

(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2014274577 B2**

(54) Title
TRANSMISSION SYSTEM ERROR DETECTION AND CORRECTION SYSTEM AND METHOD

(51) International Patent Classification(s)
G06F 11/16 (2006.01) **H04L 1/22** (2006.01)

(21) Application No: **2014274577** (22) Date of Filing: **2014.12.11**

(30) Priority Data

(31) Number	(32) Date	(33) Country
14/145,636	2013.12.31	US

(43) Publication Date: **2015.07.16**

(43) Publication Journal Date: **2015.07.16**

(44) Accepted Journal Date: **2019.07.18**

(71) Applicant(s)
General Electric Company

(72) Inventor(s)
Glosser Jr., Richard Joseph; Wise, Michael Glynn; Schmitt, Steven James

(74) Agent / Attorney
Phillips Ormonde Fitzpatrick, L 16 333 Collins St, Melbourne, VIC, 3000, AU

(56) Related Art
US 2013/0318406 A1
Nath M: "Can Protocol Eases Automotive Electronics Networking" EDN Network, Reed Business Information, Highlands Ranch Vol 43 No. 17, 17 August 1998

TRANSMISSION SYSTEM ERROR DETECTION AND CORRECTION SYSTEM AND METHOD

ABSTRACT OF THE DISCLOSURE

A method for real time error correction in a data transmission system that includes receiving a first communication signal on a first programmable logic device from a second programmable logic device via a first serial link, receiving a second communication signal on the first programmable logic device from the second programmable logic device via a second serial link, in which the first serial link and the second serial link are disposed in parallel with each other and are configured to communicate over a single conduit, and the first communication signal and the second communication signal are representative of identical information. The method further includes comparing the first communication signal and the second communication signal in the first programmable logic device to detect an error in data in the first communication signal, the second communication signal, or both, and when the error is detected, correcting the error in real time in the first programmable logic device based at least in part on the comparison of the first communication signal and the second communication signal.

124 →

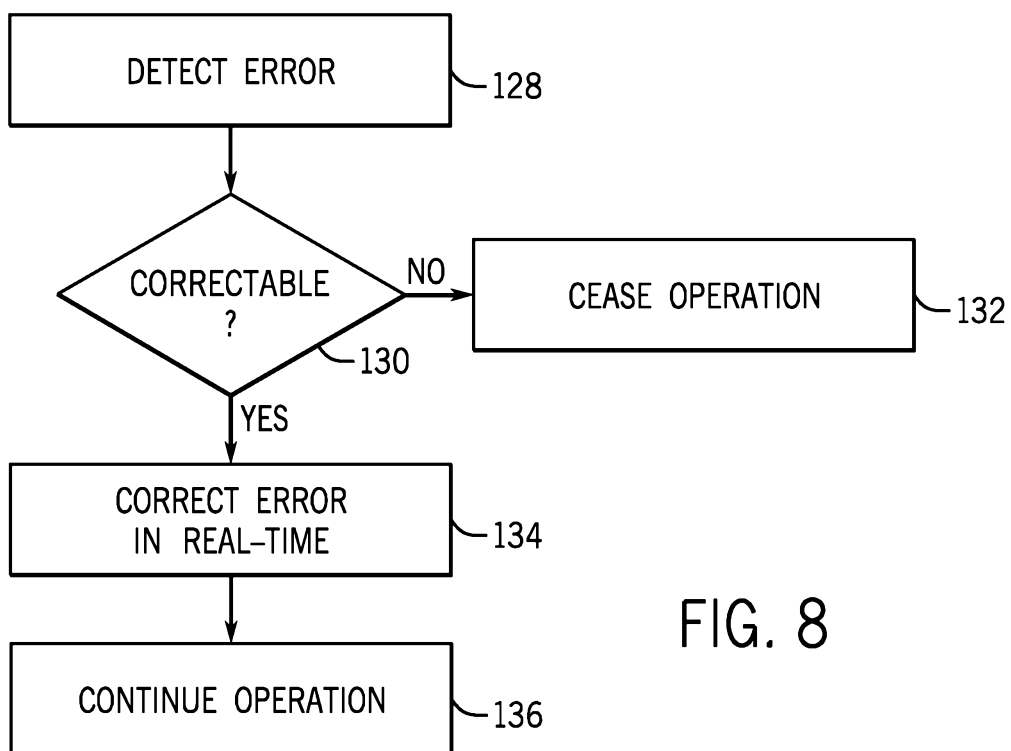


FIG. 8

TRANSMISSION SYSTEM ERROR DETECTION AND CORRECTION SYSTEM AND METHOD

[0001] This application claims priority from United States Application No. 14/145,636 filed on 31 December 2013, the contents of which are to be taken as incorporated herein by this reference.

BACKGROUND OF THE INVENTION

[0002] The subject matter disclosed herein relates to transmission systems, and more specifically to a hardware serial link fault detection system for a data transmission system.

[0003] Generally, data transmission systems may be utilized to transfer data from one input/output (I/O) device to another. For example, a transmission system may be utilized in industrial control systems to facilitate the communication between various industrial controllers, which interface with and control a variety of devices such as turbines, generators, compressors, combustors, and the like. More specifically, the transmission system may enable controllers to communicate data, such as control commands, with one another. For example, a first controller may transmit a control command to a second controller to start operation of a turbine via the transmission system. Accordingly, it would be beneficial to improve the reliability of transmission systems.

[0004] The discussion of the background to the invention included herein including reference to documents, acts, materials, devices, articles and the like is included to explain the context of the present invention. This is not to be taken as an admission or a suggestion that any of the material referred to was published, known or part of the common general knowledge in Australia or in any other country as at the priority date of any of the claims.

BRIEF DESCRIPTION OF THE INVENTION

[0005] Certain aspects commensurate in scope with the claimed invention are summarized below. These aspects are not intended to limit the scope of the claimed invention, but rather these aspects are intended only to provide a brief summary of possible forms of the invention. Indeed, the invention may encompass a variety of forms that may be similar to or different from the aspects set forth below.

[0006] A first aspect of the present invention provides a method for real time error correction in a data transmission system that includes receiving a first communication signal on a first programmable logic device from a second programmable logic device via a first serial link, receiving a second communication signal on the first programmable logic device from the second programmable logic device via a second serial link, in which the first serial link and the second serial link are disposed in parallel with each other and are configured to communicate over a single conduit, and the first communication signal and the second communication signal are representative of identical information. The method further includes detecting an error in data in the first serial link, the second serial link or both by comparing the first communication signal and the second communication signal to determine a difference, wherein when the error is detected, correcting the error in real time in the first programmable logic device based on the comparison of the first communication signal and the second communication signal; and wherein the comparison is comparing the first and second communication signals for bit by bit agreement.

[0007] A second aspect of the present invention provides a data transmission system including a programmable logic device. The programmable logic device includes a first serial port configured to be communicatively coupled to a first serial link, in which the first serial port receives a first transmission signal from the first serial link, and a second serial port configured to be communicatively coupled to a second serial link, in which the second serial port receives a second transmission signal from the second serial link. The first serial link and the second serial link are disposed in parallel with each other and are configured to communicate over a single conduit, and the first communication signal and the second communication signal are

representative of identical information. The programmable logic device further includes circuitry to detect and error in the first serial link, the second serial link, or both by comparing the first communication signal and the second communication signal to determine a difference, wherein when the error is detected, correct the error in real time based on the comparison of the first communication signal and the second communication signal; and wherein the comparison is selected from comparing the first and second communication signals for bit by bit agreement.

[0008] [Intentionally left blank]

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] These and other features and advantages of the present invention will become better understood when the following detailed description is read with reference to the accompanying drawings in which like characters represent like parts throughout the drawings, wherein:

[0010] FIG. 1 is a schematic diagram of a data transmission system, in accordance with an embodiment;

[0011] FIG. 2 is a block diagram of controllers in the data transmission system depicted in FIG. 1, in accordance with an embodiment;

[0012] FIG. 3 is a block diagram of a triple modular redundancy controller, in accordance with an embodiment;

[0013] FIG. 4 is a block diagram of a communication signal, in accordance with an embodiment;

[0014] FIG. 5 is a flowchart depicting a process to transmit communication signals to a data transmission system, in accordance with an embodiment;

[0015] FIG. 6 is a flowchart depicting a process to receive communication signals from a data transmission system, in accordance with an embodiment;

[0016] FIG. 7 is a flowchart depicting a process to detect a fault in a serial link, in accordance with an embodiment; and

[0017] FIG. 8 is a flowchart depicting a process to detect and correct an error in received communication signals, in accordance with an embodiment.

DETAILED DESCRIPTION OF THE INVENTION

[0018] One or more specific embodiments of the present invention will be described below. In an effort to provide a concise description of these embodiments, all features of an actual implementation may not be described in the specification. It should be appreciated that in the development of any such actual implementation, as in any engineering or design project, numerous implementation-specific decisions must be made to achieve the developers' specific goals, such as compliance with system-related and business-related constraints, which may vary from one implementation to another. Moreover, it should be appreciated that such a development effort might be complex and time consuming, but would nevertheless be a routine undertaking of design, fabrication, and manufacture for those of ordinary skill having the benefit of this disclosure.

[0019] When introducing elements of various embodiments of the present invention, the articles "a," "an," "the," and "said" are intended to mean that there are one or more of the elements. The terms "comprising," "including," and "having" are

intended to be inclusive and mean that there may be additional elements other than the listed elements.

[0020] Data transmission systems are often utilized in a system, such as industrial control systems and automation systems, to facilitate communication between various input/output (I/O) devices within the system. For example, a data transmission system may enable various controllers to communicate data with each other. Generally, controllers interface with and control a variety of devices such as turbines, generators, compressors, combustors, sensors, pumps, valves, other field devices and the like. Accordingly, the data communicated by a controller may include data gathered by controlled devices, control commands, device statuses, and the like. Due to the nature of systems that may utilize data transmission systems, it would be beneficial to improve the reliability of such data transmission systems.

[0021] One technique to improve the reliability of data transmission systems may include utilizing various communication protocols such as Object Linking and Embedding for Process Control (OPC), Modbus, Foundation Fieldbus, Process Field Bus, and/or BACnet. Generally, these communication protocols may use some software oversight and processing. For example, when utilizing Foundation Fieldbus, a certain level of processing may be utilized to execute function blocks for transmitting and receiving data. However, because some I/O devices may not include the desired processing capabilities for such protocols, the present disclosure includes embodiments to improve reliability of transmission systems for I/O devices regardless of processing capabilities. Accordingly, the techniques described herein provide for a low level serial link for communications in various applications, including safety system applications. The low level serial link may use less processing and resources (e.g., software and hardware resources) yet provide suitable operations to meet desired safety integrity level (SIL) certifications, including SIL 1, 2, 3, and/or 4. In certain embodiments, the low level serial link may apply a hardware fault detection scheme to provide the desired safety certification, suitable for reducing the use of processing and other resources.

[0022] Accordingly, a first embodiment describes a fault tolerant transmission system that includes a programmable logic device. The programmable logic device includes a first serial port communicatively coupled to a first serial link, in which the first serial port receives a first transmission signal from the first serial link, and a second serial port communicatively coupled to a second serial link, in which the second serial port receives a second transmission signal from the second serial link. The first serial link and the second serial link are disposed in parallel with each other and communicate over a first single conduit, and the first communication signal and the second communication signal are representative of identical information. The programmable logic device further includes circuitry that detects a fault in the first serial link, the second serial link, or both by comparing the first communication signal and the second, and if a fault is detected, determines whether the transmission system can continue operation despite the fault. In other words, receiving parallel communication signals representative of the same information enables the detection of faults in a communication link (e.g., serial link) between various devices. Moreover, the transmission system may be more fault tolerant by intelligently reacting to detected faults.

[0023] Additionally, a second embodiment describes a data transmission system including a programmable logic device. The programmable logic device includes a first serial port communicatively coupled to a first serial link, in which the first serial port receives a first transmission signal from the first serial link, and a second serial port communicatively coupled to a second serial link, in which the second serial port receives a second transmission signal from the second serial link. The first serial link and the second serial link are disposed in parallel with each other and communicate over a single conduit, and the first communication signal and the second communication signal are representative of identical information. The programmable logic device further includes circuitry to compare the first communication signal and the second communication signal to detect an error in a data included in the first communication signal, the second communication signal, or both, and when the error is detected, correct the error in real time based at least in part on the comparison of the first communication signal and the second communication signal. In other words,

receiving parallel communication signals representative of the same information enables the detection and real-time correction of the errors in the communication signals. As used herein, “real-time” is intended to mean immediately following the preceding step. For example, error correction may begin as soon as an error is detected.

[0024] As will be described in more detail below, in some embodiments, the data transmission system may utilize hardware with little or no software oversight to improve the reliability of the transmission system. Accordingly, improved reliability of the data transmission system may be provided to I/O devices having processors and those without. Furthermore, the techniques described herein enable I/O devices to comply with International Electrotechnical Commission (IEC) 61508, and in some embodiments, meet SIL 1, 2, 3, and/or 4 certifications.

[0025] By way of introduction, FIG. 1 depicts an embodiment of a data transmission system 10 disposed in an industrial control system 12. The industrial control system 12 includes multiple controllers 14 that each may interface with and control various devices in the industrial control system 12. As depicted, a first controller is coupled to and controls a turbine 16 (e.g., gas turbine, steam turbine, hydroturbine) and is communicatively coupled to a sensor 18, and a second controller is coupled to and controls a valve 20 and a pump 22. In other embodiments, the controllers 14 may be coupled to and control other devices in the industrial control system 12, such as combustors, generators, and other turbo machinery.

[0026] Furthermore, as depicted, the controllers 14 may communicate with one another via the data transmission system 10. In some embodiments of an industrial control system 12, the data transmission system 10 may include a high speed Ethernet (HSE) network or an H1 network. However, as will be described in more detail below, the techniques described herein may utilize any serial point-to-point architecture. By utilizing the data transmission system 10, the controllers 14 may communicate various types of data, such as a status of connected devices, measurements collected, and control commands. For example, the first controller may transmit measurements collected by the sensor 18 to the second controller via the

data transmission system 10. In response, the second controller may transmit a control command instructing the first controller to modify operation of the turbine 16. Accordingly, to facilitate communication, control of connected devices, and other various functions of the controller, the controllers 14 include one or more processors 24 and memory 26, which may store non-transitory machine-readable instructions executable by the one or more processors 24.

[0027] A detailed view of an embodiment of the data transmission system 10 is depicted in FIG. 2. As depicted, each controller 14 includes a data transmission subsystem 15. In some embodiment, the data transmission subsystem 15 may be a programmable logic device, such as a field programmable gate array, a programmable array logic, a programmable logic array, a generic array logic, or any combination thereof. In other embodiments, the subsystem 15 may be an application specific integrated circuit. Additionally, the data transmission subsystem 15 may be included as a hardware subsystem of each controller 14 (e.g., as a hot swappable card). One example of the controller 14 that may include the subsystem 15 is a Mark VIe Control System made available by General Electric Company of Schenectady, New York.

[0028] In the depicted embodiment, the functions performed by the data transmission subsystem 15 are represented as functional blocks, which may be performed by the controller processor 24 and/or memory 26. As depicted, the subsystem 15 includes a clock system (functional block 27), a free running counter (FRC) (functional block 28), a data I/O (functional block 30), a data comparison/replication (functional block 32), and parallel I/O ports (functional blocks 34 and 36) that couple the controllers 14 to the data transmission system 10. As will be described in more detail below, the FRC 28 may generate timing data to enable time synchronization of communication signals; the data I/O 30 may generate communication signals; and the data comparison/replication 32 may replicate communication signals, detect faults in a communication link (e.g., serial link), and detect/correct errors in received communication signals.

[0029] Furthermore, as depicted, the transmission system 10 includes a single conduit (e.g., cable) 38 and parallel serial transceivers 40 and 42 coupled to either end

of the conduit 38. More specifically, the serial transceivers 40 and 42 interface with the I/O ports 34 and 36 to enable the controllers 14 to communicate with other controllers 14 via the subsystem 15. The serial transceivers 40 and 42 may be RS-232 transceivers, RS-485 transceivers, differential bus transceivers, single ended transceivers, or other low level serial transceivers. In some embodiments, only one cable (e.g., Cat 5, Cat 5e, or Cat 6) is used to connect the two subsystems 15, thus minimizing conduit and related equipment. In other embodiments, multiple cables may be used. In some embodiments, the serial transceivers 40 and 42 may transmit communication signals at frequencies lower than radio frequencies to minimize the amount of noise in the signals caused by radio frequencies. For example, the serial transceivers 40 and 42 may communicate between 1-20 MHz, 20-30MHz, 30-40 MHz, 40-50 MHz, 50-60, 60-100 MHz, or any combination thereof to enable data transmission between 1-20 Mbits/second, 20-30 Mbits/second, 30-40 Mbits/second, 40-50 Mbits/second, 50-60 Mbits/second, 60-100 Mbits/second or any combination thereof.

[0030] Although not explicitly depicted, the single conduit 38 includes a first serial link that couples serial transceivers 40 and a second serial link that couples serial transceivers 42 in parallel. More specifically, the first and second serial link may enable the parallel transmission of communication signals between the controllers 14. For example, a first communication signal may be transmitted via the first serial link and a second communication signal may be transmitted via the second serial link in parallel and at substantially the same time. As will be described in more detail below, the first communication signal and the second communication may be representative of the same information, which facilitates the detection of faults in the serial links and the detection/correction of errors in the communication signals. In some embodiments, a suitable conduit 38 may be a Category 5 (Cat5) cable because it includes eight serial wires, which enables four wires to be used for the first serial link and four wires to be used for the second serial link. In other embodiments, the first serial link and the second serial link may be included on separate conduits. Moreover, as depicted, the single conduit 38 couples one controller to another, thus setting up a serial point-to-point architecture. As used herein, “point-to-point architecture” is

intended to describe when one device is directly coupled another device. In other words, in some embodiments, the controllers 14 may communicate exclusively via the conduit 38, which enables both controllers 14 to predetermine (e.g., anticipate or expect) identity of the other controller 14, the transmission time of a communication signal, the sequence of communication signals, or any combination thereof. Such an architecture may be suitable for communication between devices located 1-10 meters apart, 10-20 meters apart, 20-30 meters apart, 30-40 meters apart, 40-100 meters apart, or any combination thereof.

[0031] In some embodiments, the controllers 14 may be triple modular redundancy (TMR) controllers 14 as depicted in FIG. 3. Generally, TMR controllers 14 may improve reliability by utilizing three cores. More specifically, as depicted, TMR controllers 14 include an R core 46, an S core 48, and a T core 50 that each functions substantially the same. Thus, the TMR controllers 14 operate through a voting scheme to reduce the effect of a malfunctioning core (e.g., 46, 48, or 50).

[0032] As depicted, the R cores 46 in the TMR controller 14 are coupled by a first transmission system 52, the S cores 48 are coupled by a second transmission system 54, and the T cores are coupled by a third transmission system 56. More specifically, in some embodiments, each of the transmission systems (e.g., 52, 54, and 56) may function similar to the data transmission system 10 described above, for example, by including data transmission subsystem 15 and operating substantially independent from one another. More specifically, each transmission system (e.g., 52, 54, and 56) may include a first and a second serial link (or any other component of the subsystem 15) to enable parallel transmission of communication signals between each pair of cores. In other words, the first transmission system 52 may transmit parallel communication signals (e.g., two or more identical data transmissions) between the R cores 46, the second transmission system 54 may transmit parallel communication signals between the S cores 48, and the third transmission system 56 may transmit parallel communication signals between the T cores 50. Additionally, as described above, the first and second serial links may be included in a single or two separate conduits. Accordingly, in the embodiment depicted in FIG. 3, three separate conduits

or six separate conduits may be utilized. Alternatively, all six of the serial links may be included in a single conduit.

[0033] An embodiment of a communication signal 58 that may be transmitted between the controllers 14 via the data transmission system 10 is depicted in FIG. 4. As depicted, the communication signal 58 includes a communication signal data 60, a sequence number 62, a timing data 64, a sender identification 66, and error check data 68. The communication signal data 60 may include the data intended to be communicated between the controllers 14. For example, the communication signal data 60 may include control commands, login data, authentication data, voting data, measurement data, device statuses, and the like.

[0034] The sequence number 62 may represent the order the communication signal 58 was transmitted. In some embodiments, the sequence number 62 may be included in the header of the communication signal 58. Accordingly, the sequence number 58 may enable a receiving controller 14 to determine the order communication signals 58 were transmitted.

[0035] As described above, the timing data 64 may be generated by the FRC 28 to enable time synchronization of the communication signal 58. For example, in some embodiments, the timing data 64 may be a unique character representative of the count in the FRC 28. Thus, if the FRC 28 in the transmitting and receiving controllers 14 are synchronized, the timing data 64 enables the receiving controller 14 to determine when the communication signal 58 was transmitted.

[0036] The sender identification 66 may identify the transmitting controller 14. For example, in some embodiments, the sender identification may be a media access control (MAC) address, a proprietary unique module identification code, or other similar unique identifier. In some embodiments, the proprietary unique module identification code may include a unique serial number containing a date code. Accordingly, the sender identification 66 may enable the receiving controller 14 to determine the identity of the transmitting controller 14.

[0037] The error check data 68 may include data that facilitates error check techniques, such as a cyclic redundancy check, a checksum, a cryptographic hash function, or the like. For example, when a cyclic redundancy check is utilized, the error check data 68 may include a remainder of a polynomial division of the communication signal 58. In operation, the transmitting controller 14 may perform the error check technique and encode the error check data 68 (e.g., checksum) on the communication signal 58. Accordingly, the receiving controller 14 may detect errors/faults by repeating the error check technique and matching the result with the error check data 68 (e.g., calculating the checksum anew and verifying the calculation with the transmitted checksum).

[0038] As described above, the transmission system 10 enables communication signals 58 to be transmitted between controllers 14. FIG. 5 depicts one embodiment of a process 70 that may be utilized by a transmitting controller 14, and more specifically the data transmission subsystem 15 (e.g., programmable logic device), to transmit parallel communication signals. The process 70 may be implemented by non-transitory machine-readable instructions or code stored in the memory 26 and executed by the processor 24. In the depicted embodiment, the process 70 begins by generating the first communication signal (bracket 72), which includes generating the communication signal data (process block 74), encoding a sequence number (process block 76), encoding timing data (process block 78), encoding a sender identification (process block 80), and encoding error data (process block 82). The process 70 continues by generating the second communication signal (process block 84), and transmitting the first communication signal on the first serial link (process block 86) and the second communication signal on the second serial link (process block 88) in parallel.

[0039] More specifically, the first communication signal may be generated by the data I/O 30. To generate the communication signal data 60 (process block 74), the data I/O 30 may gather the data to be transmitted on the first communication signal. For example, when a device status is to be transmitted, the data I/O 30 may poll a connected device (e.g., 16, 20 or 22) for its status or retrieve measurements from sensors (e.g., 18) that monitor the device. To encode the sequence number 62

(process block 76), the data I/O 30 may include a sequential counter that increments with each communication signal transmitted. Thus, the count from the counter may be included as the sequence number 62 to represent the order a communication signal 58 is transmitted. To encode the sender identification 66 (process block 80), the data I/O 30 may retrieve a stored unique identifier identifying the transmitting controller from the transmitting controller's memory 26.

[0040] To encode the timing data 64 (process block 78), the data I/O 30 may receive the timing data 64 from the FRC 28. More specifically, the FRC 28 may generate the timing data 64 based on a timing signal received from the system clock 27. For example, the FRC 28 may count up based on the timing signal and reset the count once it reaches a particular count. Based on the count when the communication signal 58 is to be transmitted, the FRC 28 may output a unique character that represents the count. Thus, when a communication signal 58 is received, the receiving controller may compare the unique character to the count in its own FRC 28 to determine when the communication signal 58 was transmitted.

[0041] To encode the error check data 68 (process block 82), the data I/O 30 may perform a data checking method, such as a cyclic redundancy check, a checksum, a cryptographic hash function, or the like. For example, when a cyclic redundancy check is utilized, the data I/O 30 may perform a polynomial division on the communication signal 58, or more specifically the communication signal data 60, and encode the remainder of the division as the error check data 68. Thus, the receiving controller 14 may detect errors by repeating the error check technique and matching the result with the error check data 68. In other embodiments, encoding the sequence number 62 (process block 76), encoding the timing data 64 (process block 76), encoding the sender identification 66 (process block 76), and encoding the error check data 68 (process block 76) may be performed in any order.

[0042] The first communication signal may then be passed from the data I/O 30 to the data comparison/replication 32 to generate the second communication signal (process block 84). More specifically, the data comparison/replication 32 may generate the second communication signal to represent the same information as the

first communication signal, for example, by copying the first communication signal. Additionally or alternatively, to generate the second communication signal, the data comparison/replication 32 may invert the first communication signal (e.g., change a “1” to a “0” and vice versa) or reverse the first communication signal (e.g., put most significant bit of the first communication signal as the least significant bit of the second communication signal and so on). For example, if the first communication signal is “10011100” a reversed second communication signal may be “00111001” and an inverted second signal may be “01100011.”

[0043] Reversing and/or inverting the first communication signal may facilitate detecting faults in the serial links and/or detecting/correcting errors in the data transmitted. For example, when the communication signals are inverted, an error in one of the communication signals may be detected if a bit in the first and second communication signals are both “high” or both “low.” Additionally, when the second communication signal is reversed, an error may be detected if the most significant bit of the first communication signal does not match the least significant bit of the second communication signal. Furthermore, the error may be corrected by replacing an erroneous bit on the first communication signal with a non-erroneous bit from the second communication signal.

[0044] The first communication signal may be passed from the data comparison/replication 32 to the first I/O port 34 to be transmitted via the first serial link (process block 86), and the second communication signal may be passed from the data comparison/replication 32 to the second I/O port 36 via the second serial link (process block 88). As will be described in more detail below, to facilitate serial link fault detection and data error detection/correction, the first communication signal and the second communication signal may be transmitted in parallel and substantially simultaneously.

[0045] On the receiving side of the transmission system 10, the transmitted communication signal 58 is received by the receiving controller 14. FIG. 6 depicts an embodiment of a process 90 that may be utilized by the receiving controller 14, and more specifically the data transmission subsystem 15 (e.g., programmable logic

device), to receive parallel communication signals 58, for example on a single conduit (e.g., Cat 5 cable). The process 90 may be implemented by non-transitory machine-readable instructions or code stored in the memory 26 and executed by the processor 24. As depicted, the process 90 begins by waiting for a first communication signal (process block 92) and waiting for a second communication signal (process block 94). If the first communication signal is received (process block 96) and the second communication signal is received (process block 98), the first and second communication signals are compared (block 100), which may include comparing the first and second communication signals for bit by bit agreement (process block 102), comparing the encoded sequence numbers (process block 104), comparing the encoded timing data (process block 106), comparing the sender identification (process block 108), and comparing the encoded error check data (process block 110).

[0046] More specifically, the data comparison/replication 32 may wait to receive the first communication signal via the first I/O port 34 (process block 92), and to receive the second communication signal via the second I/O port 36 (process block 94). In some embodiments, this may include employing a watchdog timeout for regularly scheduled communication signals 58. In other words, the data comparison/replication 32 may wait a predetermined amount of time, such an expected arrival time for a scheduled communication signal 58, and if the communication signal 58 is not received within that amount of time, the data comparison/replication signal 58 may output a timeout signal. In some embodiments, the expected arrival time may be predetermined by the receiving controller 14 because utilizing a point-to-point architecture, as described above, may enable the controller 14 to look at the actual arrival time of previous communication signals 58. Accordingly, as depicted in FIG. 2, the data comparison/replication 32 is coupled to the clock system 27 to receive a timing signal. In some embodiments, a communication signal 58 is expected every 1-5 milliseconds, 5-8 milliseconds, 8-10 milliseconds, 10-12 milliseconds, 12-15 milliseconds, 15-30 milliseconds, or any combination thereof. Accordingly, the output of a timeout signal may indicate that a communication signal 58 has been deleted or delayed.

[0047] Once both the first and the second communication signal are received, the data comparison/replication 32 may begin to compare the first and second communication signals (bracket 100), which may facilitate the detection of faults in the serial links and detection/correction of errors in the transmitted data. More specifically, to compare the first communication signal and the second communication signal for bit by bit agreement (process block 102), the data comparison/replication 32 may invert or reverse the second communication signal before comparing it with the first communication signal because, as described above, the second communication signal may be a reversed or inverted version of the first communication signal. Accordingly, the bit by bit comparison may indicate that a bit in the communication signal 58 has been deleted, inserted, or corrupted. Additionally, as will be described in more detail below, the bit by bit comparison may facilitate real-time correction of a bit in the communication signal 58.

[0048] To compare the encoded sequence number 62 (process block 104), the data comparison/replication 32 may retrieve the sequence number 62 from the header of the communication signal 58, which enables the controller 14 to determine the order the communication signal 58 was transmitted. More specifically, the data comparison/replication 32 may compare the sequence number 62 encoded on one or both of the first and second communication signals to an expected sequence number 62. For example, a repetition error may be detected if the sequence numbers 62 encoded on the communication signals 58 are the same as a previous pair of communication signals. Additionally, a deletion, insertion or resequence error may be detected if the sequence numbers 62 encoded on the communication signals 58 are different from the expected sequence number. In some embodiments, utilizing a point-to-point serial architecture, as described above, may enable the receiving controller 14 to maintain the expected sequence number because, if there are no errors, the sequence number 62 encoded on each communication signal 58 received will have a predetermine relationship with the sequence number 62 encoded on a previously received communication signal 58. Accordingly, comparing the sequence number 62 may indicate that one or both of the communication signals 58 is repeated, deleted, inserted, or resequenced.

[0049] To compare the encoded timing data 64 (process block 106), the data comparison/replication 32 may time synchronize one or both of the first communication signal and the second communication signal. In some embodiments, this may include determining the time the communication signals 58 were transmitted by matching the timing data 64 encoded on the communication signals 58 with the stored timing data 64 in memory 26 generated by the FRC 28 on the receiving controller 14. For example, the FRC 28 on both the transmitting and receiving controller may generate a unique character “X” to indicate 3:00 pm. Additionally, a repetition error may be detected if the timing data 64 encoded on the communication signals 58 is the same as the timing data 64 encoded on a previously received communication signal 58. Furthermore, a resequence or delay error may be detected if the timing data encoded on the communication signals 58 is earlier than the timing data 64 encoded on a previously received communication signal 58. Thus, comparing the encoded timing data 64 may indicate that one or both of the communication signals 58 is repeated, resequenced, or delayed.

[0050] To compare the sender identification 66 (process block 108), the data comparison/replication 32 may determine the identity of the transmitting controller 14. More specifically, this may include matching the sender identification 66 encoded on one or both of the first communication signal and the second communication signal with a sender identification 66 stored in memory 26. For example, the controller memory 26 may store a list of sender identifications 66 that correspond to devices that the controller expects to receive communication signals 58 from. In some embodiments, utilizing a point-to-point serial architecture, as described above, may enable the receiving controller 14 to predetermine what devices the controller 14 is coupled to and store the sender identifications associated with those devices. Accordingly, comparing the encoded sender identification 66 may indicate that one or both of the communication signals 58 is inserted if the encoded sender identification 66 corresponds to a device that the controller 14 does not expect to receive communication signals 58 from.

[0051] To compare the encoded error check data 68 (process block 110), the data comparison/replication 32 may perform an error check technique, such as a cyclic

redundancy check (CRC), on one or both of the first communication signal and the second communication to determine whether the received communication signals 58 are corrupted. More specifically, the data comparison/replication 32 may replicate the error check technique performed in the transmitting controller 14 and compare its results with the encoded error check data 68. Accordingly, comparing the encoded error check data 68 may indicate that one or both of the communication signals 58 is corrupted or otherwise do not match each other.

[0052] In other embodiments, comparing the communication signals bit by bit (process block 102), comparing the encoded sequence numbers 62 (process block 104), comparing the encoded timing data 64 (process block 106), comparing the sender identification 66 (process block 108), and comparing the error check data 68 (process block 110) may be performed in any order. Moreover, in certain embodiments, detection of an insertion, deletion, delay, or any combination thereof may indicate a masquerade. As used herein, a masquerade is intended to describe an unauthenticated intrusion into the transmission system 10. For example, when a communication signal 58 is inserted, deleted, or delayed, this may indicate that an unauthorized device (e.g., controller 14) is accessing the transmission system 10 and altering the communication signals 58.

[0053] As described above, transmitting a first communication signal and a second communication signal representative of substantially identical information in parallel via the first and second serial link respectively enables the detection of faults and/or masquerades in the first serial link, the second serial link, or both. One embodiment of a process 112 that may be utilized to detect faults in the serial links is depicted in FIG. 7. The process 112 may be implemented by non-transitory machine-readable instructions or code stored in the memory 26 and executed by the processor 24. As depicted, the process 112 may begin by waiting for the first and second communication signal (decision block 114). If both communication signals are received, the first and second communication signals may be compared (process block 100). Based on whether a timeout signal is output and/or the comparison of the communication signals, a fault in the serial link may be detected (decision block 116). If no fault in the serial link is detected, the transmission system 10 continues

operation (process block 118). On the other hand, if a fault is detected, it may be determined (decision block 120) whether the transmission system 10 should continue operation despite the detected fault (process block 118) or cease operation (process block 122).

[0054] More specifically, as described above, the data comparison/replication 32 may wait a specified amount of time, such as a communication signal frame, for the communication signals 58 (process block 114). If one or both of the communication signals 58 is not received, the data comparison/replication 32 may output a timeout signal indicating that one or both of the communication signals 58 has timed out. On the other hand, if both signal 58 are received within the specified amount of time, the data comparison/replication 32 may compare the communication signals 58 (process block 100), which may include comparing the first and second communication signals 58 for bit by bit agreement (process block 102), comparing the encoded sequence numbers (process block 104), comparing the encoded timing data (process block 106), comparing the sender identification (process block 108), and comparing the encoded error check data (process block 110). As described above, comparing the communications signals may facilitate the detection of errors in data transmitted in the communication signals.

[0055] Based on the comparison of the communication signals 58 and/or whether a timeout signal was output, the data comparison/replication 32 may determine whether a fault is present in the first serial link, the second serial link, or both (decision block 116). For example, if the first communication signal has timed out, but the second communication signal has not, the data comparison/replication 32 may determine that the first serial link has a fault. Additionally, any error (e.g., repetition, deletion, insertion, resequence, corruption, delay, or masquerade) may indicate a fault in one or more of the serial links. For example, corruption of the first communication signal 58 but not the second communication signal 58 may indicate that the insulation on the first serial link is weakened and susceptible to external interference, such as stray electromagnetic waves.

[0056] Furthermore, in some embodiments, when a fault in a serial link is detected, the status of the serial link may be updated based on the fault. For example, before a fault is detected, the status of the first serial link and the second serial link is set to “good,” but when a fault in the first serial link is detected, the status of the first serial link may be updated to “faulty.” As will be described further below, in other embodiments, the status may include intermediate statuses to enable the transmission system 10 to more judiciously manage operation. For example, a serial link may have a status of “partially faulty” or “potentially faulty” if only a single error is detected or if an error is only detected periodically (e.g., not in every communication signal transmitted). For example, when a first error is detected, the status may be updated from “functional” to “potentially faulty,” and when a second error is detected three communication signals later, the status may again be updated to “faulty.”

[0057] If a fault is detected, the data comparison/replication 32 may more judiciously determine whether to cease or continue operation of the transmission system based on the circumstances associated with the fault (decision block 120). In other words, the transmission system 10 may become more fault tolerant and not immediately shut down each time a fault in the serial links is detected. In some embodiments, the decision whether to continue operation may depend on the status of the serial links and the information included in the communication signal 58. For example, if the first serial link is “functional” but the second serial link is “faulty” and the communication signal 58 merely communicates a temperature measurement, the transmission system 10 may continue operations solely based on the communication signals 58 received via the first serial link. In such an embodiment, the controller 14 may decide to continue transmitting and receiving sensor 18 measurements but may decide to cease transmission/reception of control commands.

[0058] Additionally, other factors influencing the decision whether to continue operation of the transmission system 10 may include the nature of the system utilizing the transmission system 10 and history of the errors in the communication signals. For example, a transmission system 10 in a candy manufacturing plant may be more fault tolerant than a nuclear plant and choose to operate with one “functional” serial link and one “potentially faulty” link, whereas the nuclear plant may choose to operate only

with two “functional” serial links. Additionally, if there has not been a history of errors, the transmission system 10 may attribute the error to an anomaly and continue operation until a subsequent error is detected. In such embodiments, the transmission system 10 may decide to continue operation for a set period to give an operator an opportunity to correct the faulty serial link, but cease operation if the fault has not been correct after the set period.

[0059] Furthermore, the decision whether to continue operation may depend on whether the error is correctable in real-time. For example, if the error is a deletion or a corruption of the most significant bit on both communication signals 58 and the communication signals are reversed, the transmission system 10 may choose to continue operation based on the corrected communication signals 58. The correctability of errors in communication signals 58 will be described in more detail below.

[0060] Based on the above described examples, it should be appreciated that the decision whether to continue or cease operation may be customized based on one or more of a variety of factors, such as the nature of the system utilizing the transmissions system 10, the status the serial links, whether an error in the communication signals 58 is correctable in real-time, the nature of the information transmitted in the communication signals 58, the history of errors, and the like. Functions utilized for such determinations may be performed by the data comparison/replication 32 or other systems of the processor 24 and/or memory 26.

[0061] One embodiment of a process 124 for correcting an error in real time is depicted in FIG. 8. The process 124 may be implemented by non-transitory machine-readable instructions or code stored in the memory 26 and executed by the processor 24. As depicted, when an error is detected (process block 126), a determination is made whether the error is correctable (decision block 128). If the error is uncorrectable, the transmission system 10 may cease operation (process block 132). If it is correctable, the error is corrected in real-time (process block 134) and the transmission system 10 continues operation (process block 136).

[0062] More specifically, the data comparison/replication 32 may determine whether the error is correctable based on various factors, such as the type of error and the magnitude of the error. For example, in the example described above, if the error is merely corrupting the most significant bit of each communication signal 58 and the communication signals 58 are reversed, the data comparison/replication 32 may correct the error by replacing the most significant bit on the first communication signal with the least significant bit on the second communication signal and vice versa. Additionally, if the error detected is a repetition, the transmission system 10 may correct the error simply by executing the earlier receive communication signals and disregarding the subsequently received. Similarly, if the error detected is an insertion, the transmission system may correct the error simply by disregarding the inserted communication signals 58. On the other hand, if the error detected is a masquerade, the error may not be correctable and the transmission system 10 may choose to cease operation because an intrusion into the transmission system 10 has been detected.

[0063] Based on the above described examples, the data comparison/replication 32 may correct errors affecting one or more bits of the first communication signal in real-time by replacing erroneous bits in the first communication signal with non-erroneous bits from the second communication signal. Additionally, the data comparison/replication 32 may correct errors affecting the first communication signal but not the second communication signal by disregarding the first communication signal. Furthermore, the data comparison/replication 32 may correct errors affecting both the first and second communication signal by disregarding both. It should be appreciated that the correctability of a detected error may be fact specific. Functions utilized for such determinations may be performed by the data comparison/replication 32 or other parts of the processor 24 and/or memory 26.

[0064] Moreover, the techniques described herein enable the errors to be corrected in real-time or as soon as the error is detected. For example, in the above described examples, the most significant bit of the first communication signal may be replaced with the least significant bit of the second communication signal when they are compared bit by bit, the earlier received communication signal 58 may be executed

upon receipt and the subsequently received communication signals may be disregarded upon receipt, and the inserted communication signal 58 may be disregarded upon receipt.

[0065] Although the above description describes two parallel serial links, other embodiment may include more than two parallel serial links (e.g., three or four parallel serial links). As can be appreciated, the use of more than two parallel serial links may generally utilize the techniques described above to improve the reliability of data transmission systems 10.

[0066] Technical effects of the described embodiments include increasing the reliability of a data transmission system 10. More specifically, the reliability of the data transmission system 10 may be improved by utilizing parallel serial links that each transmits a communication signal 58 representative of the same information in parallel. Based on receipt and comparison of the parallel communication signals 58, faults in the serial links and errors in the communication signals may be detected. Accordingly, the reliability of the transmission system 10 may be improved by making the transmission system 10 more fault tolerant and correcting errors in real time. Additionally, in some embodiments, the detection and/or correction of faults and errors may be provided only with the use of hardware, which enables the improved reliability to be provided to all devices regardless of processing capabilities.

[0067] This written description uses examples to disclose the invention, including the best mode, and also to enable any person skilled in the art to practice the invention, including making and using any devices or systems and performing any incorporated methods. The patentable scope of the invention is defined by the claims, and may include other examples that occur to those skilled in the art. Such other examples are intended to be within the scope of the claims if they have structural elements that do not differ from the literal language of the claims, or if they include equivalent structural elements with insubstantial differences from the literal language of the claims.

The claims defining the invention are as follows:

1. A data transmission system comprising:
 - a programmable logic device comprising:
 - a first serial port configured to be communicatively coupled to a first serial link, wherein the first serial port is configured to receive a first transmission signal from the first serial link;
 - a second serial port configured to be communicatively coupled to a second serial link, wherein the second serial port is configured to receive a second transmission signal from the second serial link, wherein the first serial link and the second serial link are disposed in parallel with each other and are configured to communicate over a single conduit, and the first communication signal and the second communication signal are representative of identical information; and
 - circuitry configured to:
 - detect an error in the first serial link, the second serial link, or both by comparing the first communication signal and the second communication signal to determine a difference, wherein
 - when the error is detected, correct the error in real time based on the comparison of the first communication signal and the second communication signal; and
 - wherein the comparison is selected from comparing the first and second communication signals for bit by bit agreement;
 - wherein the circuitry is configured to correct the the error in real time by replacing a bit in the first communication signal with a bit from the second communication signal.

2. The data transmission system of claim 1, wherein the circuitry is configured to determine whether the error is correctable based at least in part on a type of error and a magnitude of the error, and to correcting the error in real time is based at least in

part on the comparison of the first communication signal and the second communication signal and the error type.

3. The data transmission system of claim 1 or 2, wherein the circuitry is configured to detect a repetition, a deletion, an insertion, a resequence, a corruption, a delay, a masquerade, or any combination thereof in the first communication signal or the second communication signal.

4. The data transmission system of any one of the preceding claims, wherein the circuitry comprises a field-programmable gate array, a programmable array logic, a programmable logic array, a generic array logic, an application-specific integrated circuit, or any combination thereof.

5. The data transmission system of any one of the preceding claims, wherein the circuitry is configured to:

detect a sequence number encoded on the first communication signal or second communication signal which is different from an expected sequence number stored in the programmable logic device;

detect timing data encoded on the first communication signal or second communication signal which is different from a timing data stored in the programmable logic device;

detect a sender identification encoded on the first communication signal or the second communication signal which is different from an expected sender identification stored in the programmable logic device;

perform an error check technique on the first communication signal or the second communication signal and compare a result from the error check technique with error check data encoded on the first communication signal or the second communication signal; or

any combination thereof.

6. A method for real time error correction in a data transmission system comprising:

receiving a first communication signal on a first programmable logic device from a second programmable logic device via a first serial link;

receiving a second communication signal on the first programmable logic device from the second programmable logic device via a second serial link, wherein the first serial link and the second serial link are disposed in parallel with each other and are configured to communicate over a single conduit, and the first communication signal and the second communication signal are representative of identical information;

detecting an error in data in the first serial link, the second serial link, or both by comparing the first communication signal and the second communication signal to determine a difference;

wherein when the error is detected, correcting the error in real time in the first programmable logic device based on the comparison of the first communication signal and the second communication signal; and

wherein the comparison is comparing the first and second communication signals for bit by bit agreement.

7. The method of claim 6, wherein comparing the first communication signal and the second communication signal comprises determining whether the error is correctable based at least in part on a type of error and a magnitude of the error, and correcting the error in real time is based at least in part on the comparison and the error type.

8. The method of claim 6 or 7, wherein comparing the first communication signal and the second communication signal comprises detecting a repetition, a deletion, an insertion, a resequence, a corruption, a delay, a masquerade, or any combination thereof, in the first communication signal or the second communication signal.

9. The method of any one of claims 6 to 8, wherein checking bit by bit agreement between the first communication signal and the second communication signal comprises inverting or reversing the second communication signal.

10. The method of any one of claims 6 to 9, wherein correcting the error in real time comprises replacing a bit in the first communication signal with a bit from the second communication signal, or disregarding the first communication signal, the second communication signal, or both.

11. The method of any one of claims 6 to 10, wherein comparing the first communication signal and the second communication signal comprises detecting timing data encoded on the first communication signal or the second communication signal which is different from timing data stored in the first programmable logic device.

12. The method of any one of claims 6 to 11, wherein comparing the first communication signal and the second communication signal comprises detecting a sequence number encoded on the first communication signal or the second communication signal which is different from an expected sequence number stored in the first programmable logic device.

13. The method of any one of claims 6 to 12, wherein comparing the first communication signal and the second communication signal comprises detecting a sender identification encoded on the first communication signal or the second communication signal which is different from an expected sender identification stored in the first programmable logic device.

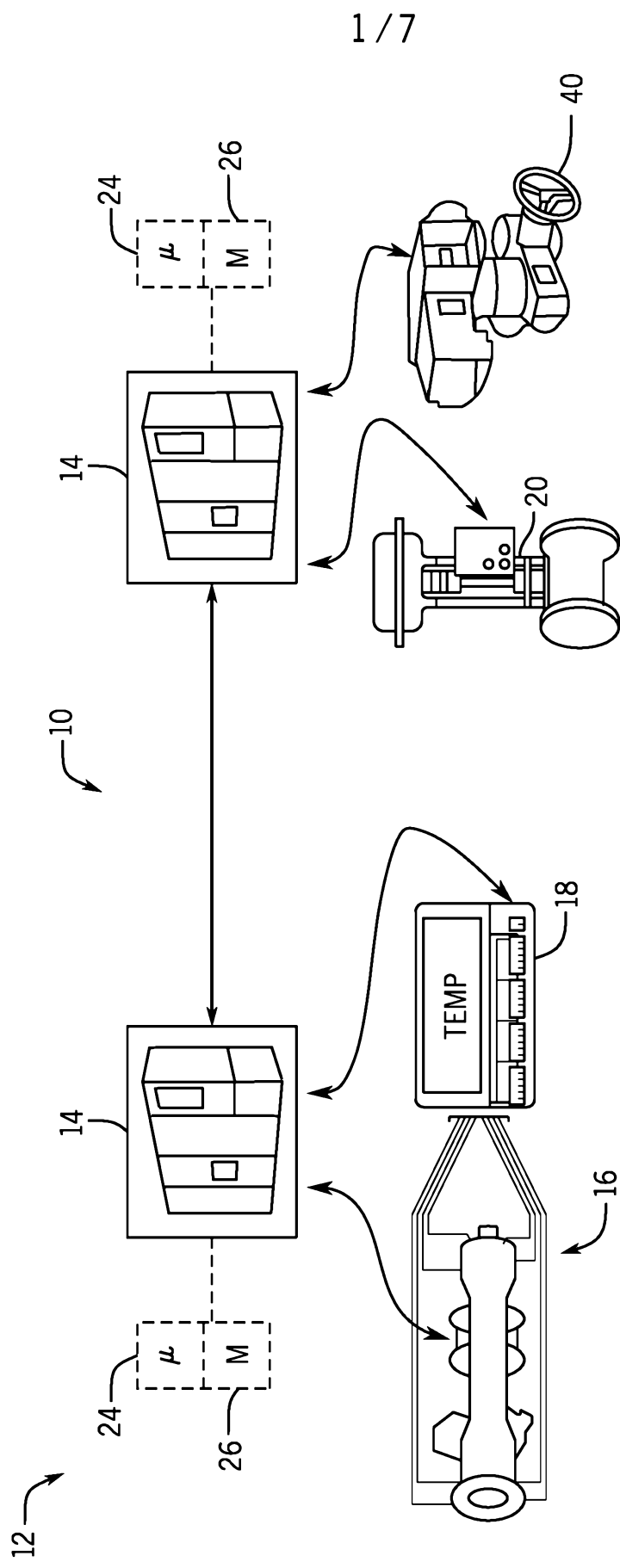


FIG. 1

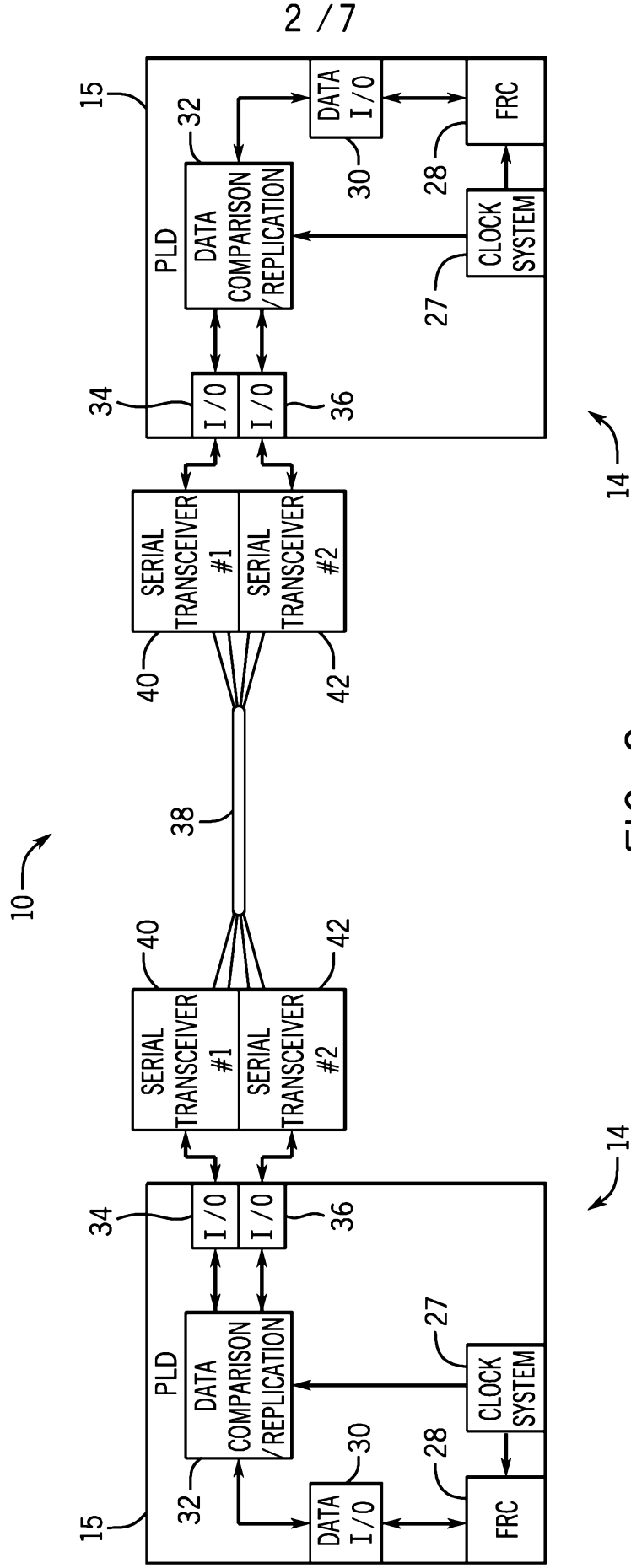


FIG. 2

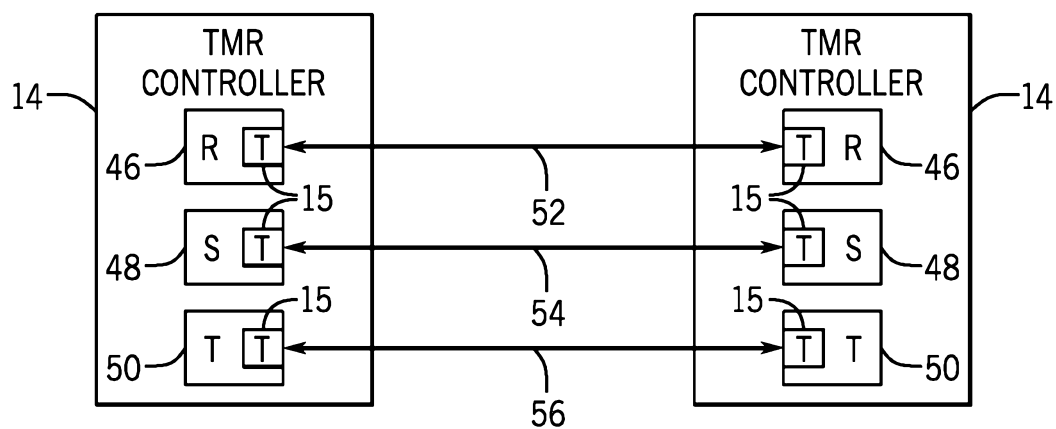


FIG. 3

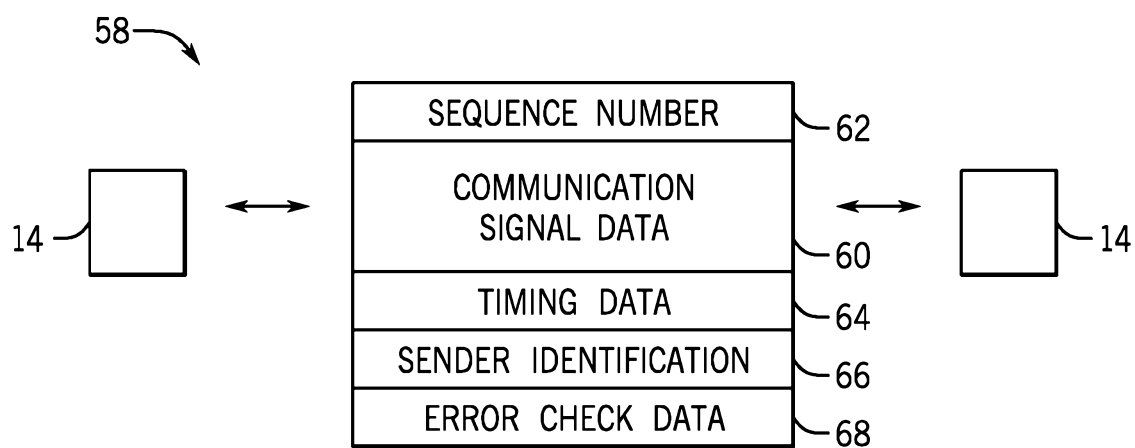


FIG. 4

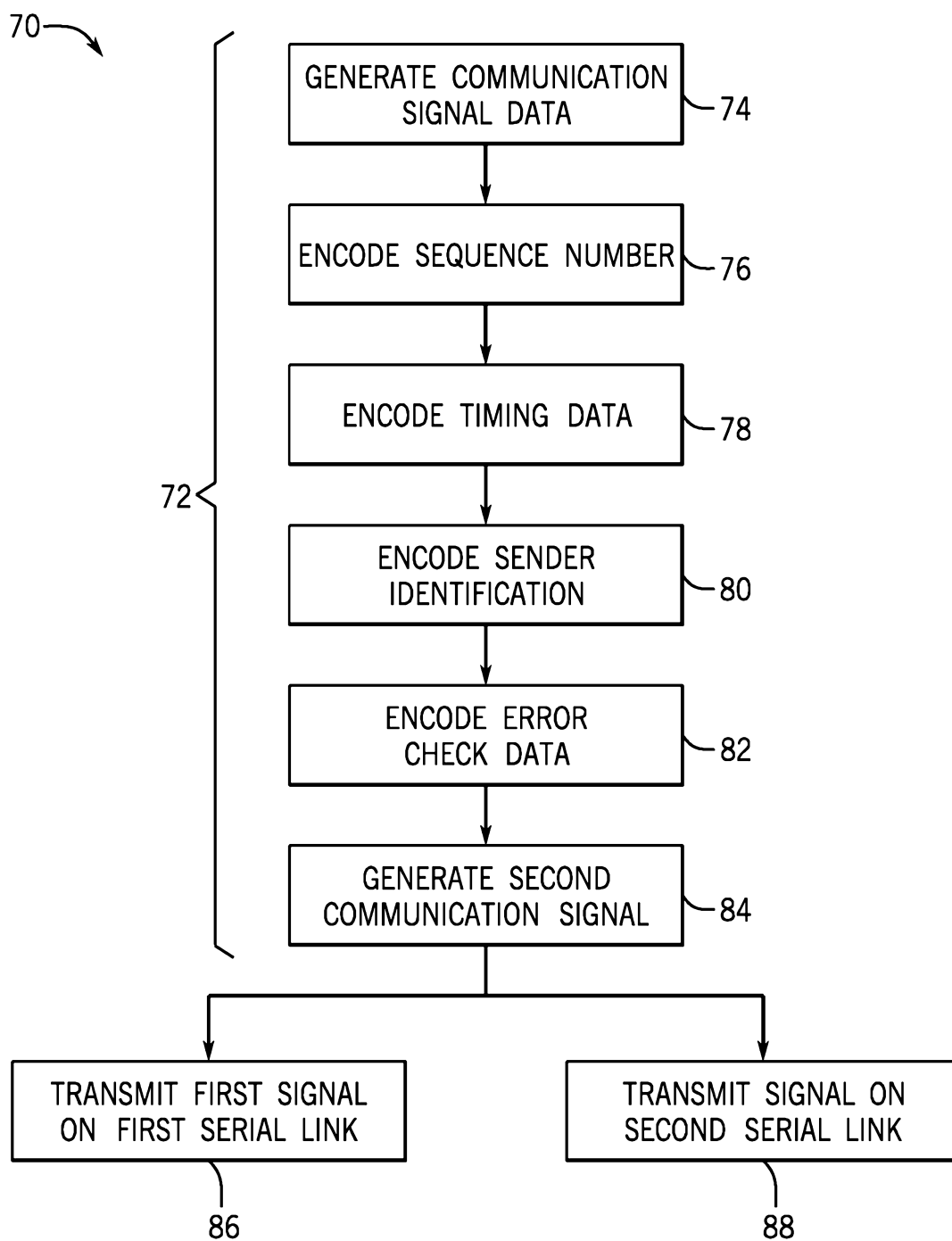


FIG. 5

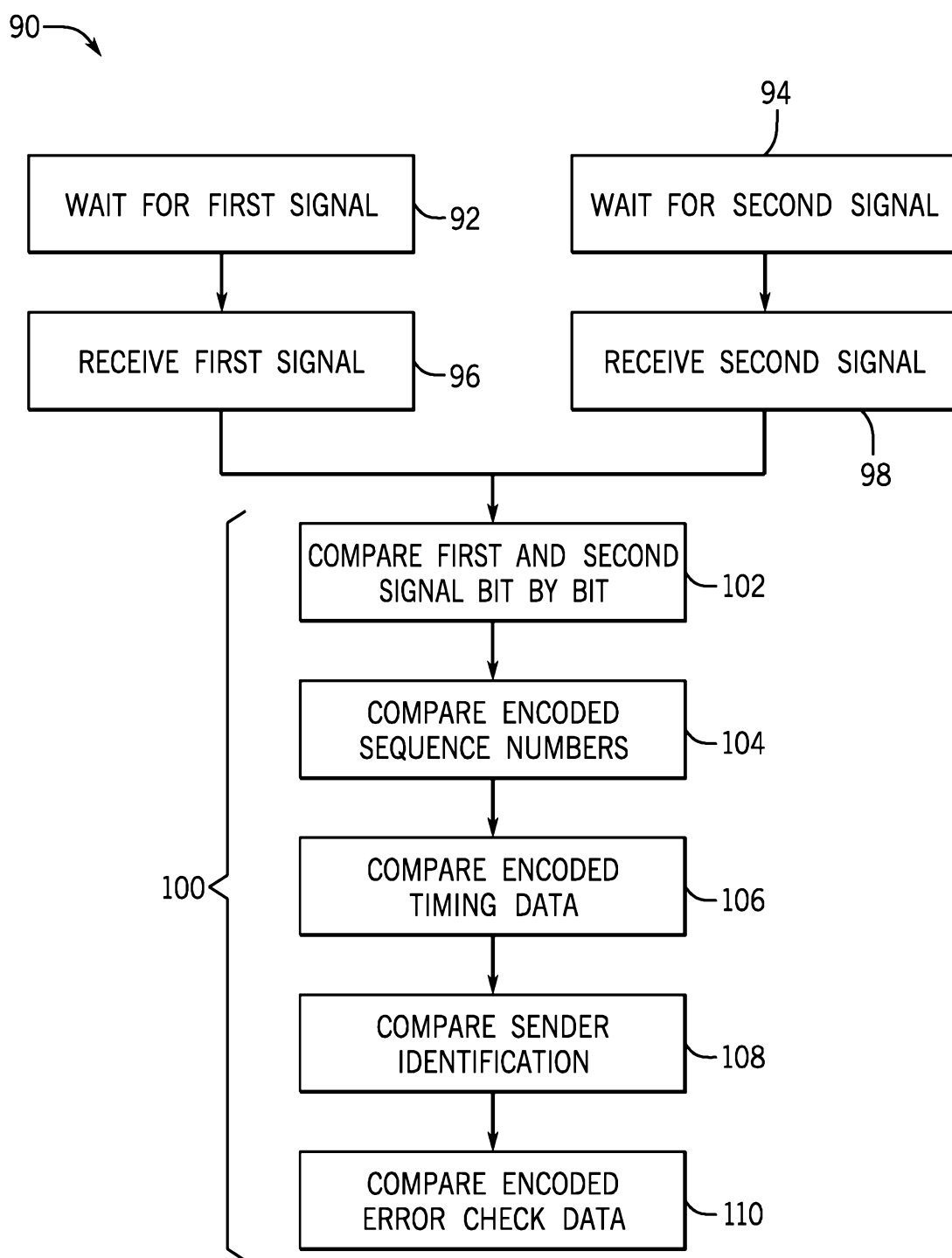


FIG. 6

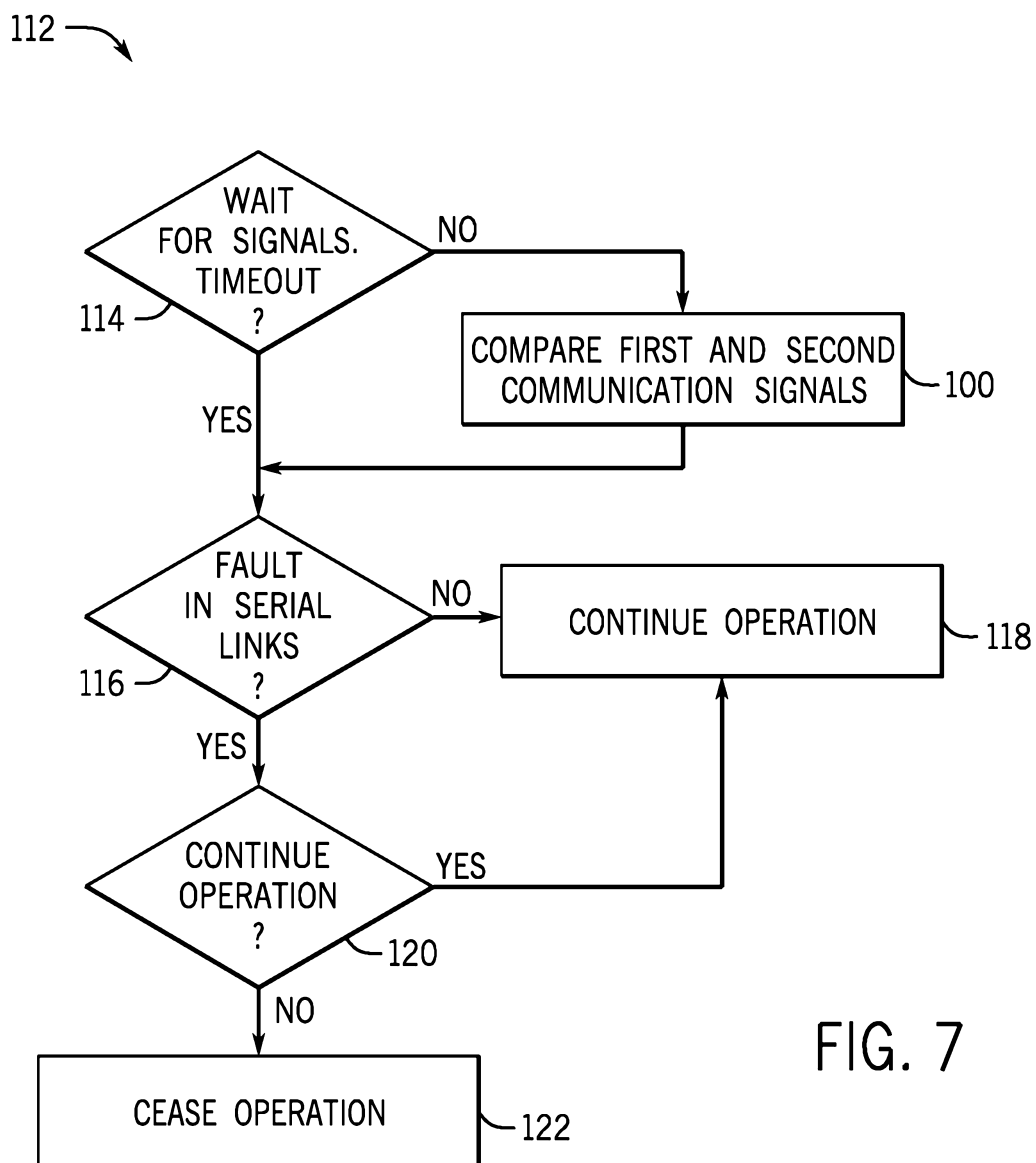


FIG. 7

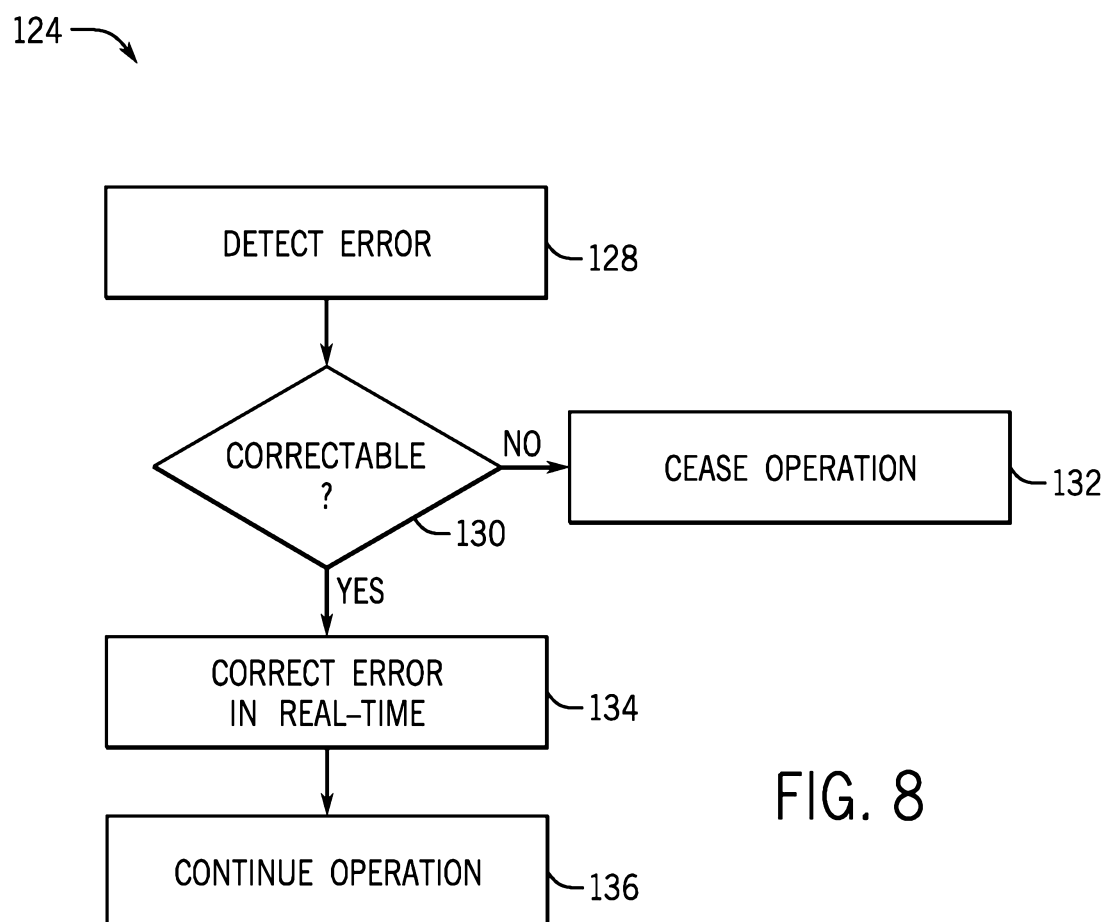


FIG. 8