132***a*, **132***b*, **132***c*, **132***d* in each card **106** are able to provide routes for data streams through the router **100**. Some of the routes may traverse only the line card on which a data stream is received while other routes may traverse various line cards, fabric cards and backplane connections. In doing so, the controller can reduce the number of requests transmitted to the supervisor, increasing the rate at which data streams can be coupled through a router, particularly when a router receives, routes and transmits a large number of data streams.

Router **100** has been described as a video router. A video router will typically receive audio/video data streams (which may be referred to as transport streams). In some embodiments, the data streams may also include non-video streams or may not include any video streams at all.

Reference is next made to FIG. **7**, which illustrates a control hierarchy between the card controllers, supervisors, other routers and other devices in a system. In some situations, a plurality of routers **700** may be coupled to provide a data transmission system. For example, routers **700** may be installed in a video processing facility such as a television studio or broadcast facility. Some or all of the routers may receive and transmit a plurality of input and output data streams. In some facilities, hundreds, thousands or even millions of data streams may be received and transmitted. The group of routers will typically be interconnected with a variety of other equipment including signal processors, analytic devices and other devices that generate or require data streams that are switched through one or more routers.

Each router **700** is coupled to a supervisor system **754**, which may include a plurality of supervisors **738**, such as a first supervisor sub-system **738***a*, a second supervisor sub-system **738***b*, a third supervisor sub-system **738***c*, a fourth supervisor sub-system **738***d*, a fifth supervisor sub-system **738***e*, a sixth supervisor sub-system **738***f*, a seventh supervisor sub-system **738***g* and an eighth supervisor sub-system **738***h*. The supervisor system **754** forms a hierarchy in conjunction with the controllers **732** in each router. As described above, a card controller **732** on a card **706**, including a line card **716** and a fabric card **740**, in a router **700** may control the configuration of the corresponding switch **724** on its card **706** and may also be authorized to control the configuration of switches **724** on other cards within the same router. The controller may send requests to a corresponding supervisor sub-system **738***a* when the controller is unable to determine a route for a data stream, for example, when the controller is unable to allocate switches or connections to set up a required route, or when a route may require coordination between routers or under other conditions, which may include instructions from a supervisor to always make a request to the supervisor when certain types of data streams are received or after a particular time or other conditions. In some cases, two or more supervisors may be assigned to each router and may act as primary and backup routers, may operate in parallel, or may operate in a distributed manner to manage the flow and latency or requests made to the supervisory system **754**.

The supervisory system **754** may itself be coupled to other devices in a facility via a controller communication network **736** to receive and provide control and status information about the routers **700**. Such control and status information may be used to control the routing of data streams within and between routers **700**. For example, the other devices in the facility may identify high priority data streams that are to be switched through one or more routers **700** to reach a particular destination. Supervisor system **754** may instruct one or more of the routers to configure an appropriate route between a port on which a high priority data stream is to be received and its destination. In such a situation, a supervisor sub-system **738** may instruct the routers to configure a route directly, without previously having received a request from a controller **732**.

FIG. **7** illustrates a control hierarchy in which the supervisory system **754** communicates with other devices, which may be at the same or a different facility as the supervisory system. Supervisors **738** in the supervisor system **754** control the routing of data streams within and between the routers and between the routers and other devices. Controllers **732** in the routers can control routes directly within the router and may request control instructions from supervisors to generate requests. The supervisory system **754** and the controllers **732** are part of a control layer **756** that provides routes for data streams.

FIG. **7** also illustrates a data layer **758** in which the data streams are transmitted. The data layer **758** includes input ports **718**, switches **724**, backplane connections **752** and output ports **720**. The control layer **756** configures the data layer so that data streams are able to traverse the data layer between input ports and output ports.

The present invention has been described here by way of example only. Various modification and variations may be made to these exemplary embodiments without departing from the scope of the invention, which is limited only by the appended claims.

The invention claimed is:

1. A priority based transmission system comprising:

a plurality of data signals;

a plurality of video routers;

a supervisory system configured to transmit one or more router configuration signals to one or more video routers, the one or more router configuration signals comprising a data signal path;

a controller communication network for coupling the plurality of video routers and the supervisory system;

wherein, each video router comprises:

a backplane including a plurality of backplane connections,

at least one line card, the line card comprising:

a plurality of input ports and output ports, each input port and output port being coupled to a respective data signal through the backplane, and

a line card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein a first plurality of input and output switch terminals are coupled to a respective plurality of input and output ports and a second plurality of input and output switch terminals are coupled to a respective plurality of backplane connections, and

at least one fabric card, each fabric card comprising: a fabric card cross-point switch having a plurality of input switch terminals and a plurality of output switch terminals, wherein the plurality of input and output switch terminals are coupled to a respective plurality of backplane connections,

wherein the data signal path comprises an input switch terminal, one or more cross-point switches from one or more video routers, and an output switch terminal, and

wherein, each line card and fabric card comprises a card controller, the card controller being coupled to one or more cross-point switches and configured to determine the path of one or more data signals based on the router configuration signals.

2. The system of claim 1 wherein a first path and a second path are determined based on a first priority and a second priority, the first path corresponding to a first data signal and the second path corresponding to a second data signal.

3. The system of claim 2 wherein the first priority and the second priority are equal.

4. The system of claim 2 wherein the first priority is higher than the second priority and the first path replaces at least a portion of the second path.

5. The system of claim 2 wherein the first priority is higher than the second priority.

6. The system of claim 3 wherein the first path is determined based on an availability of backplane connections.

7. The system of claim 3 wherein the card controller determines a selected routing from at least two proposed routings, each proposed routing comprising a first proposed route and a second proposed route, the first proposed route corresponding to the first data signal and the second proposed route corresponding to the second data signal.

8. The system of claim 7 wherein the determining the selected routing from at least two proposed routings further comprises determining a selected routing based on a priority score.

9. The system of claim 7 wherein the determining the selected routing from at least two proposed routings further comprises determining a selected routing based on a historical measurement of a property of the first path.

10. The system of claim 9 wherein the historical measurement of a property of the first path comprises a previously measured frequency of transmission failures of the first path.

11. A method of priority based routing of video signals from a plurality of input ports to a plurality of output ports using at least one video router of the data transmission system of claim 1, the method comprising:

receiving a data stream at an input port of a card, the card being a line card;

receiving one or more router configuration signals by one or more card controllers, at least one card controller being a line card controller of the line card; and

configuring one or more cross-point switches by card controllers to determine the path of one or more data signals based on the one or more router configuration signals.

12. The method of claim 11 further comprising determining a first path and a second path based on a first priority and a second priority, the first path corresponding to a first data signal and the second path corresponding to a second data signal.

13. The method of claim 12 wherein the first priority and the second priority are equal.

14. The method of claim 12 wherein the first priority is higher than the second priority and the first path replaces at least a portion of the second path.

15. The method of claim 12 wherein the first priority is higher than the second priority.

16. The method of claim 12 wherein the determination of the first path is based on an availability of backplane connections.

17. The method of claim 12 wherein the determination by the card controller of a selected routing from at least two proposed routings, each proposed routing comprising a first proposed route and a second proposed route, the first proposed route corresponding to the first data signal and the second proposed route corresponding to the second data signal.

18. The method of claim 17 wherein the determining the selected routing from at least two proposed routings further comprises determining a selected routing based on a priority score, the priority score determined from.

19. The method of claim 17 wherein the determining the selected routing from at least two proposed routings further comprises determining a selected routing based on a historical measurement of a property of the first path.

20. The method of claim 19 wherein the historical measurement of a property of the first path comprises a previously measured frequency of transmission failures of the first path.

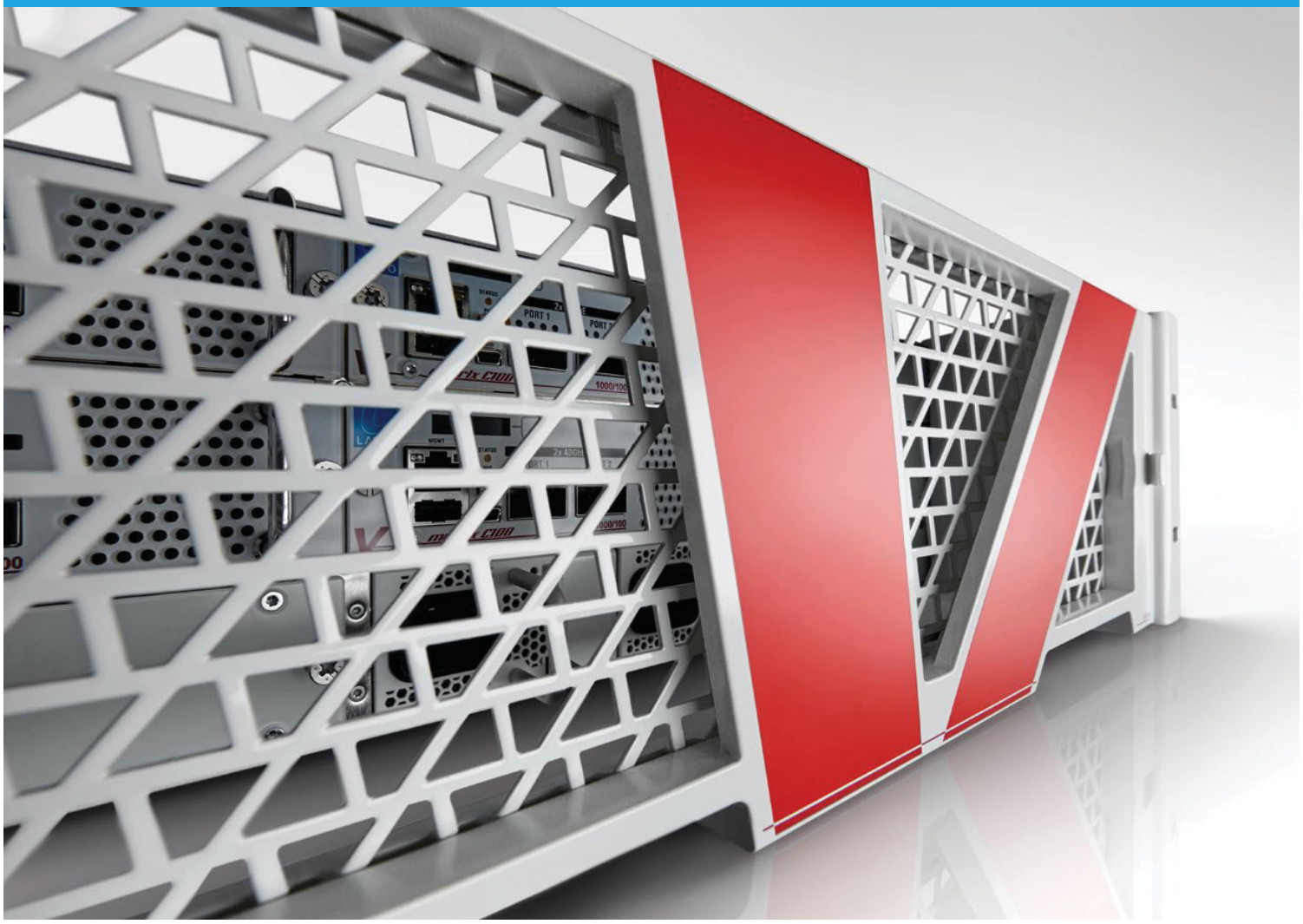
* * * * *

EXHIBIT H

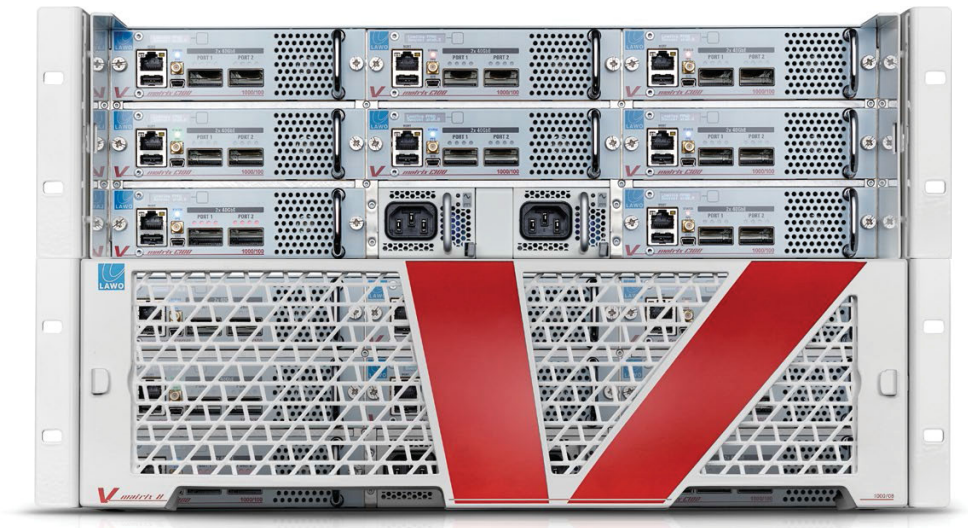
V__matrix

LAWO

SOFTWARE-DEFINED IP CORE ROUTING, PROCESSING & MULTI-VIEWING PLATFORM



FLEXIBLE
FORCEFUL
FUTURE-PROOF



V__matrix

SOFTWARE-DEFINED IP CORE ROUTING, PROCESSING & MULTI-VIEWING PLATFORM

Some call it cloud, some call it virtualization – we call it: V__matrix. This new IP broadcast video core infrastructure product will change your idea of what a broadcast facility looks like from legacy to future, quickly transforming any broadcast installation into a flexible, future-proof production facility, addressing a wide range of workflows and supporting your transition to a totally IP-based environment. V__matrix is the first of its kind. Free from the restrictions of legacy hardware platforms it offers a completely virtualized real-time routing and processing infrastructure. Instead of connecting single purpose modular products in elaborate production chains the V__matrix is based on the data center principles of flexibility, fabric computing and COTS economics and makes these available to any live production broadcast environment. Whether an OB truck, a TV studio or a broadcast operation center, V__matrix creates a fully virtualized facility infrastructure.

The V__matrix ecosystem is based on generic high capacity FPGA-based processing blades upon which Virtual Modules (VM) are loaded to create the functionality required. Multiple cores are connected through redundant 40GE (or 4x 10GE) Ethernet interfaces to an IP network to form a distributed IP routing and processing matrix that provides frame-accurate, clean switching just like a legacy baseband matrix.

The V__matrix ecosystem scales linearly from tens to thousands of I/O and processing functions which make it ideal for any size live broadcast facility, small or large. Capabilities easily scale as well. An entire production workflow can be remapped in minutes when requirements change from production to production. The functionality of any processing blade can be exchanged, enabling system capabilities to easily be modified or upgraded to address your constantly changing business requirements. The V__matrix pool of generic processing blades provides ultimate flexibility; with software-defined functionality they can be configured and called upon to handle the peaks and troughs of seasonal production demands and with Lawo's innovative licensing model, Virtual Modules can be assigned to a particular processing blade or be stored in an on-site license server allowing for unprecedented flexibility.

PROUDLY SUPPORTING

AIMS Alliance for IP Media Solutions · RAVENNA · EMBER+

V__matrix

SYSTEM OVERVIEW

OUR DEFINITION OF FUTURE-PROOF: SOFTWARE-DEFINED HARDWARE

The V__matrix ecosystem can be divided into two parts: the physical and the virtual. The physical consists of the C100 processing blade and associated hardware which provides the compute and processing capacity of the platform. Simply put: the more compute power you have the more functions you can run.

The virtual world is centered around the software which defines the functionality of the platform. The software packages are called Virtual Modules (VM) and in the V__matrix they allow the function-agnostic core processing hardware to build complex workflows by simply running the appropriate VM. I.e. typical broadcast processing functionality normally only found in dedicated hardware is instead defined by the various VMs.

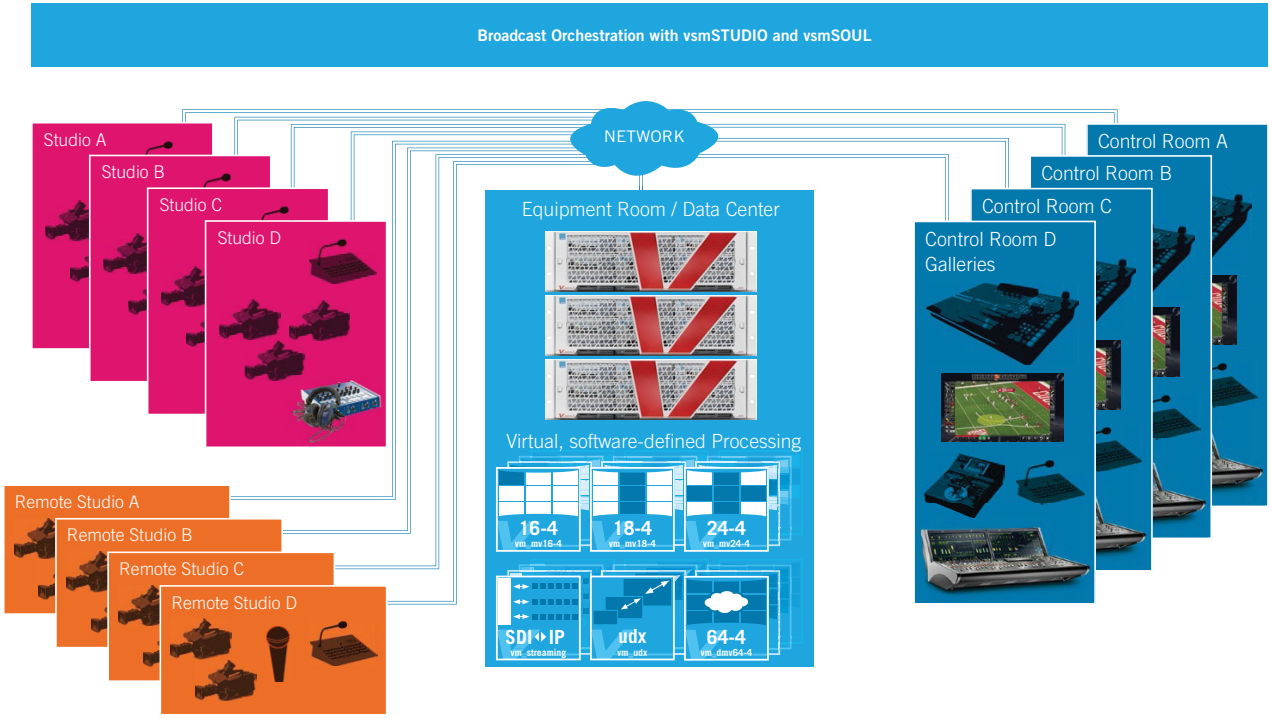
Therefore, the abilities of the system and the functionality of the signal chains are no longer defined by the hardware and its physical interconnectivity, but rather by the VMs and the way the control system connects multiple VMs together over a COTS IP network to build workflows and processing chains. And as all functions can easily be changed and all VMs are connected to each other over the IP network, these workflows and processing chains can easily be changed on-the-fly during runtime as demands change.

Since V__matrix is a fully IP-based platform, the C100 processing blade can be placed anywhere there is an IP network. It can be decentralized and spread over one or more facilities or centralized in a core facility or OB truck. A hybrid approach is also possible where some core equipment is kept on-site while a pool of processing power is kept in a remote data center. This decentralized approach allows for example the technical operation center to be situated in a purpose-built data center outside of town where space, power and cooling is inexpensive, while talent and studios can be in another area. V__matrix is not only game-changing in increasing the flexibility of broadcast installations in OB trucks, fly-away kits or studios – it delivers a truly future-proof core infrastructure based on open standards with significantly less rack-space, less power consumption and fewer restrictions for broadcast operations.

In a nutshell: V__matrix opens the door to an entire format and function agnostic infrastructure that finally provides broadcasters the flexibility and capability to cope with the challenges of the future.

V__matrix KEY FEATURES

- World's 1st virtualized broadcast ecosystem with software defined functionality based on high-capacity generic compute modules
- Unified routing, processing & multi-viewing core infrastructure
- Support for source-timed and destination-timed deterministic frame-accurate clean switching
- Distributed processing over multiple networked processing blades
- Seamless SDI to IP migration utilizing one unified control system (VSM)
- High-density IP conversion for legacy SDI equipment (up to 160 I/O in 3RU)
- Optimum utilization of resources: Significantly reduced footprint, power consumption, spare pooling and cabling
- Sophisticated multi-layer redundancy for signal, network, control and hardware layers
- Designed for both decentralized operation and data center style centralized operation
- Fully based on open standards in-line with the AIMS roadmap: ST2110-10/20/30/40, ST2022-6/-7, ST2042 (VC-2), VSF TR-01, AES67 and Ember+



V__matrix

SYSTEM OVERVIEW

LIMITED COMPONENTS, LIMITLESS POSSIBILITIES

V__matrix FRAMES

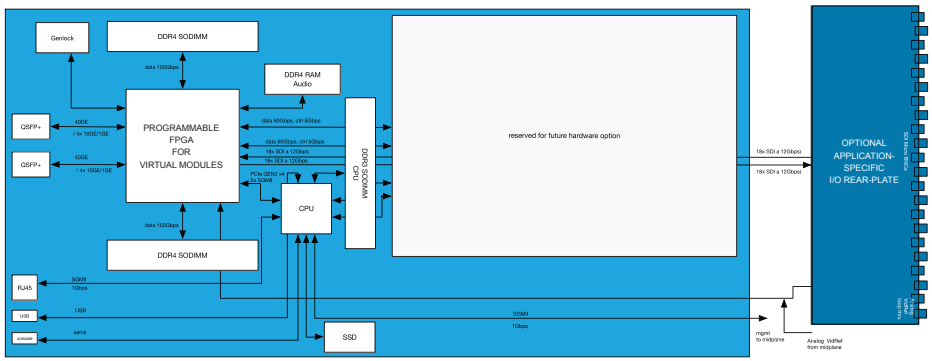
The V__matrix frames provide power and protected housing for the V__matrix processing blades. Each frame has a dedicated 1GE management port that provides connectivity for control and monitoring to all installed processing modules of the frame. Although the V__matrix ecosystem is designed for IEEE1588 / PTP synchronization, the frame also has a central video reference input (blackburst or tri-level) that optionally distributes sync to each card slot if required.

V__matrix frames are available in 1RU, 2RU and 3RU versions with slots for 2, 5 or 8 C100 processing blades respectively.

V__matrix C100 PROCESSING BLADE

The C100 processing blade forms the powerful epicenter of the V__matrix ecosystem and is the generic compute core upon which all the various virtual modules are loaded. Each module has dual front serviceable 40GE QSFP+ ports for connectivity to redundant IP core switches or for a maximum throughput of 80 Gbps. In addition, each C100 module has a dedicated 1GE management port, a USB port for saving and loading configurations and a mini-USB serial console port. The C100 processing blade slots in from the front of the V__matrix frame into the midplane and optionally into a rear mounted IO interface module. In combination with Lawo's VSM control and monitoring system, multiple C100 processing blades form a fully scalable, large, distributed routing and processing ecosystem with software-defined functions, workflows and signal chains.

C100 – Block Diagram



V__matrix REAR-PLATE I/O MODULES

The V__matrix rear-plate I/O modules provide additional interfaces on the back of the processing blade. These rear-plates house a variety of application specific physical interface connectors in order to provide connectivity to legacy broadcast equipment such as baseband video and audio components. All IP native processing functions are automatically handled from the dual front mounted 40GE QSFP+ ports. The design of the V__matrix platform allows the processing blade to be replaced from the front without touching any of the physical connectors on the back, greatly simplifying maintenance.

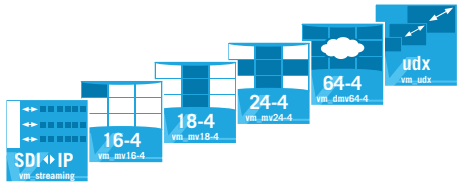
Available V__matrix rear-plates:
io_bnc_10+10: Provides 10 SDI inputs and 10 SDI outputs plus analog REF with loop-thru via micro-BNC connectors.

io_bnc_2+18: Provides 2 SDI inputs and 18 SDI outputs plus analog REF with loop-thru via micro-BNC connectors.

io_bnc_18+2: Provides 18 SDI inputs and 2 SDI outputs plus analog REF with loop-thru via micro-BNC connectors.

io_bnc_2+2+16: Provides 2 SDI inputs, 2 SDI outputs and 16 bidirectional inputs/outputs (switchable via software) plus analog REF with loop-thru via micro-BNC connectors.

All inputs and outputs are capable of 3G, HD and SD video, half of the inputs and outputs are also capable of 4K (6G and 12G) video. C100 processing blade and rear-plate IO modules are hot swappable.



V__matrix_vm VIRTUAL MODULES

The V__matrix Virtual Modules are the brains and intelligence of the V__matrix system allowing the user to build elaborate signal chains fulfilling all processing requirements in a fully virtualized environment. As all V__matrix core functionality resides in the software-based Virtual Modules, the platform is prepared from the start for the development of many more processing functions.

VSM IP BROADCAST CONTROL SYSTEM

Lawo's Virtual Studio Manager (VSM) Control System forms the orchestration and control layer of the V__matrix platform. Recognized as the most open and powerful broadcast control system on the market today, VSM enables operators to switch and route signal flows in both the IP and SDI domain. With support for a wide range of 3rd party equipment, VSM is the perfect control system to integrate a V__matrix platform to any legacy broadcast environment. Its highly intuitive and customizable user interface allows operators to continue working in a familiar environment while under the surface, production capabilities can be gradually migrated to an IP infrastructure at a pace that makes sense both logistically and economically.

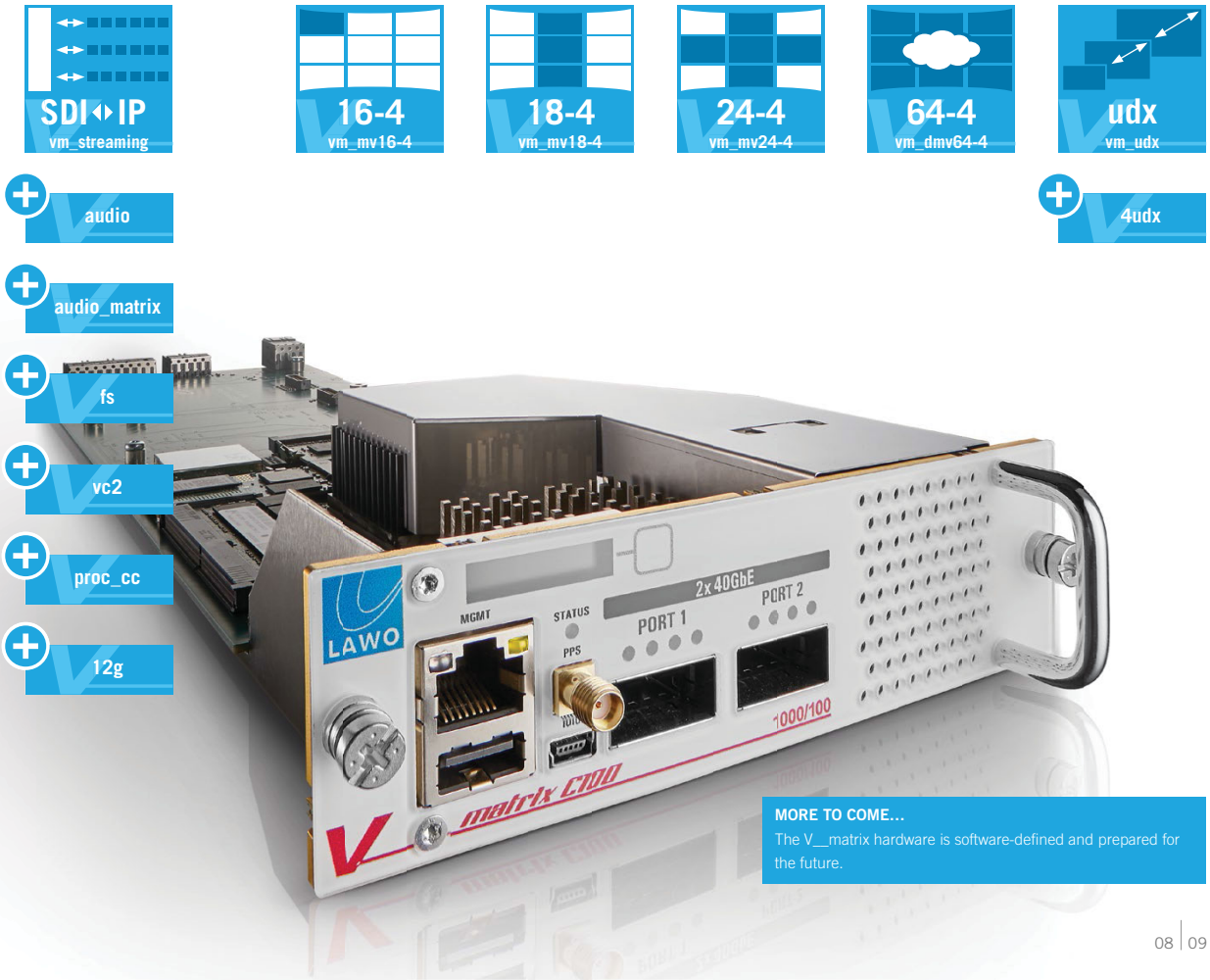
Virtual Modules

FLEXIBLE SIGNAL CHAINS – FOR TODAY, TOMORROW & BEYOND

If the processing blades are the muscle, the Virtual Modules are the brain of the V__matrix platform. Combining VMs together allows the creation of complete production chains fulfilling all broadcast requirements in a fully virtualized environment. As all functions of the V__matrix ecosystem are software-defined it is the ultimate future-proof platform. Changing the functionality of your broadcast plant is as easy as changing the software modules loaded onto the processing blades. By cascading multiple VMs together, the V__matrix scales linearly up to thousands of SDI I/O and audio/video processing functions for unparalleled scalability, flexibility, versatility and cost-efficiency.

Virtual Modules can be licensed to specific V__matrix processing blades or managed as floating licenses within an installation via a dedicated license server thus providing total flexibility for the broadcast infrastructure by accessing all the various functions as pooled resources controlled by VSM.

The current line-up of V__matrix Virtual Modules include:



MORE TO COME...
The V__matrix hardware is software-defined and prepared for the future.

4K/HDR Streaming & Processing

vm_streaming – SDI-to-IP Gateway and more

The V__matrix vm_streaming virtual module (VM) is a versatile audio and video processing module that provides routing, processing and glue functionality for the V__matrix eco-system when loaded on a c100 processing blade.

As base functionality vm_streaming provides encapsulation and de-encapsulation of 3G, HD and SD-SDI to IP SMPTE2022-6 and SMPTE2110-20/30 and provides SMPTE2022-7 hitless protection switching. In this configuration, the vm_streaming VM is the ultimate high-density gateway between legacy SDI infrastructure and an IP network with a capability of up to 160 conversions in 3RU.

Designed as a complete IP infrastructure solution vm_streaming offers fully independent IP interfaces with redundant or discrete operation modes, frame phasing of received video streams in IP, timed playout of received streams to selectable reference, IP to IP format stream conversion, clean video switching in IP using destination-timed switching (MBB & BBM) as well as source-timed switching and virtual video, audio and a/v crossbars with shuffling.

When combined with the +AUDIO option vm_streaming also supports full audio embedding/de-embedding and routing in a 512x512 mono router crosspoint as well as pooled audio sample rate conversion, audio leveling and cross-fade support. If you need even more the +AUDIO_MATRIX option increases the amount of audio transceivers to 128TX and 88RX and the audio crosspoint to 5,300 by 5,300!

The +FS option enables frame syncing functionality and delay for video as well as sample-rate conversion and delay for audio. +FS provides a pool of 8 video and 50 audio instances that can be used for both IP and SDI sources and destinations. In addition, +FS enables an enhanced receive buffer of up to 150ms for 2022-7 protection over WAN. The +PROC_CC option enables a pool of 8 RGB/YUV color correction engines and the ability to insert timecode, AFD metadata etc.

Using only the IP interfaces of the c100, an effective way of providing framesync, color correction and ANC data processing is to load a number of c100 blades with vm_streaming and either or both of these options thus making them available as pooled resources routable from the IP network. All while keeping the signals in IP without having to go back to baseband.

With the +VC2 option the vm_streaming virtual module gets 20 x VC-2/DiracPro ultra low latency video codecs @3G. This SMPTE standardized and open codec originally developed by BBC provides visually lossless compression at ratios between 2.5-4.4:1 with less than 20 lines of end-to-end delay. This makes it a great option for 4K or in remote production applications where bandwidth is limited but minimal delay is crucial.

KEY FEATURES

- High density IP/SDI gateway with up to 160 conversions in 3RU
- 4K/HDR 12G-SDI encap/decap to SMPTE2110
- Up to 5,300x5,300 audio mono matrix with full audio embedding/de-embedding and shuffling between SDI/IP, IP/IP and SDI/SDI with ST2110-30/RAVENNA/AES67.
- Provides common glue and processing functionality such as framesync, RGB/YUV color correction, timecode insertion and test signal generation
- 20 instances of VC-2/DiracPro ultra low latency codec enables high quality ultra-low latency compression
- Designed for WAN environments with hitless merge protection and large receive buffers
- Built-in programming, configuration and streaming telemetry capabilities
- Simple management and control through Lawo's VSM makes operation imperceptible from a traditional baseband environment while maintaining all of the benefits of an IP system

OPTIONS

audio Option to vm_streaming that adds embedding/de-embedding and shuffling of audio from both IP and baseband I/O with sample rate conversion. Provides 40 TX and 40 RX instances of RAVENNA/AES67/SMPTE2110-30 streaming and an audio crossbar of 512 x 512. Level adjustments, x/y-fades on all outputs.

vc2 Option to vm_streaming that adds visually lossless VC2/DiracPro ultra-low latency encoding and decoding (each 20 pooled instances @ 3G-SDI; a 12G 4K signal uses 4 instances). ST2042 low-delay profile with RAW headers. Encoding ratio typ. 2.5:1 – 4.4:1. Latency < 20 lines

audio_matrix In addition to the specifications of the +audio option, the +audio_matrix option provides a total of 88RX and 128TX instances of RAVENNA/AES67/SMPTE2110-30 streaming and an increased audio crossbar of 5,312 x 5,312. Requires +audio option.

proc_cc Option for vm_streaming that adds YUV & RGB color correction, timecode generator/inserter, test pattern generator/inserter, test tone generator as well as metadata pass-through selection and alteration. 8 pooled @ 3G-SDI (a 12G 4K signal uses 4 instances).

fs Option to vm_streaming that adds framesync, sample rate conversion and audio/video delay functionality (up to 10 frames/ 400 ms) for both IP and baseband video inputs. 20 pooled instances @ 3G-SDI (a 12G 4K signal uses 4 instances). Also provides A/V Sync Test Pattern generation and measurement functionality.

12g Option for vm_streaming that adds support for 12 Gbps video standards (2160p 23.976;24;25;29.97;50;59.94;60Hz SMPTE ST-2081,2082). Also provides cross conversion capabilities between UHDTV1 Single-Link and Quad-Link (2SI and SQD).

V__matrix

VIRTUAL MODULES: SIGNAL CONVERSION

4K Format Converter

vm_udx – IP Up/Down/Cross Converter

The V__matrix vm_udx virtual module (VM) is a software module capable of format conversion between SD, HD and 4K formats in the V__matrix eco-system when loaded on a c100 processing blade. In addition to up-, down-, and cross-conversion the vm_udx also provides audio embedding/de-embedding, frame-sync, proc-amp and RGB color correction functionality.

At its core the vm_udx provides four independent paths of format conversion between SD, HD and 4K for IP and/or SDI signals. Conversion between SD and HD formats use one path while conversion to/from 4K uses four paths.

Each path provides full video framesync and audio sample rate conversion as well as audio and video delay functionality. Every path also has audio embedding/de-embedding capabilities with audio gain, surround to stereo downmix (5.1 & 7.1) and x/y-fade functionality for all outputs.

A 5,312x5,312 mono audio router provides ample room for audio embedding/de-embedding to video as well as pure audio channel reshuffling if desired. Broadcast quality RGB and YUV color correction is provided for every processing path.

An additional four independent paths for a total of eight are available with the +4UDX option that can be added at any time.

Fundamentally designed with IP networking in mind vm:udx natively supports both ST2022-6 and ST2110-20 IP video as well as ST2110-30/AES67 and RAVENNA IP audio streams. Conversion between IP video and IP audio standards is also possible, e.g. ST2022 to ST2110. To ensure high availability ST2022-7 seamless protection switching is natively supported.

With the available io_bnc rear-plates vm_udx allows for legacy connection to SD-, HD- and 4K-SDI. For 4K both single-link 12G-SDI as well as quad-link (2SI and SQD) is supported. The vm_udx is also capable of single-link to quad-link and 2SI to SQD cross-conversion.

4udx

Option for additional four paths of up/down/cross conversion for vm_udx, resulting in a total of eight paths.

KEY FEATURES

- 4 instances of Up/Down/Cross conversion between SD/HD/4K (1 instance when converting to/from 4K)
- Additional 4 instances available with the +4UDX option
- 5,312 x 5,312 audio mono matrix with full audio embedding/de-embedding and shuffling between SDI/IP, IP/IP and SDI/SDI with SMPTE2110-30/RAVENNA/AES67
- Includes framesync, RGB/YUV color correction and timecode insertion
- 4K 12G-SDI single-link inputs/outputs when combined with the io_bnc rear-plates
- Built-in programming, configuration and streaming telemetry capabilities
- Simple management and control through Lawo's VSM makes operation imperceptible from a traditional baseband environment while maintaining all of the benefits of an IP system

4K/HDR Multiviewer

vm_mv16-4, vm_mv18-4 & vm_mv24-4

The V__matrix vm_mv24-4, vm_mv18-4 and vm_mv16-4 virtual modules (VMs) provide high quality multiviewer functionality to the V__matrix ecosystem when loaded onto a C100 processing blade.

Capable of handling both IP and SDI sources these VMs can monitor uncompressed 4K, 3G, HD and SD video as well as both embedded and discrete audio while generating pixel perfect mosaics with ultra-low latency.

The vm_mv24-4 can monitor up to 24 simultaneous sources from a combination of IP or SDI video while the vm_mv18-4 and vm_mv16-4 can monitor 18 and 16 sources respectively. All three multiviewer VMs can generate up to 4 x 3G output heads (2 @ 4K) and scale and display any source on any output head without limitation. The same source can be displayed on multiple output heads at different resolutions. The output heads can be configured as either HD, 3G or 4K and output over IP as ST2110/2022 or, using the modular I/O rear-plate, as SDI.

KEY FEATURES

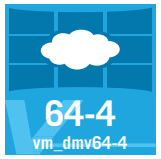
- Broadcast quality low-latency IP and SDI multiviewer VMs for the V__matrix C100 processing blade
- Supports HDR and standard 4K, 3G, HD and SD signal monitoring with 4K or 3G mosaic outputs
- Designed for monitoring of IP video and audio with support for SMPTE2022 and SMPTE2110
- Intuitive control through the touch-supported HTML5 GUI in “theWALL”
- Support for 4 mosaic outputs heads with any input to any output head without limitation
- Output head can be streamed as SMPTE2022 or SMPTE2110 IP or output over SDI using the optional C100 I/O rear-plate

theWALL – SMART DRAG&DROP MULTIVIEWER CONTROL

The V__matrix multiviewer VMs were designed to be controlled by Lawo’s groundbreaking touch operated configuration system “theWALL”. This unique HTML5 based GUI makes mosaic configuration with borders, colors, UMDs, tally etc a simple case of drag and drop.

Distributed 4K/HDR IP Multiviewer

vm_dmv64-4 – WORLD'S 1ST INFINITELY EXPANDABLE MULTIVIEWER



Lawo's vm_dmv64-4 is virtual module (VM) for Lawo's V__matrix IP routing & processing platform, complementing the existing vm_mv16-4, vm_mv18-4 and vm_mv24-4 multiviewer line-up. It turns the V__matrix C100 software-defined processing blade into a distributed, infinitely expandable, true IP multiviewer.

The vm_dmv64-4 is based on a distributed architecture where multiple modules network together. These modules could be hosted together in the same V__matrix frame, in different frames or even at different geographical locations. Basically anywhere as long as they are networked together via IP.

Every vm_dmv64-4 has an input stage capable of receiving up to 24 sources of any combination of 4K/3G/HD/SD which is limited only by the physical (up to 18 SDI inputs) or network (2x 40GbE) I/O. These sources are downscaled by the vm_mv64-4 and returned to the network as IP (RFC 4175) encapsulated mipmaps.

In parallel to the input stage, every vm_dmv64-4 also features an output stage capable of creating up to four 3G mosaics (or two 4K mosaics) with up to 64 sources/PIPs each (128 each in 4K). The output stage compiles a mosaic from the appropriate mipmaps needed, automatically taking into account the size the user requests for the PIPs. The output stage can both use mipmaps that it has generated by its own input stage or by subscribing to other mipmaps from the network.

As any vm_dmv64-4 can use any mipmaps from any other vm_dmv64-4 on the network it scales linearly with each vm_dmv64-4 that is added to the network which results in an "infinitely" expandable and distributed multiviewer.

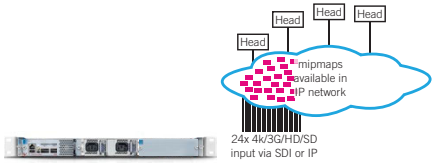
The result is not only the world's first 1st infinitely expandable multiviewer, it's also a solution that significantly reduces rack-space, weight and power-consumption. In addition, Lawo's V__matrix platform is already renowned for its software-defined functionality where C100 blades can be changed at run-time by loading different virtual modules.

KEY FEATURES

- Virtual Module (VM) for V__matrix C100 core processing blade
- Distributed multiviewer architecture with unlimited inputs and heads
- Full support of IP and SDI sources in 4K, 3G, HD and SD
- Support of embedded and discrete audio
- Pixel perfect mosaics with ultra-low latency
- Intuitive drag&drop mosaic configuration with Lawo's "theWALL"

INFINITELY EXPANDABLE

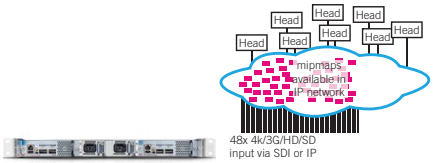
If you need to add more heads or more sources, just add another C100 with a vm_dmv64-4 virtual module installed…



24-4

1RU
1x V__matrix C100 processing blade

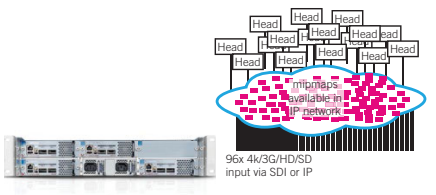
Input Stage: up to 24 x4k/3G/HD/SD signals
Mipmaps created and available: 24
Output Stage: 4x heads with max. 64 sources/PIPs each



48-8

1RU
2x V__matrix C100 processing blades
(in same frame, different frames, or at geographically distant locations)

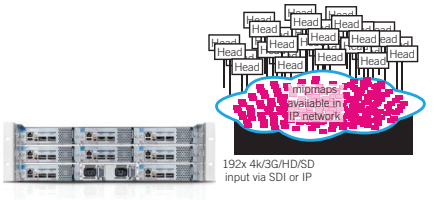
Input Stage: up to 48 x4k/3G/HD/SD signals
Mipmaps created and available: 48
Output Stage: 8x heads with max. 64 sources/PIPs each



96-16

2RU
4x V__matrix C100 processing blades
(in same frame, different frames, or at geographically distant locations)

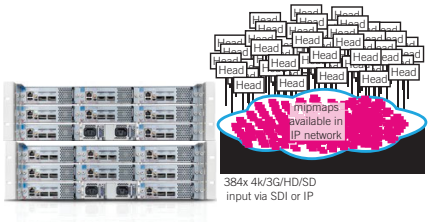
Input Stage: up to 96 x4k/3G/HD/SD signals
Mipmaps created and available: 96
Output Stage: 16x heads with max. 64 sources/PIPs each



192-32

3RU
8x V__matrix C100 processing blades
(in same frame, different frames, or at geographically distant locations)

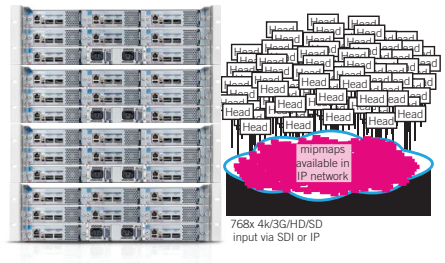
Input Stage: up to 192x 4k/3G/HD/SD signals
Mipmaps created and available: 192
Output Stage: 32x heads with max. 64 sources/PIPs each



384-64

6RU
16x V__matrix C100 processing blades
(in same frame, different frames, or at geographically distant locations)

Input Stage: up to 384x 4k/3G/HD/SD signals
Mipmaps created and available: 384
Output Stage: 64x heads with max. 64 sources/PIPs each



768-128

12RU
32x V__matrix C100 processing blades
(in same frame, different frames, or at geographically distant locations)

Input Stage: up to 768x 4k/3G/HD/SD signals
Mipmaps created and available: 768
Output Stage: 128x heads with max. 64 sources/PIPs each

HARDWARE OVERVIEW

Building a V__matrix solution is incredibly simple. Just choose the number of processing modules and associated Virtual Modules needed to meet your requirements and then populate them into the frame-size that works best for your application, be it a stagebox, an OB truck, a studio or a broadcast operations data center.

V__matrix 2

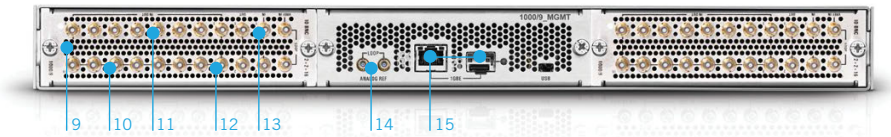
V__matrix 5

V__matrix 8

V__matrix 2 – FRONT VIEW (with front cover)

V__matrix 2 – FRONT VIEW (without front cover)

V__matrix 2 – REAR VIEW



1 Secure lock
2 Integrated cable duct
3 1 Gigabit Ethernet (control & monitoring)
4 USB Port (save & load configuration)
5 Mini-USB as serial console port
6 2x 40 Gigabit Ethernet (QSFP+)
7 Fan
8 Redundant power supplies (hot-swappable)
9 Video reference input & loop-thru (blackburst or tri-level)
10* 5x SDI inputs (12G/6G/3G/HD/SD)
11* 5x SDI inputs (3G/HD/SD)
12* 5x SDI outputs (12G/6G/3G/HD/SD)
13* 5x SDI outputs (3G/HD/SD)
14 Video reference input & loop-thru; blackburst or tri-level distributed via backplane to all I/O modules in a frame
15 1x 1 Gigabit Ethernet available through RJ45 or SFP+ for control and monitoring (allows centralized access to all core processing blades in a frame)

* V__matrix_io10+10. Actual input/output configuration depends on type of I/O card.

V__matrix

CONTROL, ORCHESTRATION AND MONITORING

VSM and vsmSOUL

UNIFIED CONTROL AND ORCHESTRATION OF V__matrix, IP AND LEGACY SDI

Mobile Production | Studio Production | Remote Production | MCR

vsmSTUDIO & vsmSOUL – UNIFIED ORCHESTRATION, CONTROL AND MONITORING SYSTEM

IP Edge Devices	Cloud Processing	IP Routing (vsmSOUL)	Baseband Migration	Audio Production
– Video-to-IP interfaces – Audio-to-IP interfaces – Based on open standards	– Software-defined workflow – High-density video processing – Real-time data center ready	– Designed for multi-vendor COTS IP switch operation. – Support for Patching, Destination and Source-timed switching – Compatible with NMOS 1.0 (and higher) - Supports SMPTE 2110, 2022-6, 2022-7, AES67, RAVENNA	– Unified system wide control – Industry leading support for 3rd party devices	– IP-native audio network & control – Cloud-based audio processing – AES67 / RAVENNA compliant

VSM is the ideal orchestration system for broadcasters with legacy baseband infrastructures that are considering expanding into SDN and IP with a hybrid IP/SDI plant. With support for more protocols and devices than any other control system on the market, VSM makes transitioning from SDI to IP a seamless, step by step process at your own pace without disruption for the operators.

For control of the V__matrix, Lawo's Seamless Orchestration and Unification Layer (vsmSOUL) is the ideal orchestration manager and deeply integrated with the V__matrix. vsmSOUL manages the routing of audio and video streams across any vendor IP network and is compatible across individual interfaces and technical solutions from 3rd party sources.

HITLESS MERGE

A network with vsmSOUL guarantees Hitless Merge (SMPTE 2022-7). This requires that a signal is packaged in two different streams and travels two separated networks, with vsmSOUL acknowledging both branches and stream addresses. Operationally, it appears that a single crosspoint is presented, but with two alarms, two sources and two multicast addresses.

ROUTING STATE RECOVERY

In case of a system failure or reboot, vsmSOUL can recall the network's routing status. After a reboot, inconsistencies in stream flows are indicated and can be corrected.

SOPHISTICATED REDUNDANCY

Both vsmStudio and vsmSOUL support active-active redundancy, meaning that two systems run in parallel with the secondary system actively monitoring all system status live. The secondary system is always ready to seamlessly assume control.

POOLING SIMPLIFIES SIGNAL MANAGEMENT

VSM simplifies and automates operation by automatically inserting a free 'pooled' device dynamically (such as a frame synchronizer) and automatically setting the device so that the signal arrives at the target in the correct format. These 'pooled' devices can include any physical 3rd party device as well as virtual devices and functions of the V__matrix. As broadcast operations are mission critical, if one of the currently used pooled devices should fail, VSM will automatically re-route the signal to another spare device without user intervention.

"BOXING" MAKES TRUCKS OR STUDIOS HANDY AS BOXES

As resources become centralized, system capabilities dramatically increase in size, thus becoming difficult to manage. Virtualizing temporary setups of both physical and virtual resources in V__matrix, which can then be recalled to any studio environment in a preset, simplifies resource management even for the largest of systems. "Boxing" the resources into virtual environments means moving complete productions from one studio to another is as simple as one button push – workflow optimization at its best! Any available studio now becomes a backup for a production even if equipment is different – the ultimate disaster/recovery solution.

KEY FEATURES OF vsmSTUDIO AND vsmSOUL

- Perfect integration with Lawo V__ units
- A single control interface for numerous devices – learn one not many
- Dynamic router and IP tie line management that includes transparent Tally logic
- "Boxing" complete studios allows fast switching to emergency backup studios on the fly
- Automatic resource management with administration and user rights
- Combine hardware and software control interfaces for simplified control
- Virtual signal paths provide unbeatable speed and flexibility for a constantly changing environment
- Bundle different signal types together logically to route multiple signals from a device at the press of a button
- Simple and flexible control panel design to adapt to or optimize existing production workflows
- Strong redundancy architecture designed for 24/7 non-stop operation
- Sophisticated SNMP and alarm management to avoid problems before they become critical
- Northbound abstraction of the network through standard router protocols
- Switch-API support southbound, with access to multicast routing
- Agnostic to various switching mechanisms. Supported switching modes: Patching, Make-before-break, Break-before-make...
- Supports ST2110, ST2022-6, ST2022-7, AES67, RAVENNA
- Intuitive GUI for fast configuration

V__matrix

SPECIFICATIONS

V__MATRIX FRAMES

CENTRALIZED VIDEO REFERENCE INPUT
- 1x Analog genlock high definition trilevel sync SMPTE-274M/296M or Analog Genlock SD 1V BB SMPTE-170M/318M or SDI, 1x reference loop-back

CENTRALIZED MANAGEMENT PORTS
- 1x RJ45 100/1000Base-T, 1x SFP slot

FRAME MECHANICS
- DIMENSIONS: (H x W x D): 44/88/132 mm, (1/2/3 RU) x 481 mm (19“) x 458 mm (18“)
- WEIGHT: V__matrix2: 5 kg (11 lb), V__matrix5: 7 kg (15.5 lb), V__matrix8: 8 kg (18 lb)
- INDICATORS: 2x power status per PSU
- POWER: Connector: 2x IEC redundant, Input Voltage: nominal 100-240 V, AC +/- 10%, 50/60 Hz, Hot swappable: Yes
- MAX. CONSUMPTION: V__matrix2 < 400 W, V__matrix5 < 1,000 W, V__matrix8 < 1,000 W

C100 PROCESSING BLADE

INTERFACES
- 2x QSFP+: Each configurable as 40GE Ethernet or 4x 10GE Ethernet, 1x RJ45 100/1000Base-T Dedicated management port, 1x USB Console port, 1x PPS pulse per second output

VIDEO REFERENCE
- IEEE1588 PTPv2 / SDI / Analog Video Ref (Tri-Level, BB) / IP Vid-stream

PROCESSING
- Max. 36 Gbps of processing in SPS mode or 54 Gbps in discrete mode

MANAGEMENT AND MONITORING
- PROTOCOLS: HTTP, SNMPv1, Ember+, Syslog, User interface: Embedded HTML5 user interface, Management interface: Out-of-band and in-band management with guaranteed min bandwidth for inband management & control
- INDICATORS: 1x status LED, 4x status LED: per QSFP+ port, 1x OLED status display: monochrome display with touch point

ENVIRONMENTAL SPECIFICATIONS
- OPERATING TEMPERATURE: 0°C to +30°C (+32°F to +86°F)
- STORAGE TEMPERATURE: -20°C to +70°C (-4°F to +158°F)
- RELATIVE HUMIDITY: < 90% non-condensing,
- VENTILATION/AIR-FLOW: front-to-back cooling, prerequisite airflow > 26 qbm/h per C100 blade, max 320 qbm
- NOISE EMISSION: < 58 dBA per C100 processing blade,
- ELECTROMAGNETIC ENVIRONMENT: E2 (EN55103-1,-2)

IO REAR-PLATES

io_bnc_10+10 IO rear-plate
- 10x 12G*/6G*/3G/HD/SD-SDI inputs via micro BNC connector, 10x 12G*/6G*/3G/HD/SD-SDI outputs via micro BNC connector, 1x micro BNC analog video reference input with micro BNC loop-back

io_bnc_18+2 IO rear-plate
- 18x 12G*/6G*/3G/HD/SD-SDI inputs via micro BNC connector, 2x 12G*/6G*/3G/HD/SD-SDI outputs via micro BNC connector, 1x micro BNC analog video reference input with micro BNC loop-back

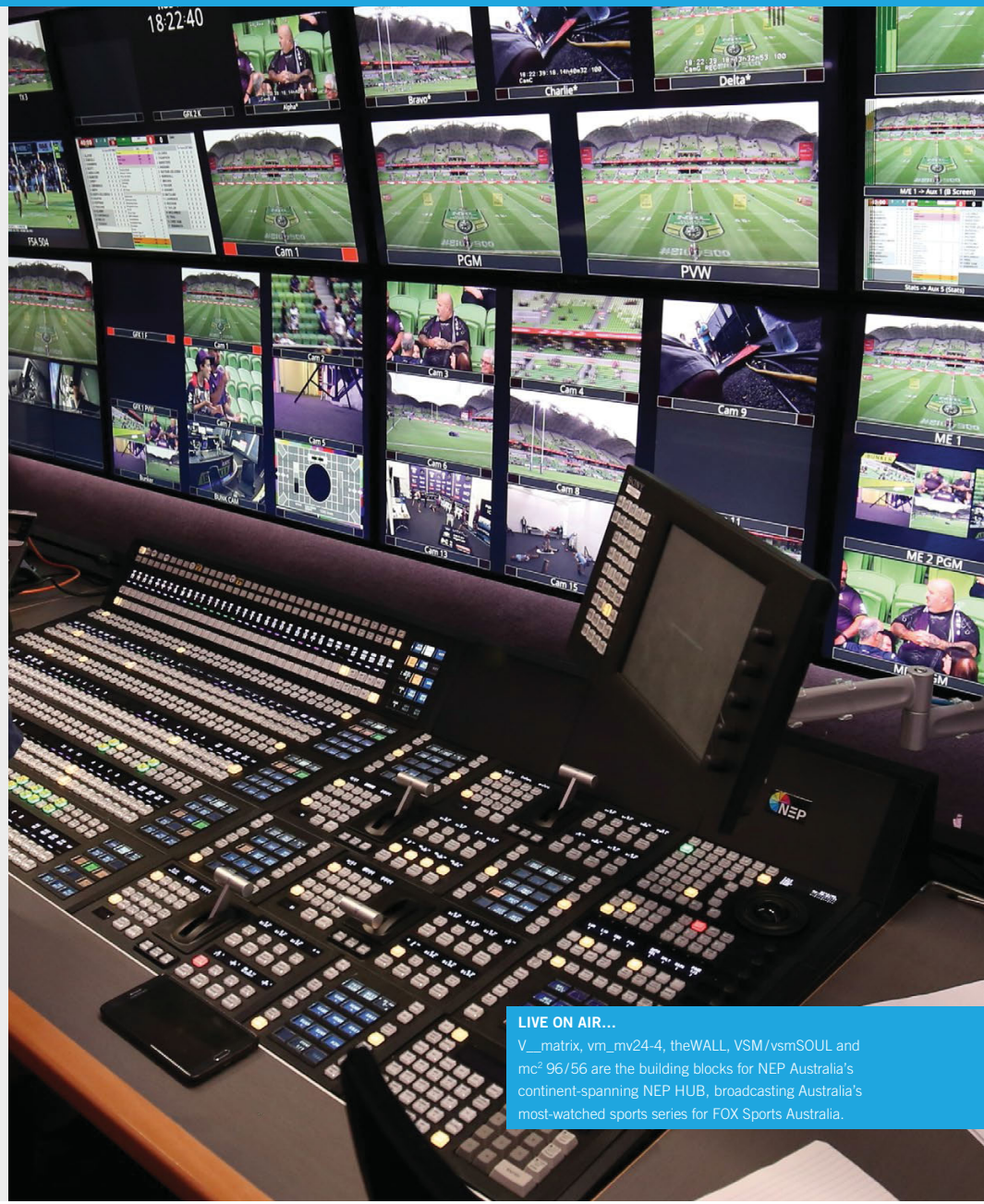
io_bnc_2+18 IO rear-plate
- 2x 12G*/6G*/3G/HD/SD-SDI inputs via micro BNC connector, 18x 12G*/6G*/3G/HD/SD-SDI outputs via micro BNC connector, 1x micro BNC analog video reference input with micro BNC loop-back

io_bnc_2+2+16 IO rear-plate
- 2x 12G*/6G*/3G/HD/SD-SDI inputs via micro BNC connector, 2x 12G*/6G*/3G/HD/SD-SDI outputs via micro BNC connector, 16x bidirectional 12G*/6G*/3G/HD/SD-SDI inputs/outputs via micro BNC connector (switchable via software), 1x micro BNC analog video reference input with micro BNC loop-back

RETURN LOSS
- SD: > 15 dB; HD: > 15 dB; 3G-HD: >15 dB 5 MHz-1.485 GHz, >10 dB 1.485 GHz-2.97 GHz

CABLE LENGTH:
- SD: >350 m (using Belden1694A), HD: >180 m (using Belden 1694A), 3G-HD: >120 m (using Belden1694A)

* 12G and 6G available on half of the connectors



LIVE ON AIR...
V__matrix, vm_mv24-4, theWALL, VSM/vsmSOUL and mc² 96/56 are the building blocks for NEP Australia's continent-spanning NEP HUB, broadcasting Australia's most-watched sports series for FOX Sports Australia.

V__matrix

SOFTWARE-DEFINED IP CORE ROUTING, PROCESSING & MULTI-VIEWING PLATFORM

HEADQUARTERS
Lawo AG
Rastatt
GERMANY
+ 49 7222 1002 0
sales@lawo.com

INTERNATIONAL OFFICES

BENELUX	+ 31 6 54 26 39 78
CANADA	+ 1 416 292 0078
CHINA	+ 86 10 6439 2518
NORWAY	+ 47 22 106110
SINGAPORE	+ 65 9818 3328
SWITZERLAND	+ 49 7222 1002 0
UK	+ 44 333 444 5296
USA	+ 1 888 810 4468

RENTAL SERVICE
+ 49 7222 1002 0
rental@lawo.com

www.lawo.com