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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 August 2008 (14.08.2008)

PCT

(10) International Publication Number
WO 2008/097731 A1

(51) International Patent Classification:
H04B 7/212 (2006.01)

(21) International Application Number:
PCT/US2008/051988

(22) International Filing Date: 25 January 2008 (25.01.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/701,690 2 February 2007 (02.02.2007) US

(71) Applicant *(for all designated States except US)*: **CISCO TECHNOLOGY, INC.** [US/US]; 170 West Tasman Drive, SJC/10/2/1, San Jose, CA 95134-1706 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SHARMA, Neil** [IN/US]; 311 Elmwood Ave, Apt. 2, Ithaca, NY 14850 (US). **LAWSON, Matthew Todd** [US/US]; 12621 Digger Pine Circle, Grass Valley, CA 95949 (US). **JACOBS, Mick R.** [US/US]; 11975 Norman Lane, Auburn, CA 95603 (US).

(74) Agents: **LAFFERTY, Wm. Brook** et al.; Scientific-Atlanta, Inc., Intellectual Property Dept., 5030 Sugarloaf Parkway, Lawrenceville, GA 30044 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(54) Title: MECHANISM FOR CHANNEL SYNCHRONIZATION

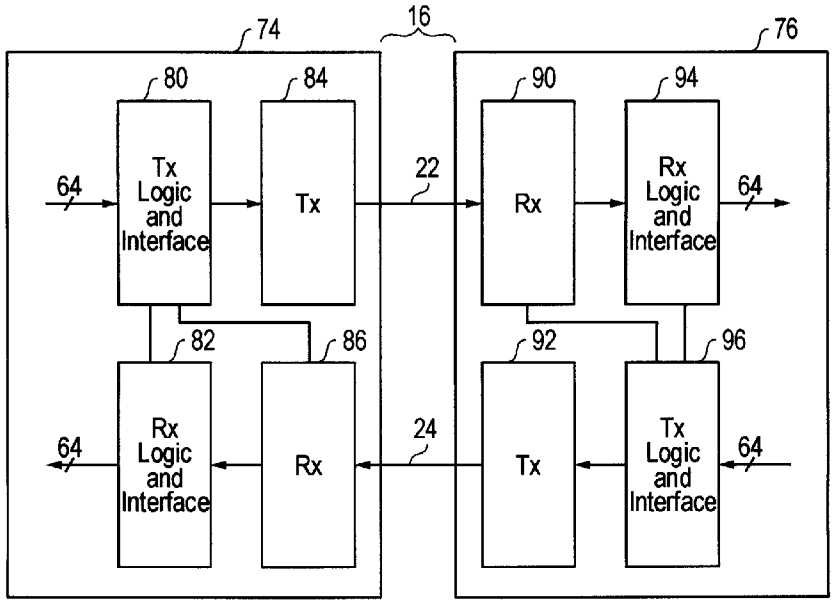


FIG. 8

(57) Abstract: A channel synchronization method in which local serializers serially transmits first n-bit codes, respectively, to remote deserializers, respectively. Also local deserializers serially receive first n-bit codes, respectively, from remote serializers, respectively. One of the first n-bit codes transmitted to one of the remote serializers indicates one of the local deserializers is not link aligned.

WO 2008/097731 A1

MECHANISM FOR CHANNEL SYNCHRONIZATION

Matthew Todd Lawson
Mick R. Jacobs
Neil Sharma

BACKGROUND OF THE INVENTION

[0001] Communication channels are employed in many types of communication systems. Communication channels transmit data between, for example, a line card and a switching fabric of a switch or a router.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] A better understanding of the present invention can be obtained when the following detailed description of the drawings is considered in conjunction with the following drawings.

[0003] Figure 1 illustrates relevant components of an exemplary communication system.

[0004] Figure 2 illustrates relevant components of one of the line cards and a switching fabric port of Figure 1.

[0005] Figure 3 illustrates relevant components of one of the receivers and one of the transmitters of Figure 2.

[0006] Figure 4 illustrates relevant components and operational aspects of one of the receivers and one of the transmitters of Figure 2.

[0007] Figure 5 illustrates relevant operational aspects of an encoder, a serializer, a deserializer and a decoder of Figure 3 or Figure 4.

[0008] Figure 6 illustrates relevant operational aspects of an encoder, a serializer, a deserializer and a decoder of Figure 3 or Figure 4.

[0009] Figure 7 illustrates relevant components of another exemplary communication system.

[0010] Figure 8 illustrates relevant components of one of the line cards and a switching fabric port of Figure 7.

[0011] Figure 9 illustrates relevant components of the receivers and the transmitters of Figure 8.

[0012] The use of the same reference symbols in different drawings indicates similar or identical items.

DETAILED DESCRIPTION

[0013] The following is intended to provide a detailed description of one or more examples of the invention and should not be taken to be limiting of the invention itself. Rather, any number of variations may fall within the scope of the invention which is defined in the claims following the description. In the following description, the present invention could be implemented as hardware although one of ordinary skill will readily recognize that the equivalent of such hardware may also be constructed in a computer program executing on one or more processors. If the invention is implemented as a computer program, the program may be stored in a conventional computer readable medium that may include, for example: magnetic storage media such as a magnetic disk (e.g., a floppy disk or a disk drive), or magnetic tape; optical storage media such as an optical disk, optical tape, or machine readable barcode; solid state electronic storage devices such as random access memory (RAM), or read-only memory (ROM); or any other device or medium employed to store computer program instructions.

[0014] Figure 1 illustrates a block diagram showing relevant components of an example communication system 10, which includes a switching fabric 12 coupled to multiple line cards via respective communication channels 16. The switching fabric 12 may include ports (not shown in Figure 1) which are coupled to respective communication channels. The line cards, in turn, may be coupled to one or more end-devices. The term end-device may be defined to include desk top computers, printers, servers, disk arrays, switches, routers, or other such components.

[0015] Communication system 10 enables transmission of data between the various end-devices coupled to line cards 14. While the present invention will be described with reference to transmitting data between end-devices via a switching fabric, the present invention should not be limited thereto. The present invention may find application in transmitting data directly between a first device (e.g., a disk array) and a second device (e.g., a server) via a communication channel.

[0016] Figure 2 illustrates relevant components of one of the line cards 14 of Figure 1. Figure 1 also shows relevant components of a switching fabric port 20 of switching fabric 12

also of Figure 1. Communication channel 16 includes unidirectional channel 22 for transmitting data from line card 14 to fabric port 20, and unidirectional channel 24 for transmitting data from fabric port 20 to line card 14. As will be more fully described below, each of the channels 22 and 24 includes n serial links, and each serial link may take form in a pair of differential lines. The present invention will be described with each of the channels 22 and 24 including 8 serial links, it being understood that the present invention should not be limited thereto.

[0017] Line card 14 includes a transmit logic and interface module (TLIM) 30 coupled to a transmitter 34, and a RLIM (RLIM) 32 coupled to a receiver 36. Although not shown, line card 14 may include a host ASIC coupled to the TLIM 30 and the RLIM 32. Fabric port 20 includes a RLIM 44 coupled to a receiver 40, and a TLIM 46 coupled to a transmitter 42. Transmitters 34 and 42 may be identical to each other, and receivers 36 and 40 may be identical to each other. Moreover, TLIMs 30 and 46 may be identical to each other, and RLIMs 32 and 44 may be identical to each other.

[0018] The end-devices of Figure 1 can communicate with each other by sending data in accordance with any one of several well known protocols. The line card and fabric port shown in Figure 2 will be described with reference to transmission of data from line card 14 to switching port 20. Data from an end-device enters line card 14 and is subsequently transmitted to fabric port 20 via transmit channel 22. It should be understood that the principals described with respect to transmission of data via transmit channel 22, apply equally to the transmission of data from fabric port 20 to line card 14 via receive channel 24.

[0019] The combination of TLIM 30, transmitter 34, channel 22, receiver 40, and RLIM 44 defines a communication path through which data is transmitted from line card 14 to fabric port 20. In operation, TLIM 30 receives 64-bit wide data values from the line card's host ASIC (not shown) via a 64-bit wide parallel bus in response to the host ASIC receiving data packets or frames from an end-device coupled thereto. Figure 2 shows TLIM 30 receiving 64-bit wide data values A-C in sequential order. If the communication path consisting of TLIM 30, transmitter 34, channel 22, receiver 40, and RLIM 44 is operating properly, RLIM 44 will subsequently output the 64-bit wide data values A-C in sequential order onto a 64-bit wide parallel bus as shown. However, as will be more fully described

below, if the communication path components are not operating properly, the 64-bit wide data values A-C received by TLIM 30 will not emerge in order from RLIM 44. For example, if receiver 40 is not link and/or channel aligned, then the 64-bit wide data values A-C will not emerge from RLIM 44; rather, RLIM will output invalid 64-bit wide data values. The present invention will be described with reference to transmitting 64-bit wide data values A-C, but the present invention should not be limited thereto. The present invention may find application to transmitting data values having fewer bits (e.g. 32-bits) or more data bits (e.g. 128-bits) through the communication path consisting of TLIM 30, transmitter 34, channel 22, receiver 40, and RILM 44.

[0020] Figure 3 illustrates relevant components of the transmitter 34 and receiver 40 in block diagram form. Figure 3 also shows channel 22 of Figure 2, which consists of 8 serial links 50. In one embodiment, each link 50 consists of a pair of differential lines for serially transmitting data as will be more fully described below. Transmitter 34 consists of eight serializers 52 coupled to eight 8-bit/10-bit encoders 60 via respective 10-bit wide parallel buses. Receiver 40 includes eight deserializers 54 coupled to eight 10-bit/8-bit decoders 62 via respective 10-bit wide parallel buses. The 8-bit/10-bit encoders 60 are coupled to respective outputs of the TLIM 30 via 8-bit wide parallel data buses as shown. Similarly, the outputs of the 10-bit/8-bit decoders 62 are coupled to respective inputs of the RLIM 44 via 8-bit wide parallel buses as shown. The relevant purposes and functions of the various components of transmitter 34 and receiver 40 shown in Figure 3 will be described below.

[0021] Figure 3 shows TLIM 30 sequentially receiving three 64-bit wide data values A-C from the host ASIC (not shown) via a 64-bit wide parallel bus. TLIM 30 divides the 64-bit wide data values into 8 bytes (A0-A7), which in turn are outputted at the same time and subsequently transmitted to the eight encoders 60 via respective 8-bit wide parallel buses. TLIM 30 then receives the next 64-bit wide data value B. TLIM 30 divides B into eight bytes (B0-B7), which in turn are transmitted to respective encoders 60. Lastly, TLIM 30 receives 64-bit wide data value C. TLIM 30 divides C into eight bytes (C0-C7), which in turn are transmitted at the same time to respective encoders 60. Figure 3 illustrates bytes of each 64-bit data value being transmitted across the 8-bit wide buses between the TLIM 30 and encoders 60.

[0022] As noted above, channel 22 consists of eight links, each of which includes a pair of differential lines for serially transmitting data bits from a serializer 50 to a respective deserializer 54. Each of the serial links 50 are subject to run length and/or DC imbalance problems described in U.S. Patent Application 11/364,303 (the 303 Application), entitled FABRIC CHANNEL CONTROL APPARATUS AND METHOD, which was filed February 28, 2006, and which is incorporated herein by reference in its entirety. These run length and/or DC imbalance problems can arise when too many bits of the same value (e.g., logical 1 or logical 0) are consecutively transmitted across a serial link. The 8-bit/10-bit encoders 60 prevent these run length and/or DC imbalance problems as described in the 303 Application. In operation, each 8-bit/10-bit encoders maps each 8-bit value into a corresponding 10-bit code, which in turn is subsequently transmitted to a respective serializer 52 via a 10-bit wide parallel data bus coupled therebetween. The serializer 52, in turn, serially transmits each bit of the 10-bit code to its respective deserializer 54 via a serial link 50. Deserializer 54, in response to serially receiving the 10 bits of the 10-bit code, aggregates the 10 bits to recreate the 10-bit code, which in turn is sent to its respective 10-bit/8-bit decoder 62 via a 10-bit wide parallel data bus coupled therebetween. Decoder 62 decodes the 10-bit code into the corresponding 8-bit value, which in turn is sent in parallel to a respective input of RLIM 44 via an 8-bit wide parallel data bus.

[0023] If the deserializers 54 are operating properly (e.g., operating in link and channel alignment), the eight bytes of a 64-bit wide data value inputted to respective 8-bit/10-bit encoders 60 on the line card side, will subsequently emerge in parallel (i.e., at the same time with respect to each other) from respective 10-bit/8-bit decoders 62 on the fabric port side. Figure 3 illustrates that deserializers are operating in link and channel alignment. More particularly, Figure 3 shows that bytes A0-A7 of data value A inputted at the same time to respective encoders 60, subsequently emerge in parallel from respective 10-bit/8-bit decoders 62 at time t_1. Figure 3 also shows bytes B0-B7 of data value B emerging from the 10-bit/8-bit decoders 62 in parallel at the same time t_2 just after bytes A0-A7 emerge in parallel from respective 10-bit/8-bit decoders. Likewise, Figure 3 shows bytes C0-C7 of data value C emerging in parallel at the same time t_3 from respective 10-bit/8-bit decoders 62. Because deserializers are link and channel aligned, RLIM 44 recombines the outputs of the 10-bit/8-bit decoders 62 and sequentially transmits data values A-C onto the 64-bit wide parallel bus coupled thereto. However, RLIM 64 will output 64-bit wide invalid data if one or more of

the deserializers 54 are not link aligned and/or if one or more of the deserializers 54 are not channel aligned with the remaining deserializers 54 as will be more fully described below.

[0024] Figure 4 illustrates the effects of a lack of channel alignment between deserializers 54 of Figure 3. Figure 4 shows the same sequential input of 64-bit wide data values A-C to TLIM 30. For purposes of explanation, it will be presumed the 8-bit/10-bit encoders 60 and the serializers 52 operate as described with reference to Figure 3. Moreover, deserializers 54 will be presumed to be link aligned. However, deserializer 54(1) is presumed to be operating out of channel alignment. In general, channel alignment can mean that deserializers 54 output corresponding 10-bit codes in parallel or at the same time. For purposes of explanation only, channel misalignment will take form in deserializer 54(1) outputting its 10-bit code by one clock cycle after the remaining deserializers 54 output their respective and corresponding 10-bit codes. It should be understood that the term channel misalignment should not be limited to that shown in Figure 4. Rather, channel misalignment may take form in a deserializer 54 outputting its 10-bit code one clock cycle before the remaining deserializers output their respective and corresponding 10-bit codes. Further, channel misalignment may take form in two or more of the deserializers 54 outputting their 10-bit codes at the same time, but one clock cycle before or after the remaining deserializers 54 output their respective and corresponding 10-bit codes.

[0025] As noted, Figure 4 illustrates the effect of channel misalignment of deserializer 54(1). Specifically, Figure 4 shows that byte B1 of B is outputted from decoder 62(1) at the same time t_3 that byte C0 and bytes C2-C7 are outputted from decoders 62(0) and 62(2)-62(7), respectively. Figure 4 also shows that byte A1 of A is outputted from decoder 62(1) at the same time t_2 that byte B0 and bytes B2-B7 are outputted from decoders 62(0) and 62(2)-62(7), respectively. Lastly, byte X1 of a 64-bit wide data value X (not shown) inputted to the TLIM 30 prior to 64-bit wide data value A, is outputted from decoder 62(1) at the same time t_1 as byte A0 and bytes A2-A7 are outputted from decoders 62(0) and 62(2)-62(7), respectively. Bytes A0, X, and A2-A7 are received in parallel and combined by RLIM 44 to produce a 64-bit wide invalid data value. Likewise, bytes B0, A1, and B1-B7, outputted at the same time by decoders 62, respectively, are subsequently received in parallel and combined by RLIM 44 to produce a 64-bit wide invalid data value.

[0026] As noted above, link misalignment can also cause improper function of the communication path shown in Figure 3. Figures 5 and 6 illustrate aspects of deserializer link

alignment and misalignment, respectively. Figures 5 and 6 illustrate encoder 60(0), serializer 52(0), deserializer 54(0), and decoder 62(0) of Figure 3 in greater detail. Figure 5 shows that bytes A0-C0 are sequentially inputted into encoder 60(0) as described above. Encoder 60 translates bytes A0-C0 into respective 10-bit codes G-E; 10-bit code E consists of bits e0-e9, 10-bit code F consists of bits f0-f9, and 10-bit code G consists of bits g0-g9. The 10-bit codes E-F are transmitted in sequence to serializer 52(0) over the 10-bit wide parallel bus therebetween as shown. Serializer 52(0), in response, serially transmits bits e0-e9 to deserializer 54 via serial link 50 in response to receiving 10-bit code E. Just after bit e9 is transmitted, serializer 52(0) serially transmits bits f0-f9 to deserializer 54 via serial link 50. Just after bit f9 is transmitted, serializer 52(0) serially transmits bits g0-g9 to deserializer 54 via serial link 50. Figure 5 shows the bits of codes E-F as they are transmitted in order across link 50(0) to deserializer 54(0). Because deserializer 54(0) is link aligned, deserializer 54 outputs the 10-bit codes E-F in the same manner in which codes E-F were inputted to serializer 52(0). Thus, Figure 5 shows that bits e0-e9 of 10-bit code E are outputted in parallel by deserializer 54(0) at time t_1 and transmitted to decoder 62(0) via the 10-bit wide parallel bus therebetween. Subsequently, deserializer transmits bits f0-f9 of 10-bit code F in parallel at time t_2 to decoder 62(0) via the 10-bit wide parallel bus therebetween. Lastly, bits g0-g9 of 10-bit code G are outputted in parallel at time t_3 by deserializer 54(0) and transmitted to decoder 62(0). Encoder 62, in turn, decodes the 10-bit codes E-F into bytes A0-C0, respectively, as they are received.

[0027] However, when deserializer 54(0) is out of link alignment, deserializer 54(0) may aggregate bits of two 10-bit codes it serially receives via link 50(0) and, accordingly, output erroneous 10-bit values onto the bus coupled to decoder 62(0). Figure 6 illustrates operational aspects of deserializer 54(0) when it is operating in link misalignment in one embodiment. In Figure 6 the same serial stream of bits is transmitted to deserializer 54(0). Because deserializer 54(0) is out of link alignment, the deserializer 54(0) outputs 10-bit values Z1-Z3 in parallel, which are not the same as the 10-bit codes E-F outputted by deserializer 54(0) in Figure 5. Rather, 10-bit values Z1-Z3 are a mixture of bits of 10-bit codes G-F. For example, value Z1 consists of bit e9 of 10-bit code E and bits f0-f8 of 10-bit code F. The 10-bit values Z1-Z3 outputted by deserializer are transmitted in sequential order to decoder 62(0) and subsequently translated into 8-bit values which do not equate with A0-C), respectively. Rather, decoder 62(0) translates 10-bit values Z1-Z3 into 8-bit invalid data values, which are subsequently sent to RLIM 44 (not shown in Figure 6).

[0028] The 64-bit data values outputted from RLIM 44 will be invalid if any of the deserializers 54 are channel misaligned and/or if any of the deserializers 54 are link misaligned. Channel or link misalignments can be detected. When detected, TLIM 30 is directed to enter an alignment mode in which it periodically (e.g., between 64-bit wide data values of data packets sequentially received by TLIM30) transmits special 8-bit codes called K28.5 comma characters in parallel to each of the encoders 60. The encoders map the comma characters to corresponding 10-bit codes, which in turn are subsequently transmitted to deserializers 54. Each of the deserializers 54 may include an alignment circuit that is always active and searching for the 10-bit code corresponding to the comma character. The alignment circuits use the 10-bit codes corresponding to the comma characters to align themselves either at the link level or at the channel level. This alignment circuit will reframe the 10-bit codes corresponding to the comma character if the 10-bit code is detected out of alignment, either at the link level or at the channel level. Unfortunately, when one or more of the deserializers 54 enter a misalignment state, either at the link level or the channel level, RLIM 44 will output invalid data until link and/or channel alignment is obtained.

[0029] Figure 7 illustrates an alternative communication 70. This communication system is similar to communication system 10 shown in Figure 1, but with switching fabric 12 replaced by switching fabric 72, and line cards 14 replaced by line cards 74. The end-devices are the same between Figures 1 and 7 as are channels 16.

[0030] Figure 8 illustrates in block diagram form, relevant components of one of the line cards 74 shown in Figure 7 and relevant components of a switching fabric port 76 of switching fabric 72. More particularly, line card 74 shown in Figure 8 includes a TLIM 80 coupled to a transmitter 84, and a RLIM 82 coupled to a receiver 86. Fabric port 76 includes a receiver 90 coupled to RLIM 94, and a transmitter 92 coupled to TLIM 96. Transmitter 84 and receiver 90 are coupled to each other via the same type of transmit channel 22 shown in Figure 2. Likewise, transmitter 92 is coupled to receiver 86 via the same type of receive channel 24 shown in Figure 2.

[0031] Like TLIM 30 shown in Figure 2, TLIM 80 sequentially receives 64-bit wide data values from a host ASIC (not shown) via a 64-bit wide data bus for subsequent transmission to switch fabric 72 via the transmit communication path, which includes transmitter 84, transmit channel 22, receiver 90, and RLIM 94. RLIM 94 sequentially outputs the same 64-bit wide data values onto a 64-bit wide parallel data bus coupled to it if the transmit

communication path is operating properly. Similarly, TLIM 96 of port 76 sequentially receives 64-bit wide data values via a 64-bit wide data bus for subsequent transmission to an end-device (not shown) coupled to line card 74 via the communication path that includes transmitter 92, receive channel 24, receiver 86, and RLIM 82. RLIM 82 sequentially outputs the same 64-bit wide data values onto a 64-bit wide parallel data bus coupled to it if the receive communication path is operating properly. Like RLIM 44 described above, neither RLIM 82 nor 94 will output valid 64-bit wide data values onto their respective 64-bit wide buses if deserializers contained in the receive and transmit communication paths, respectively, are link and/or channel aligned.

[0032] The 303 Application describes a protocol for data communication between line cards and a switching fabric. Line cards employing the 303 Application protocol do not use 8-bit/10-bit encoders or 10-bit/8-bit decoders to prevent run length and DC bias problems created by the serial links described above. Moreover, the 303 Application protocol does not use K28.5 comma characters for channel and/or link alignment operations. However, as more fully described below, deserializers contained within receiver 86 and/or 90 are capable of realigning themselves either on the link level or at the channel level. In one embodiment of the present invention, TLIM 80 can transmit local channel alignment status information of receiver 82, and TLIM 96 can transmit local channel alignment status information of receiver 94. As will be more fully described below, TLIM 80 can generate alignment codes based upon the alignment status of receiver 86 and/or receiver 90, and TLIM 96 can generate alignment codes based upon the alignment status of receiver 90 and/or receiver 86.

[0033] Figure 9 illustrates components of Figure 8 in greater detail. Specifically, Figure 9 shows transmitter 84 containing serializers 102, and transmitter 92 containing serializers 114. Figure 9 also shows receiver 86 containing deserializers 112, and receiver 90 containing deserializers 104. The serializers of Figure 9 operate in a manner that is the same or similar to the operation of serializers 52 described above. The deserializers of Figure 9 operate in a manner that is the same or similar to the operation of deserializers 54 described above. Transmitters 84 and 92 or receivers 86 and 90 may include integrated circuit components other than that shown in Figure 9.

[0034] Deserializers 104 and 112 may include link level alignment logic for aligning the deserializers on the link level should they enter a state of link level misalignment. Deserializers 104 and 112 may also include channel level alignment logic, which collectively

align the deserializers on the channel level should they enter a state of channel level misalignment. Alternatively, the link level and channel level alignment logic may take form in integrated circuits that are external to the deserializers 104 and 112, but which operate on the outputs of the deserializers 104 and 112 and provide link level and/or channel level alignment. In this alternative embodiment, the outputs of deserializers 104 can be coupled to respective inputs of link level alignment logic circuits (not shown) via respective 10-bit wide parallel buses (not shown), the outputs of the link level alignment logic circuits can be coupled to respective inputs of the channel level alignment logic circuits (not shown) via respective 10-bit wide parallel buses (not shown), and the outputs of the channel level alignment logic circuits can be coupled to respective inputs of the RLIM 94 via respective 10-bit wide parallel buses. On the line card side the outputs of deserializers 112 can be coupled to respective inputs of link level alignment logic circuits (not shown) via respective 10-bit wide parallel buses (not shown), the outputs of the link level alignment logic circuits can be coupled to respective inputs of the channel level alignment logic circuits (not shown) via respective 10-bit wide parallel buses (not shown), and the outputs of the channel level alignment logic circuits can be coupled to respective inputs of the RLIM 82 via respective 10-bit wide parallel buses. The link level and/or channel level alignment logic, whether contained in the deserializers 112 or configured as separate integrated circuits in the data path between the deserializers 112 and the RLIM 82, may be in data communication with TLIM 80, which enables the link level and/or channel level alignment logic to report alignment status (i.e., aligned or misaligned on the channel and/or link level) as will be more fully described below. Similarly, the link level and/or channel level alignment logic, whether contained in the deserializers 104 or configured as separate integrated circuits in the data path between the deserializers 104 and the RLIM 94, may be in data communication with TLIM 96, which enables the link level and/or channel level alignment logic to report alignment status (i.e., aligned or misaligned on the channel and/or link level) as will be more fully described below. The deserializers 104 and 112 can also report to TLIM 96 and TLIM 80, respectively, the types of 10-bit alignment codes deserializers 104 and 112 receive from their respective serializers 102 and 92 as will be more fully described below.

[0035] In operation, TLIM 80 sequentially receives 64-bit wide data values (e.g., A-C) of, for example, a data packet via the 64-bit wide data bus coupled to a host ASIC (not shown). TLIM 80 may operate in accordance with the protocol described in the 303 Application. In particular, when TLIM 80 receives the three 64-bit wide data values (e.g., A-C), TLIM 80

generates a frame. The frame may consist of a header (more fully described in the 303 Application) and the contents of the three 64-bit wide data values (e.g., A-C).

[0036] TLIM 80 outputs eight 10-bit portions of the frame in parallel for subsequent transmission to respective serializers 102 via the 10-bit wide parallel buses, respectively shown in Figure 9. After the first eight 10-bit portions of the frame are transmitted to respective serializers, the next eight 10-bit portions of the frame are transmitted. The third eight 10-bit frame portions outputted in parallel by TLIM 80 may contain data from the next frame generated by TLIM 80.

[0037] Serializers 102, in turn, transmit the respective 10-bit frame portions they receive from TLIM 80 in serial fashion to respective deserializers 104 via respective serial links 50. If deserializers 104 are in link and channel alignment, deserializers 104 will output in parallel the same 10-bit frame portions that were respectively received in parallel by the serializers 102. The eight 10-bit frame portions respectively outputted by deserializers 104 are transmitted to RLIM 94 via respective 10-bit wide parallel buses coupled therebetween.

[0038] RLIM 94 may process the 10-bit frame portions it receives to recreate the frames. RLIM 94 may then process the frames or frames it recreates in accordance with the protocol of the 303 Application in order to generate the original 64-bit wide data values A-C. Eventually, RLIM 94 sequentially outputs A-C onto the 64-bit wide parallel data bus coupled thereto. The receive communication path including TLIM 96, serializers 114, receive channels 100, deserializers 112, and RLIM 82 operate in substantially the same way as the transmit communication path including TLIM 80, serializers 102, transmit channels 50, deserializers 104, and RLIM 94 described above. Thus, if the transmit communication path is operating properly, 64-bit wide data values Z-X sequentially inputted to TILM 96, will eventually emerge from RLIM 82 as shown.

[0039] Deserializers 104 and 112 include link alignment and channel alignment logic, for initial aligning or for realigning the deserializers on the link level and/or channel level as noted above. An initial link and/or channel alignment may need to occur when components shown in Figure 9 are first powered up or rebooted, while realignment may need to occur when one or more of the deserializers are discovered to be operating out of alignment either on a link level an/or a channel level basis. For purposes of explanation, the present invention will be described below with respect to realigning the deserializers after one or more are

discovered to be operating out of alignment on both the link level and channel level, it being understood that the present invention should not be limited thereto. In contrast to the system described in Figures 3 and 4, TLIM 80 may suspend the transmission of 64-bit wide data received from the host ASIC until deserializers 104 are realigned on the link and/or channel level, or TLIM 80 may suspend the transmission of 64-bit wide data values received from the host ASIC until deserializers 104 and 112 are realigned on the link and/or channel level. TLIM 96 may also suspend the transmission of 64-bit wide data it receives under the same circumstances.

[0040] When it is discovered that one or more of the deserializers 104 or 112 are out of alignment, TLIM 80 and TLIM 96 output a first 10-bit alignment code for subsequent transmission to each of the serializers 102 and 114, respectively. TLIM 80 transmits the first 10-bit alignment code in parallel (i.e., at the same time) to each of the serializers 102, and TLIM 96 transmits the same first 10-bit alignment code in parallel to each of the serializers 114. The first 10-bit code indicates that local deserializers are out of link alignment (deserializers 112 are local to TLIM 80 while deserializers 104 are local to TLIM 96). Thus, the first 10-bit alignment code transmitted to serializer 102 indicates that one or more deserializers 112 are out of link alignment. Likewise, the first 10-bit alignment code transmitted in parallel (i.e., at the same time) to serializer 114 indicates that one or more deserializers 104 are out of link alignment. For purposes of explanation, it will be presumed that only one of the deserializers 104 is out of link level alignment when the first 10-bit alignment codes are transmitted to serializers 102 and 114. In one embodiment, the first 10-bit alignment code may take form as "1010100011," where the least significant bits "011" are the active part of the 10-bit alignment code that defines the status of the local deserializers as out of link alignment.

[0041] TLIM 80 will continue to send the first 10-bit alignment codes in parallel to respective serializers 102 for subsequent transmission to deserializers 104, respectively, until TLIM 80 receives an indication from deserializers 112 that each is link level aligned. It is noted TLIM 80 may send one or more 10-bit values in between successive transmissions of the first 10-bit alignment codes. Similarly, TLIM 96 will continue to send the first 10-bit alignment codes in parallel to serializers 114 for subsequent transmission to deserializers 112, respectively, until TLIM 96 is notified that each of the deserializers 104 are in link level alignment.

[0042] The link level alignment logic (not shown) of deserializers 104 can use the first 10-bit alignment codes received from respective serializers 102 to align deserializers 104 on the link level. Once a deserializer 104 is aligned on the link level, the deserializer will send TLIM 96 a signal indicative thereof via the connection shown in Figure 9. Although Figure 9 only shows one connection between deserializer 104(7) and TLIM 96, it is presumed that each of the deserializers 104 has a connection to TLIM 96 over which it can transmit their link level alignment status. Similarly, the link level alignment logic (not shown) of deserializers 112 use the first 10-bit alignment codes received from respective serializers 114 to align deserializers 112 on the link level. Once a deserializer 112 is link aligned, the deserializer will send TLIM 80 a signal indicative thereof via the connection shown in Figure 9. Although Figure 9 only shows one connection between deserializer 112(0) and TLIM 80, it is presumed that each of the deserializers 112 has a connection to TLIM 80 over which it can transmit their link level alignment status.

[0043] When TLIM 96 receives verification that each of the deserializers 104 is in link level alignment, TLIM 96 may transmit a second 10-bit alignment code in parallel to each of the serializers 114. The second 10-bit code indicates that each of the local deserializers (i.e., deserializers 104) is link aligned. Similarly, when TLIM 80 receives verification that each of the deserializers 112 is in link level alignment, TLIM 80 will transmit the second 10-bit alignment code to each of the serializers 102. The second 10-bit code indicates that each of the local deserializers 104 is in link level alignment. In one embodiment, the second 10-bit alignment code may take form as "1010100110," where the least significant bits "110" are the active part of the second 10-bit alignment code that defines the status of the local deserializers as link aligned. Serializers 114 transmit the second 10-bit alignment code in parallel to respective deserializers 112, and serializers 102 transmit the second 10-bit alignment code in parallel to respective deserializers 104.

[0044] The link level alignment logic (not shown) of deserializers 112 can use the second 10-bit alignment code they receive, respectively, to link align themselves if not already in link alignment, and the link level alignment logic (not shown) of deserializers 104 deserializers 104 can use the second 10-bit alignment code they receive, respectively, to link align themselves if not already in link alignment. As noted above, deserializers 104 and 112 inform TLIM 96 and TLIM 80, respectively, when they are link aligned. In addition, deserializers 104 and 112 inform TLIM 96 and TLIM 80, respectively of the alignment status

of deserializers 112 and 104, respectively, by virtue of the alignment codes they receive. Thus, when deserializers 104 receive second 10-bit alignment codes, deserializers 104 transmit a signal to TLIM 96 indicating that deserializers 112 are in link alignment, and when deserializers 112 receive second 10-bit alignment codes, deserializers 112 transmit a signal to TLIM 80 indicating that deserializers 104 are in link alignment. As an aside, the first and second 10-bit alignment codes are selected to insure that links 50 and 100 do not experience run length or DC balance problems. In one embodiment, the first and second 10-bit alignment codes contain an equal number or logical ones and zeros.

[0045] Once TLIM 80 is informed by local deserializers 112 that all deserializers 112 and 104 are link aligned, TLIM 80 may transmit third 10-bit alignment codes in parallel (i.e., at the same time) to respective serializers 102. Likewise, once TLIM 96 is informed by local deserializers 104 that all deserializers 104 and 112 are link aligned, TLIM 96 may transmit third 10-bit alignment codes in parallel (i.e., at the same time) to respective serializers 102. The third 10-bit alignment code indicates the local deserializers are not channel aligned. Thus, the third 10-bit alignment codes sent by TLIM 96 indicate that deserializers 104 are not in channel level alignment. Similarly, the third 10-bit code sent by TLIM 80 indicates that deserializers 112 are channel aligned. The third 10-bit alignment codes may be part of a frame generated in accordance with the protocol of the 303 Application, which frame may also include stuff code. In one embodiment, the third 10-bit alignment code may take form as "1010100100," where the least significant bits "100" are the active part of the 10-bit alignment code that defines the status of the local deserializers as out of channel alignment..

[0046] Serializers 102 serially transmit the respective third 10-bit alignment code they receive to respective deserializers 104. Channel level alignment logic of the deserializers 104, in response, channel align the deserializers 104 using the third 10-bit alignment codes they receive in accordance with well known principles. Once the deserializers 104 are in channel level alignment, a signal to that effect may be sent to TLIM 96. Likewise, serializers 114 serially transmit the respective third 10-bit alignment codes they receive to respective deserializers 112. Channel level alignment logic of the deserializers 112 use the third 10-bit alignment codes they receive to channel align the deserializers 104 in accordance with well known principles. Once the deserializers 112 are in channel level alignment, a signal to that effect may be sent to TLIM 80. Channel level alignment logic of the deserializers 104 should be able to channel align the deserializers with the first set of third 10-bit alignment codes they

receive. Likewise, channel level alignment logic of the deserializers 112 should be able to channel align the deserializers with the first set of third 10-bit alignment codes they receive.

[0047] When TLIM 80 receives verification that local deserializers 112 are in channel alignment, TLIM 80 may transmit fourth 10-bit alignment codes in parallel (i.e., at the same time) to respective serializers 102. This fourth bit code indicates that local deserializers (e.g., deserializers 112) are channel aligned. Serializers 102, in response, transmit the fourth 10-bit alignment code to their respective deserializers 104. When deserializers 104 receive the fourth 10-bit codes, respectively, deserializers 104 may send a signal to TLIM 96 indicating that deserializers 112 are channel aligned. Similarly, when TLIM 96 receives verification that local deserializers 104 are in channel alignment, TLIM 96 may transmit fourth 10-bit alignment codes in parallel (i.e., at the same time) to respective serializers 114. Serializers 114, in response, transmit the fourth 10-bit alignment code to their respective deserializers 112. When deserializers 112 receive the fourth 10-bit code, the deserializers 112 may send a signal to TLIM 96 indicating that deserializers 104 are in channel alignment. The fourth 10-bit alignment codes may be part of a frame generated in accordance with the protocol of the 303 Application, which frame may also include stuff code. In one embodiment, the fourth 10-bit alignment code may take form as "1010100000," where the least significant bits "000" are the active part of the fourth 10-bit alignment code that defines the status of the local deserializers as channel aligned.

[0048] Once TLIM 96 receives confirmation that deserializers 112 are in channel level alignment via the fourth 10-bit codes received via serializers 102, TLIM 96 can exit its realignment mode of operation and resume transmitting data to RLIM 82 using, for example, the protocol described in the 303 Application, and TLIM 96 can resume transmitting data before or after TLIM 96 receives confirmation that deserializers 104 are in link and/or channel alignment. Likewise, Once TLIM 80 receives confirmation that deserializers 104 are in channel level alignment via the fourth 10-bit codes received via serializers 114, TLIM 80 can exit its realignment mode of operation and resume transmitting data to RLIM 94 using, for example, the protocol described in the 303 Application, and TLIM 80 can resume transmitting data before or after TLIM 80 receives confirmation that deserializers 112 are in link and/or channel alignment. Alternatively, and perhaps in the preferred embodiment, TLIM 96 and TLIM 80 will begin transmitting data to each other using the protocol described

in the 303 Application once TLIM 96 and TLIM 80 receive confirmation that all local and remote deserializers are in channel level and/or link level alignment.

[0049] Although the present invention has been described in connection with several embodiments, the invention is not intended to be limited to the specific forms set forth herein. On the contrary, it is intended to cover such alternatives, modifications, and equivalents as can be reasonably included within the scope of the invention as defined by the appended claims. END

WHAT IS CLAIMED IS:

1. A method comprising:

local serializers serially transmitting first n-bit codes, respectively, to remote deserializers, respectively;

local deserializers serially receiving first n-bit codes, respectively, from remote serializers, respectively;

wherein one of the first n-bit codes transmitted to one of the remote serializers indicates one of the local deserializers is not link aligned.

2. The method of claim 1 wherein one of the first n-bit codes received by one of the local serializers indicates one of the remote deserializers is not link aligned.

3. The method of claim 1 further comprising the local serializers serially transmitting second n-bit codes, respectively, to the remote deserializers, respectively, after the local serializers serially transmit the first n-bit codes, wherein each of the second n-bit codes indicates all the local deserializers are link aligned.

4. The method of claim 1 further comprising:

the local serializers serially transmitting second n-bit codes, respectively, to the remote deserializers, respectively, after the local serializers serially transmit the first n-bit codes, wherein each of the second n-bit codes transmitted by the local serializers indicates all the local deserializers are link aligned;

the local deserializers serially receiving second n-bit codes, respectively, from the remote serializers, respectively, wherein each of the second n-bit codes received by the local deserializers indicates all the remote deserializers are link aligned.

5. The method of claim 1 further comprising:

a first of the remote serializers receiving an n-bit value in parallel;

the first remote serializer serially transmitting the n-bits of the n-bit value to a first of the local deserializers in response to the first remote serializer receiving the n-bit value in parallel;

wherein the first local deserializer outputs the n-bit value in parallel in response to serially receiving the n-bits of the n-bit value from the first remote serializer when the first local deserializer is link aligned;

wherein the first local deserializer does not output the n-bit value in parallel in response to serially receiving the n-bits of the n-bit value from the first remote serializer when the first local deserializer is not link aligned.

6. The method of claim 3 further comprising:

the local serializers serially transmitting third n-bit codes, respectively, to the remote deserializers, respectively;

the local deserializers serially receiving third n-bit codes, respectively, from the remote serializers, respectively;

each of the local deserializers outputting in parallel the third n-bit code it receives from its respective remote serializer;

wherein one of the third n-bit codes transmitted to one of the remote deserializers indicates one of the local deserializers does not output its third n-bit code in parallel at the same time the other local deserializers output their respective third n-bit codes in parallel.

7. The method of claim 6 further comprising:

each of the remote deserializers outputting in parallel the third n-bit code it receives from its respective local serializer;

wherein one of the third n-bit codes received by a respective local deserializer indicates one of the remote deserializers does not output its third n-bit code in parallel at the same time the other remote deserializers output their respective third n-bit codes in parallel.

8. The method of claim 6 further comprising:

a device receiving the third n-bit codes outputted by the local deserializers, wherein one of the third n-bit codes is not received by the device at the same time the rest of the third n-bit codes is received by the device;

the device aligning all of the third n-bit codes the device receives so that all of the third n-bit codes can be outputted from the device in parallel at the same time.

9. The method of claim 8 further comprising the local serializers serially transmitting fourth n-bit codes, respectively, to the remote deserializers, respectively, after the local serializers serially transmit the third n-bit codes, wherein each of the fourth n-bit codes indicates all of the third n-bit codes have been channel aligned.

10. The method of claim 1 further comprising:

one of the remote deserializers using the first n-bit code it receives to link align itself;

one of the local deserializers using the second n-bit code it receives to link align itself.

11. A system comprising:

a first node;

a line card coupled to the first node, wherein the line card comprises local serializer circuits for serially transmitting first n-bit codes, respectively, to remote deserializer circuits, respectively, contained in the first node;

wherein the line card further comprises local deserializer circuits for serially receiving first n-bit codes, respectively, from remote serializer circuits, respectively, contained in the first node;

wherein one of the first n-bit codes transmitted to one of the remote serializer circuits indicates one of the local deserializer circuits is not link aligned.

12. The system of claim 11 wherein the local serializer circuits is configured to serially transmit second n-bit codes, respectively, to the remote deserializer circuits, respectively, after the local serializer circuits serially transmit the first n-bit codes, wherein each of the second n-bit codes indicates all the local deserializer circuits are link aligned.

13. The system of claim 12:

wherein the local serializer circuits are configured to serially transmit third n-bit codes, respectively, to the remote deserializer circuits, respectively;

wherein the local deserializer circuits are configured to serially receive third n-bit codes, respectively, from the remote serializer circuits, respectively;

wherein each of the local deserializer circuits is configured to output in parallel the third n-bit code it receives from its respective remote serializer circuit;

wherein one of the third n-bit codes transmitted to one of the remote deserializer circuits indicates one of the local deserializer circuits does not output its third n-bit code in parallel at the same time the other local deserializer circuits output their respective third n-bit codes in parallel.

14. The system of claim 13 further comprising:

wherein each of the remote deserializer circuits is configured to output in parallel the third n-bit code it receives from its respective local serializer circuit;

wherein one of the third n-bit codes received by a respective local deserializer circuit indicates one of the remote deserializer circuits does not output its third n-bit code in parallel at the same time the other remote deserializer circuits output their respective third n-bit codes in parallel.

15. The system of claim 13 further comprising:

a device receiving the third n-bit codes outputted by the local deserializer circuits, wherein one of the third n-bit codes is not received by the device at the same time the rest of the third n-bit codes is received by the device;

the device aligning all of the third n-bit codes the device receives so that all of the third n-bit codes can be outputted from the device in parallel at the same time.

16. The system of claim 15 wherein the local serializer circuits are configured to serially transmit fourth n-bit codes, respectively, to the remote deserializer circuits, respectively, after the local serializer circuits serially transmit the third n-bit codes, wherein each of the fourth n-bit codes indicates all of the third n-bit codes have been channel aligned.

17. The system of claim 11:

wherein one of the remote deserializer circuits is configured to use the first n-bit code it receives to link align itself;

wherein one of the local deserializer circuits is configured to use the second n-bit code it receives to link align itself.

18. A system comprising:

a first node;

a line card coupled to the first node, wherein the line card comprises a plurality of first means for serially transmitting first n-bit codes, respectively, to a plurality of second means, respectively, contained in the first node;

wherein the line card further comprises a plurality of third means for serially receiving first n-bit codes, respectively, from a plurality of fourth means, respectively, contained in the first node;

wherein one of the first n-bit codes transmitted to one of the fourth means indicates one of the third means is not link aligned.

19. The system of claim 11 wherein the plurality of first means is configured to serially transmit second n-bit codes, respectively, to the plurality of second means, respectively, after the plurality of first means serially transmit the first n-bit codes, wherein each of the second n-bit codes indicates all the plurality of third means are link aligned.

20. The system of claim 12:

wherein the plurality of first means are configured to serially transmit third n-bit codes, respectively, to the plurality of fourth means, respectively;

wherein the plurality of third means are configured to serially receive third n-bit codes, respectively, from the plurality of fourth means, respectively;

wherein each of the plurality of third means is configured to output in parallel the third n-bit code it receives from its respective fourth means;

wherein one of the third n-bit codes transmitted to one of the plurality of second means indicates one of the plurality of third means does not output its third n-bit code in parallel at the same time the other of the plurality of local means output their respective third n-bit codes in parallel.

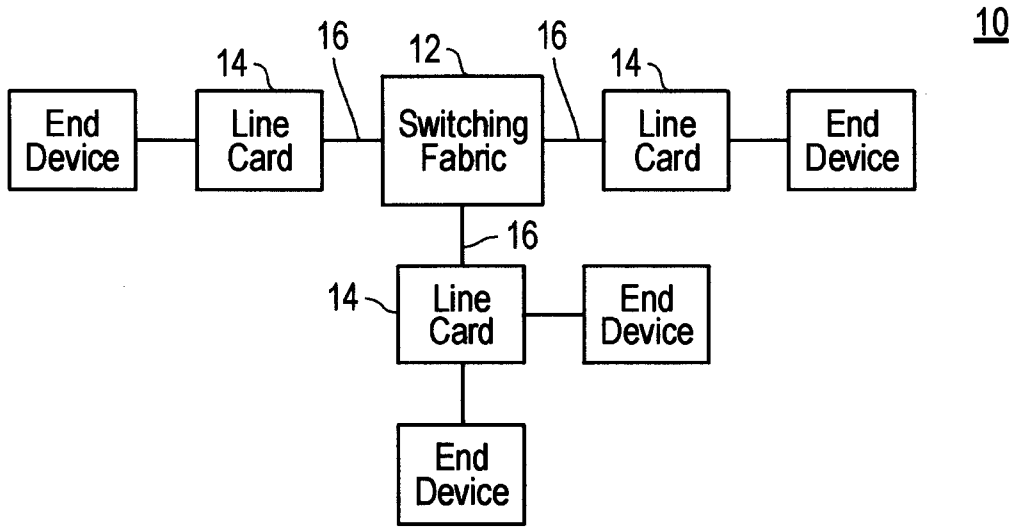

10
12
14
16
End Device
Line Card
Switching Fabric
Line Card
End Device
End Device
Line Card
End Device
End Device

FIG. 1

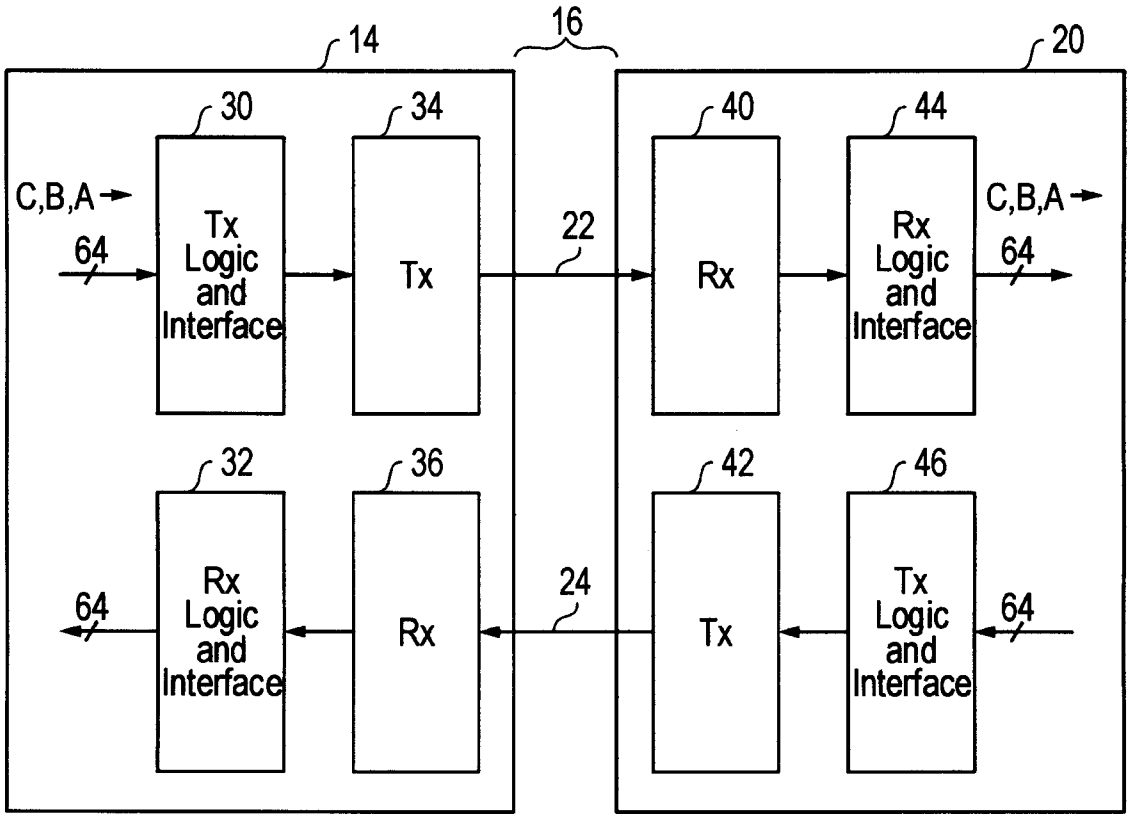

14
16
20
30
34
40
44
C,B,A
64
Tx Logic and Interface
Tx
22
Rx
Rx Logic and Interface
C,B,A
64
32
36
42
46
64
Rx Logic and Interface
Rx
24
Tx
Tx Logic and Interface
64

FIG. 2

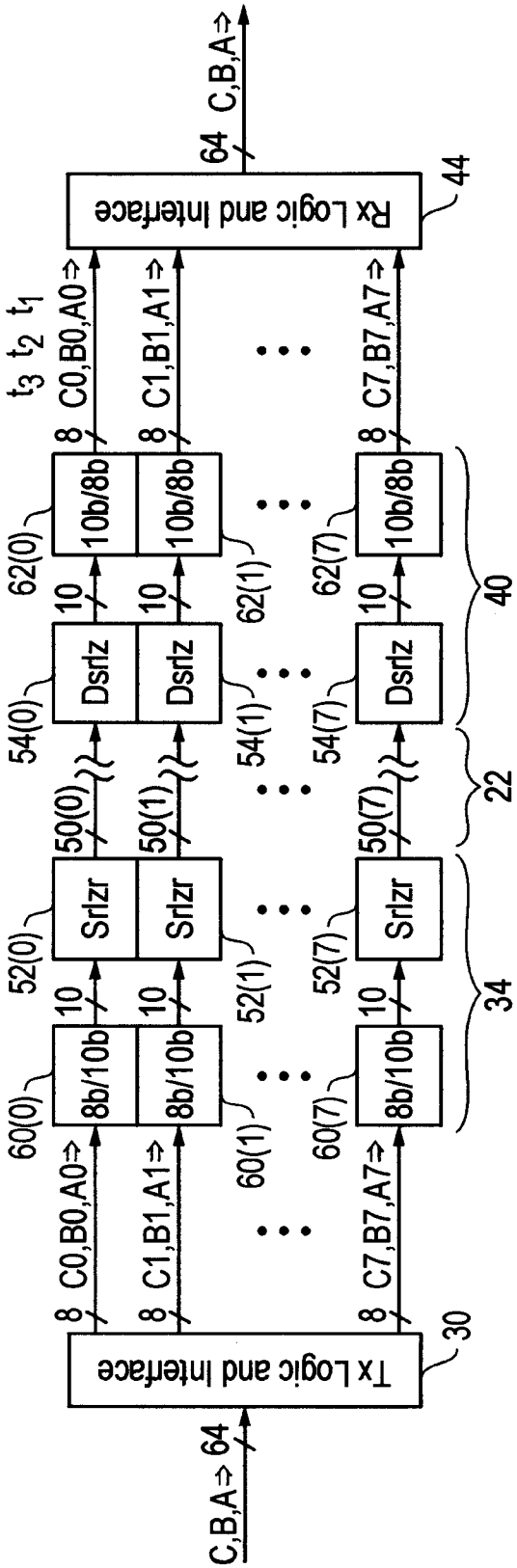

C,B,A⇒ 64
Tx Logic and Interface
30
8 C0,B0,A0⇒
8 C1,B1,A1⇒
8 C7,B7,A7⇒
60(0)
60(1)
60(7)
8b/10b
10
52(0)
52(1)
52(7)
Srlzr
50(0)
50(1)
50(7)
34
22
54(0)
54(1)
54(7)
Dsrlz
62(0)
62(1)
62(7)
10b/8b
8 C0,B0,A0⇒
8 C1,B1,A1⇒
8 C7,B7,A7⇒
t3 t2 t1
40
Rx Logic and Interface
44
64 C,B,A⇒

FIG. 3

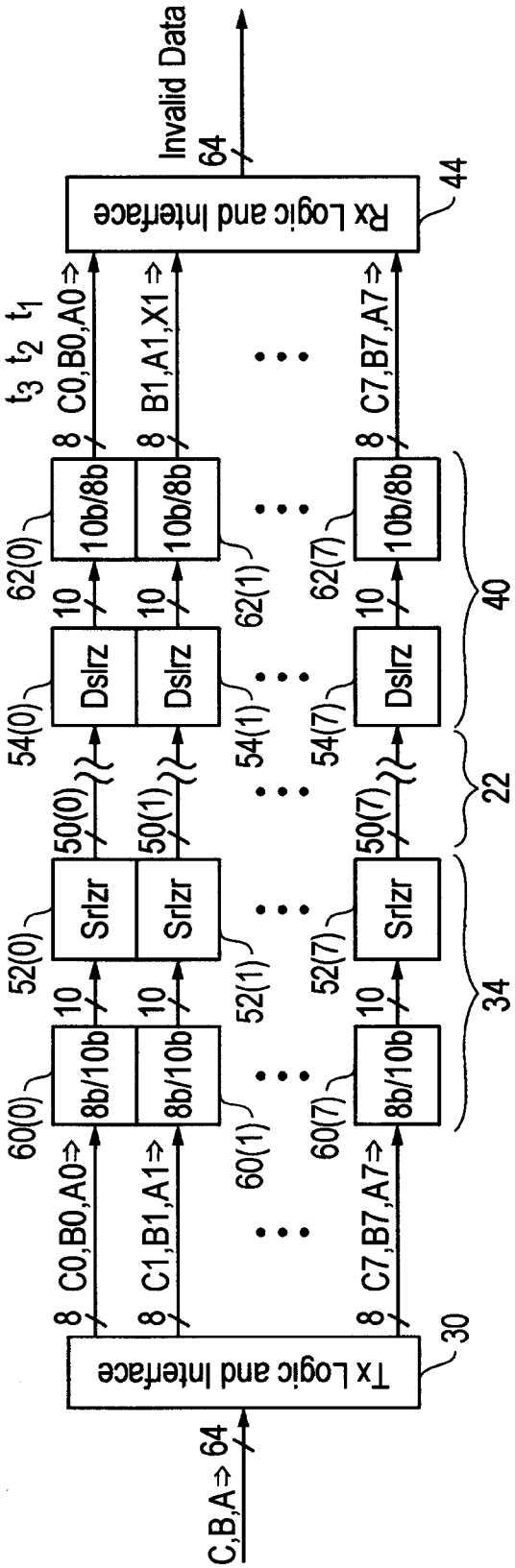

C,B,A⇒ 64
Tx Logic and Interface
30
8 C0,B0,A0⇒
8 C1,B1,A1⇒
8 C7,B7,A7⇒
60(0)
60(1)
60(7)
8b/10b
10
52(0)
52(1)
52(7)
Srlzr
50(0)
50(1)
50(7)
34
22
40
54(0)
54(1)
54(7)
Dslrz
62(0)
62(1)
62(7)
10b/8b
t3 t2 t1
8 C0,B0,A0⇒
8 B1,A1,X1⇒
8 C7,B7,A7⇒
Rx Logic and Interface
44
Invalid Data
64

FIG. 4

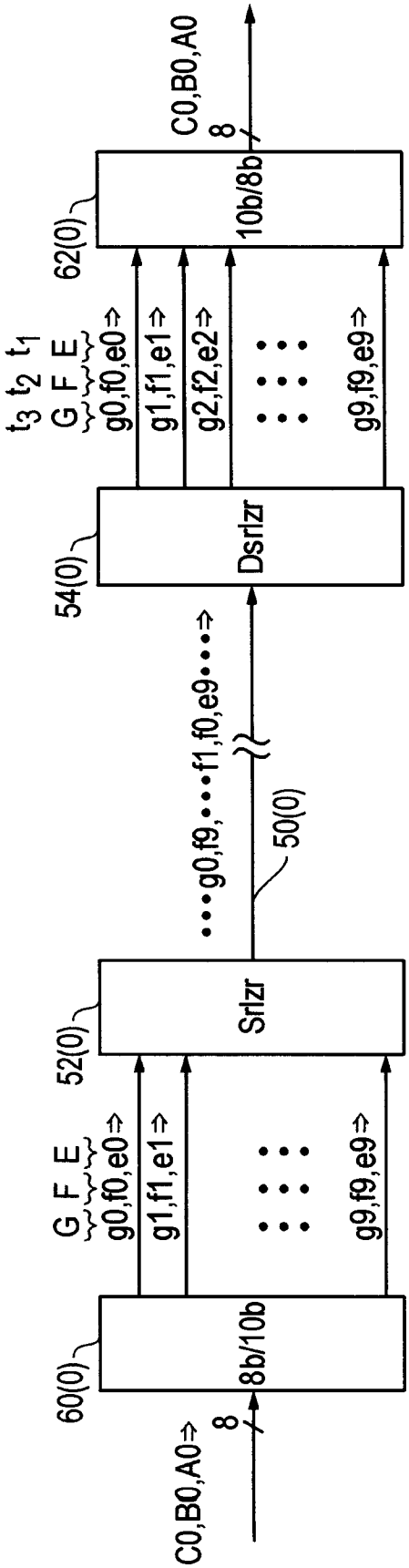

C0,B0,A0⇒
8
60(0)
8b/10b
G F E
g0,f0,e0⇒
g1,f1,e1⇒
g9,f9,e9⇒
52(0)
Srlzr
•••g0,f9,••f1,f0,e9•••⇒
50(0)
54(0)
Dsrlzr
t3 t2 t1
G F E
g0,f0,e0⇒
g1,f1,e1⇒
g2,f2,e2⇒
g9,f9,e9⇒
62(0)
10b/8b
8
C0,B0,A0

FIG. 5

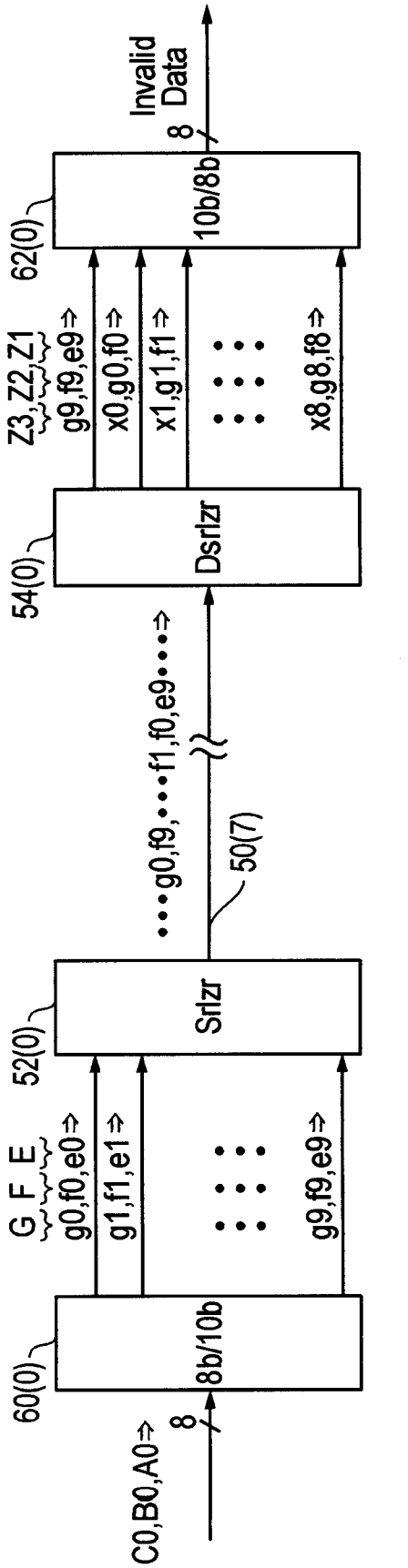

C0,B0,A0⇒
8
60(0)
8b/10b
G F E
g0,f0,e0⇒
g1,f1,e1⇒
g9,f9,e9⇒
52(0)
Srlzr
•••g0,f9,••f1,f0,e9•••⇒
50(7)
54(0)
Dsrlzr
t3 t2 t1
Z3,Z2,Z1
g9,f9,e9⇒
x0,g0,f0⇒
x1,g1,f1⇒
x8,g8,f8⇒
62(0)
10b/8b
8
Invalid
Data

FIG. 6

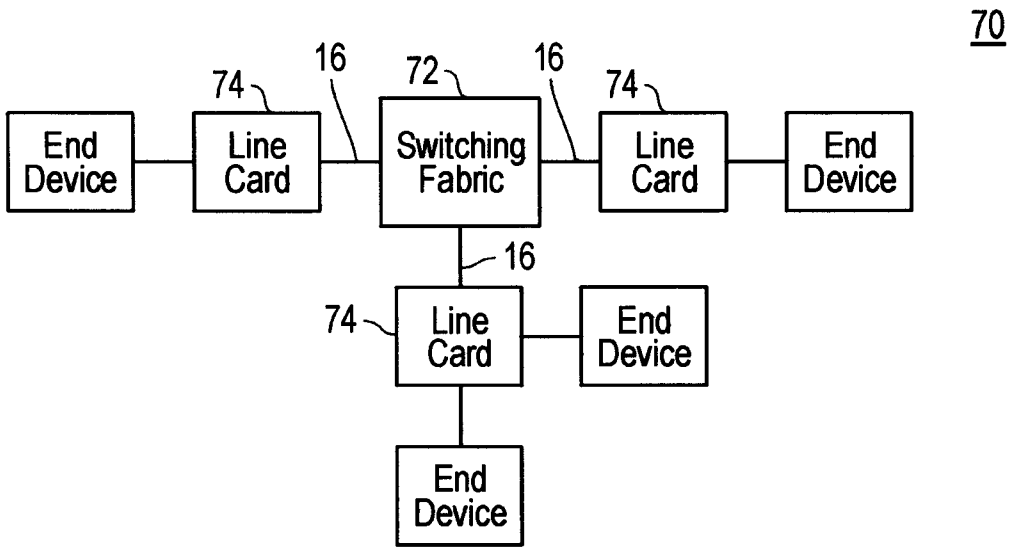

70
74
16
72
16
74
End Device
Line Card
Switching Fabric
Line Card
End Device
16
74
Line Card
End Device
End Device

FIG. 7

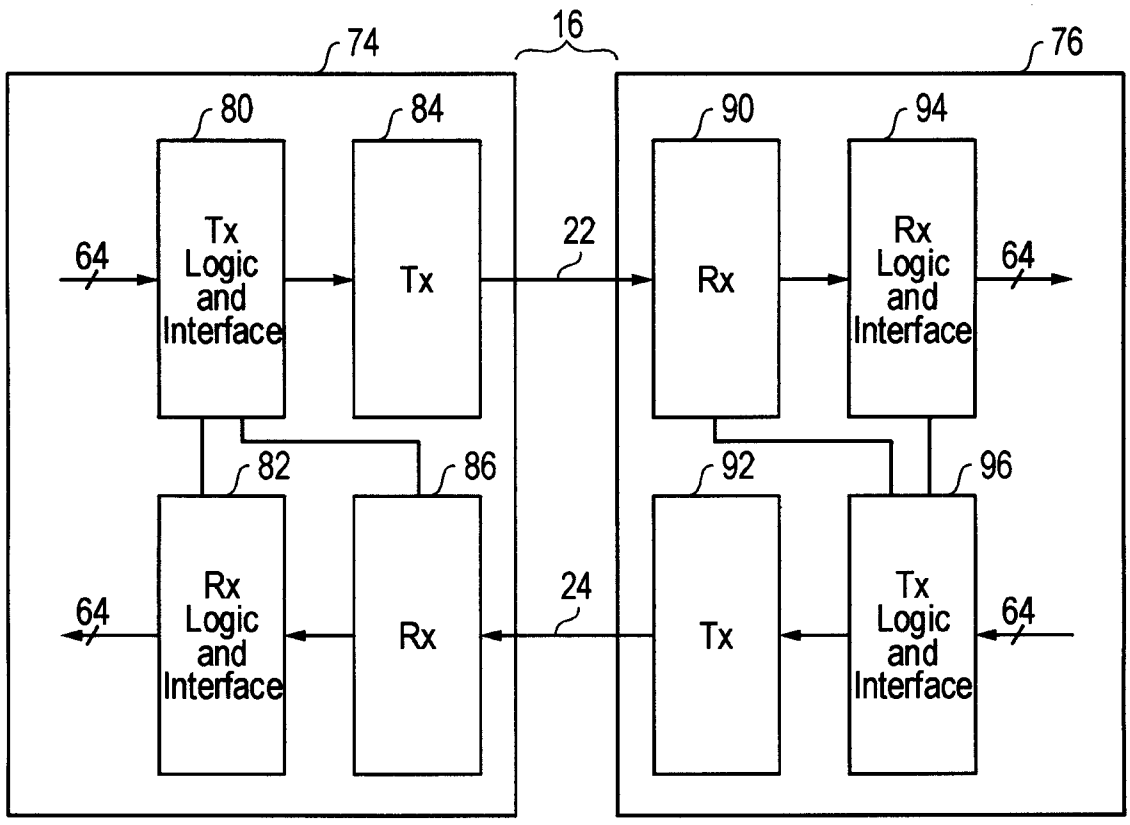

74
16
76
80
84
90
94
64
Tx Logic and Interface
Tx
22
Rx
Rx Logic and Interface
64
82
86
92
96
64
Rx Logic and Interface
Rx
24
Tx
Tx Logic and Interface
64

FIG. 8

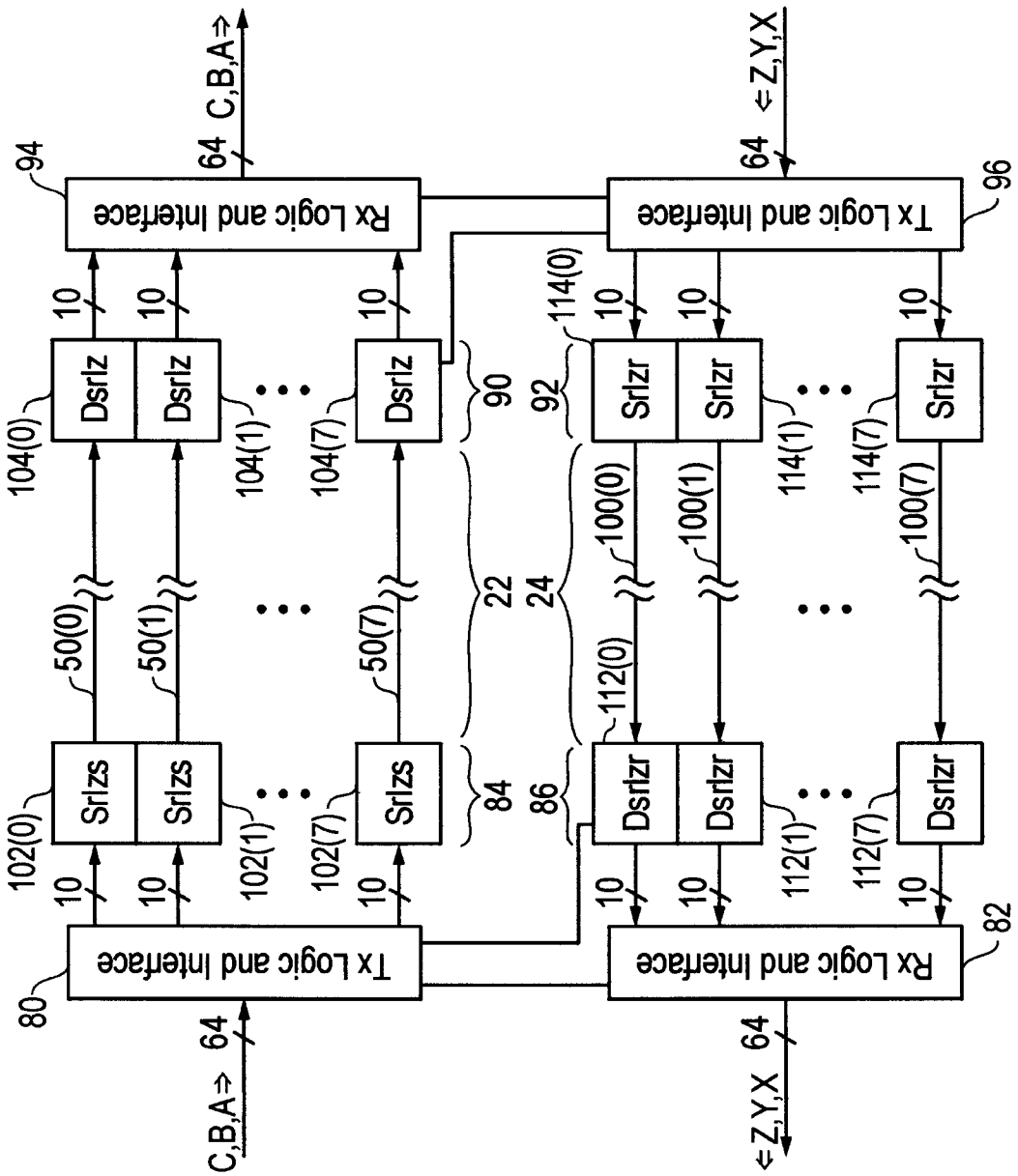

C,B,A⇒ 64
80
Tx Logic and Interface
10
102(0)
Srlzs
50(0)
102(1)
102(7)
84
22
86
24
Dsrlzr
112(0)
100(0)
Srlzr
114(0)
112(1)
112(7)
100(7)
114(1)
114(7)
82
Rx Logic and Interface
64 ⇐Z,Y,X
⇐Z,Y,X 64
96
Tx Logic and Interface
64 C,B,A⇒
94
Rx Logic and Interface
104(0)
Dsrlz
50(1)
104(1)
104(7)
50(7)
90
92

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US08/51988

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04B 7/212 (2008.04)
USPC - 370/324
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - H03M 9/00, 1/72; H04B 7/212; H04J 3/00, 3/06 (2008.04)
USPC - 341/100, 101, 146; 370/324, 476, 514

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Database: PatBase

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6,459,393 B1 (NORDMAN) 01 October 2002 (01.10.2002) entire document	1-20
A	US 2003/0033459 A1 (GARNETT) 13 February 2003 (13.02.2003) entire document	1-20
A	US 6,662,254 B1 (TAL et al) 09 December 2003 (09.12.2003) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
06 May 2008	23 MAY 2008
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2005)