

EFFICIENT ERROR HANDLING ON A LINK USING ARQ AND MULTIPLE NACKS ASSOCIATED WITH MULTIPLE ERROR THRESHOLDS

The present invention relates to electronic multi-component systems, methods and devices providing optimized communication on a link with in-order data delivery.

5

Background of the invention

It is desirable in electronic multi-component systems that components used to build up the systems can be seamlessly integrated, thereby keeping the integration effort as low as possible, while at the same time having good performance. Ideally this should be the case independent of who the component manufacturer is. To form a standard, several manufacturers have formed a group which is called the Mobile Industry Processor Interface Alliance (MIPI®). The MIPI Alliance has agreed on a high-speed interface technology called UniProSM which is for instance used for high speed communication between electronic components in mobile devices. In the future it will incorporate Phy symbol encoding which is available in D-Phy which is using 8b9b encoding as well as M-Phy that will be using 8b10b encoding.

In general UniProSM is defined to be a general purpose protocol that deals with solutions of common issues like error handling, flow-control and routing or arbitration.

In the past and at present UniProSM however provides a fairly complex error handling to ensure reliable communication. UniProSM error handling is based on error detection using cyclic redundancy check (CRC), error reporting using Negative Acknowledgement Control (NAC) frames and data frame retransmissions in case of errors. Furthermore sequence numbers ensure that data frames are received in the proper order and prevent frames from being lost or duplicated. Sequence numbers are also used for management of a retransmission buffer on the transmitter side, assisting in removal of the frames from the retransmission buffer once they have been correctly received.

Further to the CRC error detection in UniProSM, timers are used to address error corner cases. At present, two timers are used to protect against the loss of frame acknowledgements and flow-control credits respectively.

When Phy-encoding is used, UniProSM makes use of the Phy-specific error detection to trigger standard UniProSM error recovery mechanism.

Summary of the Invention

It is an object of the present invention to improve an error handling performance in a communication protocol.

A first aspect of the invention is a method for handling an error at a receiver in a communication system that uses a frame-based communication protocol having a first error handling mechanism (error recovery protocol) responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol. The method comprises:

- 5 - receiving, via a receiver, a sequence of symbols and determining whether a count of incorrect or non-expected symbols in the sequence equals or exceeds a first error threshold greater than 1, and in the affirmative the receiver initiating the first error handling mechanism.

10 The receiver thus triggers the error recovery protocol (the error recovery protocol is initiated) when multiple incorrect or unexpected symbols are received within a sequence of symbols. A significant advantage of this method is that the first error recovery protocol is invoked less than it otherwise would. Known methods will employ the error recovery protocol in response to a single incorrect/non-expected symbol.

15

 In some embodiments, wherein the first error handling mechanism involves the receiver sending a first Negative Acknowledgement Control (NAC) frame to a transmitter responsible for sending the sequence of symbols, the first NAC not requiring a re-synchronization between the transmitter and the receiver.

20 In some embodiments, the communication protocol also has a second error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol, and the method further comprises:

- determining whether a count of incorrect or non-expected symbols in the sequence equals or exceeds a second error threshold higher than the first error
25 threshold, and in the affirmative the receiver initiating the second error handling mechanism.

 This embodiment is advantageous because it causes the second error handling mechanism to be initiated only when the number of errors equals or exceeds the
30 second error threshold, in a sense "postponing" it.

 In some embodiments, the second error handling mechanism involves the receiver sending a second Negative Acknowledgement Control (NAC) frame to the transmitter responsible for sending the sequence of symbols, and that second NAC
35 requires a re-synchronization between the transmitter and the receiver. This embodiment further illustrates the advantage of using a second threshold for a second error handling mechanism. In the case where the second error handling mechanism

involves re-synchronization, the invention provides that this action is not taken until a higher number of errors have occurred. An example of a value of the second threshold could be 10 or 50 or 100, depending on the requirements in the given application/environment.

5

In some embodiments, the error handling mechanism is triggered if the sequence comprises two adjacent incorrect or non-expected symbols. This has turned out to improve error handling substantially, especially under more noisy conditions where electromagnetic interference (EMI) causes a significant number of errors.

10 Triggering the error recovery protocol after three adjacent incorrect or non-expected symbols is advantageous for more efficiently handling serious errors. Adjacent errors could be indicative of something other than EMI, but if the problem is significant but intermittent, then an acceptance of multiple errors before invoking the error recovery protocol can improve performance significantly. Other values of the first threshold can
15 be selected, for instance with the goal of optimizing the error handling in a certain application/environment.

In another embodiment, the count is determined in between a complete frame has been received and another one not yet begun. In between frames, more errors can be tolerated, and the invention can therefore adhere to standard error handling within
20 frames and a less strict handling between frames. From a standards perspective, this is a very attractive embodiment feature.

In another embodiment, the sequence in which errors are counted is received after any outstanding frames have been completely received. In another embodiment the sequence of incorrect or non-expected symbols is received after a last symbol of a
25 frame was received and before an ongoing frame is continued with a continuation-of-frame (COF) symbol. This particular embodiment specifically counts errors (in the sense of the invention) in ongoing frames.

The methods described above are highly relevant when the communication protocol is Unified Protocol, UniPro. In some embodiments, then, the error recovery
30 protocol is an error recovery protocol in the sense of the UniPro standard. Other protocols, or developments of UniPro, are likely to be compatible with the present invention and thus the invention will be applicable for such protocols as well.

A second aspect of the invention provides a receiver symbol parser for use in a communication system that uses a frame-based communication protocol having a first
35 error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol. The symbol parser is adapted to perform a method accordance in accordance with the first aspect of the invention. According to the second aspect of

the invention, the receiver symbol parser is adapted to receive a sequence of symbols and determining whether a count of incorrect or non-expected symbols in the sequence equals or exceeds a first error threshold greater than 1, and in the affirmative initiating the first error handling mechanism.

5 The considerations related to the first aspect apply to the second aspect as well. For instance, in some embodiments of the second aspect, the communication protocol also has a second error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol, and the receiver symbol parser is further adapted to determine whether a count of incorrect or non-expected symbols
10 in the sequence equals or exceeds a second error threshold higher than the first error threshold, and in the affirmative to initiate the second error handling mechanism.

 A third aspect of the invention provides a receiver for use in a communication system that uses a frame-based communication protocol having an error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol
15 symbol. A receiver in accordance with this aspect employs a receiver symbol parser in accordance with the second aspect of the invention.

 The considerations related to the first and second aspects apply to the third aspect as well.

20

Brief description of the drawings

Below embodiments of the present invention are presented, with reference to the examples shown in drawings, wherein

25 Fig. 1 shows an example of a length-based data format,
 Fig. 2 shows an example of an acknowledgement and flow control frame,
 Fig. 3 shows an example of a negative acknowledgment control frame,
 Fig. 4 shows an example of a common receiver symbol parser error detection,
 Fig. 5 shows an example of a procedure called following error detection,
30 Fig. 6 shows an example of a procedure called when detecting the start of an NAC frame,

 Fig. 7 presents an example of a procedure called when detecting the start of an AFC frame,

35 Fig. 8 shows an example of a procedure called when detecting the start of a data frame in traffic class 1,

 Fig. 9 gives an example of a procedure called when detecting the start of a data frame on traffic class 0,

Fig. 10 gives a schematic example a receiver symbol parser that employs error handling according to the present invention,

Fig. 11 gives another schematic example of a receiver symbol parser that employs error handling according to the present invention,

5 Fig. 12 shows a configuration of a transmitter and a receiver building up a system interconnected by a serial bus with in-order data delivery

Fig. 13 shows another schematic example of a receiver symbol parser that employs error handling according to the present invention.

10 Description of selected embodiments

The present invention will be described with respect to particular embodiments and with reference to certain drawings but the invention is not limited thereto but only by the claims. The drawings described are only schematic and are non-limiting. In the drawings, the size of some of the elements may be exaggerated and not drawn on
15 scale for illustrative purposes.

Where the term "comprising" is used in the present description and claims, it does not exclude other elements or steps. Furthermore, the terms first, second, third and the like in the description and in the claims, are used for distinguishing between similar elements and not necessarily for describing a sequential or chronological order.
20 It is to be understood that the terms so used are interchangeable under appropriate circumstances and that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein.

As shown in Fig. 1 the data frame format based on frame length may be used in the UniProSM version 1.5. Throughout the description of the drawings the same
25 reference numerals are used for the same descriptive items and a redundant description thereof is omitted for the sake of efficiency.

Here the frame length is used instead of the EOF/EOF_PAD control symbol of a former frame format. The first two symbols and the last symbol of the UniProSM data frame are the frame header and trailer respectively. The header consists of control
30 symbol identified by the 16th bit with a value of 1 which has the following fields: an escape byte 110, which will be converted to an escape symbol in an encoded Phy, and denotes a control symbol, the control ID field CTRLJ D 120, which in the case of a data frame header has the value 0, denoted by SOF (Start of Frame), the traffic class TCx, 130, which may take the value of 0 and 1 at UniProSM 1.5 and furthermore three
35 reserved bits 140. The second header symbol is a data symbol identified by the 16th bit with a value of 0, which has the following fields: the frame length 150, the frame sequence number 170. The frame payload includes the Layers 3 and 4 headers (the

L3s bit 180, the destination device ID 182, the L4s bit 184, the destination port 186, the FCT bit 188 and the EoM bit 190), and the payload 192. The frame trailer consists of the error check code 194.

Fig. 2 gives an example of an acknowledgement and flow control frame format
200. For instance, the Acknowledgement and Flow Control (AFC) frame consists of
5 three symbols: The AFC control symbol where the 16th bit is 1 and two data symbols.
The AFC control symbols consist of an escape byte 210, as for the SOF control symbol
in the header, where the symbol control ID field 220, which has a value of 6 denoted by
AFC, the traffic class TCx 225, for which UniProSM 1.0 allows the values 0 and 1, a Req
10 bit 235 used to request the peer side to send an AFC control frame at this traffic class,
and two reserved bits 235. 240 denotes the frame sequence number, 245 is for
reserved bits and 250 for flow control credits, whereas 255 denotes the cyclic
redundancy check code.

Fig. 3 gives an example of the frame format for a negative acknowledgement
15 control frame 300. This NAC control frame for instance consists of two symbols: the
NAC control symbol, where the 16th bit is 1, and one data symbol, the CRC is used for
error detection. The NAC control symbol consists of: an escape byte 310, as for the
SOF control symbol in the header, the symbol control ID field 315, which has the value
of 5 denoted by NAC (negative acknowledgement), as well as four reserved bits 320
20 and an RRR bit 325 used to request the pier side to resynchronize the link.

Fig. 4 shows an example of a receiver symbol parser for error detection.
Generally for instance the link error handling scheme of UniProSM is based on error
detection and data frame retransmission, and consists of two safety nets. The first
safety net is acting quickly and handles typical errors, whereas the second safety net is
25 slower and associated to the error handling of lost control frames. Generally UniProSM
adds a cyclic redundancy check error detection code CRC to each transmitted frame.
The receiver re-computes the CRC and verifies it against the CRC that was received at
the end of the frame. In case the received frame is correct the receiver commits it to
the receive buffer and acknowledges the frame to the transmitter (for a data frame), or
30 processes it (for a control frame). In case the receiver detects an error, this is reported
back to the transmitter using an NAC frame.

Furthermore all transmitted data frames are assigned a sequence number,
which is also transmitted in the data frame, and is used by the receiver as an
assignment to identify the acknowledgement. For instance, the sequence number has
35 5 bits incremented with every frame and consequently wraps around at 31.

For instance, in UniProSM a maximum of 16 frames can be transmitted without
receiving an acknowledgement. Frame acknowledgement is performed using an ACK

control frame. In case of an acknowledgement, a sequence number in the AFC frame acknowledges all unacknowledged data frames up to and including the one identified by the sequence number. The transmitter saves all the transmitted data frames in a retransmission buffer until they are acknowledged. If an NAC control frame is received, reporting an error, all the data frames in the retransmission buffer are retransmitted starting with the oldest unacknowledged frame.

The second safety net is used if control frames are lost. The consequence is a loss of either acknowledgement and/or link-level flow control credits contained in an AFC frame, or an error indication contained in an NAC frame. In order to handle such errors, which rarely occur, two timers are used for each traffic class: one for data, called replay timer, and one for credits, called credit timer. In case the replay timer expires, all the unacknowledged frames in that traffic class are retransmitted. In case the credit timer expires, an AFC frame with the Req bit set is sent for that traffic class. However, in both cases UniProSM version 1.0 requires the link to be resynchronized. An NAC frame with the RRR bit is sent to also require the incoming link to be resynchronized, an activity that requires bandwidth and lowers the efficiency of the connection.

Apart from the CRC errors for instance numerous other errors may be detected in case of the first safety net by the receiver which errors can be: a receiver buffer overflow for instance due to a bit transmission error in the TCx field 130; an incoming frame having a larger length than the maximum frame size due to a transmission error changing the EOF control symbol into a valid data symbol; incorrect sequence number in a data frame; an AFC frame not followed by two data symbols; an NAC frame not followed by one data symbol; an EOF/EOF_PAD not followed by one data symbol; reception of a COF, EOF or EOF_PAD control symbol when no data frame has been started; reception of an SOF symbol when the data frame of the same traffic class is already ongoing and the data frame is not currently pre-empted; reception of an SOF symbol called TC0, in case a TC1 data frame is already ongoing; reception of a COF symbol during a data frame of a same traffic class, wherein the data frame has not been pre-empted; reception of a COF symbol continuing a data frame of a different traffic class; reception of an EOF, EOF_PAD or data symbol when the data frame is pre-empted and the pre-empted frame has finished; reception of a control symbol with invalid values for defined fields, for instance undefined CTRLJD or TC; reception of an error indication from the Phy adapter.

In case of Phy-encoding the Phy adapter indicates an error for instance in the following cases:

a bad Phy symbol was received,

a correct exception Phy symbol was received, but it is not mapped to a valid UniProSM symbol,

a correct exception Phy symbol was received immediately after another exception Phy symbol, when a data Phy symbol was expected, and

5 an incorrect value or valid data Phy symbol is received following an ESC_PA.

In case of any of the above errors being detected, an NAC control frame is transmitted, which may be preceded by a pair of AFC control frames to prevent unnecessary retransmissions.

Fig. 4 gives an example of an embodiment of a part of the UniProSM error
10 handling mechanism.

A top level view of a symbol parser as it may be implemented in the receiver is described. The symbol parser maintains three variables tc0 and tc1 in case there is an ongoing data frame at traffic class TCO and TC1 respectively, and a variable NACGenOk in order to indicate if the NAC generation is validated indicated by a true,
15 or not, indicated by false as marked by reference sign 401 .

As there is no ongoing data frame initially, hence TCO and TC1 are both set to false. Also NAC generation is temporarily disabled after an error was reported with an NAC control frame until a frame with a good CRC is received. Initially there is no detected error and thus NACGenOk is set to true. As initially the symbol parser has not
20 started any frame it may for instance wait in a state NoFrame indicated by reference sign 403. In the first line the start symbols are indicated 405 "NAC", 407 "AFC", 409 "SOF/TC=1", 411 "SOF/TC=0", 413 "DATA", 415 "COF.EOF" and 417 "PAErr" which call the respective procedures when received described below and connected by arrows respectively as indicated with reference signs from 425 to 437. If any other
25 symbol is received, an error procedure is called with the parameter false. For any state, when an error indication is received from the Phy adapter, an error procedure is called with the parameter *false* as indicated by reference signs 450 to 460.

Fig. 5 shows an error procedure. It takes a parameter RRR indicated by 505 and determines whether NAC generation is enabled (step 510). If this is not the case
30 no NAC frame is generated, and the symbol parser returns to state NoFrame at 530. If NAC generation is enabled, an NAC frame is triggered at 515 and the NACGenOk value is set to false at 520. Then the symbol parser returns to state "NoFrame" at 525.

Fig. 6 shows an example of a procedure that is called when the start of an NAC frame is detected as indicated at 605. The procedure waits in a state 610 "NAC" until
35 the data symbol, the CRC is received at 615. In case of another symbol being received at 630 an error routine is called at 635. If the computed CRC at 620 is not correct an error is as well reported at 625. If it is correct 640 the value of the parameter NACGen

Ok is set to 2 at 645. Then at 650 the traffic class having a higher priority, TC1, is evaluated and in this case it is branched to the state of 655 "TC1_CoF" whereas on the other hand if this is not the case it is evaluated if the lower traffic class, TC0, is applicable at 660. If this is not the case the state of 670 "NoFrame" is assumed. On the other hand if TC0 is applicable the state 665 "TC0_CoF" is adopted.

Fig. 7 gives an example of a procedure that for instance is called wherein, if the start of an AFC frame is detected in the beginning at 705 this procedure is called and transits into a state 710 "AFC" where it is waiting on the reception of data. In this case it is expecting the reception of the second data symbol of the AFC frame. In case another symbol is received at 775 an error routine is called at 780. If the correct symbol is received then the state 720 is assumed by the parser "AFC_crc". Again if any other symbol is received at 785 then an error routine is called at 790. If the second symbol a data symbol, which is actually the AFC frames' CRC then the data are verified and if they are correct the procedure 740, "AFC_crcOK" is called. Else the error procedure is called at 735. In case of a correct CRC computation and comparison result the AFC frame is committed with AFCcrcOK, where the proper actions following the AFC receipt are taken at 740. Following this activity, NACGenOk is set to true at 745 and then as explained at the previous figure the traffic classes are evaluated. Here the explanation for reference signs 750 to 770 are the same as the explanations given at Fig. 6 for reference signs 650 to 670.

Fig. 8 gives an example of a procedure called when detecting the start of a data frame associated to TC1. In this case after the parser receives an SOF symbol associated to the traffic class which is set to 1, the procedure for handling of the associated traffic class TC1 is called at 803. The procedure first sets the tc1 variable to true at 808 to indicate that a TCV1 frame has started, and then moves to state TC1_data at 823. In state 823 "TC1_DATA", the procedure is in the progress of receiving corresponding data symbols. While data is received at 835, two errors can occur which are not shown in the figure and relate to a buffer overflow and a frame exceeding the DL_MTU, which lead to the reporting of "Error". If no error occurred when processing the data symbol, the procedure returns to state TC1_data at 823. From the state 823 the frame can be ended with an EOF symbol or correspondingly with an EOF_PAD symbol, which is not shown here. After an EOF indicated by 838 a CRC is expected at 848. The frame is finished in which case TC1 becomes false at 863. The CRC is evaluated at 868 if any other symbol is received at 860 this is reported as an error at 865. The frame is committed in case the CRC check proves that at 868 otherwise an error is indicated at 870. In case of a correct error evaluation at 873 the variable NACGenOk is set to true at 875 then the TC0 data frame pre-emption

is checked using tcO at 878. If this is the case the TCO data frame reception is continued in a state 880 "TC0_CoF". Otherwise the parser returns to the state 885 of "NoFrame". In case of NAC symbols 813, AFC symbols 815, SOF symbols 818 or wrong symbols 820 being received corresponding procedures 825 to 833 as indicated by the connecting arrows are called. If in the state 823 a pre-emption occurs due to receiving NAC or AFC symbols the corresponding procedures are called 840 causes 850 to be called and 843 causes 853 to be called as is the case for 840 and 850 as well as 845 and 855. Any other symbol received in the state 823 causes the report of an error. In case a TC1 data frame was pre-empted and the pre-empting frame is finished, the parser turns to the state 805 "TC1_CoF". In this state the frame can be continued while a CoF symbol is received with the TCx field set to 1 as indicated at 810, and leads to returning to the state 823. From state 805, another pre-emption may occur by an NAC or AFC symbol, associated to 813 and 815. Any other symbol will be reported under using the error procedure 833, or an SOF symbol with the TC field set to 1 can be received indicating potential retransmission.

Fig. 9 shows a procedure that is called for instance when the start of data frame for traffic class 0 is detected. This procedure is very similar to the procedure for the higher traffic class TC1 apart from the evaluation for traffic class 0 at 878 and from there returning to the state 880. Consequently the description given for Fig. 8 applies with the exception, that wherever TC1 appears it should be replaced by the TCO as the drawings are virtually identical apart from the different traffic classes that are handled a double description thereof is omitted for the sake of efficiency.

Here however TCO frame may be pre-empted by a TC1 frame which is shown by allowing an SOF symbol with the TC field being set to 1 at 918 to be received during the pre-emption of the state 905 "TC0_CoF" or during the state 925 "TC0JDATA" where the SOF is indicated by 950. Secondly when the TCO frame is finished the parser preferably always returns to the state 998 "NoFrame" as there is no other frame that can be continued contrary to the TC1 case, TCO could be pre-empted, therefore a check is needed to determine if a TCO frame needs to be continued in state 905 "TC0_CoF", or there is no ongoing frame in which case the parser returns to the state 988.

Fig. 10 schematically shows an example of a receiver symbol parser that employs error handling according to the present invention, which allows a data symbol and/or occasional errors in-between frames.

Allowing data symbols in-between frames is an advantageous alternative to sending IDLE symbols, which is good for EMI reasons, as it allows more randomness

in the Phy symbol patterns, as opposed to a repetition creating unwanted spectral harmonics.

On the other hand preferably also occasional errors in-between frames may be ignored, as on the one hand they are likely not to be related to useful data. If on the other hand errors would hide a useful symbol, the UniProSM error handling scheme would demonstrate to be robust enough to detect the error at a later detection stage. However, in most of the cases that were observed, the Phy transmission error does not need to be reported, and bandwidth and/or power use is thereby reduced.

Apart from the modifications this figure is directly comparable to the parser shown in Fig. 4 and explained there for instance box 401 compares to box 1005 whereas the first line from 1010 to 1023 compares to the first line of Fig. 4 from 405 to 417, concerning the received symbols. The last line from 425 to 431 directly compares to the boxes in Fig. 10 from 1058 to 1065 and therefore requires no additional comment or explanation. This also holds true for the flow shown from 450 to 460 which directly compares to the flow from 1070 to 1080.

The states "COF", "EOF", "PAErr", "PAIdle" and "DATA" indicated by 1020 to 1025 are handled in the state 1008 "NoFrame". I.e. in-between frames a counter "ErrCnt" is used to count errors in-between frames. When a frame started at 1010 - 1018, or when a "PAIdle" 1025 is received the "ErrCnt" 1043 is reset to 0 as shown from 1028 to 1038, if the Phy signals an error via "PAErr", 1023 the "ErrCnt" is incremented, and preferably if it reaches a first threshold TH_PA_ERR for instance like 3 a Phy resynchronization is triggered in using "Error (false)", 1053 if the error persists and ErrCnt reaches a second threshold an NAC frame requesting Phy resynchronization is triggered using "Error (true)" as shown in 1055.

This scheme can further be improved by allowing unexpected control symbols to be received like COF or EOF in-between frames, where the error counting is done as it is done for "PAErr" like shown at box 1020.

Typically PAErr indicates a decoding error issued by the Phy Adapter, and may e.g. consist of one of the following possible errors:

- BAD_PHY_SYMBOL: An invalid Phy encoded symbol was received from the Phy. For Phy's using the 8b9b encoding (e.g., MIPI D-PHY) or 8b10b encoding (MIPI M-PHY), this means that an invalid 9-bit or 10-bit was received, respectively.
- UNMAPPED_PHY_ESC_SYMBOL: A valid symbol according to the Phy encoding was received, however the symbol is used neither by the Phy nor by the protocol. Being unused, such a symbol is e.g. not transmitted, and, therefore, receiving one indicates a transmission error.

- UNEXPECTED_PHY_ESC_SYMBOL: A valid and used Phy encoded symbol is received, however at an unexpected time. For UniPro v1.10, this means receiving C400, C417 or C600 as defined by the MIPI D-PHY for PA_PDU[7:0] - bits [7:0] in Figures 1-3.
- 5 - BAD_PA_PARAM: For UniPro v1.10, this error indicates a Phy Adapter symbol (e.g. introduced by ESC_PA in bits [15:0] of a symbol, where the bit combination for bits [7:0] is invalid. This is conventionally the only detectable decoding error in the case of a Phy without symbol encoding .

10 As it is shown here both types of errors are combined.

Fig. 13 schematically shows another receiver symbol parser that employs error handling in accordance with the invention. It is in many ways similar to the parser in Fig. 10. However, for further flexibility in the error handling and detection it may be desirable to allow data symbols (1324 in Fig. 13) to be received in-between frames for
 15 e.g. EMI reasons. If the system is configured by setting the Cfg_DataErr (1341 in Fig. 13) to false (1347 in Fig. 13) to accept data symbols in-between frames, the error counter is kept to 0 (1043 in Fig. 13) when such a data symbol is received in-between frames. These data symbols are simply introduced on purpose by the transmitter to preferably prevent the Phy from autonomously introducing IDLE symbols because of
 20 lack of data to be transmitted. If Cfg_DataErr is set to true (1346 in Fig. 13), meaning that data symbols should not be sent in-between frames, data symbols are treated like the other unexpected events in-between frames (PAErr, COF and EOF) by incrementing the ErrCnt (1040 in Fig. 13) and triggering NAC transmission without or with Phy resynchronization when ErrCnt exceeds TH_PA_ERR and MAX_PA_ERR,
 25 respectively. Since in this embodiment data and PAIdle are treated differently, element 1025 in Fig. 10 is substituted with element 1326, which deals with PAIdle, see Fig. 13.

The same mechanisms may also be applied in the same manner to the states "TC1_COF" and "TC0_COF", where UniProSM is in-between frames during a pre-emption and expects a COF symbol. The difference to the above is that the COF and
 30 SOF symbols have opposite roles, as the COF symbol is an expected symbol and the SOF symbol is not. As can also be seen in Fig. 10 in the initialization of the procedure the ErrCnt is defined 0 as indicated in 1005 and the corresponding value is checked before calling up corresponding procedures in response to the reception of respective symbols as shown in 1028, 1030, 1035 and 1038.

35 Fig. 11 schematically shows another example of an embodiment of a receiver symbol parser for error handling according to the present invention. Apart from box 1105, where a variable "BadCnt" is defined to 0 the figure is identical from 1110 to

1138 with Fig. 10 from 1010 to 1065. Therefore with this regard it requires no further explanation. This also holds true for the explanations regarding 1070 to 1080 which correspond to the reference numerals 1180 to 1185 in Fig. 11.

Contrary to the previous embodiment here the "PAErr" 1143 is defined in form of bad symbol error indications signalling a bad symbol and is treated differently from the other errors which are triggered by valid Phy symbols received when not expected. The bad Phy symbols are treated as before explained corresponding to Fig. 10 at reference signs 1040, 1045, 1050, 1053 and 1055. Herein states using a counter BadCnt and two thresholds TH_PA_BAD which may be for instance set to 3 and MAX_PA_BAD for instance 100 are used. In this embodiment other errors preferably are counted with the ErrCnts counter 1155 and when they reach a threshold MAX_PA_ERR, which for instance is set to 3 they trigger the transmission of a NAC frame without Phy resynchronization as shown with 1178. In this case no resynchronization is needed, because the received Phy symbols are valid. Also here like it was explained for the first embodiment a similar scheme can be applied for the states 1145 "TC1_CoF" and "TC0_CoF" for instance in-between frames during a pre-emption.

Beneficially the invention may be applied to UniProSM network-based products for use in mobile devices, such as mobile phones.

Fig. 12 shows a configuration built from a transmitter 1215 and a receiver 1225 wherein this system is indicated by 1200. The receiver is in accordance with the invention and employs a receiver parser in accordance with the invention. The transmitter comprises an active component 1210 and a bus interface to a serial bus with in-order data delivery 1217. The bus itself is marked by 12500 and connects the transmitter 1215 and the receiver 1225 via their interfaces 1217 and 1219 the receiver is also shown to have an active device such as a processing unit indicated by 1220. For instance, the transmitter may comprise a processor 1210 that inserts valid data symbols via a data scheduler to the bus 12500, whereas the receiver may have a processor running the program for a receiver parser in the active component 1220.

30

Claims

1. A method for handling an error at a receiver in a communication system, the communication system using a frame-based communication protocol having a first error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol, the method comprising:
- 5 - receiving, via a receiver, a sequence of symbols and determining whether a count of incorrect or non-expected symbols in the sequence equals or exceeds a first error threshold greater than 1, and in the affirmative the receiver initiating the first error handling mechanism.
- 10 2. A method in accordance with claim 1, wherein the communication protocol has a second error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol, the method further comprising:
- 15 - determining whether a count of incorrect or non-expected symbols in the sequence equals or exceeds a second error threshold higher than the first error threshold, and in the affirmative the receiver initiating the second error handling mechanism.
- 20 3. A method in accordance with claim 1, wherein the error handling mechanism is initiated if the sequence comprises two adjacent incorrect or non-expected symbols.
- 25 4. A method in accordance with claim 1, wherein the error handling mechanism is initiated if the sequence comprises three adjacent incorrect or non-expected symbols.
5. A method in accordance with one of the preceding claims, wherein the sequence of incorrect or non-expected symbols is received after any outstanding frames have been completely received.
- 30 6. A method in accordance with any of the claims 1 to 5, wherein the sequence of incorrect or non-expected symbols is received after a last symbol of a frame was received and before an ongoing frame is continued with a continuation-of-frame (COF) symbol.
- 35 7. A method in accordance with any of the preceding claims, wherein the communication protocol is Unified Protocol, UniPro.

8. A method in accordance with claim 7, wherein a frame is a UniPro data frame or UniPro control frame.

9. A method in accordance with any of the preceding claims, wherein the first error

5 handling mechanism involves the receiver sending a first Negative Acknowledgement Control (NAC) frame to a transmitter responsible for sending the sequence of symbols, the first NAC not requiring a re-synchronization between the transmitter and the receiver.

10 10. A method in accordance with any of claims 2-9, wherein the second error handling mechanism involves the receiver sending a second Negative Acknowledgement Control (NAC) frame to a transmitter responsible for sending the sequence of symbols, the second NAC requiring a re-synchronization between the transmitter and the receiver.

15 11. A receiver symbol parser for use in a communication system that uses a frame-based communication protocol having a first error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol, wherein the receiver symbol parser is adapted to receive a sequence of symbols and determining
20 whether a count of incorrect or non-expected symbols in the sequence exceeds an error threshold greater than 1, and in the affirmative to initiate the first error handling mechanism.

25 12. A receiver symbol parser in accordance with claim 11, wherein the communication protocol has a second error handling mechanism responsive to receipt of an incorrect or non-expected Phy and/or protocol symbol, and the receiver symbol parser is further adapted to determine whether a count of incorrect or non-expected symbols in the sequence exceeds a second error threshold higher than the first error threshold, and in the affirmative to initiate the second error handling mechanism.

30 13. A receiver symbol parser in accordance with claim 11 or 12, wherein the first error handling mechanism involves sending a first Negative Acknowledgement Control (NAC) frame to a transmitter responsible for sending the sequence of symbols, the first NAC not requiring a re-synchronization between the transmitter and the receiver.

35 14. A receiver symbol parser in accordance with claim 12 or 13, wherein the second error handling mechanism involves sending a second Negative Acknowledgement

Control (NAC) frame to a transmitter responsible for sending the sequence of symbols, the second NAC requiring a re-synchronization between the transmitter and the receiver.

- 5 15. A receiver symbol parser in accordance with one of claims 11-14, wherein the communication protocol is Unified Protocol, UniPro.

16. A receiver for use in a communication system that uses a frame-based communication protocol, characterized in that the receiver employs a receiver symbol
10 parser in accordance with one of claims 11-15.

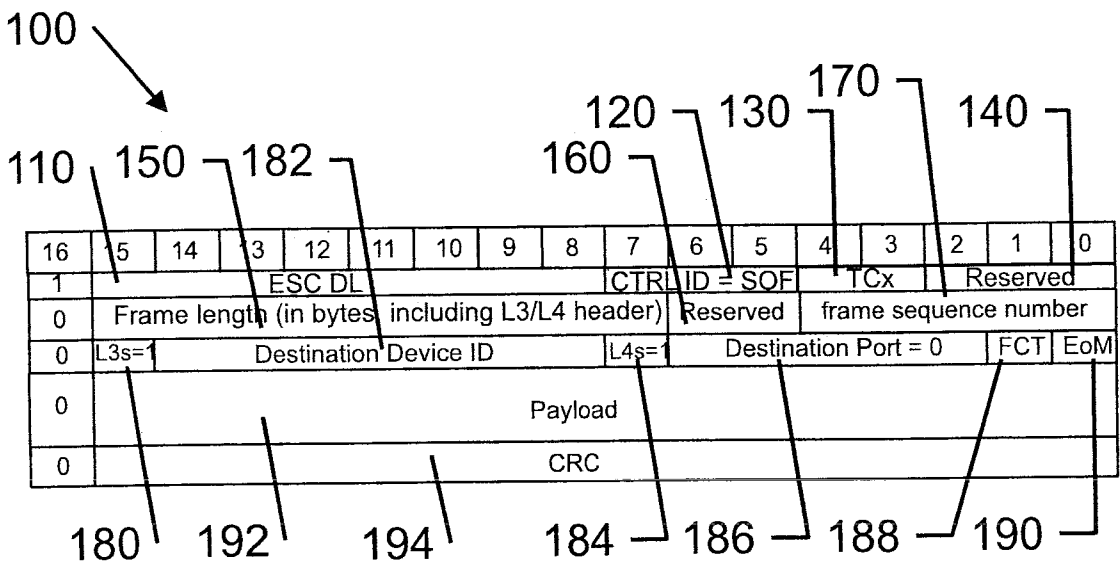


Fig. 1

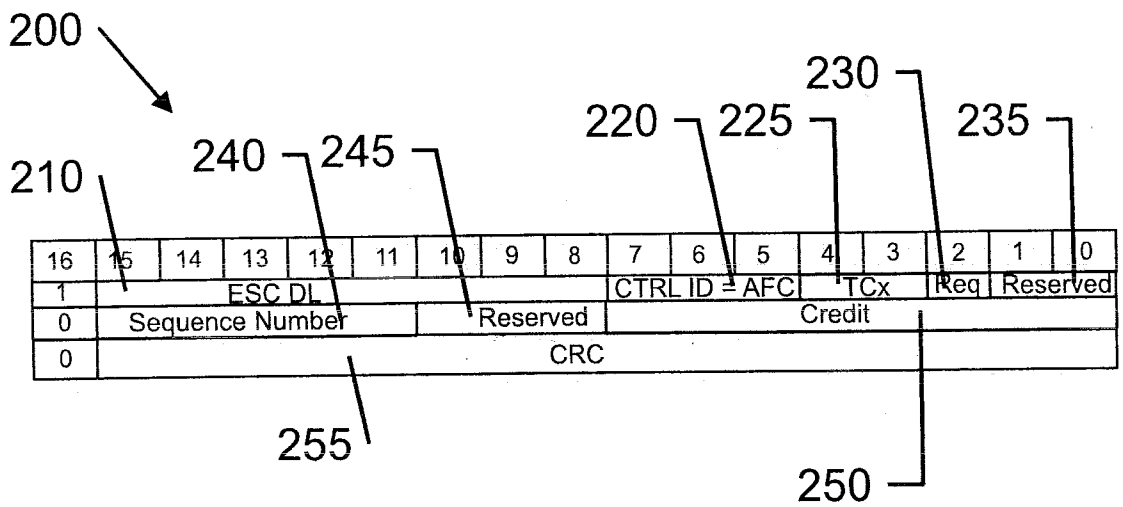


Fig. 2

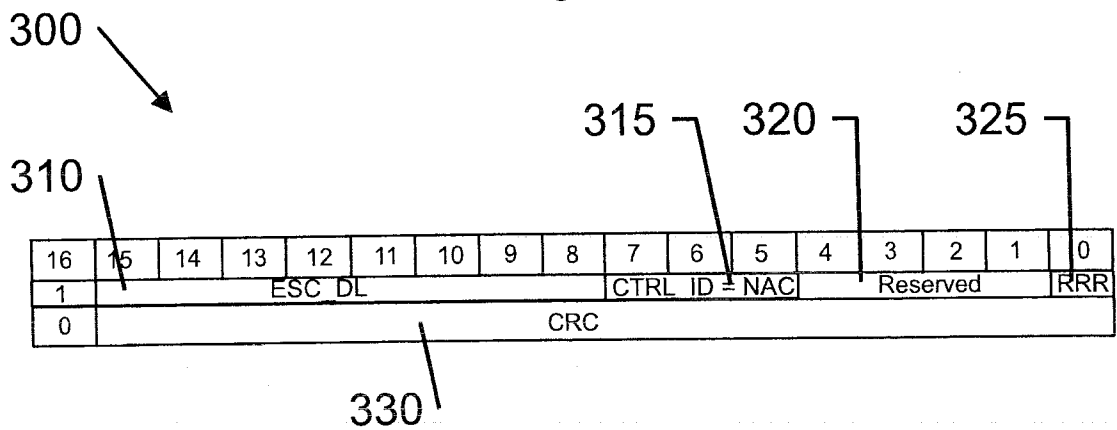


Fig. 3

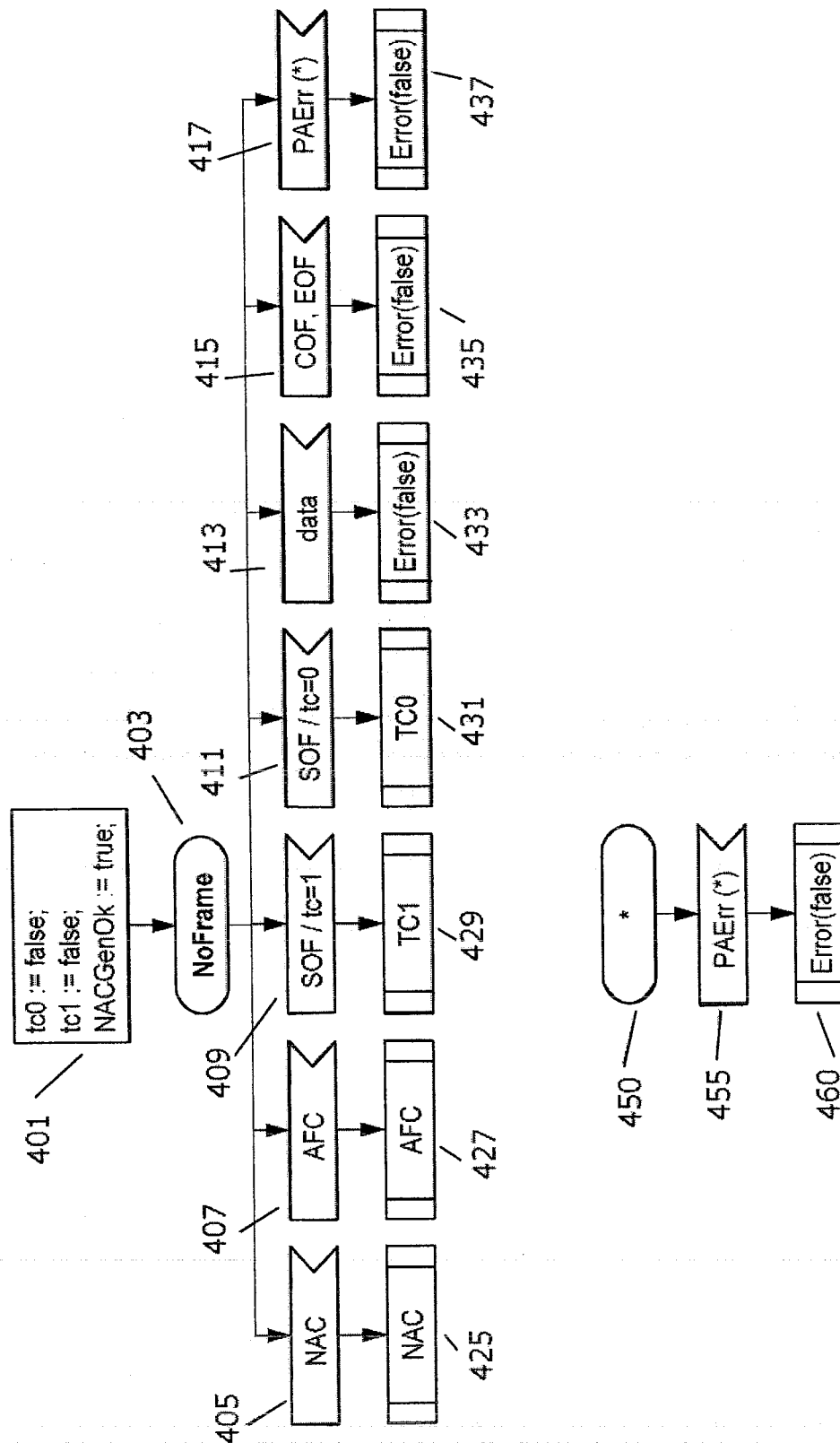


Fig. 4

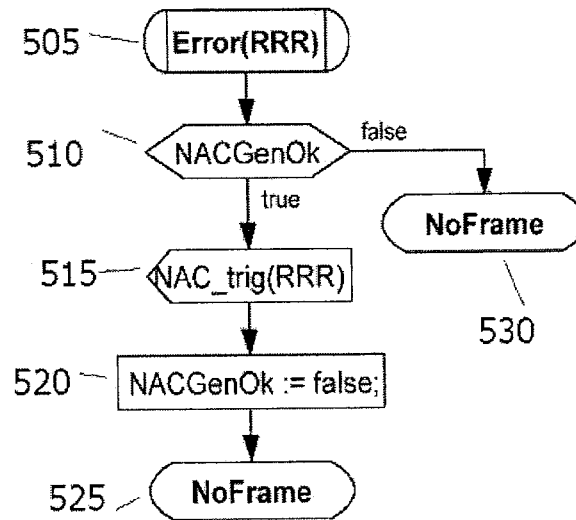


Fig. 5

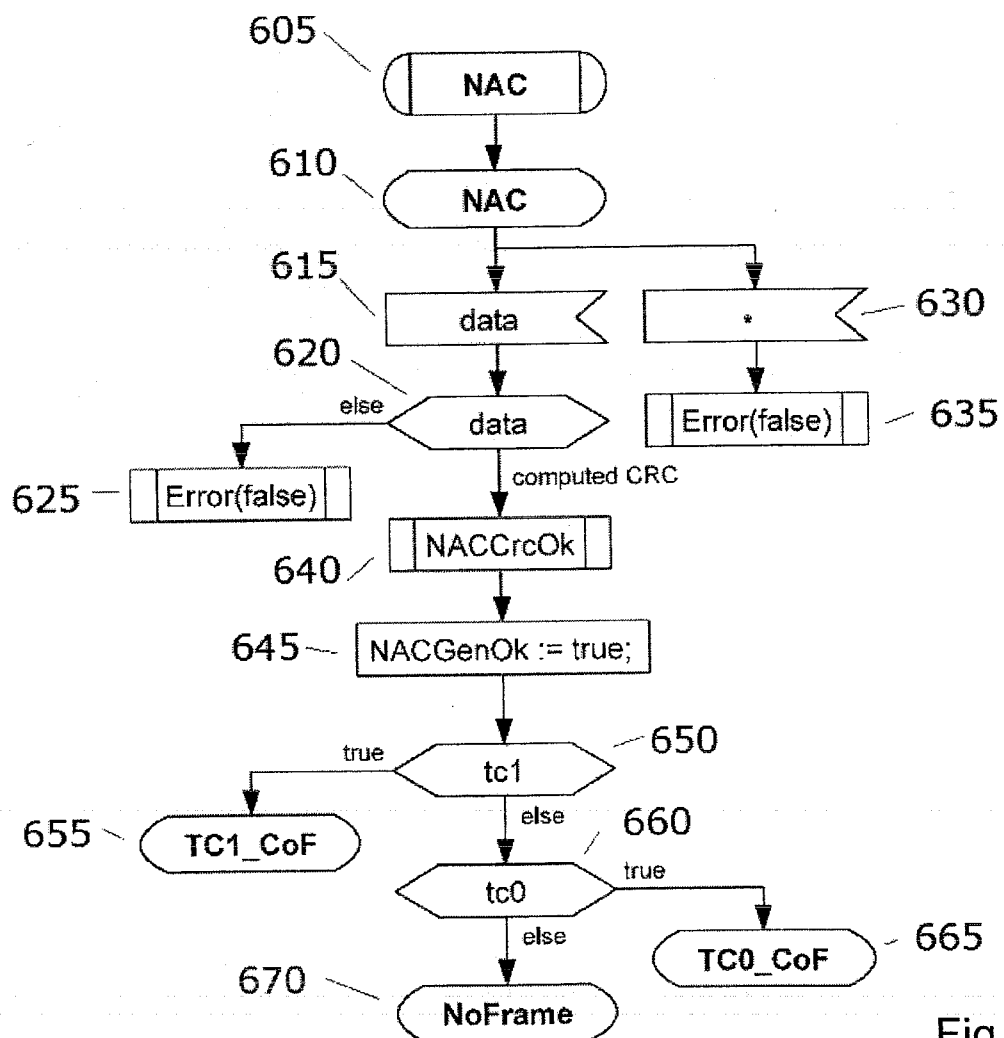


Fig. 6

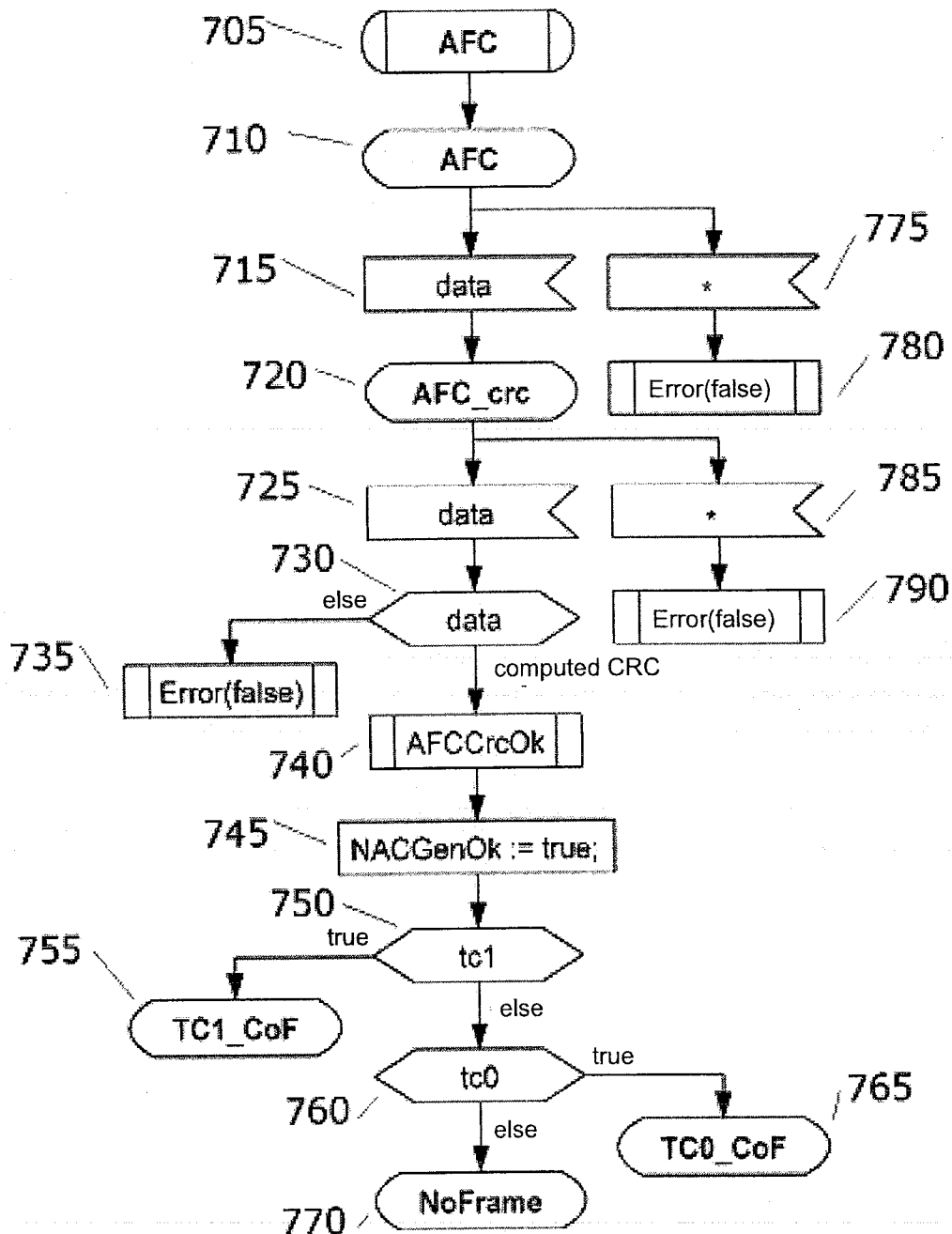


Fig. 7

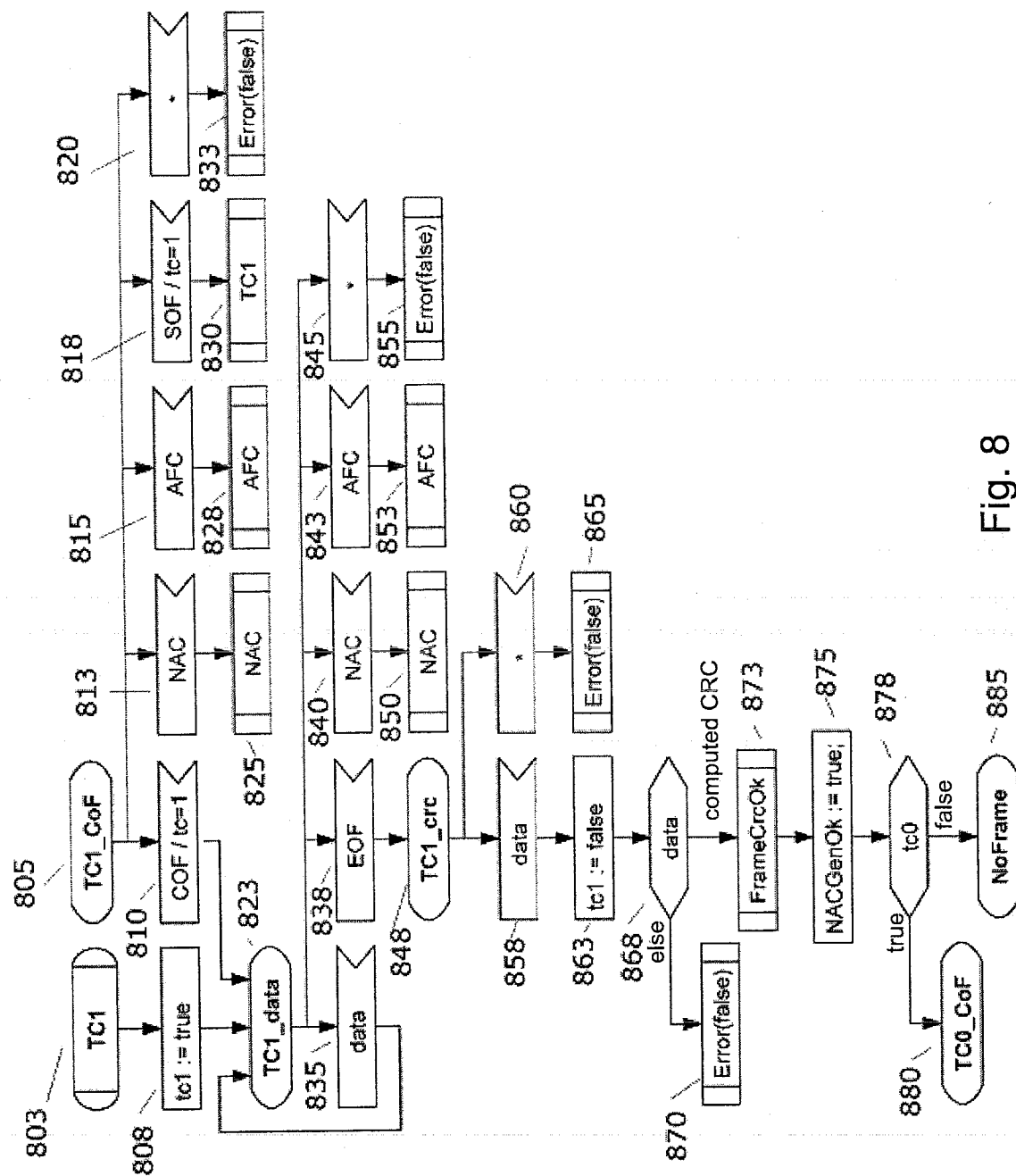


Fig. 8

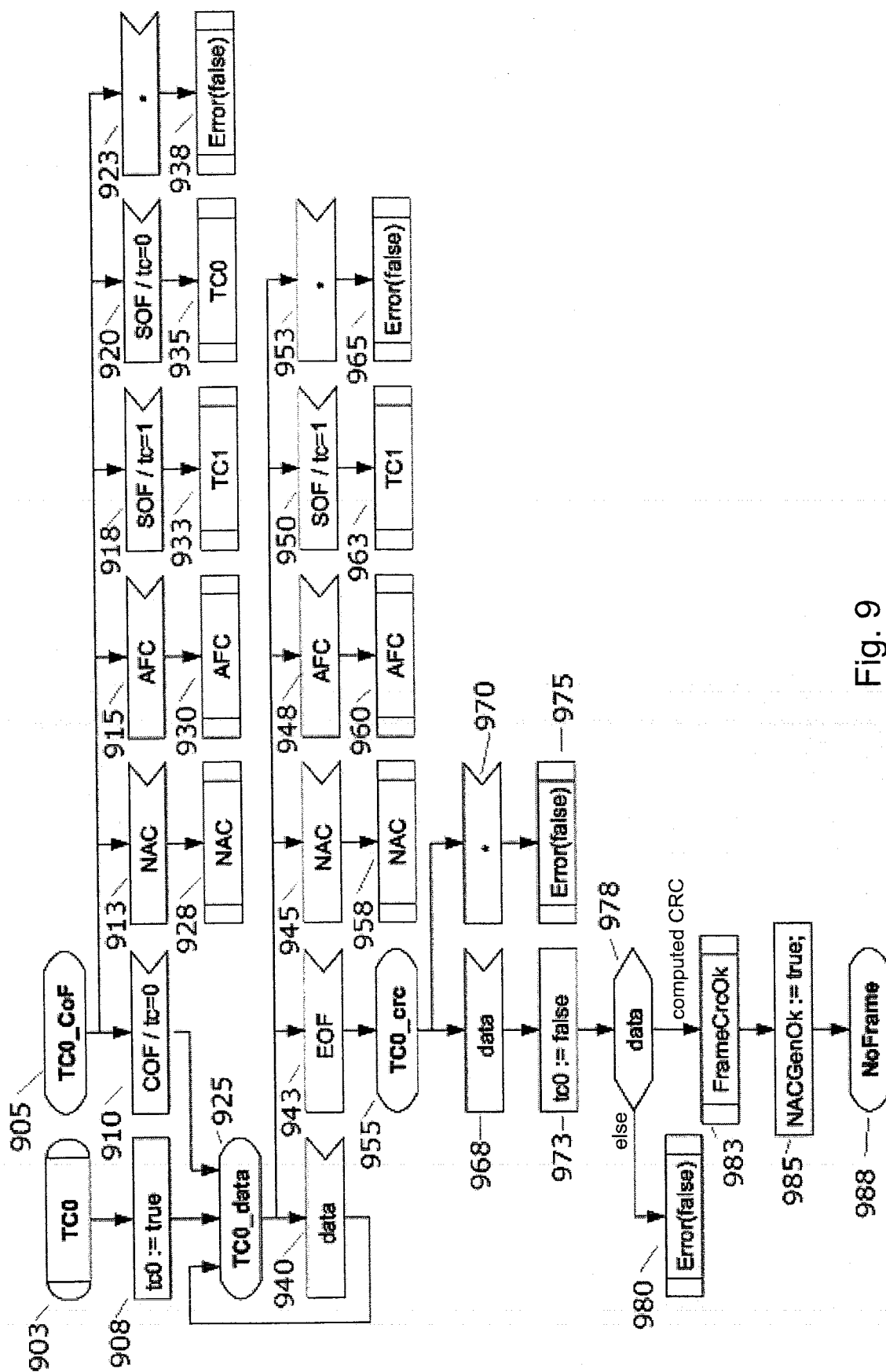


Fig. 9

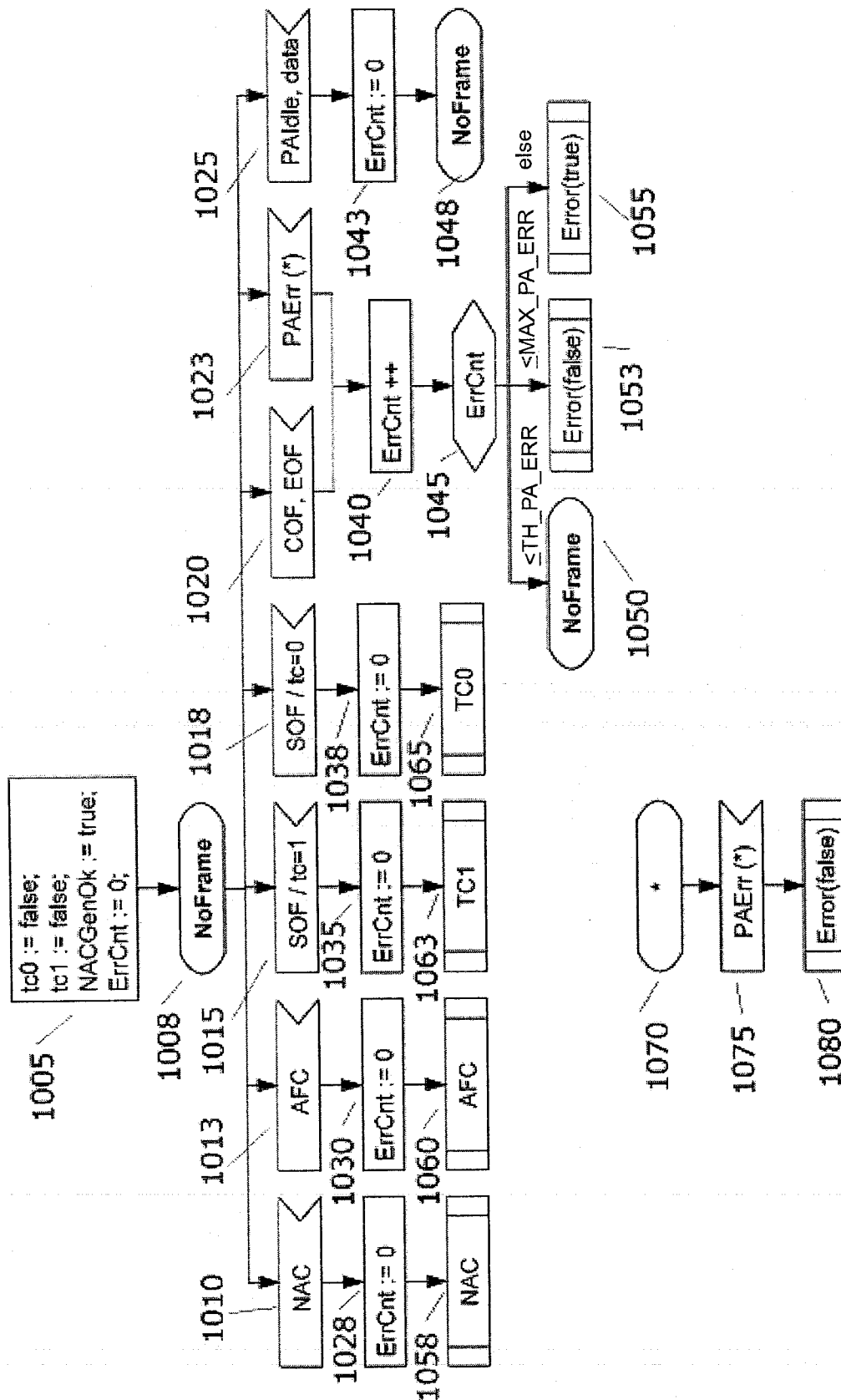


Fig. 10

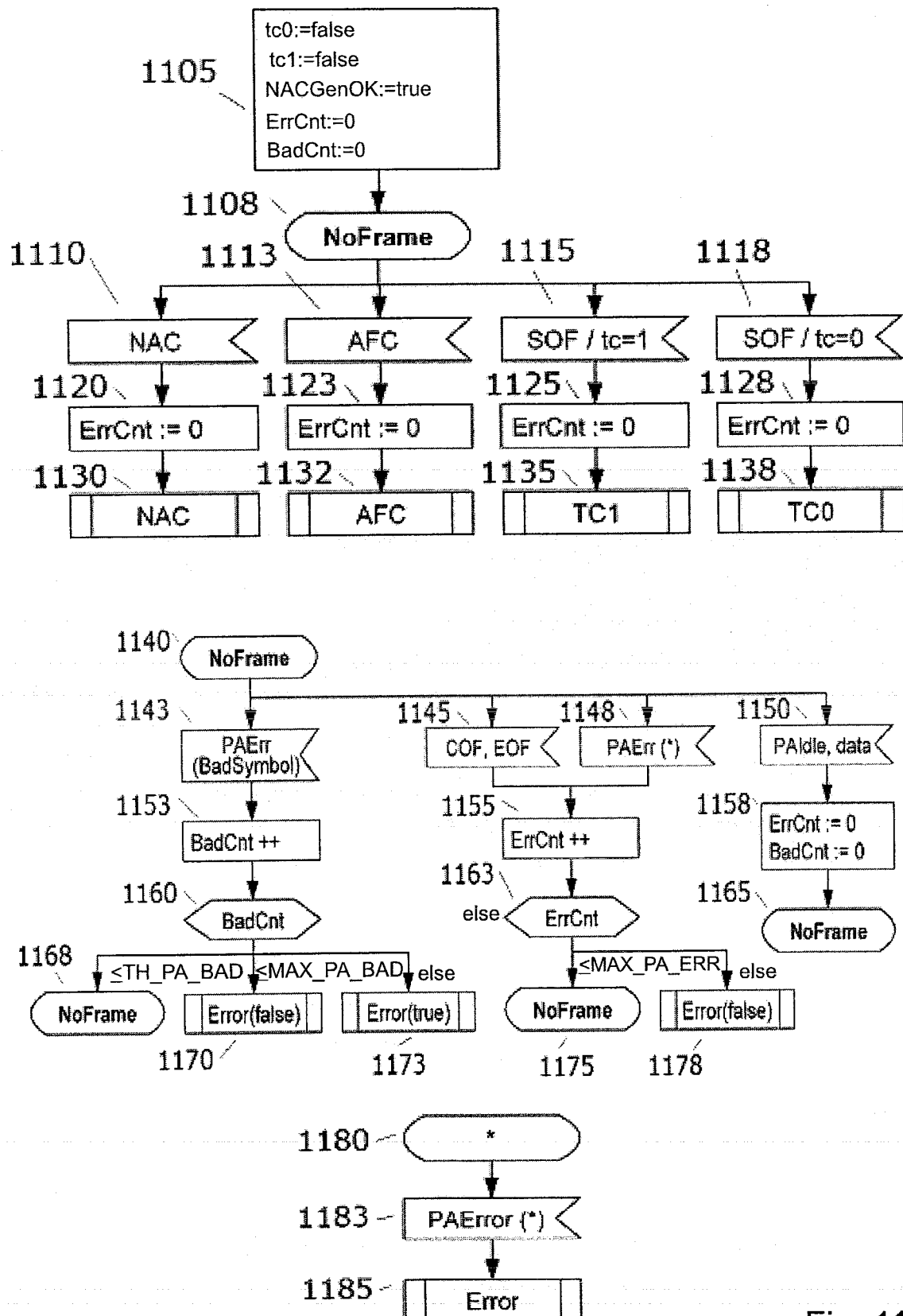


Fig. 11

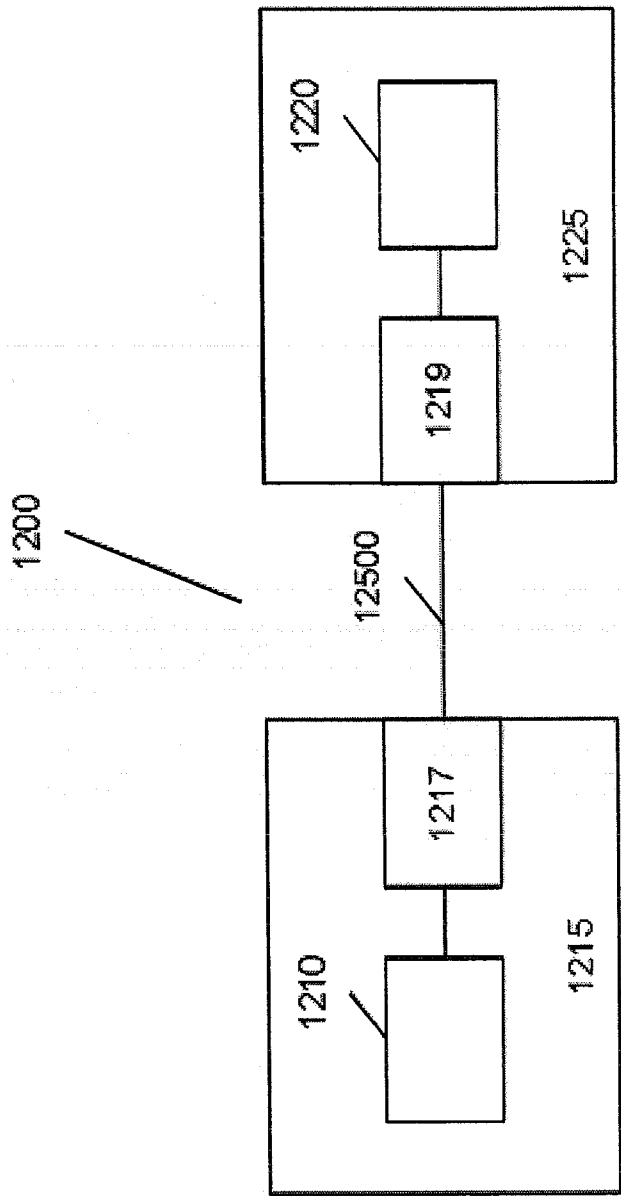


Fig. 12

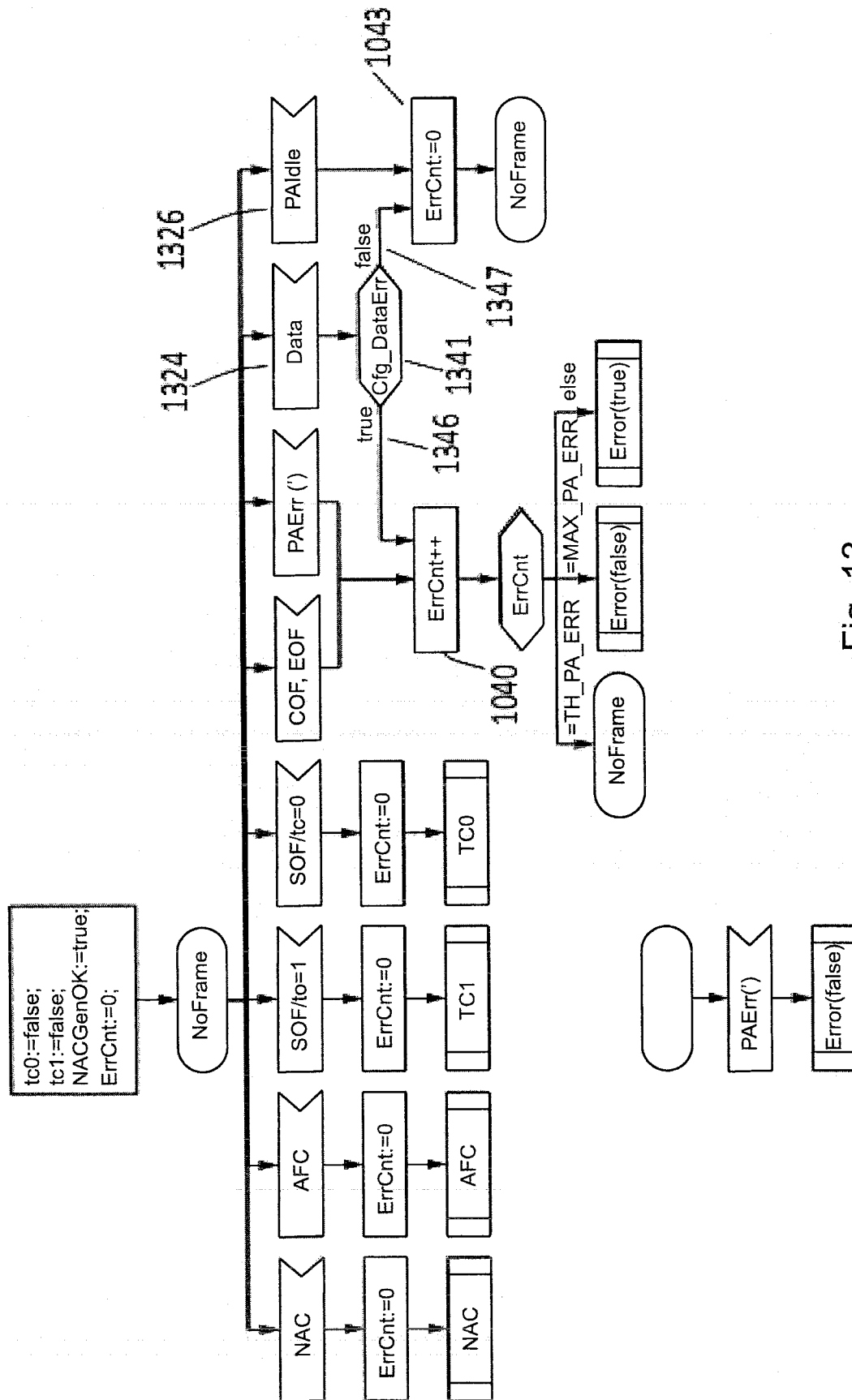


Fig. 13

International application No
PCT/EP2011/058763

According to International Patent Classification (IPC) or to both national classification and IPC

Minimum documentation searched (classification system followed by classification symbols)
H04L

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	wo 2004/021658 AI (PULSE LINK INC [US]) 11 March 2004 (2004-03-11) page 1, line 5 - page 2, line 2 page 2, line 16 - line 33 page 3, line 10 - line 19 page 5, line 28 - page 6, line 9 page 6, line 29 - page 7, line 18; figure ? - page 7, line 31 - page 8, line 9 page 11, line 3 - page 12, line 6; figure 8 -----	1,3-11 , 13-16 2, 12
X A	GB 2 313 268 A (MOTOROLA LTD [GB]) 19 November 1997 (1997-11-19) abstract page 3, line 23 - page 6, line 27; figures 3,4a, 4b ----- -/--	1,2,5-16 3,4

☒

Further documents are listed in the continuation of Box C.

☒

See patent family annex.

* Special categories of cited documents :

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

"P" document published prior to the international filing date but later than the priority date claimed

Date of the actual completion of the international search

Date of mailing of the international search report

14 October 2011

26/10/2011

Name and mailing address of the ISA/
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer _____

Courvi lle, Ni col as

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2011/058763

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 1 507 369 A1 (SONY CORP [JP]) 16 February 2005 (2005-02-16) abstract page 7, column 11, paragraph 64 - column 12, paragraph 70; figure 7 page 8, column 14, paragraph 83 - page 9, column 15, paragraph 96; figure 11 page 10, column 18, paragraph 114 - page 11, column 19, paragraph 118; figure 17 -----	1-16
A	FENGMIN GONG ET AL: "An Application-Oriented Error Control Scheme for High-Speed Networks" , IEEE / ACM TRANSACTIONS ON NETWORKING, IEEE / ACM, NEW YORK, NY, US, vol . 4, no. 5, 1 October 1996 (1996-10-01) , XP011039017 , ISSN: 1063-6692 page 671, right-hand column, paragraph III.C - page 672, left-hand column, paragraph III., -----	1-16

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2011/058763

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
Wo 2004021658	AI	11-03 -2004	
		AU 2002230913 AI	19-03-2004
		EP 1423954 AI	02-06-2004
		US 2004161052 AI	19-08-2004

GB 2313268	A	19-11 -1997	
		AT 251364 T	15-10-2003
		AU 2957497 A	09-12-1997
		DE 69725305 D1	06-11-2003
		DE 69725305 T2	29-04-2004
		Wo 9744927 AI	27-11-1997
		EP 0898820 AI	03-03-1999

EP 1507369	AI	16-02 -2005	
		CN 1656750 A	17-08-2005
		WO 03098884 AI	27-11-2003
		JP 4000905 B2	31-10-2007
		JP 2003338841 A	28-11-2003
		US 2005182850 AI	18-08-2005
