(12) UK Patent Application (19) GB (11) 2 120 055 A

(21) Application No **8311274**
(22) Date of filing **26 Apr 1983**
(30) Priority data
(31) **8212266**
(32) **28 Apr 1982**
(33) **United Kingdom (GB)**
(43) Application published **23 Nov 1983**
(51) **INT CL[3] H04L 11/16**
(52) Domestic classification **H4P** PL
(56) Documents cited **None**
(58) Field of search **H4P H4K**
(71) Applicants
International Computers Limited,
(United Kingdom),
ICL House,
Putney,
London SW15 1SW
(72) Inventors
Donald Bell
Stuart O'Connor
(74) Agent and/or Address for Service
R. G. Brodie
International Computers Limited,
Group Patent Services,
Cavendish Road,
Stevenage,
Herts SG1 2DY

(54) Data communication system

(57) Stations (3) are connected in a series configuration. A station which wishes to transmit waits until it detects a "token", a signal pattern present only when no station is transmitting, converts it into a "connector" and then transmits its own data. The token contains two instances of a first sequence of signals (GO AHEAD) which does not occur elsewhere in the transmissions of the system. To convert the token into a connector each instance of this sequence is changed as it passes in its last bit position into a second sequence (FLAG). This arrangement offers good security against corruption of transmitted data causing a connector to be mistaken for a token while at the same time keeping short the time needed to recognise and change the token.

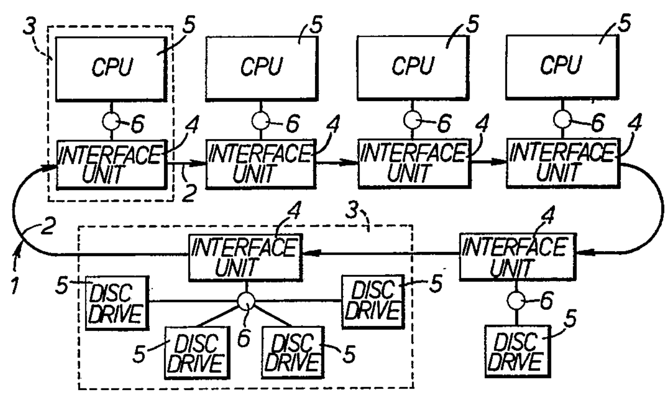

FIG. 1.

GO AHEAD 01111111 TOKEN 01111111¦01111111
FLAG 01111110 CONNECTOR 01111110¦01111110

FIG. 3.

GB 2 120 055 A

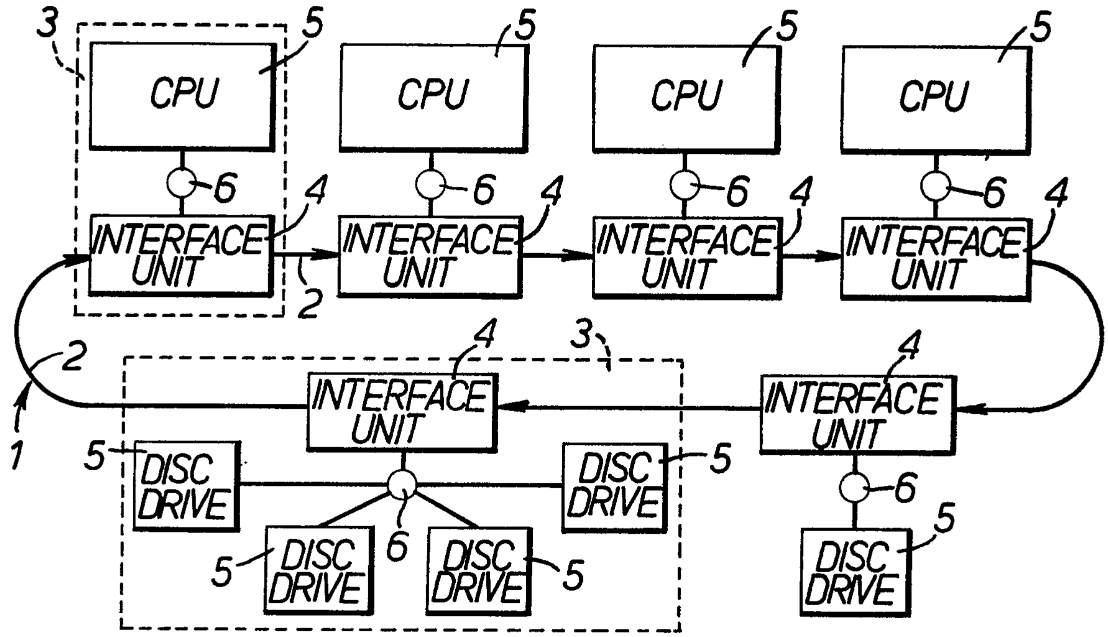

FIG. 1.

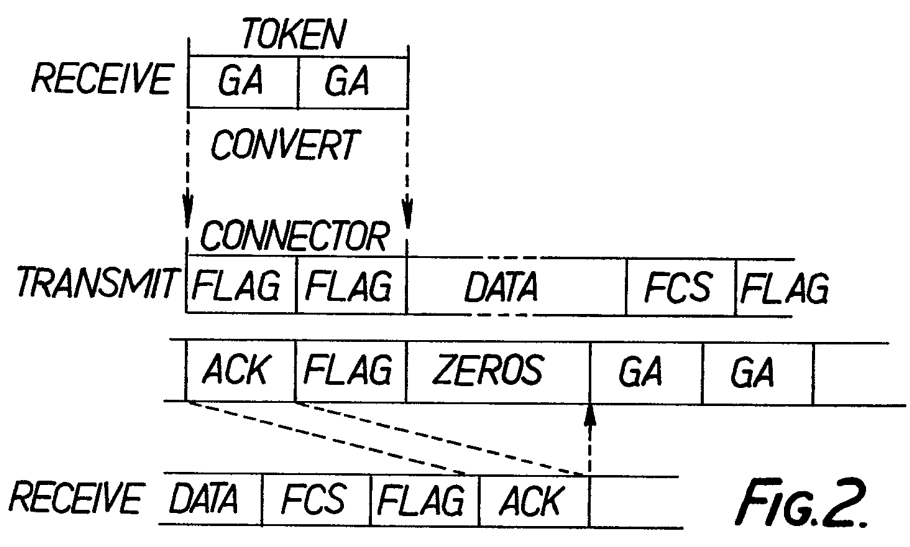

FIG. 2.

GO AHEAD	01111111	TOKEN	01111111 ¦ 01111111
FLAG	01111110	CONNECTOR	01111110 ¦ 01111110

FIG. 3.

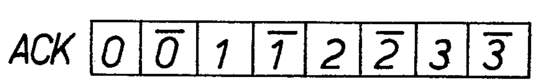

FIG. 4.

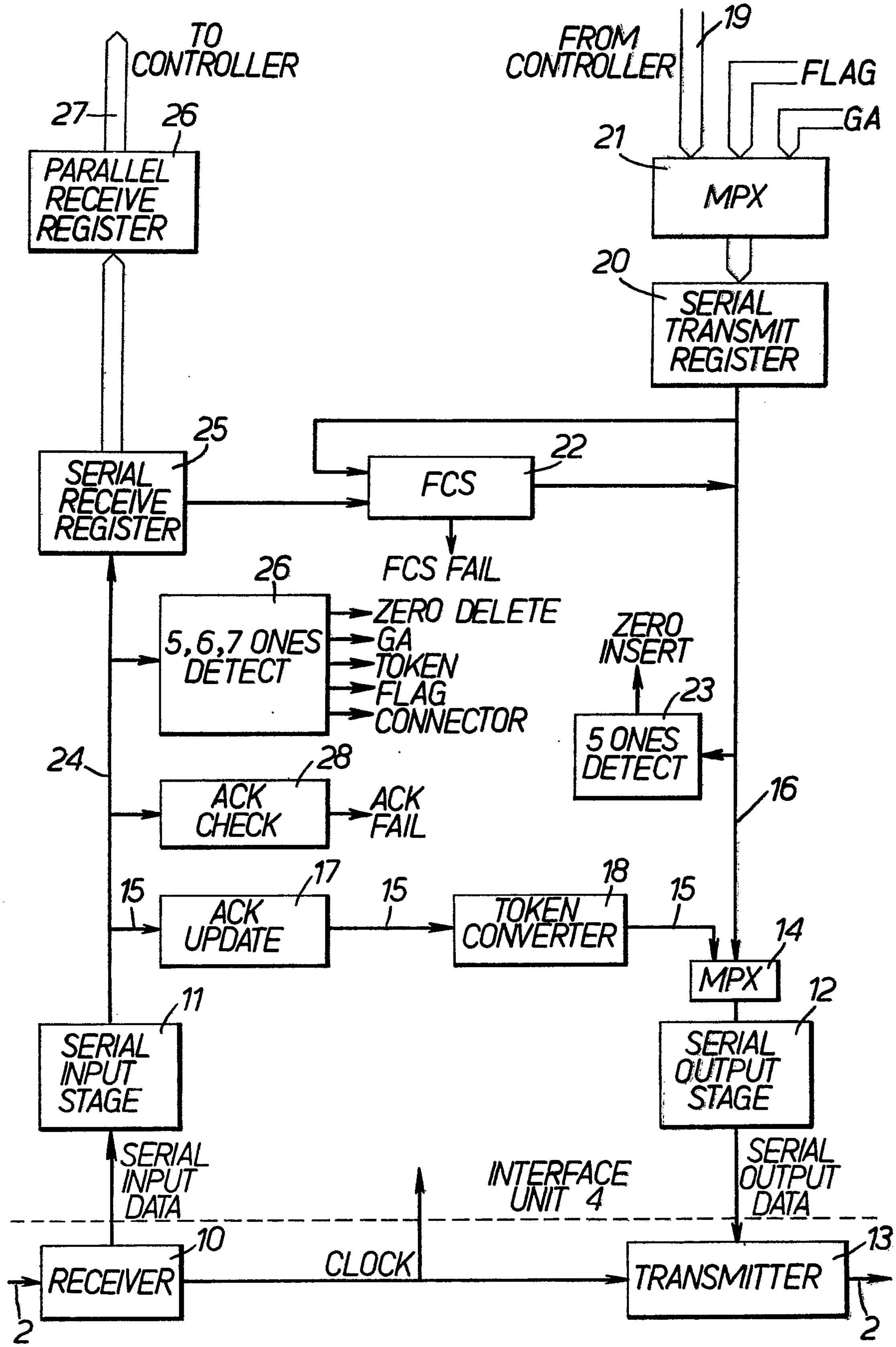
TO CONTROLLER
27
26
PARALLEL RECEIVE REGISTER
FROM CONTROLLER
19
FLAG
GA
21
MPX
20
SERIAL TRANSMIT REGISTER
25
SERIAL RECEIVE REGISTER
22
FCS
FCS FAIL
26
5,6,7 ONES DETECT
ZERO DELETE
GA
TOKEN
FLAG
CONNECTOR
ZERO INSERT
23
5 ONES DETECT
24
28
ACK CHECK
ACK FAIL
16
15
17
ACK UPDATE
15
18
TOKEN CONVERTER
15
14
MPX
11
SERIAL INPUT STAGE
12
SERIAL OUTPUT STAGE
SERIAL INPUT DATA
INTERFACE UNIT 4
SERIAL OUTPUT DATA
10
RECEIVER
CLOCK
TRANSMITTER
13
2
2

FIG. 5.

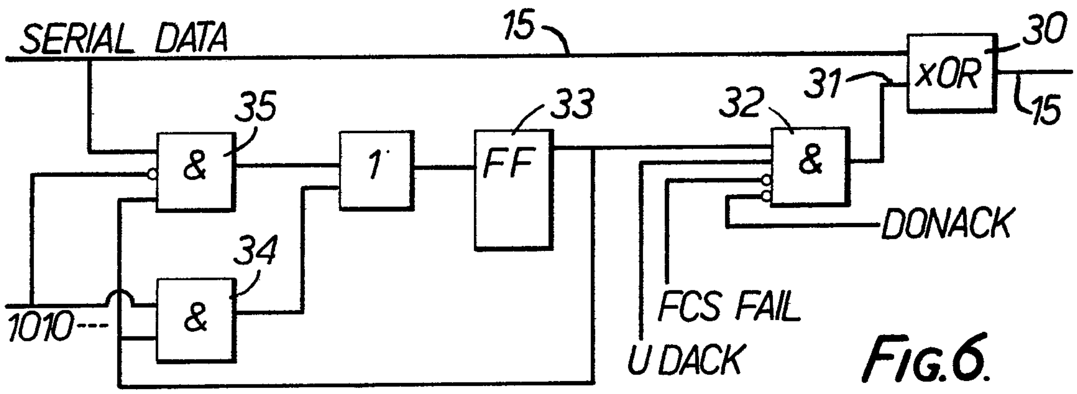

FIG.6.

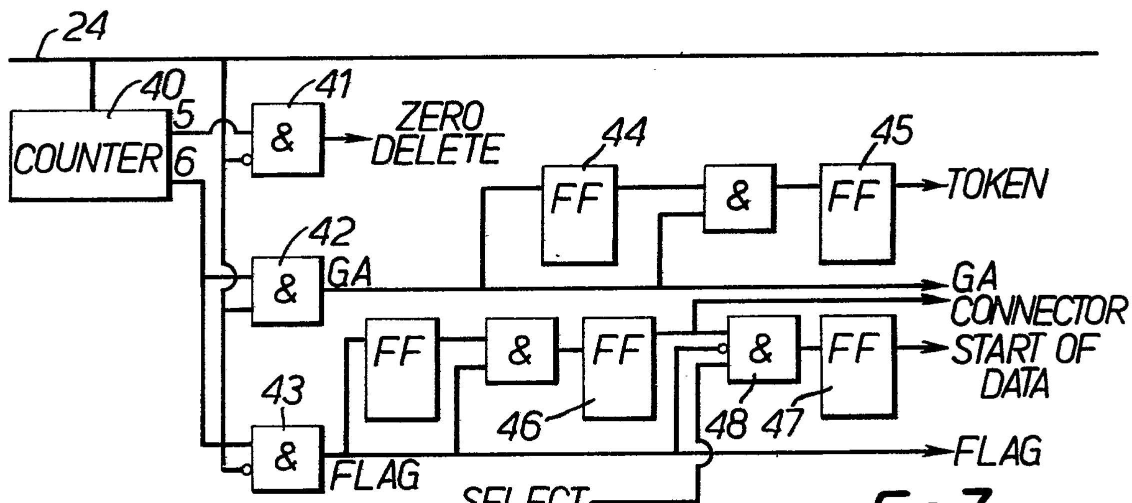

FIG.7.

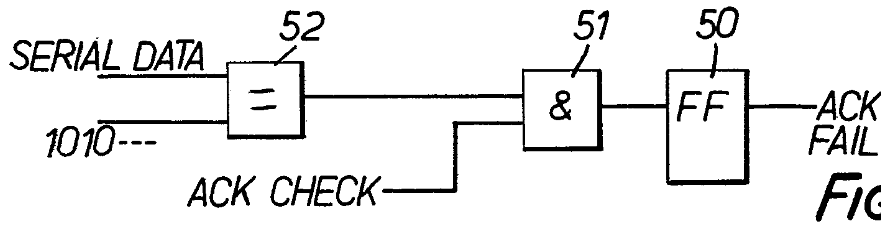

FIG.8.

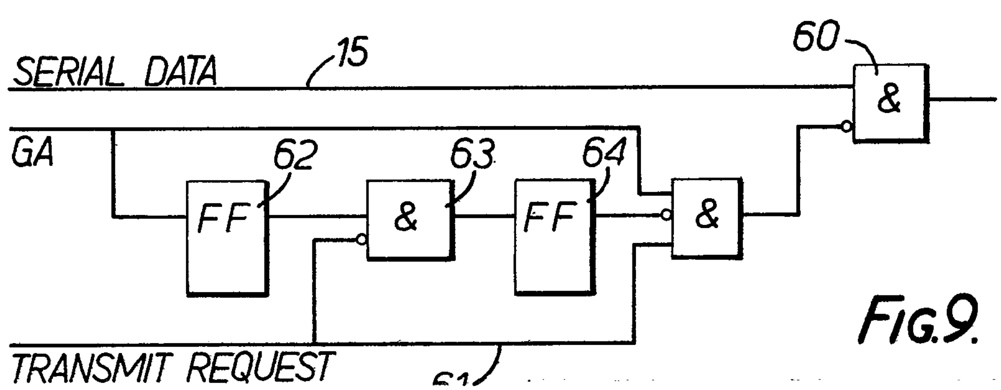

FIG.9.

SPECIFICATION

Improvements in or relating to data communication systems

This invention relates to data communication systems in which a plurality of stations are connected transmission links in a single series configuration such as a ring or loop.

One known control procedure for allowing a station access to the transmission medium when it wishes to transmit uses what will be termed herein a "token", namely a pattern of signals which is present only when no station is transmitting. A station which wishes to transmit changes the token to another pattern which will be termed a "connector" and then transmits its data.

In order to ensure that the minimum delay to the message flow is introduced in changing the token into a connector, the token and the connector differ only at the final bit position. However, the system is relatively vulnerable to noise or erroneous operation since only one bit need be corrupted to change a connector into a token or vice versa.

According to this invention there is provided a method of data transmission in a system comprising a plurality of stations connected in a series configuration by transmission links, in which method data is transferred serially in one direction only along the series configuration of stations and a station starts to transmit data from itself in response to detection of a token comprising a plurality of first sequences of signals, each instance of a first sequence occurring only in a token and not elsewhere in transmission over the transmissions over the transmission links, the station converting the token as it passes through the station into a connector in which each first sequence is changed into a second sequence by altering the signal in the last bit position of the first sequence, the station thereafter transmitting the said data from itself.

This method offers greater protection than the prior method described above against corruption of a token or connector causing the wrong control code to be recognised. But it does not introduce any extra delay, since in normal operation a first sequence occurs only in a token and therefore, if recognised, indicates that a token is being received. It may therefore be changed as it passes through the station without the need to delay it long enough to allow the end of the token to arrive.

There is also provided, according to the invention an interface apparatus suitable for forming the coupling between a station and an incoming and an outgoing serial transmission link, the element comprising:
means for receiving data from the incoming link;
means for outputting data for transmission over the outgoing link;
loop-through means providing a path for the transfer of data from the receiving means to the outputting means;
means for detecting the reception at the receiving means of a token comprising a plurality of first sequences of signals;
means responsive at least to an indication that the station is ready to transmit and the detection of a token for enabling the station to supply data from itself to the output means in place of data from the loop-through path; and
means responsive at least to an indication that the station is ready to transmit for changing a token into a connector, the means changing each first sequence of signals into a second sequence by altering the signal in the last bit position of the sequence;
the last two enumerated means each being inhibited from operating in response to a token if the ready-to-transmit indication is received too late to alter the initial first sequence of the token.

An example of a system operation in accordance with the invention will now be described in greater detail with reference to the accompanying drawings, in which

Figure 1 is an overall block diagram of the system;
Figure 2 illustrates the message format;
Figure 3 illustrates certain control patterns;
Figure 4 illustrates the acknowledgement field;
Figure 5 is a block diagram of the interface unit;
Figure 6 is a logic diagram of the acknowledgement update circuit;
Figure 7 is a logic diagram of the 5,6,7 ones detection circuit;
Figure 8 is a logic diagram of the acknowledgement field check circuit, and
Figure 9 is a logic diagram of the token-to-connector converter circuit.

Overall system

Referring to Figure 1, the system allows a number of devices to communicate with one another over a ring 1. The ring 1 consists of transmission links 2 joining stations 3 each of which contains an interface unit 4 to which the links 2 are joined, one or more devices 5, and a controller 6 connecting the device or devices to the interface unit 4.

Figure 1 is purely an illustration of one possible configuration and shows four stations each containing a CPU and two stations in which the controller is an input-output controller for one or more disc drives. Other devices may of course be used. One special case is a device acting as a gateway to another network. In this case the device, although originating or receiving messages from the point of view of the ring network, actually transfers the information to or from another network.

Data passes round the ring bit-serially and in one direction only. Each interface unit 4 normally functions as an active repeater and retransmits onto its outgoing link 2 the data it receives from its incoming link 2, unless its station is transmitting, in which case it substitutes data from its controller 6 for that received. The interface unit also acts as a serialiser-deserialiser, converting serial data from the ring into parallel data for the controller and vice versa.

The ring uses a token control mechanism in which a station is able to transmit when it detects what will be termed a "token". If it wishes to transmit, it changes the token to what will be termed a "connector". It then sends its message out and finally relinquishes control by sending another token.

Messages may be sent either to one individual device or to several. For example, one CPU may broadcast a message to all the other CPUs in order to ensure that all the CPUs hold the same version of a common data item. It is especially important in such a multiprocessor system that the station originating a message should be assured that the message has been correctly received by all the stations for which it is intended, and the system being described uses an acknowledgement scheme carried out by the interface units to provide this assurance.

Message format and transmission

Referring to Figure 2, assume a station wishes to transmit a message. Then the interface unit 4 of that station examines the data passing round the ring until it detects the token, which indicates that no other station is transmitting. In this system the token consists of two identical eight-bit bytes, each of which will be termed herein a "go-ahead", abbreviated to GA in various of the figures. (See also figure 3, which shows the bit patterns of the various control codes.) The go-ahead consists of a zero followed by seven ones. When the interface unit detects the token it changes it to a connector to prevent its being recognised by any other station. The connector is derived from the token by changing the last bit of each byte to a zero. The resulting pattern 01111110 for each byte will be termed herein a "flag".

After the interface unit has transmitted the connector it ceases to retransmit the data it receives from the ring and instead transmits data provided by its controller, preceded if desired by additional flags. Typically the data will contain the address of the source of the data, the address of the intended recipient or recipients, control information and the actual information to be passed from the source to the recipient or recipients.

The data is followed by a frame checking sequence (FCS) calculated from the data and designed to allow the detection of errors introduced into the data. The frame checking sequence is followed by a flag.

In order to ensure that the flag and go-ahead patterns are recognised as such it is arranged that a sequence of six ones does not occur elsewhere in any message over the links 2. However, it is inconvenient if this limitation is also imposed on the data, which is therefore allowed to take any form but undergoes a procedure known as "bit stuffing" in the interface unit. The interface unit breaks up sequences of six or more ones in the data to be transmitted by inserting a zero after every sequence of five ones. Then, on receipt, it recovers the data by removing every zero found after a seqeunce of five ones. Bit stuffing is also appplied to the frame checking sequence.

The use of a flag (as herein defined) to determine the start end of a message, together with bit stuffing, is well known and is, for example, a part of HDLC, the high-level data link control procedure that is the subject of the International Standard ISO 3309-1976.

The flag following the frame check sequence is followed by an acknowledgement field ACK. This field is used for the acknowledgement scheme and is set by the originating station to hold a count equal to the number of stations taking part in the scheme and expected to receive the message.

The acknowledgement field is followed by a flag, which concludes the useful data included in the message. A normal token ring would at this point generate the token to allow the next unit to transmit as soon as possible. However, in this system the token is withheld and zeroes are output while the message circulates round the ring.

Each interface unit which is not transmitting monitors the data on the ring for the connector. When it detects the connector it knows that a message is starting. There may be additional leading flags; when data that is not a flag is detected the significant past of the data has started to arrive and the interface unit starts to transfer it to the controller 6. It also deletes inserted zeros and starts to recalculate the frame checking sequence. The next flag indicates that the transmitted frame checking sequence has been received, and that the next field is the acknowledgement field. Assume the station is taking part in the acknowledgement scheme. Then if its value is the one expected if the data has been correctly transmitted, and the station has not failed in some other way to accept the message, the interface unit acknowledges successful receipt by decreasing the count in the acknowledgement field by one. If an error is detected or the message cannot be accepted the field is left unchanged. The same is done at each of the other destination stations in the acknowledgement scheme.

When the message has made a complete circuit of the ring it returns to the originating station. That monitors the incoming data to detect the acknowledgement field but does not re-emit the data. If the count is zero the interface unit assumes that all the expected stations in the acknowledgement scheme have received the message successfully and it outputs the token onto the ring for the next station that wishes to transmit. But if the acknowledgement field does not contain zero it knows that an error has occurred and may then, since it is still in control of the ring, carry out a recovery procedure. For example it may send a repeat of the message, indicated as such by a control field, to each expected destination station in turn. Each station responds immediately to its message to indicate whether or not it needed the re-transmission and if it did whether it has now received it successfully. When all expected destinations have been polled in this manner, if they have all succeeded in receiving the message the token is issued. Otherwise a more drastic action, such as reconfiguration to eliminate a faulty station, may be necessary.

Acknowledgement field

Referring to Figure 4, the acknowledgement field holds the count as eight bits. The first, third, fifth and seventh in order of transmission are the bits of the binary value of the count in sequence, least significant bit first. Each of these bits is followed by its inverse. So in Figure 4 the bit labelled O is transmitted first and holds the coefficient of 2° in the binary representation of the count, the bit labelled O holds its inverse and so on. Thus 5, for example, is held as 10011001. This format ensures that spurious control codes will not be generated even though the field is outside the section of the message which undergoes bit stuffing.

Interface unit

Referring to Figure 5, the incoming transmission link 2 for a station ends in a receiver 10 which extracts a data signal and a clock signal from the signal transmitted over the link. The clock signal derived from the incoming data is used by the interface unit only in receiving incoming data but also in transmitting its own outgoing data.

The construction of the ring may be as described in copending European Application No. 83300716.4. One property of this ring is that if a station is not operative it is bypassed by the ring.

Data from the receiver 10 is entered in the interface unit 4 into a serial input stage 11. Data for the ring is held in a serial output stage 12 from where is passes to a transmitter 13 at the start of the outgoing transmission link 2.

Data for the serial output stage 12 is selected by a multiplexer 14 from either a path 15 or a path 16.

The path 15 is connected to the serial input stage 11 and allows data on the ring to loop through the interface unit. It contains an acknowledgement update circuit 17 which normally leaves data unchanged but can decrease the acknowledgement count by one when required. The path 15 also contains a token converter 18 which also normally leaves data unchanged but when the interface unit wishes to transmit converts the final one bit of each of the go-aheads in the token to the zero value appropriate to a connector.

Data to be transmitted from the station is supplied to the interface unit 4 by the controller 6 over a highway 19. The controller 6 is a processor-controlled unit that buffers the data to be transferred between the interface unit 4 and the appropriate device 5 and exchanges control signals with the interface unit 4. Data supplied over the highway 19 is entered into a serial transmit register 20.

The register 20 may alternatively, when required, receive the flag or go-ahead pattern, as selected by a multiplexer 21.

The contents of the serial transmit register 20 are clocked out serially a bit at a time onto the line 16, from where they go to the serial output stage 12 via the multiplexer 14. They are also entered into a frame checking sequence (FCS) circuit 22 and, when the attached device has supplied all its data, the frame checking sequence is unloaded from the circuit 22 onto the line 16. The frame checking sequence may be a modified cyclic redundancy check of the kind described in ISO 3309-1976. Operation of such circuits is well known and the construction of the circuit will not be described further.

To allow bit stuffing a circuit 23 contains a counter, reset by every zero, which counts ones in the bit stream on the line 16 and issues a ZERO INSERT signal each time it detects five consecutive ones. Operation of this circuit is inhibited if flags or go-aheads are being output on the lines 16. The ZERO INSERT signal prevents data being clocked out of the serial transmit register 20 or FCS circuit 22 (whichever is the current source) for one beat, and thus inserts a zero into the bit stream output onto the ring.

The incoming serial data passes over a path 24 to a serial receive register 25. It also passes to a 5,6,7 ones detect circuit 26 which issues signals when it detects the various control codes and also a ZERO DELETE signal when it detects a zero following five ones. The ZERO DELETE signal stops that bit being clocked into the serial receive register 25, which is equivalent to deleting it.

Receipt of two flags indicates th start of a message. When non-flag data starts it is transferred, whenever eight bits have been assembled, from the serial receive register 25 in parallel to a parallel receive register 26 from where it passes to the controller over a highway 27. It is also transferred from the serial receive register 25 into the FCS circuit 22. That data is therefore also stripped of inserted zeros. The next flag indicates that the FCS itself has now been entered into the FCS circuit 22 (but not the flag, which is still held in the serial receive register). Provided the data has not been corrupted the FCS circuit will now hold a predetermined pattern. If it holds any other valve an FCS FAIL signal will be produced indicating that the received data is not reliable.

If the message is one that has been originally transmitted from this station the acknowledgement field willl be checked in a circuit 28 following the receipt of the flag after the FCS. If the acknowledgement field holds other than a count of zero an ACK FAIL signal will be output to the controller and recovery action will be taken as described. If the ACK FAIL signal is not produced it is assumed that all destination stations have acknowledged receipt.

Acknowledgement update circuit

Referring to Figure 6, in the acknowledgement update circuit 17 the serial data on the path 15 is input to an exclusive-OR gate 30. The gate 30 inverts the data if there is a one on its other input, a line 31, but otherwise leaves it unchanged. It can therefore only change the data if a number of signals applied to an AND gate 32 connected to the line 31 are in the right state. First, a signal UDACK to update the acknowledgement field, produced when a station is receiving data and in response to the flag following the FCS, must be present. This indicates that the right part of the message has been reached. Secondly the FCS FAIL signal from the FCS circuit 22 must be absent. Thirdly a signal DONACK from the controller must be absent. This signal is asserted either long-term if the station does not wish to take part in the acknowledgement scheme or for a particular message if the controller does not have buffer space to accept it. These last two signals are the means by which the station controls the update circuit 17 to withhold acknowledgement.

Provided acknowledgement is to take place the imput 31 of the exclusive-OR gate 30 receives the output of a bistable 33. Now the change to the acknowledgement field required is to reduce the count it holds by one. The bistable 33 is preset to hold a one, and the first bit of the acknowledgement field (the least significant of the biary representation of the count) is therefore changed by the exclusive-OR gate 30 from one to zero or zero to one, as required.

The output of the bistable 33 is also applied to two AND gates 34 and 325 which are enabled in turn by an alternating bit stream 1010.... The bistable 33 is therefore reloaded through the gate 34 with its initial one for the second bit of the serial stream, which is the mirror to the first bit and is also changed by the exclusive-OR gate 30. The AND gate 35 is then enabled and will reload the contents of the bistable 33 only if the signal on the line 15 is a one. This bit is however the mirror bit. If the initial bit was a one, the mirror bit will be a zero, the bistable 33 will be cleared, and from then on the serial stream will be unchanged, which is the effect required. A count of 5, for example, held as 10011001 will be changed to 01011001, representing 4. If the initial bit was a zero, the bistable 33 is reloaded with a one for the third and fourth bits, which will also be changed. In fact the value held in the bistable 33 can be regarded as a borrrow which is propagated while the unmirrored bits of the binary representation are zero until it meets a one, which with its mirror bit is inverted and the borrow then extinguished. Thus a count of 4, 01011001, is changed to 10100101, which is 3 as required.

5,6,7 ones detect circuit

Referring to Figure 7, the 5,6,7 ones detect circuit applies the serial input data received from the path 24 to a counter 40, which is counted up by each one nd reset by each zero. It therefore holds a count of the number consecutive ones received. It outputs a signal when a count of five is reached; this signal and the inverse of the current data bit are applied to an AND gate 41 which outputs the ZERO DELETE signal if the bit following the five ones is a zero.

The counter 40 outputs another signal when it detects six ones. This signal is applied to an AND gate 42 with the current data bit and to an AND gate 43 with its inverse. The AND gate 42 outputs the go-ahead recognition signal GA when the bit following the six ones is itself one and the flag recognition signal FLAG when it is zero.

The arrangement is chosen to give as early a recognition as possible of the bit sequences requiring action.

The GA signal sets a bistable 44 so that two successive go-aheads set a bistable 45 to produce a signal TOKEN to indicate the detection of a token. Two successive FLAG signals similarly set a bistable 46 to output a CONNECTOR signal indicating the detection of the connector. A bistable 47 is then set to indicate the start of significant data, additional leading flags being eliminated by combining the CONNECTOR signal in an AND gate 48 with the inverse of the FLAG signal and a SELECT signal that is produced only at those times a FLAG signal might occur.

Acknowledgement checking circuit

Referring to Figure 8 the acknowledgement checking circuit 28 produces the ACK FAIL signal from a bistable 50. This bistable is enabled to be set by a signal ACK CHECK applied to an AND gate 51 and produced, providing the unit is transmitting, in response to the flag following the FCS and for the duration of the acknowledgement field. The serial input data stream is applied to a comparator 52 together with the alternating stream 1010..... A count of zero is represented by 01010101 and therefore will never yield equality and an output signal from the comparator 52. But any departure from this pattern will produce a signal that sets the bistable 50 and causes the ACK FAIL signal to be passed to the controller.

Token-to-connector converting circuit

Referring to Figure 9, broadly, the token converter circuit 18 applies the serial input data on the line 15 and the inverse of the go-ahead recognition signal GA to an AND gate 60, which is thus disabled when the go-ahead is recognised, preventing the bit following the six ones from being entered into the serial output stage 12. It is thus changed from one to zero, converting the go-ahead to a flag. This action is normally enabled by a transmit request signal from the controller on a line 61. However, the action must be carried out for either both go-aheads of the token or neither. If the controller requests transmission after one go-ahead has already passed the next must not be changed. To ensure this requirement is met the GA signal, preserved in a bistable 62, passes through a gate 63 and sets a bistable 64 if it arrives before the transmit request. That inhibits the application of the next GA signal to the AND gate 60 and also inhibits transmission of the message in response to detection of the token.

General

The acknowledgement field is supplied by the controller set to the expected number of destination stations in the acknowledgement scheme. For example if one of the set which would normally receive the message is inoperative and by-passed by the ring, the count will be adjusted accordingly. The scheme is simple to operate because the action each station must take to carry out the acknowledgement is essentially the same - to decrease the count by one - and does not depend on the position or address of the station. And it is very simple to check, since the final value of the ACK field does not depend on which particular stations are concerned.

Since each station must make a positive acknowledgement, if the count returns as zero the originating station has a high level of assurance that all the expected stations are present and have accepted the message. If any station becomes inoperative and drops out of the ring the originating station cannot remain unaware of the fact since the final value of the count will change. The reason may then be discovered using the recovery procedure and stations will ajust the number of stations they expect to reply.

As has been mentioned, not every station need take part in the acknowledgement scheme. For instance in a configuration like that shown in Figure 1, it might be preferred to restrict it to just the CPUs and use separate messages for acknowledging disc transfers. In that case DONACK will be permanently set in a station not in the group participating in the acknowledgement scheme. Alternatively it may have a simpler interface unit with no acknowledgement update or checking circuit. Messages from a station not taking part should include a dummy acknowledgement field which may be changed by participating stations but will not be checked.

As so far described, a participating station asserts DONACK if it cannot accept a message. That leads to a system in which acknowledgements are expected from all participating stations. However, DONACK can in addition be asserted unless the controller recognises its address in the message. That leads to a system in which acknowledgements are expected only from stations to which the message is addressed.

For greater security additional checks on the data may be performed. For example if the data part of messages (prior to bit stuffing) should equal a whole number of bytes the received data after bit stripping may be counted to ensure that this condition is still met.

Since the acknowledgement field is outside the FCS-checked part of the message and errors introduced on transmission will not be detected, the acknowledgement field may be repeated, for example being sent three times in all. The count must then be zero each time if successful receipt is to be assumed.

The system described combines error checking and message acceptance into one acknowledgement field. If desired a separate count may be used for each. And an additional count may be introduced for address recognition, a station acknowledging in this count when it recognises that it is addressed by an individually addressed or broadcast message whether or not it can accept the message. It may then be desired to transfer the latter count to the body of originating station to allow it to ascertain the number of stations active in the ring which respond to a given address. A count may simply acknowledge presence.

It will be realised that while a station is transmitting the ring is broken into a loop starting and ending at that station. The invention may of course be applied in a system arranged as a permanent loop, but only messages originating at the station to which the ends of the loop are connected can be sensibly acknowledged, since messages from other stations will not return past that station.

The token chosen, in which each of two bytes must be changed, has the advantages that it is more secure than either a single go-ahead or one of the same length that is changed at only one bit position. Yet on conversion to a connector it remains compatible with HDLC, which allows multiple flags at the start of a message. It also ensures that the control codes in the middle of messages - the single flags - will be distinguishable from the start of a message, which has multiple flags, without the need to multiply the patterns that must be recognised. Because the go-ahead occurs only in a token the first go-ahead can be changed without the need to delay the signals received in an interface unit for long enough to detect the complete token.

While this embodiment distinguishes control codes by the presence of more than 5 ones, it will be recognised that control codes may be distinguished instead by including violations of the normal binary encoding rule used by the transmitter. A token may include additional information besides the go-aheads. For example, it may start with a priority, a station being enabled to respond to detection of a token to transmit only if the message equals or exceeds the priority of the token. If a token or connector should be corrupted so as to include only a single go-ahead, error recovery procedures in the stations are responsible for detecting the absence of a token for more than a predetermined time and for generating a new token. While it is advantageous for the token to contain two identical go-aheads, the patterns to be changed may if desired by different.

CLAIMS

1. A method of data transmission in a system comprising a plurality of stations connected in a series configuration by transmission links, in which method data is transferred serially in one direction only along the series configuration of stations and a station starts to transmit data from itself in response to detection of a token comprising a plurality of first sequences of signals, each instance of a first sequence occurring only in a token and not elsewhere in transmissions over the transmission links, the station converting the token as it passes through the station into a connector in which each first sequence is changed into a second sequence by altering the signal in the last bit position of the first sequence, the station thereafter transmitting the said data from itself.

2. A method as claimed in claim 1, in which second sequences used as control codes occur elsewhere than at the start of a station's transmission.

3. A method as claimed in claim 1 or claim 2 in which each first sequence is the bit string 01111111 and each second sequence is the bit string 01111110.

4. A method as claimed in claim 1 or claim 2 in which each first sequence contains a violation of the normal binary encoding rule.

5. A method as claimed in any one of the preceding claims in which the first sequences in a token are identical to one another.

6. Interface apparatus suitable for forming the coupling between a station and an incoming and an outgoing serial transmission link, the element comprising:

means for receiving data from the incoming link;

means for outputting data for transmission over the outgoing link;

loop-through means providing a path for the transfer of data from the receiving means to the outputting means;

means for detecting the reception at the receiving means of a token comprising a plurality of first sequences of signals;

means responsive at least to an indication that the station is ready to transmit and the detection of a token for enabling the station to supply data from itself to the output means in place of data from the loop-through path; and

means responsive at least to an indication that the station is ready to transmit for changing a token into a connector, the means changing each first sequence of signals to a second sequence by altering the signal in the last bit position of the sequence;

the last two enumerated means each being inhibited from operating in response to a token if the ready-to-transmit indication is received to late to alter the initial first sequence of the token.

7. Interface apparatus as claimed in claim 6 in which there is provided means responsive to the detection of a second sequence received at the reception means not at the start of a transmission from another station to control operation of the interface apparatus.

8. Interface apparatus as claimed in claim 6 or 7 in which the first sequence is the bit string 01111111 and the second sequence is the bit string 01111110.

9. Interface apparatus as claimed in claim 6 or 7 in which each first sequence contains a violation of the normal binary encoding rules.

10. Apparatus as claimed in any one of claims 6 to 9 in which the first sequences in a token are identical to one another.

11. A data communication system comprising a plurality of stations connected in a series configuration by serial data transmission links, each station being coupled to an incoming and an outgoing one of the links by interface apparatus as claimed in any one of claims 6 to 10.

12. A system as claimed in claim 11 in which the series configuration is a ring.

13. A system as claimed in claim 11 in which the series configuration is a loop connected between an input and an output of an additional station not providing a loop-through path for data through itself.

14. An interface unit substantially as hereinbefore described with reference to Figures 1 to 9 of the accompanying drawings and as shown in Figures 5 to 9 of those drawings.

15. A system substantially as hereinbefore described with reference to Figures 1 to 9 of the accompanying drawings and as shown in Figures 1, and 5 to 9 of those drawings.

Printed for Her Majesty's Stationery Office by The Tweeddale Press Ltd., Berwick-upon-Tweed, 1983.
Published at the Patent Office, 25 Southampton Buildings, London WC2A 1AY, from which copies may be obtained.