KLNI ERROR SPEC

LSG/CSSE

Joe Holewa

REV 0.4

4/20/83

Table of Contents

1.0 PURPOSE

The purpose of this ERROR SPEC is to document the KLNI Error Working Group's interpretation of certain errors which will occur or end up being detected in the KLNI. This document will also cover errors occurring on the NI wire, H4000 transceiver, and the PLUTO. This document in the reviewed final form will be used as reference for the working group for associated error analysis, coding, reporting, and recovery.

2.0 KLIPA PORT ERRORS

2.1 CRAM PAR ERR (CSR 06 Non Planned)

2.1.1 DETECTION - A non planned CRAM PAR ERR results from the CRAM WORD picking up or dropping a single bit. It is detected by the KLIPA CRAM PARITY logic, and when detected forces a Non-Vectored Interrupt to occur.

2.1.2 REPORTED - This error will only be reported to the console (BUGCHK/BUGINF KLIERR) if it is a hard error (i.e. two or more CRAM PARITY errors in a given time period TBS). This error will always be reported to ERROR.SYS by the MONITOR and will contain the following KLIPA information.

- o All CSR bits
- o KLIPA Microcode Version
- o Failing Cram Address and contents of the word
- o Channel Logout Area

Plus all the other standard SPEAR related information DATE/TIME etc.

2.1.3 ACTION - When notified of a CRAM PARITY error via non-vectored interrupt, the MONITOR will attempt to reload the CRAM and restart the KLIPA. The KLIPA has to be restarted due to the possible execution of the bad micro-instruction by the 2901s. A hard CRAM PARITY error will be declared if another CRAM PARITY error occurs within a given time period (TBS).The MONITOR will shut down the KLIPA if a hard CRAM PARITY error occurs or if two KLIERR errors occur.

2.2 CRAM PAR ERR (CSR 06 Planned)

2.2.1 DETECTION - A planned CRAM PARITY error is the result of executing a known bad parity micro-instruction. These instructions are purposely written to the CRAM with bad parity and are only used when certain error conditions appear(TBS). It will be detected by the CRAM PARITY logic and will force a non-vectored interrupt when detected.

2.2.2 REPORTED - This error will be reported to the console (BUGCHK/BUGINF KLIERR). The MONITOR will also make an entry to ERROR.SYS. The ERROR.SYS entry will contain the standard ERROR information as shown previous.

2.2.3 ACTION - When notified of a CRAM PARITY error via non-vectored interrupt, the MONITOR will attempt to reload the CRAM and restart the KLIPA. The KLIPA has to be restarted due to the possible execution of the bad micro-instruction by the 2901s. A hard CRAM PARITY error will be declared if another CRAM PARITY error occurs within a given time period (TBS). The MONITOR will shut down the KLIPA if a hard CRAM PARITY error occurs.

2.3 MBUS ERROR (CSR07)

2.3.1 DETECTION - A MBUS ERROR is caused by two or more MBUS drivers being asserted at the same time. This error is detected by the MBUS ERROR logic. When detected, a non-vectored interrupt will occur.

2.3.2 REPORTED - A MBUS ERROR will cause a (BUGCHK/BUGINF KLIERR) to occur. The MONITOR will also make an entry to ERROR.SYS with the information shown previous.

2.3.3 ACTION - Since MBUS ERRORS are not correctable, the MONITOR will only restart the KLIPA when a MBUS ERROR occurs. If two KLIERR errors occur within a given time period (TBS), the MONITOR will shutdown the KLIPA.

2.4 EBUS PARITY ERROR (CSR24)

2.4.1 DETECTION - A EBUS PARITY ERROR will be detected in the KLIPA when a data word with bad parity is received from the KL10. A non-vectored interrupt will result when the EBUS PARITY logic detects bad parity.

2.4.2 REPORTED - Assuming the KL10 is still operational after the EBUS PARITY error, the MONITOR will issue a (BUGCHK/BUGINF KLIERR) and will also make an entry to ERROR.SYS with the data shown previous.

2.4.3 ACTION - When an EBUS ERROR is detected the KLIPA will generate a planned CRAM PARITY Error. Since EBUS ERRORS are not correctable, the MONITOR will restart the KLIPA. If two KLIERR errors occur within a given time period(TBS), the MONITOR will shut down the KLIPA.

2.5 PLI PARITY ERROR

2.5.1 DETECTION - The PLI PARITY logic on the CBUS-PLI interface module will detect a parity error whenever a PLI DATA bit is either dropped or picked up on a PLI BUS READ. The error forces CCPLIPARERR to come up.

2.5.2 REPORTED - PLI PARITY errors will only be reported when the PLI PARITY error threshold count is exceeded. Originally, the threshold count is set to 5. When the threshold exceeds it limits, a planned CRAM PARITY ERROR at a specified address will occur. The MONITOR will issue a (BUGCHK/BUGINF KLIERR) and will also make an entry to ERROR.SYS.

2.5.3 ACTION - If a PLI PARITY error occurs within the allowed threshold, the KLIPA microcode will abort the sequence being performed and restart it. Since PLI PARITY errors occur only on PLI BUS reads, there is a very good chance that recovery can be made assuming multiple PLI PARITY errors do not occur. SPEAR WILL NOT be NOTIFIED of recoverable PLI PARITY errors. Corrupted PLI BUS data which the KLIPA has read will NOT leave the CBUS-PLI module!!

2.6 MOVER PARITY CHECK ERROR

2.6.1 DETECTION - If a bit gets dropped or picked up in the MOVER/FORMATTER logic, a MOVER PARITY CHECK ERROR will occur. The MOVER/FORMATTER PARITY CHECK logic will force the condition code CCMVRPARCHK to go true.

2.6.2 REPORTED - MOVER/FORMATTER CHECK errors will only be reported when the MOVER/FORMATTER CHECK error threshold count is exceeded. Originally, the threshold count is set to 5. When the threshold exceeds it limits, a planned CRAM PARITY ERROR at a specified address will occur. The MONITOR will issue a (BUGCHK/BUGINF KLIERR) and will also make an entry to ERROR.SYS.

2.6.3 ACTION - On MOVER/FORMATTER CHECK errors, the microcode will complete the transfer knowing that the transfer will be restarted again by the KLIPA microcode. When the threshold is exceeded, once notified of the condition from the KLIPA via planned CRAM PARITY error, the MONITOR will restart the KLIPA.

NOTE

> On all planned CRAM PARITY errors the MONITOR can believe what's on the RESPONSE QUEUE (The DATA will be valid).

2.7 CBUS PARITY ERROR

2.7.1 DETECTION - When the data coming into the KLIPA from the CBUS either drops or picks up a bit, the CBUS IN PARITY checking logic will generate the condition code CCCBUSPARERR.

2.7.2 REPORTED - CBUS PARITY errors will only be reported when the CBUS PARITY error threshold count is exceeded. Originally, the threshold count is set to 5. When the threshold exceeds it limits, a planned CRAM PARITY ERROR at a specified address will occur. The MONITOR will issue a (BUGCHK/BUGINF KLIERR) and will also make an entry to ERROR.SYS.

2.7.3 ACTION - When a CBUS PARITY error occurs which hasn't exceeded the error threshold, the KLIPA microcode will start a duplicate CBUS READ (same starting address) to try to recover from the CBUS PARITY error. When the threshold is exceeded, once notified of the condition from the KLIPA via planned CRAM PARITY error, the MONITOR will restart the KLIPA.

2.8 CHANNEL ERROR

2.8.1 DETECTION - A CHANNEL error can be caused by the following conditions:

- o ADDRESS PARITY ERROR
- o NXM
- o MEMORY PARITY ERROR
- o SHORT WORD COUNT

The condition code CCCHANERR will be asserted whenever the signal CBUS ERROR is detected on the CBUS.

2.8.2 REPORTED - When the MONITOR is made aware of a CHANNEL ERROR (CBUS ERROR) it will update ERROR.SYS with the standard KLIPA ERROR information.

2.8.3 ACTION - When CCCHANERR gets asserted it causes the KLIPA microcode to do two things. The first being putting an ERROR packet on the RESPONSE QUEUE. The second is setting DATA PATH ERR (CSR 26) and forcing an interrupt. The interrupt forces the MONITOR to do a CONICSR and from there the MONITOR will look at the CHANNEL LOGOUT WORD.

2.9 CCBUSAVAIL

2.9.1 DETECTION - This condition is called an error when it exceeds the timeout limit as set by the microcode. This timeout limit will be supplied by Hardware Engineering.

2.9.2 REPORTED - If this failure exceeds the threshold limit set up by the microcode(currently 5), the microcode will place a packet on the Response Queue with the appropriate Status Bit set reflecting the error. When the Monitor receives the error indication, it will supply the Channel Logout Area, Opcode word, and the CCW word from the PCB to Error.Sys.

2.9.3 ACTION - When CCBUSAVAIL is not assertec for a certain time period(TBD), the microcode will restart the transfer for up to 5 times. If the transfer fails 6 times the microcode will notify the Monitor via EBUS and place itself in the Disabled State.

2.10 CCEBUSRQST

2.10.1 DETECTED - This condition is called an error when it exceeds the timeout limit as set by the microcode. The timeout limit will be supplied by Hardware Engineering.

2.10.2 REPORTED - The Monitor will report this error to Error.Sys when it receives a planned CRAM PARITY error from the KLIPA. The Monitor will update Error.sys with the typical planned CRAM PARITY error information.

2.10.3 ACTION - The MONITOR will attempt to restart the KLIPA after it is notified of the CCEBUSRQST error.

2.11 CCGRANTCSR

2.11.1 DETECTION - This condition is called an error when it exceeds the timeout limit as set by the microcode.

2.11.2 REPORTED - The Monitor will report this error to Error.Sys when it receives a planned CRAM PARITY error from the KLIPA. The Monitor will update Error.sys with the typical planned CRAM PARITY error information.

2.11.3 ACTION - The MONITOR will attempt to restart the KLIPA after it is notified of the CCGRANTCSR error.

3.0 KLNI ERRORS

3.1 XMIT PAR ERR DFT H

3.1.1 DETECTION - The Transmit Buffer will be parity checked after the PORT issues a Transmit Frame command. Parity is re-generated for the data leaving the Transmit buffer and is compared to the parity stored for each byte in the buffer. If a parity error does occur, the PORT will be notified via the TRANSMIT STATUS register once Transmit Attention comes up. The PORT will keep a counter of XMIT PARITY errors.

3.1.2 REPORTED - XMIT PARITY errors will only be reported when the XMIT PARITY error threshold count is exceeded. Originally the threshold count is set to 5. When the threshold exceeds its limits, a planned CRAM PARITY ERROR at a specified address will occur. The MONITOR will issue a (BUGCHK/BUGINF KLNxxx) and will also make an entry to ERROR.SYS.

3.1.3 ACTION - If a XMIT PARITY ERROR does occur within the allowed threshold, the PORT will reload the TRANSMIT BUFFER and start another transfer. The transmission in error will be aborted.

3.2 PLI PARITY ERROR (KLNI)

3.2.1 DETECTION - On any PLI PARITY error (except one caused by a FREE BUFFER/USED BUFFER parity error), the PORT will follow the same algorithm explained under PLI PARITY Error referenced in section (2.5 - 2.5.3) of this spec.

3.2.2 REPORTED - See above.

3.2.3 ACTION - See above.

3.3 FB PAR ERR

3.3.1 DETECTION - The FB PARITY ERROR is caused by the FREE BUFFER picking or dropping a single bit. The error is detected by the FREE BUFFER PARITY logic. The PORT is made aware of the condition via the RECEIVE STATUS register when RECEIVE ATTENTION comes up. A PLI PARITY error will a also occur.

3.3.2 REPORTED - FREE BUFFER PARITY errors will only be reported when the FREE BUFFER PARITY error threshold count is exceeded. Originally the threshold count is set to 5. When the threshold exceeds its limits, a planned CRAM PARITY ERROR at a specified address will occur. The MONITOR will issue a (BUGCHK/BUGINF KLNxxx) and will also make an entry to ERROR.SYS.

3.3.3 ACTION - If a FREE BUFFER PARITY error occurs within the allowed threshold, the PORT will discard the corrupted Free Buffer List Location and continue to use entries from the FREE BUFFER LIST. The incoming Frame will be discarded.

3.4 USED BUFFER LIST PARITY ERROR

3.4.1 DETECTION - A USED BUFFER PARITY error is caused by the USED BUFFER dropping or picking up a single bit. It is detected when the PORT receives a PLI PARITY error when reading the USED BUFFER.

3.4.2 REPORTED - USED BUFFER PARITY errors will only be reported when the USED BUFFER PARITY error threshold count is exceeded. Originally the threshold count is set to 5. When the threshold exceeds its limits, a planned CRAM PARITY ERROR at a specified address will occur. The MONITOR will issue a (BUGCHK/BUGINF KLNxxx) and will also make an entry to ERROR.SYS.

3.4.3 ACTION - If a USED BUFFER PARITY error occurs within the allowed threshold, the PORT will discard the corrupted Used Buffer List Location and continue to use entries from the USED BUFFER LIST. The incoming frame is discarded.

3.5 INIT FAILURE

3.5.1 DETECTION - This error condition occurs when the PORT Microcode detects an error during self-test. This error may be caused by any number of things in the PORT hardware (2901's, MBUS, Local Storage Rams, etc.).

3.5.2 REPORTED - This error will be reported to ERROR.SYS when the MONITOR is notified of the INIT FAILURE via Planned CRAM PARITY error. The standard Plan CRAM PARITY information will be provided.

3.5.3 ACTION - When detected by the PORT Microcode, the PORT Microcode will force a Planned Cram Parity error. This essentially shuts down the KLNI hardware. Once notified of this error, the MONITOR will NOT attemp to reload/restart the KLNI due to the possibility of propagation of corrupted data.

3.6 SEND FAILED--EXCESSIVE COLLISIONS

3.6.1 DETECTED - A Collision is defined as one or more nodes are transmitting on the NI wire at the same time. All Collisions are detected by the H4000 transceiver which passes Collision Detect back to the NIA module. When the NIA gets Collision Detect back it waits a random delay (Truncated Binary Exponential Backoff Algorithm) and re-transmits. The NIA allows up to 16 successive collisions before it determines the collision rate is EXCESSIVE. EXCESSIVE collisions are possibly caused by an overload condition on the Ethernet.

3.6.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver). The TDR count also gets entered into ERROR.SYS.

3.6.3 ACTION - When an EXCESSIVE collision occurs, the NIA stops re-transmitting and notifies the PORT microcode via the Transmit Status Register. The PORT microcode builds a Response Packet with the EXCESSIVE collision information in it. The PORT microcode also returns the value of the TDR count to the PORT driver. The PORT Driver hands this information over to the next higher level of software to decide if another Transmission will be attempted. The Error Event Logger will log this error to ERROR.SYS(TBD, could be the PORT driver).

3.7 SEND--CARRIER CHECK FAILED

3.7.1 DETECTION - This error is detected by the NIA when it did not sense the the presence of Carrier when it transmitted a frame. This failure caused by one of the following: broken NIA hardware, bad H4000, or bad Tranceiver cable.

3.7.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver).The TDR count is also entered into ERROR.SYS

3.7.3 ACTION - Upon the detection of Carrier Check Failed the NIA will notify the PORT microcode by setting the LOCE bit in the Transmit Status Register and setting Transmit Attention. The NIA will not attempt a re-transmit. The PORT microcode will build a Response Packet and notify the PORT driver. The PORT microcode will also will return the valueof the TDR to the PORT driver. The PORT driver will notify the next higher software level of the error.

3.8 SEND--COLLISION DETECT CHECK FAILED(HEARTBEAT)

3.8.1 DETECTION - The H4000 has a Collision Presence Test function which tests the Collision sensing logic in the H4000. The rational behind this testing is to further insure that the frame sent out of the H4000 was received without a collision.When this error occurs the last frame sent from the H4000 could have had an undetected Collision. The H4000 enables this test after every transmission. If a Collision is not detected when expected, the NIA notifies the PORT microcode by setting the CIF bit in the Transmit Status Register and setting Transmit Attention.

3.8.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TED, could be the PORT driver).

3.8.3 ACTION - The PORT microcode notifies the PORT driver of the error via Response Packet. The PORT Driver notifies the next higher level of software of the error.

3.9 SEND--SHORT CIRCUIT

3.9.1 DETECTION - The KLNI can not at this time differentiate between SHORT CIRCUITS and other Ethernet Coax problems, therefor no explanation is provided.

3.9.2 REPORTED - None

3.9.3 ACTION - None

3.10 SEND--OPEN CIRCUIT

The KLNI can not at this time differentiate between OPEN CIRCUITS and other Ethernet Coax problems, therefor no explanation is given.

3.11 SEND--FRAME TOO LONG

3.11.1 DETECTION - This error is detected in the NIA module. The logic checks to see if the transmitted frame is larger than 1536 bytes. If the frame is larger than 1536 bytes the error occurs.

3.11.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver).

3.11.3 ACTION - When detected, the NIA will set the TOTL bit in the Transmit Status Register along with Transmit Attention. The PORT microcode will build a Response Packet for the PORT Driver.

3.12 SEND--REMOTE FAILURE TO DEFER

3.12.1 DETECTION - When a collision is detected on the NI wire, the H4000 will assert the Collision Detected signal. If the collision is detected after 64 bytes of information have been transmitted, the NIA will assert the LCF signal in the Transmit Status Register along with the Transmit Attention signal.

3.12.2 REPORTED - This error will be reported to ERROR.SYS by the Event Logger(TBD, could be the PORT driver).

3.12.3 ACTION - At the minimum, the PORT will build a Response Packet and will include the LCF error condition. No retrys are attempted and the transmission is aborted.

3.13 RECEIVE FAILED--BLOCK CHECK ERROR

3.13.1 DETECTION - A CRC error results from an incoming NI FRAME dropping or picking up bits. This error is detected on the NIA module by generating CRC and comparing it to the CRC sent over with the NI FRAME.

3.13.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver). The error also gets reported to the user of the NI which owns the bad frame along with the frame data.

3.13.3 ACTION - When this error is detected by the NIA module, it sets the CRCE error bit in the Receive Status Register and sets Receive Attention. The PORT microcode will build a response packet with the appropriate error information.

3.14 RECEIVE--FRAMING ERROR

3.14.1 DETECTION - This error is detected by the NIA when an incoming frame contained a non-interger multiple of 8 bit bytes and the crc value at the last byte boundry was in error.

3.14.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver).

3.14.3 ACTION - When the NIA dectects this error, it notifies the PORT microcode by setting the FFE bit in the Receive Status Register and setting Receive Attention. The PORT microcode will build a Response Packet with the appropriate error information.

3.15 RECEIVE--DATA OVERRUN

This error is detected by the NIA module when a Buffer Overlow (Receive Buffer is full) happens. The NIA no longer has room to store incoming messages.

3.15.1 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver).

3.15.2 ACTION - When the NIA detects this error, it notifies the PORT Microcode by setting the FBUFE bit in the Receive Status Register and setting Receive Attention. The PORT microcode will build a Response Packet with the appropriate error infomation.

3.16 RECEIVE--USER BUFFER UNAVAILABLE

3.16.1 DETECTION - This error is detected when the PORT microcode was unable to get an entry off the Free Queue in KL10 memory. This error is not hardware error related.

3.16.2 REPORTED - This error gets reported back to the NI user.

3.16.3 ACTION - The PORT Microcode will set the FREE QUEUE error bit in the CSR and interrupt the KL10.

3.17 RECEIVE--UNRECOGNIZED FRAME DESTINATION

3.17.1 DETECTION - Using the current TOPS-20 architecture this error should never occur.

3.17.2 REPORTED - Using the current TOPS-20 architecture this error should never occur.

3.17.3 ACTION - Using the current TOPS-20 architecture this error should never occur.

3.18 RECEIVE--FRAME TOO LONG

3.18.1 DETECTION - This error is detected in the NIA module. The logic checks to see if the received frame is larger than 1536 bytes. If the frame is larger than 1536 bytes the error occurs.

3.18.2 REPORTED - This error gets reported to ERROR.SYS by the Error Event Logger(TBD, could be the PORT driver).

3.18.3 ACTION - When detected, the NIA will set the FLE bit in the Receive Status Register along with Receive Attention. The PORT microcode will build a Response Packet for the PORT Driver. The PORT microcode will also discard the FRAME.
