

DESCRIPTION

INFORMATION PROCESSING APPARATUS AND INFORMATION PROCESSING METHOD

FIELD OF THE INVENTION

[0001]

The present invention relates to an information processing apparatus comprising plural devices for executing processing mutually independently and an information processing method, and particularly to an information processing apparatus having high reliability and an information processing method.

BACKGROUND ART

[0002]

A plant control system for managing and controlling field devices arranged in a plant has been known. Also, a safety system for ensuring safety of the plant is introduced in such a plant. The safety system is a system for taking necessary measures while providing notification of an alarm when abnormality is observed in the field device, and is disposed as a part of the plant control system or independently of the plant control system.

JP-A-2000-347706 is seen as a related art.

DISCLOSURE OF THE INVENTION

PROBLEMS TO BE RESOLVED BY THE INVENTION

[0003]

The safety system requires extremely high reliability from its intended function. For example, a situation in which notification of wrong information is provided or it is recognized that a system is safe though abnormality occurs must be avoided where possible. Also, it is necessary to select processing of the safe side in the case of indicating the possibility of abnormality though the abnormality cannot be recognized obviously.

[0004]

An object of the invention is to provide an information processing apparatus having high reliability, and an information processing method.

MEANS OF SOLVING THE PROBLEMS

[0005]

The invention provides an information processing apparatus comprising: a first device and a second device of which each executes the same processing independently; a sending part which sends a data generated by the second device from the second device to the first device; and a data collating part which collates the data sent from the sending part with a data generated by the first device and determines as

abnormal when these data are in disagreement with each other.

According to this information processing apparatus, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0006]

The information processing apparatus may comprise a processing stop part which stops a processing of the first device when determined as abnormal by the data collating part.

In this case, processing of the first device is stopped in the case of abnormality. A method for stopping the processing is not limited. For example, a signal may be broken at an output stage or the middle of processing or an operation may be stopped by resetting the first device.

[0007]

In the information processing apparatus, the sending part may be disposed in the second device.

[0008]

The invention also provides an information processing apparatus comprising: a first device and a second device of which each executes the same processing independently; a sending part which sends a data generated by the first device from the first device to the second device; and a data collating part which collates the data sent from the sending part with a data generated by the second device and determines as

abnormal when these data are in disagreement with each other.

According to this information processing apparatus, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0009]

The information processing apparatus may comprise a processing stop part which stops a processing of the first device when determined as abnormal by the data collating part.

In this case, processing of the first device is stopped in the case of abnormality. A method for stopping the processing is not limited. For example, a signal may be broken at an output stage or the middle of processing or an operation may be stopped by resetting the first device.

[0010]

In the information processing apparatus, the sending part may be disposed in the first device.

The information processing apparatus may comprise a synchronizing part which synchronizes a processing of the first device and a processing of the second device by use of a timing of sending of the data sent from the sending part.

In this case, processing of the first device and processing of the second device are synchronized, so that data can be collated correctly. Also, processing of the first device and processing of the second device are synchronized using timing of sending of

data, so that waiting time necessary for synchronization can be reduced and a reduction in efficiency is prevented.

[0011]

In the information processing apparatus, the synchronizing part may be disposed in the first device or the second device.

[0012]

The invention also provides an information processing apparatus comprising: a first device and a second device of which each executes the same processing in synchronization independently; an acquiring part which acquires a data generated by the first device and a data generated by the second device in real time; and a data collating part which collates the data generated by the first device with the data generated by the second device acquired by the acquiring part and determines as abnormal when these data are in disagreement with each other.

According to this information processing apparatus, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0013]

The information processing apparatus may comprise a processing stop part which stops a processing of the first device when determined as abnormal by the data collating part.

[0014]

The invention also provides an information processing apparatus comprising: a first device and a second device of which each executes the same processing independently; a first code generating part which generates a first code for error detection based on a data generated by the first device; a second code generating part which generates a second code for error detection based on a data generated by the second device; and a code collating part which collates the first code for error detection generated by the first code generating part with the second code for error detection generated by the second code generating part and determines as abnormal when both the codes are in disagreement with each other.

According to this information processing apparatus, a first code for error detection generated based on data generated by the first device is collated with a second code for error detection generated based on data generated by the second device and when a mismatch between these codes for error detection occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0015]

The information processing apparatus may comprise a processing stop part which stops a processing of the first device when determined as abnormal by the code collating part.

In this case, processing of the first device is stopped in the case of abnormality. A method for stopping the processing is not limited. For example, a signal may be broken at an output stage or the middle of processing or an operation may be stopped by resetting the first device.

[0016]

In the information processing apparatus, the first code generating part may be disposed in the first device and the second code generating part may be disposed in the second device.

[0017]

In the information processing apparatus, the code collating part may be disposed in the first device or the second device.

[0018]

The invention also provides an information processing apparatus comprising: a first device and a second device of which each executes the same processing independently with respect to a data having a code for error detection; a first inspecting part which inspects the presence or absence of abnormality of a data having a first code for error detection received by the first device by use of the first code for error detection; a second inspecting part which inspects the presence or absence of abnormality of a data having a second code for error detection received by the second device by use of the second code for error detection; a code collating part which collates the first code for error detection used in the first inspecting part with the second code for error detection used in the second inspecting part; and a determining part which determines as abnormal when: a result of the inspection by the first inspecting part or a result of the inspection by the second inspecting part indicates abnormality; or both the codes for error detection are in disagreement with each other as a result of the collation by the code collating part.

In this case, it is determined that it is abnormal when a mismatch between both the codes for error detection occurs as a result of collation by the code collating part or when a result of inspection by the first inspecting part or a result of inspection by

the second inspecting part indicates abnormality, so that the abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

In addition, a first inspecting part generates a code for error detection based on data with a first code for error detection received by the first device and the received code for error detection is compared with the generated code for error detection and thereby, the presence or absence of abnormality is inspected. A second inspecting part generates a code for error detection based on data with a second code for error detection received by the second device and the received code for error detection is compared with the generated code for error detection and thereby, the presence or absence of abnormality is inspected.

The "codes for error detection used in the first inspecting part and the second inspecting part" include both of the code for error detection generated by the first inspecting part or the second inspecting part and the code for error detection received by the first inspecting part or the second inspecting part.

[0019]

The information processing apparatus may comprise a processing stop part which stops a processing of the first device when determined as abnormal by the determining part.

In this case, processing of the first device is stopped in the case of abnormality. A method for stopping the processing is not limited. For example, a signal may be broken at an output stage or the middle of processing or an operation may be stopped by resetting the first device.

[0020]

In the information processing apparatus, the first inspecting part may be disposed in the first device and the second inspecting part may be disposed in the second device.

[0021]

In the information processing apparatus, the code collating part may be disposed in the first device or the second device.

[0022]

The invention also provides an information processing apparatus comprising: a first device and a second device of which each executes the same processing independently; a data collating part which collates a data generated by the first device with a data generated by the second device; a counting part which adds a count number to the data generated by the first device; a count stop part which stops the addition of the count number by the counting part when both the data are in disagreement with each other as a result of the collation by the data collating part; and a determining part which determines abnormality based on the count number added by the counting part.

In this case, a determining part determines abnormality of the device based on the count number added by the counting part, so that the abnormality can be detected without referring to data other than the data generated by the first device.

[0023]

The information processing apparatus may comprise a processing stop part which stops a processing of the first device or a processing of the second device when determined as abnormal by the determining part.

In this case, processing of the first device or processing of the second device is stopped in the case of abnormality. A method for stopping the processing is not limited. For example, a signal may be broken at an output stage or the middle of

processing or an operation may be stopped by resetting the first device or the second device.

[0024]

In the information processing apparatus, the first device and the second device may be separate semiconductor devices.

[0025]

In the information processing apparatus, the first device and the second device may be separate CPUs.

[0026]

In the information processing apparatus, the first device and the second device may be mounted in an insulated state each other.

[0027]

The invention also provides an information processing method which uses a first device and a second device of which each executes the same processing independently, comprising: a step of sending a data generated by the second device from the second device to the first device; and a step of collating the data sent from the second device with a data generated by the first device and determining as abnormal when these data are in disagreement with each other.

According to this information processing method, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0028]

The invention also provides an information processing method which uses a first device and a second device of which each executes the same processing independently, comprising: a step of sending a data generated by the first device from the first device to the second device; and a step of collating the data sent from the first device with a data generated by the second device and determining as abnormal when these data are in disagreement with each other.

According to this information processing method, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0029]

The information processing method may comprise a step of synchronizing a processing of the first device and a processing of the second device by use of timing of sending of the data sent from the second device.

In this case, processing of the first device and processing of the second device are synchronized, so that data can be collated correctly. Also, processing of the first device and processing of the second device are synchronized using timing of sending of data, so that waiting time necessary for synchronization can be reduced and a reduction in efficiency is prevented.

[0030]

The invention also provides an information processing method which uses a

first device and a second device of which each executes the same processing in synchronization independently, comprising: a step of acquiring a data generated by the first device and a data generated by the second device in real time; and a step of collating the data generated by the first device with the data generated by the second device and determining as abnormal when these data are in disagreement with each other.

According to this information processing method, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0031]

The invention also provides an information processing method which uses a first device and a second device of which each executes the same processing independently, comprising: a first code generating step of generating a first code for error detection based on a data generated by the first device; a second code generating step of generating a second code for error detection based on a data generated by the second device; and a step of collating the first code for error detection with the second code for error detection and determining as abnormal when both the data are in disagreement with each other.

According to this information processing method, a first code for error detection generated based on data generated by the first device is collated with a second code for error detection generated based on data generated by the second device and

when a mismatch between these codes for error detection occurs, it is determined that it is abnormal, so that abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0032]

The invention also provides an information processing method which uses a first device and a second device of which each executes the same processing independently with respect to a data having a code for error detection, comprising: a first inspection step of inspecting the presence or absence of abnormality of a data having a first code for error detection received by the first device by use of the first code for error detection; a second inspection step of inspecting the presence or absence of abnormality of a data having a second code for error detection received by the second device by use of the second code for error detection; a code collation step of collating the first code for error detection used in the first inspection step with the second code for error detection used in the second inspection step; and a step of determining as abnormal when: a result of the inspection by the first inspection step or a result of the inspection by the second inspection step indicates abnormality; or both the codes for error detection are in disagreement each other as a result of the collation by the code collation step.

In this case, it is determined that it is abnormal when a mismatch between both the codes for error detection occurs as a result of collation by the code collation step or when a result of inspection by the first inspection step or a result of inspection by the second inspection step indicates abnormality, so that the abnormality can be detected surely. Processing in each of the devices is not limited to arithmetic

processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

[0033]

The invention also provides an information processing method which uses a first device and a second device which executes the same processing independently, comprising: a step of collating a data generated by the first device with a data generated by the second device; a step of adding a count number to the data generated by the first device; a step of stopping the addition of the count number when both the data are in disagreement each other as a result of the data collation; and a step of determining abnormality based on the added count number.

In this case, abnormality is determined based on the added count number, so that the abnormality can be detected without referring to data other than the data generated by the first device. Processing in each of the devices is not limited to arithmetic processing, and includes all the processing such as processing for receiving or sending data, processing for outputting data, processing for transferring data or processing for storing data.

ADVANTAGEOUS EFFECTS OF THE INVENTION

[0034]

According to the information processing apparatus and the information processing method, data generated by the first device is collated with data generated by the second device and when a mismatch between these data occurs, it is determined that it is abnormal, so that abnormality can be detected surely.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035]

[Fig. 1] (a) and (b) are block diagrams functionally showing an information processing apparatus according to the invention.

[Fig. 2] (a) and (b) are block diagrams functionally showing an information processing apparatus according to the invention.

[Fig. 3] (a) and (b) are block diagrams functionally showing an information processing apparatus according to the invention.

[Fig. 4] is a block diagram showing a configuration of a safety system to which an information processing apparatus of a first embodiment is applied.

[Fig. 5] is a block diagram showing a part of the configuration of the information processing apparatus of the first embodiment.

[Fig. 6] is a diagram showing sequence of communication processing.

[Fig. 7] (a) and (b) are diagrams showing configurations of communication frames and Fig. 7(a) shows a configuration of individual communication frames and Fig. 7(b) shows an operation of the case where a communication state is normal.

[Fig. 8] is a block diagram showing a configuration of an information processing apparatus of a second embodiment.

[Fig. 9] is a block diagram showing a configuration of an information processing apparatus of a third embodiment.

[Fig. 10] is a block diagram showing a part of a configuration of an information processing apparatus of a fourth embodiment.

DESCRIPTION OF THE REFERENCE NUMERALS AND SIGNS

[0036]

- 101 Sending Part
- 102 Data Collating Part
- 103 Processing Stop Part
- 104 Synchronizing Part
- 105 Acquiring Part
- 106 First Code Generating Part
- 107 Second Code Generating Part
- 108 Code Collating Part
- 109 Processing Stop Part
- 111 First Inspecting Part
- 112 Second Inspecting Part
- 113 Code Collating Part
- 114 Determining Part
- 115 Processing Stop Part
- 116 Data Collating Part
- 117 Counting Part
- 118 Count Stop Part
- 119 Determining Part
- 120 Processing Stop Part

BEST MODE FOR CARRYING OUT THE INVENTION

[0037]

Figs. 1(a) to Fig. 3(b) are block diagrams functionally showing an information processing apparatus according to the invention.

[0038]

In a form shown in Fig. 1(a), a sending part 101 sends data generated by a second device from the second device to a first device. A data collating part 102 collates data sent from the sending part 101 with data generated by the first device, and when a mismatch between these data occurs, it is determined that it is abnormal.

[0039]

A processing stop part 103 stops processing of the first device when the data collating part 102 determines that it is abnormal.

[0040]

A synchronizing part 104 synchronizes processing of the first device and processing of the second device using timing of sending of data sent from the sending part 101.

[0041]

In a form shown in Fig. 1(b), the sending part 101 sends data generated by the first device from the first device to the second device. The data collating part 102 collates data sent from the sending part 101 with data generated by the second device, and when a mismatch between these data occurs, it is determined that it is abnormal.

[0042]

The processing stop part 103 stops processing of the first device when the data collating part 102 determines that it is abnormal.

[0043]

The synchronizing part 104 synchronizes processing of the first device and

processing of the second device using timing of sending of data sent from the sending part 101.

[0044]

In a form shown in Fig. 2(a), an acquiring part 105 acquires data generated by the first device and data generated by the second device in real time. The data collating part 102 collates data generated by the first device with data generated by the second device acquired by the acquiring part 105, and when a mismatch between these data occurs, it is determined that it is abnormal.

[0045]

The processing stop part 103 stops processing of the first device when the data collating part determines that it is abnormal.

[0046]

In a form shown in Fig. 2(b), a first code generating part 106 generates a first code for error detection based on data generated by the first device. A second code generating part 107 generates a second code for error detection based on data generated by the second device. A code collating part 108 collates the first code for error detection generated by the first code generating part 106 with the second code for error detection generated by the second code generating part 107, and when a mismatch between both the codes for error detection occurs, it is determined that it is abnormal.

[0047]

A processing stop part 109 stops processing of the first device when the code collating part 108 determines that it is abnormal.

[0048]

In a form shown in Fig. 3(a), a first inspecting part 111 inspects the presence or absence of abnormality of data with a first code for error detection received by the

first device using the first code for error detection. A second inspecting part 112 inspects the presence or absence of abnormality of data with a second code for error detection received by the second device using the second code for error detection. A code collating part 113 collates the first code for error detection used in the first inspecting part 111 with the second code for error detection used in the second inspecting part 112. A determining part 114 determines that it is abnormal when a mismatch between both the codes for error detection occurs as a result of collation by the code collating part 113 or when a result of inspection by the first inspecting part 111 or a result of inspection by the second inspecting part 112 indicates abnormality.

[0049]

A processing stop part 115 stops processing of the first device when the determining part 114 determines that it is abnormal.

[0050]

In a form shown in Fig. 3(b), a data collating part 116 collates data generated by the first device with data generated by the second device. A counting part 117 adds a count number to the data generated by the first device. A count stop part 118 stops addition of the count number by the counting part 117 when a mismatch between both the data occurs as a result of collation by the data collating part 116. A determining part 119 determines abnormality based on the count number added by the counting part 117.

[0051]

A processing stop part 120 stops processing of the first device or processing of the second device when the determining part 119 determines that it is abnormal.

[0052]

First to fourth embodiments of the information processing apparatus

according to the invention will be described below with reference to Figs. 4 to 10.

First Embodiment

[0053]

Fig. 4 is a block diagram showing a configuration of a safety system to which the information processing apparatus of the first embodiment is applied. This safety system is configured as a part of a plant control system.

[0054]

As shown in Fig. 4, the plant control system comprises a controller 2 for integrally managing and controlling field devices 1, 1,... such as a sensor or an electromagnetic valve arranged in each part of a plant, and input-output devices 3, 3,... interposed between the controller 2 and the field devices 1. The input-output devices 3, 3,... are connected to the controller 2 through a network 4. Also, the field devices 1, 1,... are connected to the input-output devices 3 through terminal boards 5.

[0055]

As shown in Fig. 4, input-output units 3a, 3b,... for executing interface processing between the field devices 1 and the controller 2 are mounted in the input-output devices 3. As described below, in these input-output units 3a, 3b,..., the same processing is executed mutually independently for the purpose of improving reliability.

[0056]

Fig. 5 is a block diagrams showing apart of the configuration of the input-output unit 3a. In Fig. 5, an example of a unit for processing an input value inputted from the side of the field device 4 which is a downstream process and outputting a PV value (process value) to the side of the controller 2 which is an upstream process is shown.

[0057]

As shown in Fig. 5, this unit comprises a master CPU 10 and a slave CPU 20, and the respective CPU 10 and CPU 20 execute the same processing mutually independently. Also, the CPU 10 and the CPU 20 execute diagnosis of peripheral circuits mounted in the respective peripheries.

[0058]

As shown in Fig. 5, an input value from the field device 1 is inputted to the master CPU 10 through an input part 71 and an input buffer 72. A peripheral circuit 74 of the periphery of the master CPU 10 is diagnosed by a diagnostic circuit 75. Also, a signal outputted from the input buffer 72 is inputted to the diagnostic circuit 75 and the presence or absence of abnormality of the signal is diagnosed. The presence or absence of abnormality of the peripheral circuit 74 and the presence or absence of abnormality of the signal outputted from the input buffer 72 are inputted to the master CPU 10 as diagnostic information from the diagnostic circuit 75.

[0059]

Similarly, the same input value from the field device 1 is inputted to the slave CPU 20 through the input part 71 and an input buffer 73. A peripheral circuit 76 of the periphery of the slave CPU 20 is diagnosed by a diagnostic circuit 77. Also, a signal outputted from the input buffer 73 is inputted to the diagnostic circuit 77 and the presence or absence of abnormality of the signal is diagnosed. The presence or absence of abnormality of the peripheral circuit 76 and the presence or absence of abnormality of the signal outputted from the input buffer 73 are inputted to the slave CPU 20 as diagnostic information from the diagnostic circuit 77.

[0060]

As shown in Fig. 5, the master CPU 10 comprises a PV value processing part

11 for executing arithmetic processing with respect to an input value inputted via the input buffer 72 and making conversion into a PV value (process value) of a format capable of processing in an upstream process which is the side of the controller 2, and a diagnostic part 12 for receiving diagnostic information from the diagnostic circuit 75 and executing abnormal detection and determining and generating a status which is a diagnostic result.

[0061]

Also, the master CPU 10 comprises a communication block 13 for executing communication with the slave CPU 20, and a code generating part 14 for adding a CRC (Cyclic Redundancy Check) code and an update counter to a PV value and a status.

[0062]

Also, the slave CPU 20 comprises a PV value processing part 21 for executing arithmetic processing with respect to an input value inputted via the input buffer 73 and making conversion into a PV value (process value) of a format capable of processing in the upstream process which is the side of the controller 2, and a diagnostic part 22 for receiving diagnostic information from the diagnostic circuit 77 and executing abnormal detection and determining and generating a status which is a diagnostic result.

[0063]

Also, the slave CPU 20 comprises a communication block 23 for executing communication with the master CPU 10, and a code generating part 24 for adding a CRC (Cyclic Redundancy Check) code and an update counter to a PV value and a status.

[0064]

Next, an operation of the present unit will be described.

[0065]

In the master CPU 10, a status generated by the diagnostic part 12 and a status

which is generated by the diagnostic part 24 of the slave CPU 20 and is acquired through communication by the communication block 23 and the communication block 13 are compared and equalized in an equalization part 15. Equalization is processing for equalizing a status handled by the master CPU 10 and a status handled by the slave CPU 20. In the equalization part 15, OR information about the statuses is generated. That is, when either status indicates abnormality in the equalization part 15, its abnormality is changed to the captured status and is passed to the code generating part 14. As described below, the statuses handled by the master CPU 10 and the slave CPU 20 are shared by performing similar processing also in the slave CPU 20.

[0066]

A PV value generated by the PV value processing part 11 is given to the code generating part 14. However, when abnormality of a status is detected based on processing in the equalization part 15, an input of the PV value to the code generating part 14 is broken by a breaking part 16.

[0067]

In the code generating part 14, a CRC code is generated based on the status generated by the equalization part 15 and the inputted PV value. Also, every time new PV value and status are inputted, a count number is updated and a code added to a CRC code is generated. In the code generating part 14, a frame made of the PV value, the status, the CRC code and the count number is generated by adding the code generated thus to the PV value and the status. The count number is incremented every update of the PV value and the status.

[0068]

A frame similar to the frame created by the code generating part 14 is similarly generated by the code generating part 24 of the slave CPU 20 and is acquired

through communication by the communication block 23 and the communication block 13. The frame created by the code generating part 14 and the frame created by the code generating part 24 are collated in a comparing part 17. The comparing part 17 decides that it is abnormal when a mismatch between both the frames is detected. As described below, by performing similar processing also in the slave CPU 20, the master CPU 10 and the slave CPU 20 mutually collate the other processing result with my processing result and decide that it is abnormal when a mismatch occurs. When all the processing in the master CPU 10 and the slave CPU 20 is normal, both the frames match as a result of collation in the comparing part 17.

[0069]

The frame generated by the code generating part 14 is outputted to an output part 78 which is an upstream process. However, a mismatch between both the frames is detected in the comparing part 17 and in the case of deciding that it is abnormal, an output of the frame is broken by a breaking part 18. Also, as described below, when a mismatch between the frames is detected in a comparing part 27 of the slave CPU 20, an output of the frame is broken in a fail-safe part 79.

[0070]

On the other hand, in the slave CPU 20, a status generated by the diagnostic part 22 and a status which is generated by the diagnostic part 14 of the master CPU 10 and is acquired through communication by the communication block 13 and the communication block 23 are compared and equalized in an equalization part 25. In the equalization part 25, OR information about the statuses is generated. That is, when either status indicates abnormality in the equalization part 25, its abnormality is changed to the captured status and is passed to the code generating part 24.

[0071]

A PV value generated by the PV value processing part 21 is given to the code generating part 24. However, when abnormality of a status is detected based on processing in the equalization part 25, an input of the PV value to the code generating part 24 is broken by a breaking part 26.

[0072]

In the code generating part 24, a CRC code is generated based on the status generated by the equalization part 25 and the inputted PV value. Also, every time new PV value and status are inputted, a count number is updated and a code added to a CRC code is generated. In the code generating part 24, a frame made of the PV value, the status, the CRC code and the count number is generated by adding the code generated thus to the PV value and the status. The count number is incremented every update of the PV value and the status.

[0073]

The frame created by the code generating part 24 is collated with a frame which is similarly generated by the code generating part 14 of the master CPU 10 and is acquired through communication by the communication block 13 and the communication block 23 in a comparing part 27. It is decided that it is abnormal when a mismatch between both the frames is detected in the comparing part 27.

[0074]

When the mismatch between the frames is detected in the comparing part 27, a fail-safe signal indicating abnormality is outputted from the comparing part 27 and is given to the fail-safe part 79. In this case, in the fail-safe part 79, an output of a frame from the CPU 10 is broken and a new frame to the output part 78 is inhibited. Instead of breaking an output by the fail-safe part 79, a reset signal may be outputted from the CPU 20 to a reset circuit of the CPU 10. In this case, the CPU 10 receiving the reset

signal is forcedly reset and an output to the output part 78 is inhibited.

[0075]

When the output to the output part 78 is inhibited, update of the count number is stopped, so that it can be recognized that an output of information is stopped by only referring to the count number in an upstream process of a subsequent stage after the output part 78.

[0076]

Next, a method of communication between the master CPU 10 and the slave CPU 20 will be described. As described above, in the master CPU 10 and the slave CPU 20, data is exchanged in real time and the data is collated. As a result of this, when timing of processing in both the CPUs is off, separate processing results different in a time axis direction are compared and a mismatch of collation occurs. As a result of this, it is necessary for both the CPUs to always execute the same operation in the apparatus of the present embodiment. Therefore, control is performed so that a timing phase trigger is sent from the side of the master CPU 10 using asynchronous communication (UART) and the slave CPU 20 can execute processing in the same sequence in synchronization.

[0077]

Fig. 6 is a diagram showing sequence of communication processing. Figs. 7(a) and Fig. 7(b) are diagrams showing configurations of communication frames, and Fig. 7(a) shows a configuration of individual communication frames, and Fig. 7(b) shows an operation of the case where a communication state is normal.

[0078]

As shown in Fig. 6, the master CPU 10 sends a command with a timing phase trigger to the slave CPU 20 in a constant cycle. The slave CPU 20 receiving the

command replies a response to the master CPU 10. After such processing, both the CPUs execute the same phase and thereby processing of both the CPUs is synchronized.

[0079]

As shown in Fig. 7(a), the master CPU 10 sends a command with a timing phase trigger through the communication block 13. The slave CPU 20 receives the command through the communication block 23. In the slave CPU 20, a phase (sequence number) represented in the received command is compared with an expected phase (sequence number), that is, a phase (phase 1 in Fig. 7(a)) to be processed next and when both the phases match, it is recognized that a communication state is normal. In the case of recognizing that the communication state is normal, the slave CPU 20 replies a response including information (sequence number) indicating the phase (phase 1 in Fig. 7(a)) to the master CPU 10 through the communication block 23. The master CPU 10 recognizes that a communication state is normal when the response from the slave CPU 20 is received within a certain time and a phase (sequence number) represented in the received response is proper.

[0080]

After a command trigger period for which a communication state is recognized by sending and receiving of the command and the response, it shifts to a full-duplexing communication period. For the full-duplexing communication period, the master CPU 10 and the slave CPU 20 respectively execute processing of the same phase (phase 1 in Fig. 7(a)), and data MA is sent from the master CPU 10 to the slave CPU 20 and data SL is sent from the slave CPU 20 to the master CPU 10, concurrently. The data MA and the data SL include the frame (frame made of a PV value, a status, a CRC code and a count number) sent and received for collation and the status sent and received for equalization described above, respectively.

[0081]

As shown in Fig. 7(b), by sequentially repeating such phases, the same processing is executed in synchronization with each other in the master CPU 10 and the slave CPU 20.

[0082]

As described above, in the first embodiment, CRC codes are generated by both the CPUs and collation is executed by data (frames) with CRC, so that reliability of the collation increases. Also, only when both the data match as a result of the collation by both the CPUs, information is notified of an upstream process and when any one of the CPUs determines that it is abnormal, an output of information to the upstream process is surely prevented. As a result of this, reliability of the information outputted to the upstream process can be increased. That is, the fact that a CRC code is normal by inspection in the upstream process means that its data is data with high reliability collated between the CPUs inside the input-output unit 3a. Also, in the upstream process, by inspecting the CRC code, the presence or absence of abnormality can be diagnosed again over all the processes of handling data with CRC inside the CPU.

[0083]

Also, using communication between CPUs, data with CRC is sent and received and is collated by both the CPUs, so that it is unnecessary to separately make a check of frames of communication between CPUs, for example, a check of frame sum, parity, etc.

[0084]

The first embodiment is constructed so as to synchronize processing every phase using timing of communication between CPUs. As a result of this, it is

unnecessary to perform useless processing for synchronization and reduction in performance resulting from synchronization processing does not occur. That is, extra processing is not required by only inserting a sequence number for identifying a phase into the primarily essential contents of communication. Also, both the CPUs always execute the same phase in synchronization with each other by such synchronization, so that accuracy of data collation between the CPUs can be increased. On the other hand, both the CPUs simultaneously follow a transient state change caused by a factor of the outside of the CPU, so that a sudden collation mismatch between data does not occur. Also, the CPU 20 basically has a relation of executing a phase specified from the CPU 10, so that return can be made easily even in the case of going out of synchronization between phases.

[0085]

In the first embodiment, statuses generated by each of the CPUs are equalized at the previous stage of mutually collating data of both the CPUs. As a result of this, when abnormality of the status is detected in either CPU, both the CPUs share recognition that the status is abnormal by equalization of the statuses. Because of this, at the time of abnormality of the status, a collation mismatch between data at the subsequent stage does not occur and the status abnormality can be recognized separately from the collation mismatch between data. As a result of this, notification of an abnormal state can be provided correctly. Such status equalization is particularly more effective in the case where it is necessary to loosely couple CPUs by decreasing a common circuit part between the CPUs even though an independent peripheral circuit is increased every CPU. Also, it is effective in the case where a peripheral circuit can be diagnosed in only one CPU.

[0086]

The processing of equalization, addition of a CRC code, sequence of data collation, etc. shown in the first embodiment can be implemented easily by a program of the CPU. Also, communication between CPUs can be implemented easily using an asynchronous communication (UART) function normally mounted in the CPUs. Also, by using serial communication as the communication between CPUs, mounting is facilitated even when two CPUs are mounted in the side of a controller and the side of a field device insulated mutually.

Second Embodiment

[0087]

An information processing apparatus of a second embodiment shows an example in which a master CPU and a slave CPU are mounted in a mutually insulated state.

[0088]

Fig. 8 is a block diagram showing a configuration of an input-output unit as the information processing apparatus of the second embodiment.

[0089]

In the plant control system as shown in Fig. 4, the controller 2 and the field devices 1 are usually spaced. Therefore, grounds of the field devices 1 and the controller 2 are disposed in a mutually separated state for the purpose of escaping the influence of a thunderbolt or noise propagating the surface of earth. As a result of this, it is necessary to maintain an electrically insulated state between the side of the controller 2 and each of the field devices 1.

[0090]

In an input-output unit 3b shown in Fig. 8, a master CPU 10A is disposed in

the side (upstream side) of the controller 2 and a slave CPU 20A is disposed in the side (downstream side) of the field devices 1 through an insulation boundary L. An analog input value from the downstream side is converted into a digital signal in an AD conversion part 33 via an input circuit 32 and is inputted to the slave CPU 20A. Information, which shares a ground potential with the slave CPU 20A, from a peripheral circuit 35 is given to a diagnostic part 34 and also an input value outputted from the input circuit 32 is given to the diagnostic part 34. In the diagnostic part 34, diagnostic information based on the input value outputted from the input circuit 32 and the information from the peripheral circuit 35 is given to the slave CPU 20A.

[0091]

On the other hand, the input value from the downstream side is also inputted to the master CPU 10A through a photo coupler 37. The analog input value is binarized in the photo coupler 37. Processing based on a binarized signal (digital signal) is executed in the master CPU 10A.

[0092]

In addition, the input value is constructed of plural channels (for example, 8 channels), and units of the number corresponding to the number of channels are prepared in the input circuit 32, the AD conversion part 33 and the photo coupler 37.

[0093]

The master CPU 10A and the slave CPU 20A send and receive a command with a timing phase trigger and a response to the command through a communication block 11A between CPUs and a communication block 21A between CPUs disposed respectively. Consequently, the same processing is executed in synchronization mutually independently. Also, the master CPU 10A and the slave CPU 20A send and receive mutual data through the communication block 11A between CPUs and the

communication block 21A between CPUs and respectively collate the data. Further, diagnostic information obtained in the slave CPU 20A is also sent to the master CPU 10A and statuses are equalized.

[0094]

In the master CPU 10A, a host block 38 is notified of data through a host communication block 12A only when both the data match as a result of collation between the data of the master CPU 10A and the slave CPU 20A without indicating abnormality of the equalized statuses. Also, when a mismatch between data of the master CPU 10A and the slave CPU 20A is detected in the slave CPU 20A, a reset signal is sent out to the master CPU 10A. In this case, the master CPU 10A is forcibly changed in a reset state by the reset signal and notification of data from the master CPU 10A to the host block 38 is inhibited. In addition, transmission lines of the reset signal and a path of communication between CPUs are insulated between the master CPU 10A and the slave CPU 20A by a photo coupler etc. in the insulation boundary L.

[0095]

In the unit 3b shown in Fig. 8, the slave CPU 20A is mounted in the side of the field devices 1, so that an input value via the input circuit 32 or information from the peripheral circuit 35 mounted in the side of the field devices 1 can be captured easily and a detailed diagnosis can be made. Also, by using asynchronous communication (UART) etc., the communication between CPUs can be executed by a small number of communication lines, so that an insulated state can be maintained easily. As a result of this, the statuses which are diagnostic information are equalized between both the CPUs by the communication between CPUs and thereby, a detailed diagnostic result in the slave CPU 20A can also be effectively utilized in the master CPU 10A.

Third Embodiment

[0096]

An information processing apparatus of a third embodiment shows an example of an input-output unit in which an analog signal from the side of the field devices 1 (Fig. 4) is converted into a digital signal and two CPUs receive the common digital signal and data is outputted to the side of the controller 2 (Fig. 4). Fig. 9 is a block diagram showing a configuration of the input-output unit as the information processing apparatus of the third embodiment.

[0097]

An input-output unit 3c shown in Fig. 9 comprises a master CPU 10B and a slave CPU 20B for executing the same processing mutually independently, a main multiplexer 41 and a sub multiplexer 42 for receiving analog signals of plural channels (for example, 8 channels) and selecting one signal, an input amplifier 43 for receiving the analog signal from the multiplexer 41, and an AD converter 44 for converting a signal outputted from the input amplifier 43 into a digital signal and giving the signal to the master CPU 10B and the slave CPU 20B.

[0098]

Output signals of the main multiplexer 41 and the sub multiplexer 42 are respectively compared in the master CPU 10B and the slave CPU 20B and in the case of a mismatch, it is decided that a status is abnormal. Also, the statuses are exchanged by communication between CPUs and the statuses are equalized. Consequently, soundness of an operation of the multiplexer 41 is diagnosed. Also, the input amplifier 43 and the AD converter 44 are common to all the channels, and the statuses are monitored by inputting a reference voltage to the input amplifier 43 through the multiplexer 41 in a constant cycle and respectively checking an output of the AD

converter 44 by the master CPU 10B and the slave CPU 20B. Also in this case, the statuses are exchanged by communication between CPUs and the statuses are equalized.
[0099]

The master CPU 10B and the slave CPU 20B send and receive a command with a timing phase trigger and a response to the command through a communication block 11B between CPUs and a communication block 21B between CPUs disposed respectively and thereby, the same processing is executed in synchronization mutually independently. Also, statuses obtained by the master CPU 10B and the slave CPU 20B are exchanged by communication between CPUs and the statuses are equalized. Further, the master CPU 10B and the slave CPU 20B send and receive mutual data through the communication block 11B between CPUs and the communication block 21B between CPUs and collate the data.

[0100]

In the master CPU 10B, a host block 45 is notified of data through a communication block 12B only when both the data match as a result of collation between the data of the master CPU 10B and the slave CPU 20B without indicating abnormality of the equalized statuses. Also, when a mismatch between data of the master CPU 10B and the slave CPU 20B is detected in the slave CPU 20B, a reset signal is sent out to the master CPU 10B. In this case, the master CPU 10B is forcibly changed in a reset state by the reset signal and notification of data from the master CPU 10B to the host block 45 is inhibited.

Fourth Embodiment

[0101]

An information processing apparatus of a fourth embodiment shows an

example of an input-output unit in which data with a CRC code inputted from the side (upstream side) of the controller 2 (Fig.4) is outputted to the side (downstream side) of the field devices 1 (Fig. 4). Fig. 10 is a block diagram showing a part of a configuration of the input-output unit as the information processing apparatus of the fourth embodiment.

[0102]

The input-output unit shown in Fig. 10 comprises a CPU 50 and a CPU 60 for executing processing mutually independently.

[0103]

The CPU 50 comprises a code inspecting part 51 for inspecting a CRC code based on data inputted via an input part 81, a communication block 52 for executing communication with the CPU 60, and a set value processing part 53 for converting the inputted data into a format used in the side of the field devices 1.

[0104]

The CPU 60 comprises a code inspecting part 61 for inspecting a CRC code based on data transferred from the CPU 50, a communication block 62 for executing communication with the CPU 50, and a set value processing part 63 for converting the inputted data into a format used in the side of the field devices 1.

[0105]

Next, an operation of the input-output unit shown in Fig. 10 will be described.

[0106]

Data from the side of the controller 2 is inputted to the CPU 50 through the input part 81. This data includes a set destination for identifying the field device 1 (for example, an electromagnetic valve), a set value (for example, an opening of an electromagnetic valve) to be set in the set destination, and a CRC code created based on

data of the set destination and the set value.

[0107]

The data inputted to the CPU 50 is given to the code inspecting part 51. In the code inspecting part 51, a CRC code is created based on data of the set destination and the set value received. Then, a CRC code received as data is compared with the CRC code created in the code inspecting part 51. The code inspecting part 51 decides that it is abnormal when a mismatch between both the CRC codes occurs, and in this case, data is broken in a breaking part 54.

[0108]

On the other hand, the data inputted to the CPU 50 is sent through the communication block 52. The sent data is received by the CPU 60 through the communication block 62.

[0109]

The data received by the CPU 60 is given to the code inspecting part 61. In the code inspecting part 61, a CRC code is created based on data of the set destination and the set received. Then, a CRC code received as data is compared with the CRC code created in the code inspecting part 61. The code inspecting part 61 decides that it is abnormal when a mismatch between both the CRC codes occurs, and in this case, data is broken in a breaking part 64.

[0110]

Then, the CRC code created in the code inspecting part 51 of the CPU 50 is sent through the communication block 52. The sent CRC code is received by the CPU 60 through the communication block 62. Also, the CRC code created in the code inspecting part 51 of the CPU 50 is compared with the CRC code sent from the CPU 60 in a comparing part 55.

[0111]

The comparing part 55 decides that it is abnormal when both the CRC codes do not match as a result of comparing both the CRCs. In this case, data is broken in a breaking part 56.

[0112]

In the CPU 60, the CRC code created in the code inspecting part 61 is sent through the communication block 62. The sent CRC code is received by the CPU 50 through the communication block 52. This CRC code is compared with the CRC code created in the code inspecting part 51 in the comparing part 55 as described above.

[0113]

Also, the CRC code created in the code inspecting part 61 of the CPU 60 is compared with the CRC code sent from the CPU 50 in a comparing part 65.

[0114]

The comparing part 65 decides that it is abnormal when both the CRC codes do not match as a result of comparing both the CRCs. In this case, data is broken in a breaking part 66.

[0115]

Then, in the CPU 50, the data received from the input part 81 is given to the set value processing part 53. However, in the case of deciding that it is abnormal as described above, delivery of data is inhibited in the breaking part 54 or the breaking part 56 and processing in the set value processing part 53 is stopped.

[0116]

When data is inputted, data of a set destination and a set value are converted into a format used in the side of the field devices 1 in the set value processing part 53.

[0117]

On the other hand, in the CPU 60, the data sent from the CPU 50 is given to the set value processing part 63. However, in the case of deciding that it is abnormal as described above, delivery of data is inhibited in the breaking part 64 or the breaking part 66 and processing in the set value processing part 63 is stopped.

[0118]

When data is inputted, in the set value processing part 63, data of a set destination and a set value are converted into a format used in the side of the field devices 1 and the data of the set destination and the set value are outputted to an output part 82.

[0119]

The data of the set destination and the set value outputted to the output part 82 are inputted to a comparing part 67 of the CPU 60 through a diagnostic circuit 83. Also, the data of the set destination and the set value outputted from the set value processing part 63 are directly inputted to the comparing part 67.

[0120]

In the comparing part 67, the data of the set destination and the set value at a stage outputted from the set value processing part 63 are compared with the data of the set destination and the set value via the diagnostic circuit 83 and when a mismatch between both the data occurs, it is decided that it is abnormal. In this case, the data of the set destination and the set value are broken in a breaking part 68 and an output of the data of the set destination and the set value to the output part 82 is inhibited.

[0121]

Also, the data of the set destination and the set value outputted from the CPU 60 to the output part 82 are inputted to a comparing part 57 of the CPU 50 through a diagnostic circuit 84. Further, the data of the set destination and the set value

outputted from the set value processing part 53 are directly inputted to the comparing part 57.

[0122]

In the comparing part 57, the data of the set destination and the set value outputted from the set value processing part 53 are compared with the data of the set destination and the set value outputted from the CPU 60 and when a mismatch between both the data occurs, it is decided that it is abnormal. In this case, the data of the set destination and the set value are broken in a fail-safe circuit 85 and an output of the data of the set destination and the set value from the CPU 60 to the output part 82 is inhibited.

[0123]

Thus, in the fourth embodiment, the side of the CPU 60 executes a data output to the output part 82, but in the CPU 60, the outputted data is itself traced and in the case of deciding that it is abnormal, the data output is stopped. Also, in the CPU 50, the data outputted to the output part 82 by the CPU 60 is simultaneously traced and in the case of deciding that it is abnormal, an output by the CPU 60 from the side of the CPU 50 is inhibited. As a result of this, when either CPU decides that it is abnormal, the data output is inhibited, so that wrong data can surely be prevented from being outputted to the output part 82.

[0124]

The scope of application of the invention is not limited to the embodiments described above. Also, the invention can be widely applied to an information processing system for handling various information as well as a safety system.

[0125]

The present application is based on Japanese Patent Application (No.2005-021423) filed on January 28, 2005, the contents of which are incorporated herein by reference.

CLAIMS

1. An information processing apparatus comprising first and second devices each executing the same processing in a duplex manner independently, said information processing apparatus comprising:

first determining means which determines a state of the information processing apparatus in the first device;

second determining means which determines the state of the information processing apparatus in the second device;

sending means which sends a determination result of the first determining means toward the second device; and

collating means which collates determination results of the first determining means and the second determining means in the second device and then determines as abnormal when both the determination results are not conform with each other.

2. The information processing apparatus according to claim 1, further comprising:

stop means which stops a function of the first device when determined as abnormal by the collation of the collating means.

3. The information processing apparatus according to claim 1 or 2, further comprising:

synchronizing means which synchronizes a processing of the first device and a processing of the second device using timings of sending and receiving of the data sent by the sending means.

4. An information processing apparatus comprising first and second devices each executing the same processing in a duplex manner independently, said information processing apparatus comprising:

first determining means which determines a state of the information processing apparatus in the first device;

first code generating means which generates a code for error detection based on a data of a determination result of the first determining means in the first device;

second determining means which determines the state of the information processing apparatus in the second device;

second code generating means which generates a code for error detection based on a data of a determination result of the second determining means in the second device;

sending means which sends the code for error detection generated by the first code generating means toward the second device; and

collating means which collates the code for error detection generated by the first code generating means with the code for error detection generated by the second code generating means in the second device and then determines as abnormal when both the codes for error detection are not conform with each other.

5. The information processing apparatus according to claim 4, further comprising:

stop means which stops a function of the first device when determined as abnormal by the collation of the collating means.

6. The information processing apparatus according to any one of claims 1 to 5,

further comprising:

equalization means which equalizes a state of the information processing apparatus between the first device and the second device,

wherein the first determining means and the second determining means determine the state equalized by the equalization means.

7. The information processing apparatus according to any one of claims 1 to 6, wherein the first device and the second device are constructed as separate CPUs.

8. An information processing method comprising first and second devices each executing the same processing in a duplex manner independently, said information processing method comprising the steps of:

determining a state of the information processing apparatus in the first device;

determining the state of the information processing apparatus in the second device;

sending a determination result in the first device toward the second device; and

collating a determination result in the first device and a determination result in the second device and then determining as abnormal when both the determination results are not conform with each other.

9. The information processing method according to claim 8, further comprising the step of:

stopping a function of the first device when determined as abnormal by the collation in the collating step.

10. The information processing method according to claim **8** or **9**, further comprising the step of:

synchronizing a processing of the first device and a processing of the second device using timings of sending and receiving of the data sent by the step of sending the determination result in the first device toward the second device.

11. An information processing method comprising first and second devices each executing the same processing in a duplex manner independently, said information processing method comprising the steps of:

determining a state of the information processing apparatus in the first device;

generating a code for error detection based on a data of a determination result of first determining means in the first device;

determining the state of the information processing apparatus in the second device;

generating a code for error detection based on a data of a determination result of second determining means in the second device;

sending the code for error detection generated in the first device toward the second device; and

collating the code for error detection generated in the first device with the code for error detection generated in the second device in the second device and then determining as abnormal when both the codes for error detection are not conform with each other.

12. The information processing method according to claim **11**, further

comprising the step of:

stopping a function of the first device when determined as abnormal by
collation of collating means.