M.I.T. Laboratory for Computer Science — Network Implementation Note No. 23

November 10, 1980

Version II LNI UNIBUS Interface Programming Specification

from J.H. Saltzer

The attached programming specification was developed over the course of the last year with contributions from many different people who have been working on or near the local network interface project. Questions about why the design is the way it is are probably best directed to me or Dave Clark, while questions on how one programs a driver for this device are probably best directed to Noel Chiappa or Larry Allen.

This note describes the programming of the proposed Version II LNI UNIBUS interface, which is the version of the new LNI for the DEC PDP-11 and other computers.

I. Introduction

The Version II Local Network Interface UNIBUS interface consists of one major and two lesser portions which are loosely interconnected. The first is the controller, which manages the ring interface module of the LNI, transfers data between that module and the second portion, packet buffers which temporarily hold data blocks between the computer and network, and directs the third, a full duplex direct memory access (DMA) section which transfers data between the computer's memory and the packet buffers. The input and output halves of the interface are essentially unconnected and can operate simultaneously; the DMA and interface sections can also switch operation without software intervention.

The actual ring and the ring interface module are described elsewhere, in [Saltzer1], which explains in more detail the actual operation of the network, which is a series of repeating interfaces connected in a ring by point to point links. Interfaces are joined into the ring through a relay which opens automatically if they are powered off, and must be reset under program control; when the relay is open the interface's neighbours are connected directly to each other, and the interface is looped back to itself as a diagnostic and testing aid.

To perform input, an input operation is enabled, and when a message passes through the LNI which is destined for the connected host, (i.e. either the destination address of the packet is either this interface or the packet is a broadcast or the interface is a "monitor") the LNI will write a copy of the data on the ring into the input packet buffer. At some point the data must be moved from the packet buffer by enabling a DMA operation, which will cause data to flow from the input packet buffer into a memory buffer. The process is similar for an output operation: the packet to be sent is loaded into the output packet buffer via DMA; when the packet has been loaded and output is enabled the LNI waits until the access control token passes into it, and then it removes the token, sends the message, taking data from the packet buffer as it goes, and appends a new token; when the message returns around the ring the LNI deletes it. A feature of the LNI makes it possible in both input and output for the second operation to occur automatically after the first has occurred.

"Logical" messages suitable for transmission over the ring network consist of the following sequence of fields:

(i) an 8 bit destination

(ii) a 8 bit source (filled in by the hardware from the host address switches in the LNI)

(iii) actual text of the message (some number of bytes)

This is the format of messages when they are stored in memory, both for output and input; bytes are stored in memory in the order in which they are received from or sent to the ring. The UNIBUS version of the LNI restricts the length of the message to an even number of bytes, since it is a word interface. On input, the message will be padded out to an even number of bytes if necessary. It also restricts the maximum number of data bytes in the message to 2044 (decimal), although there is no limit on the size of a packet in the network itself.

Although the user sends and receives logical messages, the LNI actually sends to the ring and receives from the ring "packaged" messages which consist of logical messages with begin and end flags and other control bits.

One of these bits is the "refused" bit; this bit has been included to provide some low level flow control. It is set by the destination LNI on a non-broadcast message if it was unable to store a copy of the message.

Another control bit, the "link parity bit", is intended to help isolate intermittent links in the ring. The mechanism is as follows: each message contains a parity bit; when an LNI detects a parity error in any message being received (whether the LNI is repeating, copying or draining) it sets an error flag in the local interface (which can be inspected by the host, and is cleared only under program control), and retransmits the message with the correct parity (if repeating or copying). Periodic inspection of the flags in all the interfaces by software reveals error prone links; any error was introduced in the "up-ring" connecting link.

Note that since the two halves of the LNI are capable of independent operation, the following scenarios are possible: originating a message while copying one, copying a message while waiting to originate one, and having an interface originate a message to itself.

The LNI is also capable of forcing a message followed by a token onto the ring; this feature is included for use if the circulating token is destroyed.

As a warning to readers, note that this document describes the operation of the UNIBUS interface as seen by the programmer, without distinguishing whether the operations described are performed by the UNIBUS interface or by the ring control module. Interested readers should study [Saltzer1] and [Ludwig] before attempting to study the internal operation of the UNIBUS interface.

II. Device Registers

The LNI contains two sets of device registers; one for each half. Most of them are identical for the input and output halves; where possible information is not duplicated. In the register descriptions below, bit 0 is the least significant bit of a word, and bit 15 is the most significant. All registers are 16 bits long.

Input Control and Status Register

Bit 0: Enable Copy

Setting this bit enables the copying of a message from the ring into the input packet buffer. Copy must be enabled for each message to be read from the ring. Enabling copying causes any data left in the input packet buffer to be flushed; thus, if data was left in the packet buffer from the previous message, it will be thrown away. There is currently no way of telling if or how many messages for the connected host went by while copying was not enabled. This bit is cleared by a reset (either a UNIBUS INIT or the appropriate Reset bit), by the copy completing or by the Ready bit setting; the user cannot clear it once set. This is thus a read/write to one bit.

Bit 1: Enable Input DMA

Setting this bit enables the transfer of data from the input packet buffer to the memory. DMA will occur as long as the DMA count is non-zero, there is data in the buffer, and no errors occur. If a DMA operation does not transfer all the data in the input packet buffer, another DMA operation will continue with the next byte. If both this bit and the Copy Enable bit are set, the LNI will wait for a message to be read in and then start a DMA operation without host intervention. The Enable DMA bit is cleared by a reset or when the Ready bit sets; this is thus a read/write to one bit.

Bit 2: Host Enable

Setting this bit closes the relay that inserts the LNI in the ring. It is a read/write bit, and is cleared by a UNIBUS INIT.

Bit 3: Self Test Enable

Setting this bit causes the LNI to disregard the Not in Ring bit so that the LNI may be tested looped back locally, either through the relay or by use of the Loop Back feature. See the description of the Not In Ring bit for further details. It is a read/write bit, and is cleared by a UNIBUS INIT.

Bit 4: Loop Back

Setting this bit causes the LNI to loop itself back locally in the digital portion of the LNI, so that it may be tested without using the analog portion of the interface, primarily for error localization. Note that setting this bit automatically open the relay, so that the setting of the Host Enable bit is ignored when this bit is set. It is a read/write bit, and is cleared by a UNIBUS INIT.

Bit 5: Reset

Setting this bit causes the input side of the LNI to be reset; Ready is set, the packet buffer is cleared and the ring interface portion returns to its idle state. This bit is a write only bit; it is cleared by a reset and reads as zero.

Bit 6: Interrupt Enable

If this bit is set when Ready sets, the LNI will generate an interrupt to the LNI receive vector. It is a read/write bit, and is cleared by a reset.

Bit 7: Ready

If this bit is on then the input side of the LNI is ready to accept commands; either a DMA command or an input command may be given. When the command finishes (either an error has occurred, a packet has been read, and/or the DMA has run as far as it is able) this bit will set. Note that as stated above it is possible to issue two commands concurrently; in this case Ready will come on when the second operation is completed. (This assumes that they both completed OK; if an error occurred Ready would have set when the error occurred.) This bit is a read only bit, and is set by a reset or command completion; it clears when an operation is started successfully. The software should thus check this bit to make sure that the operation started successfully after enabling one.

Bit 8: Data Present

This bit is on when there is unread data in the input packet buffer. It is read-only, and is cleared by a reset, enabling copy or when all the data has been removed from the packet buffer.

Bit 9: Non Existent Memory (NXM)

This bit is set when a DMA cycle experiences a non existent memory fault. It causes DMA to cease and Ready to set. It is a read-only bit, and is cleared by enabling DMA or a reset.

Bit 10: Overrun

This bit is set when a packet being copied from the ring is too large to fit into the input packet buffer. The packet up to the point at which the overflow occurred is left in the input packet buffer and may be read out by an Input DMA operation. It causes Ready to set. It is read only, and is cleared by enabling input or a reset.

Bit 11: Odd Byte Count

This bit is set when there are an odd number of bytes in the received packet; it is for use by programs that wish to determine the exact length of a packet sent them, since inspection of the word count register cannot provide this. It is read-only, and is cleared by enabling copy or a reset.

Bit 12: Link Data Error

This bit is set when the LNI detects a parity error on any message that passed through it, whether addressed to this station or not. This bit is a read/write-to-zero bit; it is also cleared by a UNIBUS INIT.

Bit 13: Ring Not OK

This bit indicates whether the network interface thinks that ring appears to be operating (idling or sending a message) or not. It is a read-only bit.

Bit 14: Bad Format

This bit is set when the LNI detects a link level protocol error as it is copying a message. The packet up to the error is left in the packet buffer. It causes Ready to set; it is read-only, and is cleared by enabling input or a reset.

Bit 15: Not in Ring

This bit is on when the LNI's relay is not closed. If this bit is on when Enable Input is set, Ready will not clear if the LNI is not in Self Test mode. It only affects input operations, not DMA operations. It is a read-only bit.

Output Control and Status Register

Bit 0: Enable Originate

Setting this bit enables the origination of a message when the ring is available for use. The origination operation will be considered to have completed either when the message has been removed from the ring, or the LNI has timed out waiting for it. Originate must be reenabled for each message to be sent. This bit is cleared by a reset or when Ready sets; it is thus a read/write-to-one bit.

Bit 1: Enable Output DMA

Setting this bit enables the transfer of data from memory into the output packet buffer; if there is data already in the packet buffer the new data is added to the end of the data already present. DMA will occur as long as there is room in the packet buffer, the DMA count is non-zero, and no errors occur. If both this bit and the Output Enable bit are set, the LNI will wait for the DMA to complete before doing the output. This bit is cleared by a reset, by output DMA completing or by the Ready bit setting; it is thus a read/write-to-one bit.

Bit 2: Clear Packet Buffer

Setting this bit clears the output packet buffer. If both this bit and Enable Output DMA are set, this acts first. It is a write-only bit, and reads as zero.

Bit 3: Unused

Bit 4: Initialize Ring

If this bit is set when Enable Originate is set, the LNI sends the message immediately, without waiting for a token to pass by; the message is ended with a token. This operation is provided to enable the ring to be reinitialized if a token is lost. It is a write-only bit, and reads as zero. It is cleared (internally) by a reset or Ready setting.

Bit 5: Reset

This bit functions the same way as the corresponding bit on the input side. If the LNI is originating a message when this bit is set, the message is ended immediately with a token and no end flag; i.e. it will produce a link level protocol error at the receiving LNI.

Bit 6: Interrupt Enable

This bit functions the same way as the corresponding bit on the input side.

Bit 7: Ready

This bit functions the same way as the corresponding bit on the input side.

Bit 8: Refused

This bit indicates that the message just originated was refused by the recipient; it is a read-only bit, and is cleared by a reset or setting Output Enable.

Bit 9: NXM

This bit functions the same way as the corresponding bit on the input side.

Bit 10: Overrun

This indicates that there was an attempt to load too many bytes into the output packet buffer. If this bit is on when Enable Output is is set, it has no effect on the operation of the LNI. It causes Ready to set, and is a read only bit; it is cleared by a reset or setting Clear Packet Buffer.

Bit 11: Unused

Bit 12: Output Timeout

This bit sets if the LNI has been waiting to originate a message for 300msec. It causes Ready to set. This is a read-only bit; it is cleared by setting Output Enable or a reset.

Bit 13: Ring Not OK

This bit functions the same way as the corresponding bit on the input side.

Bit 14: Bad Format

This bit functions much the same as the corresponding bit on the input side. It can also mean that the LNI was unable to successfully remove from the ring the message it just sent. At present no specific protocol errors are reported; if it becomes useful to know such things they may be added as separate status bits. This is a read only bit, and it is cleared by a reset or setting Output Enable.

Bit 15: Not in Ring

This bit functions the same way as the corresponding bit on the input side. If Enable Originate is set when this bit is on and Self Test Enable is off, Ready will not clear (unless a DMA operation is also requested, in which case Ready will set when the DMA completes). This bit has no effect on output DMA.

Input Address Register Low
Input Address Register High (Two least significant bits)

The address of the buffer to which a DMA input transfer from the input packet buffer is to be performed. Note that this is a word address; therefore bit zero of the doubleword is ignored and should be zero. These registers are read/write; the unused bits will always read out as ones.

Input Word Count Register

The two's complement of the length, in words, of the DMA input operation to be performed. At any instant, the value of the input count register is the two's complement of the number of words **remaining** to be transferred. After a DMA transfer has completed or the packet buffer has emptied, the value of the register will be zero or the number of words remaining unused. Remember that this number must be less than or equal to 1023 (decimal). If this count is zero when this DMA is enabled, the zero will be taken as the two's complement of two to the sixteenth. This is a read/write register.

Output Address Register Low
Output Address Register High (Two least significant bits)

These registers function the same as the corresponding registers in the input side.

Output Word Count Register

This register functions the same as the corresponding register in the input side.

III. Programming Issues

This section considers design issues which are relevant to programming the LNI. Some of the characteristics of the LNI make it an atypical device to program.

Ring Initialization

Care should be taken when initializing the ring, since incorrect strategies can cause dynamic deadlock. The simplest correct strategy is that it is all right to initialize the ring if the Ring Not OK bit is on and either a) output of a message was just requested, or b) the Output Timeout bit set.

Basically, synchronization in attempts to reinitialize the ring should be avoided, so do **not** use the Ring Not OK bit turning on as a signal for reinitialization, since this event is synchronous in all LNI's.

There is a proposal [Saltzer2] to have the ring interface module automatically perform ring initialization, which if implemented will remove the responsibility of performing this action from the host software, in addition to changing the meaning of the Ring Not OK bit to "call for service".

Microcoding

In general the LNI can be micro-programmed to do several things at once, and will do them in the correct order; thus, on output, Clear Packet Buffer, Enable DMA and Enable Originate can all be set at once, and the LNI will perform the operations in the correct order and interrupt when the complete sequence is done. Likewise, on input, standard practice would be to enable both copy and DMA, and the DMA will start when a packet has been received.

Scatter/Gather Operations

It is possible in cases where the DMA completes with data still in the input packet buffer to empty the packet buffer without losing the packet, or to add data to the output packet buffer, since the design of the DMA section is such that scatter-gather operations may be performed. It is intended that they be used mostly on input, although the capability exists on output as well; to restart the DMA, reset the DMA control registers and set the DMA Enable bit.

Buffering

Note that the LNI is only single buffered; thus, if two packets destined for the same interface arrive in tandem, the second will be lost. If the software does not reenable input fairly quickly, other packets destined for that interface may be lost as well.

Error Detection and the Parity Bit

Note that this is **not** a data protection feature; the parity is recomputed on each link, and if an error occurs the parity will be recomputed, so that when the packet gets to its destination the parity will be correct. In any case the bit is set by any message, not just ones addressed to the station in question, so there is no relation between the Link Data Error bit and any copy performed by the LNI. This feature is included **only** to detect intermittent links in the ring. Data integrity is the responsibility of higher levels of protocol, and is **not** assured by this feature.

Retransmit Tactics

As was mentioned above, interfaces (especially loaded ones) can lose input packets through not being enabled. Although the higher level protocol can correct for this by retransmitting, a great deal of load can be taken off the system if this is done by the lower level. The contents of the output buffer can be retransmitted without reloading the packet buffer; it was so that this could be done that the output packet buffer must be cleared by the program. If a packet was not accepted (the Refused bit will be on) the first time it was transmitted, simply reenabling output will resend it.

Actual retransmission strategies can be quite complex, depending on the environment and application, and are still a subject of research. Contemporary work in this field should be studied before implementing anything complex. The ability to retransmit a packet can also be used if transmission was obviously garbled (e.g. if the Bad Format bit was on). Be careful and remember to clear the packet buffer when sending a new packet.

UNIBUSINIT

The packet buffers are cleared by a UNIBUS INIT, but except for the bits (as specified above) in the Control and Status registers, no other registers are cleared by an INIT due to the implementation method. Thus, do not rely on this typical behaviour pattern of DEC devices when programming the LNI.

III. Hardware

This section contains hardware notes that are specific to the UNIBUS interface version of the LNI.

Interrupt locations

The LNI has two interrupt locations, XX0 and XX4, where the XX portion of the vector (bits 3-7) is switch selectable. The XX0 vector is the vector for the input side, and the XX4 location is for the output side. If both sides signal an interrupt simultaneously, two interrupts will occur, one after the other as soon as the first has been handled. The input interrupt will always occur before the output interrupt, since it important to get input reenabled in cases where that is possible.

Register Locations

The locations of the LNI registers are as follows, where bits 4-12 are switch selectable:

7XXX00	Input CSR
7XXX02	Input Word Count Register
7XXX04	Input Address Register (Low)
7XXX06	Input Address Register (High)
7XXX10	Output CSR
7XXX12	Output Word Count Register
7XXX14	Output Address Register (Low)
7XXX16	Output Address Register (High)

References

[Saltzer1] Saltzer, J. H., "Version Two Local Network Interface Design Considerations," M.I.T. Laboratory for Computer Science Network Implementation Note No. 18, Cambridge, Massachussetts, June 1979.

[Salzter2] Saltzer, J. H., "Communication Ring Initialization Without Central Control," M.I.T. Laboratory for Computer Science Network Implementation Note, Cambridge, Massachussetts, In Preparation.

[Ludwig] Ludwig, C. R., and Saltzer, J.H., "Local Network Control Module Interface Specification (Revision 1)," M.I.T. Laboratory for Computer Science Network Implementation Note No. 30, Cambridge, Massachussetts, July 1980.