M.I.T. Laboratory for Computer Science

Network Implementation Note
No. 30
July 22, 1980

Local Network Control Module Interface Specifications (Revision 1)

by J. H. Saltzer, H. Arbour, G. Koss, and C. Ludwig

This note specifies the interface between the version two local network module, V2.LNI.CTL, and the host-specific buffer module, V2.LNI.HSB. Several versions of V2.LNI.HSB are being designed, for the UNIBUS (V2.LNI.UNIBUS,) the nu-bus (V2.LNI.NU,) the S-100 bus (V2.LNI.S-100,) and the Q-bus (V2.LNI.Q.) Also, it is intended that more than one V2.LNI.CTL design be usable with all of these host-specific buffer modules. It should even be possible to implement a version of the control module for a passive broadcast (e.g., ETHERNET) network. To control the interface, this document separately identifies several different categories of information:

1. The agreed-upon interface specification. The meaning of agreed-upon is that all implementations either meet this specification or have a plan to evolve to it.

2. Proposed revisions to the specification, not yet agreed upon.

3. Known differences between the agreed-upon specification and the various implementations and designs.

4. Questions that are not resolved by the specifications.

Agreed-upon Network Control Module Interface Specification

1. The format of the data passed across the interface between V2.LNI.CTL and V2.LNI.HSB is the V2.LNI 0.2 packet format. The first byte passed is the destination address.

2. Data is passed between V2.LNI.CTL and V2.LNI.HSB in 8-bit parallel form, one byte at a time. (V2.LNI.HSB insures that data bit zero is the most significant bit of the character representation on the host. V2.LNI.CTL maintains the bit numbering so that the data bit sent as bit zero is received as bit zero, etc.)

3. The length of the data packet is not passed explicitly; the end of the packet is signalled with separate status lines. (This arrangement permits the entire V2.LNI.CTL to get along without data length counters.)

4. Signals between the two modules (35 lines):

Set by HSB, read by CTL:

	Default		
JOINR	negate	Join network request	
ENMOD	negate	leave local loop back test	
INITR	negate	Initialize network	
ORIGR	negate	Originate request	
ORST*	negate	Originate reset	
COPYR	negate	Enable copy request	
LBO	assert	Last byte out	
OD0...OD7	--	Data out	Bit numbered 0 is MSB

Set by CTL, read by HSB:

REFUS*	assert	Originating packet refused	
RNOK*	assert	Network not operating correctly	
JOIND	negate	Node has joined network	
MESLOST	assert	Originating packet lost	
COPYC*	negate	Copy operation complete	
ORIGC*	negate	Originate operation complete	
ORGING	negate	Originate operation in progress	
NBI*	negate	Next byte in	
NBO	negate	Next byte out	
LBI	negate	Last byte in	
BDFORM*	assert	Packet out of format	
PERR*	assert	Parity error	
ID0...ID7	--	Data in	Bit numbered 0 is MSB

The sense of all lines is high-asserted. Asterisks in the above signal names identify those lines whose senses are inverted.

The "default" column specifies which direction the receiving end of the line should pull the line if the sending end of the line is neither asserting nor negating. This choice of default, together with the signal senses in the above list, is intended to guarantee that nothing untoward happens if the cable is accidentally disconnected. The intention is that the ring control module will do nothing, and the host-specific buffer module will return all possible error status to the host.

5. No explicit clock signals pass across the interface. To the extent possible, the interface is self-timed, with a line in one direction initiating some action and a return line that reports compliance. The exception is data transfer, which goes on at a rate determined by the ring transmission rate. The control module has an internal clock that it keeps synchronized with the network data transmission rate; that clock provides pulses that are used to request transfer of data across the interface. For all lines, the timings relative to other lines are described in the following paragraphs and summarized in the next section.

6. The "originate request" line carries a level signal, asserted by the buffer module when it has a packet ready to originate. Data to be originated is transferred according to the following typical sequence (some steps can go on in parallel or in a different order; figure 1 and the summary of line timings specify the allowable range of such sequences):

   a) HSB asserts the "originate request" line.

   b) CTL waits until the network is available, then switches to originate mode, and asserts "originate operation in progress".

   c) HSB sets ODATA(0)...ODATA(7) and holds them constant until CTL negates "next byte out".

   d) CTL asserts "next byte out" for one bit time, then negates it.

   e) Steps c) and d) are repeated as many times as necessary to transfer all bytes. Step c) must be completed within 7 bit times after step d).

   f) HSB asserts "last byte out" simultaneously with ODATA(0)...ODATA(7), and negates it after CTL negates "next byte out".

   g) CTL transmits last byte and terminates packet.

   h) HSB negates the "originate request" line (unless it is prepared to send another packet.)

   i) CTL detects that this packet has cleared the network.

   j) CTL asserts "originate operation complete" line for one bit time, then negates it.

   k) CTL negates "originate operation in progress".

   The buffer module must negate its "originate request" line sometime before the control module negates "originate operation in progress," unless it is prepared to send another packet. If the "originate request" line remains asserted, V2.LNI.CTL will attempt to recapture the transmission medium at the end of the packet just originated, and expect to send another packet. This design is intended to allow V2.LNI.CTL to be used with double buffered buffer modules for higher performance.

7. The "enable copy request" line carries a level signal, asserted by HSB when it is prepared to accept a received packet. If this level is asserted at the instant that CTL matches a packet destination address, CTL will begin to transfer data to HSB, according to the following sequence as illustrated in figure 2:

   a) HSB asserts the "enable copy request" line.

b) CTL detects destination address match.

c) CTL asserts "next byte in" for 1 bit time, then negates it. Data is available on IDATA(0)...IDATA(7) 20 ns. after negation of "next byte in". From the time that "next byte in" is negated HSB has seven bit times to read IDATA(0)...IDATA(7), before CTL goes on to step d).

d) step c) is repeated as many times as necessary to transfer all bytes.

e) CTL detects the end of packet signal.

f) CTL asserts "last byte in".

g) CTL asserts "copy operation complete" for 1 bit time, then negates it.

h) HSB negates the "enable copy request" line, if it does not want another packet.

If either a packet format or link data error occurred during the packet copy operation, CTL will have asserted the corresponding status line for one bit time before it negates "copy complete". (The purpose of "last byte in" is to permit a CTL implementation to report the presence of the last byte before it has finished format and link data error checking. This early report in turn gives a double-buffered buffer module more time to decide whether or not it can immediately accept another packet.) The buffer module must negate the "enable copy request" line sometime after the "last byte in" signal comes from CTL, and before recognition of the destination address field of the next packet (At least one byte time and as much as 2.5 byte times if "last byte in" is signalled immediately upon detection of end of packet), unless it is prepared to accept two packets in a row. If it is so prepared, it has that interval before the first byte of the next packet arrives.

8. Negation of "originate operation in progress" occurs after the "originating packet refused" and "originating packet lost" status signals have been set and the link parity bit has been checked on the returned version of the just-originated packet. CTL sets "originating packet refused" and "originating packet lost" by asserting the corresponding line for one bit time.

9. The "link error" line is asserted by CTL for one bit time whenever a link parity error is noticed, independent of whether copying, origination, or just repeating is going on. The intent is that HSB either set a latch or bump a counter, either of which are readable and resettable by the host. When a packet has been copied or originated, the corresponding "in progress" signal is negated after link errors have been checked and, if necessary, signalled.

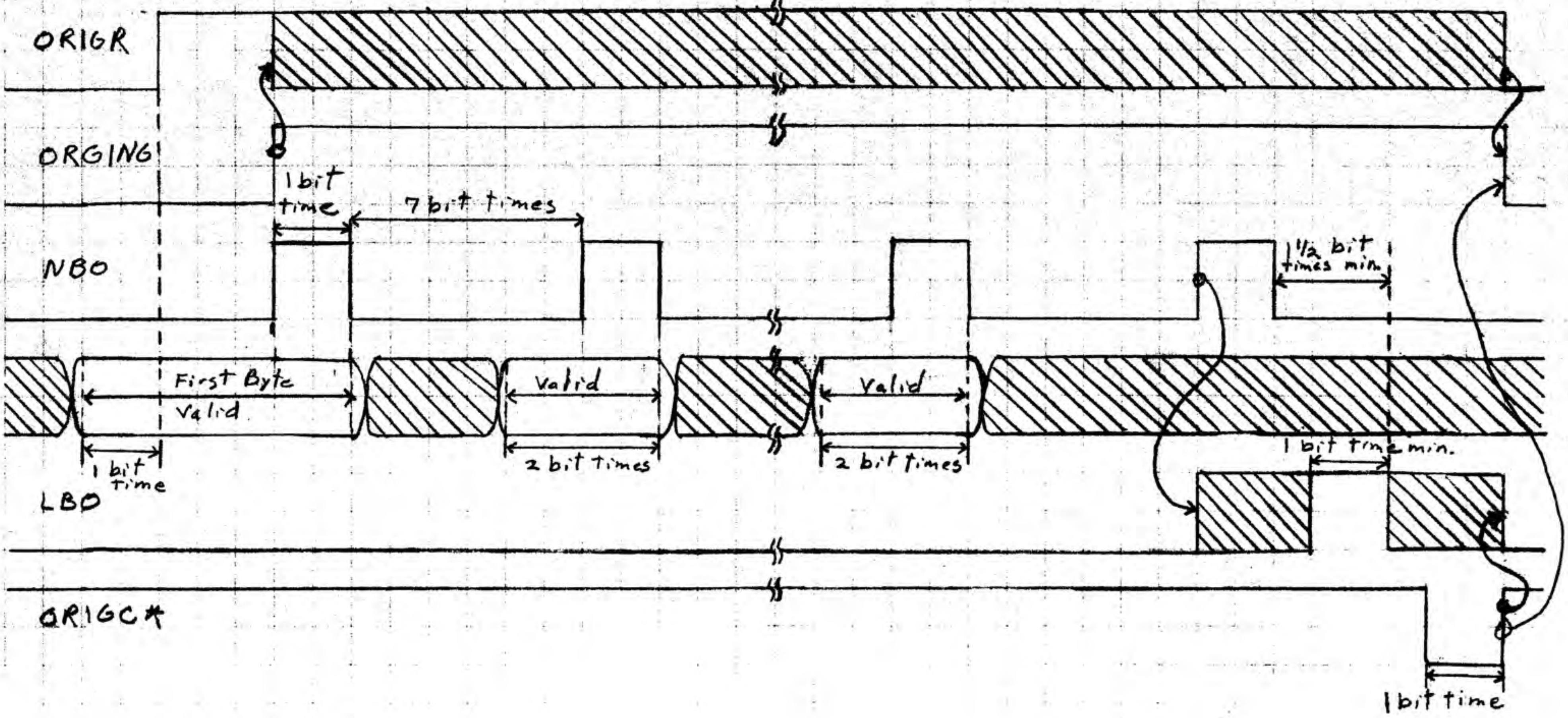
10. The "join network" request line carries a level signal that is asserted by HSB to indicate that CTL should leave analog loop back mode and join the ring. When HSB negates the level, CTL should return to analog loop back mode.

11. The "enable modem" line carries a level signal that is asserted by HSB to indicate that CTL should leave digital loopback mode. It remains asserted as long as that mode of operation is required. Both "enable modem" and "join network" must be asserted in order to participate in the network.

12. The "node has joined network" line carries a level signal that CTL asserts at all times that it appears to be participating in the network.

13. A bit time may range from 75 to 200 ns.

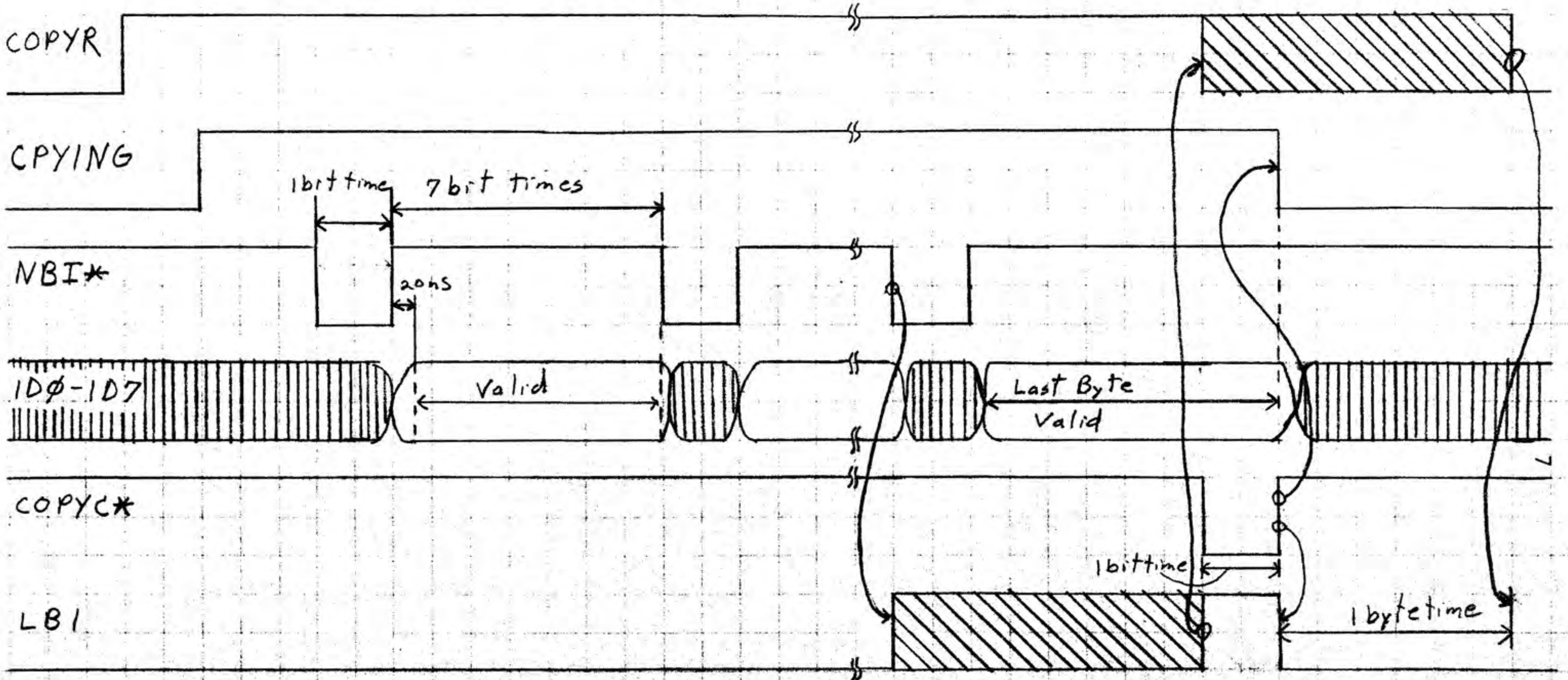
Fig. 1 Originate Mode Timing



Rev. 2 7-22-80

Rev 1
4-1-80
CVRL

Figure 2 - - Copy Mode Timing



Rev 1 7-22-80

3-27-80
CVRL

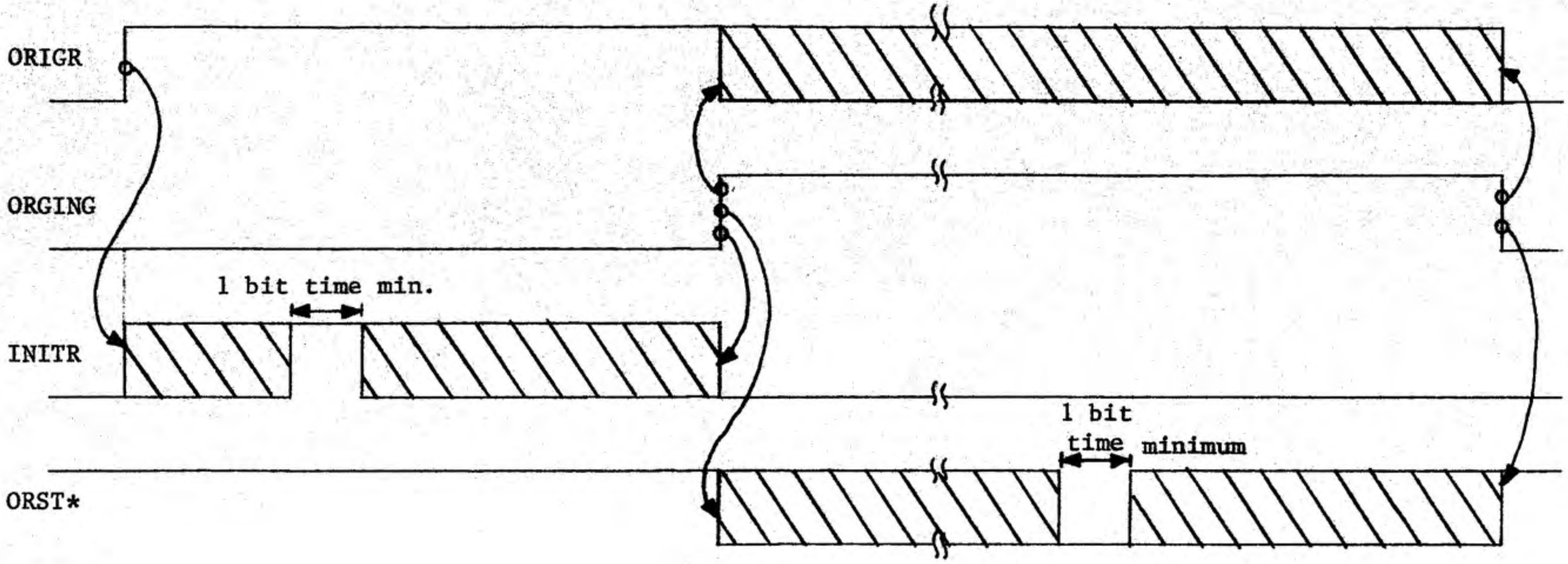
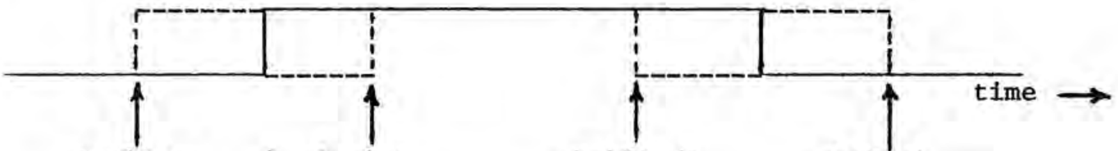


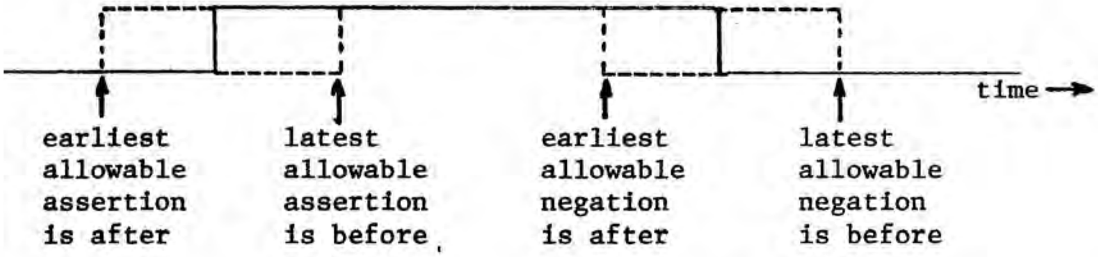
Figure 3 -- Initialize and Reset Mode Timing

Summary of line timings



	earliest allowable assertion is after	latest allowable assertion is before	earliest allowable negation is after	latest allowable negation is before
★ JOINR ★ ENMOD		(under program control, no timing limits specified)		
ORIGR	first byte available to transfer	no limit	ORGING asserted	ORGING negated (unless another packet is ready to transfer)
ORST*	ORGING asserted	no limit	1 bit time after assertion	ORIGR negated
★ INITR	RNOK* asserted	no limit	1 bit time after assertion(?)	next RNOK* asserted
★ COPYR*	prepared to accept data	first bit of address is received	COPYC* asserted (?)	COPYC* negated + 1 byte time (unless HSB prepared to accept another packet)
★ LBO	first assertion of NBO after last valid data byte	negation of NBO + ½ bit time	1½ bit times after negation of NBO	next NBO (?)
★ first ODO...7	no limit	1 bit time before ORIGR asserted	negation of NBO	negation of NBO + 6 bit times
later ODO...7	negation of NBO	negation of NBO + 6 bit times	next negation of NBO	next negation of NBO + 6 bit times
REFUS*	ORIGR asserted	(pulse asserted for 1 bit time)		ORGING negated
RNOK*	no token detected for 300 ms or no flag detected for 1.2 ms	same as earliest	token or flag detected	same as latest

continued on next page


time →
earliest allowable assertion is after
latest allowable assertion is before
earliest allowable negation is after
latest allowable negation is before

JOIND		(timing not specified)		
MESLOST	ORIGR asserted	(Pulse asserted for 1 bit time)		ORGING negated
COPYC*	7 bit times after last NBO negation	same as earliest	(Pulse asserted for 1 bit time)	
ORIGC*	9 bit times after last NBO negation	same as earliest	(Pulse asserted for 1 bit time)	
ORGING	ORIGR asserted	no limit	8 bit times after last NBI* negation	same as earliest
first NBI*	COPYR asserted	no limit	(Pulse asserted for 1 bit time)	
later NBI*	7 bit times after last NBI negate	same as earliest	(Pulse asserted for 1 bit time)	
first NBO	ORGING asserted	no limit	(Pulse asserted for 1 bit time)	
later NBO	7 bit times after last NBO negate	same as earliest	(Pulse asserted for 1 bit time)	
LBI	last NBI asserted	(Pulse asserted for 1 bit time)		COPYC* negated
BDFORM*	?	?	(Pulse asserted for 1 bit time)	
PERR	(Pulse asserted for 1 bit time)		same as latest	9 bit times after last NBO negation
first IDO...ID7	COPYR* asserted	(Same as later IDO...ID7)		
later IDO...ID7	NBI* asserted	NBI* negated	NBI* asserted + 7 bit times	NBI* negated + 7 bit times

Proposed Changes to interface specifications

1. Originate and copy protocols could be made more symmetric, two lines of the specified interface eliminated, and the originate protocol simplified, with the following change:

    a) eliminate the HSB-originated pulse signal LBO.

    b) use negation of ORIGR in response to some NBO pulse as a signal that the last byte has already been transmitted. A double-buffered HSB could re-assert ORIGR for the next packet anytime after that NBO is completed and before CTL negates ORGING.

    For symmetry, on the copy side, the current V2.LNI.UNIBUS implementation (which already omits LBI, using COPYC* instead) would be made standard, and the LBI line eliminated from the specification.

2. The three lines ORIGC*, COPYC*, and ORGING could be condensed to two, a timing-dependent specification eliminated, and the originate and copy protocols made symmetric by the following change:

    a) Add a line "copy operation in progress," or CPYING, which is specified as follows:

CPYING	COPYR* asserted	7 bit times before first NBI*	10 bit times after last NBO negation	same as earliest

    b) Eliminate ORIGC* and COPYC*. All HSB circuits that use them would respond instead to negation of ORGING and CPYING, respectively.

    c) Revise the specifications of COPYR* and LBI to refer to CPYING rather than COPYC*.

Known Differences between specification and V2.LNI.CTL--V2.LNI.UNIBUS Implementation:

1. JOINR, ENMOD, and JOIND lines are implemented, but do not do anything yet.

2. LBI is not provided (HSB is using COPYC* for this purpose. Note that originate and copy are thereby more asymmetric protocols than necessary.)

3. An extra line, "card enable", is implemented. CTL will not operate unless card enable is held at ground level. CTL pulls this line high if it is neither asserted nor negated.

4. The CPYING line is implemented as proposed above. However, V2.LNI.UNIBUS still makes use of ORIGC* and COPYC*.

Questions

1. The timing when BDFORM* is meaningful has not been specified.

2. The latest timing limit on ORIGR negation may be too late to permit CTL to recapture the just-originated token in a ring network controller. This problem would reduce effectiveness of a double-buffered HSB. It may be necessary to specify ORIGR negation no later than LBO assertion, or perhaps one-byte time thereafter.

3. The specified timing limits on JOINR and ENMOD are questionable.

4. INITR, ORST, and LBO, which are set by HSB, are specified as pulses of minimum width one bit time, but HSB does not have any clock that allows it to measure a bit time. It may be necessary to change these to a self-timed protocol.

5. The proper sequence for INITR and ORIGR is in dispute; the question revolves around whether or not RNOK* assertion causes ORIGR to be dropped, an issue in the interface between HSB and the programmer.

6. The operation of CTL is not specified if either ORIGR or COPYR is dropped prematurely. Some specification may be appropriate.

7. Thirteen of the interface lines are specified to have default values that require pull-downs rather than pull-ups. With TTL logic, pull-downs are a bad design practice; these lines should be inverted. (JOINR, ENMOD, INITR, ORIGR, COPYR, REFUS*, RNOK*, JOIND, ORGING, NBO, NBI, BDFORM*, PERR*)