A Driver/Receiver Unit for an Intercomputer Communications Link

W. A. Lushbaugh
Communications Systems Research Section

One of the most likely configurations for the computational complement of the tracking stations of the future is a number of small computers, each performing a dedicated function, interfaced with some specific subset of the tracking station equipment; each computer would be interfaced with the other computers. A previous article in this report series presented a general discussion of the intercomputer communications requirements for such a system and a system-level description of equipment for providing this intercomputer communications capability. This article describes in some detail the line-interface or driver/receiver (D/R) unit for intercomputer communications. This unit is capable of receiving and transmitting digital information a distance of up to 600 m via coaxial cable. The trilevel signaling format used to code the information is discussed as well as the various handshakes that are involved between computer and D/R unit and the transmitting D/R unit and the receiving D/R unit. Status information is supplied to the associated computer via the status register, and a bit-by-bit description of this status register is given.

I. Introduction

One of the most likely configurations for the computational complement of the tracking stations of the future is a number of small computers, each performing a dedicated function, interfaced with some specific subset of the tracking station equipment; each computer would be interfaced with the other computers. A previous article (Ref. 1) in this report series presented a general discussion of the intercomputer communications requirements for such a system and a system-level description of equipment for providing this intercomputer communications capability. This article describes in some detail the line-interface or driver/receiver (D/R) unit for intercomputer communications.

The D/R unit functions as both a transmitter and receiver of digital information. The driver (transmitting) function of the D/R unit is to convert digital signals from a computer interface buffer into a serial data stream, add a parity bit for each 8-bit byte, and transmit these bytes as a trilevel analog signal a distance of up to 600 m (2000 ft) via coaxial cable. As a receiver, the D/R unit converts the received analog signal back to digital form, checks the associated parity bit, delivers the assembled

byte to the receiving computer interface, and transmits a 2-bit control message as a request for the next byte when buffer space is available.

Software control of the D/R unit is provided through a status register. A carrier enable bit in this register allows the programmer to disable the D/R unit by unconditionally cutting off its outgoing carrier. If the carrier enable bit is set, a *start transmitting* command delivered to the status register will enable the transmitting function, while incoming carrier will cause the D/R to go into the *receive* mode. The programmer also has a reset command available and control over the value of the 2-bit control message used in the *receive* mode.

The D/R unit supplies status to the software concerning its *receiving/transmitting* state, presence of incoming carrier, end of message, and certain error conditions. In addition, two interrupts are associated with each computer, one data interrupt and one for end-of-message or error condition.

II. Signaling Format

There are two coaxial cables between any two D/R units. One is for transmitting and the other is for receiving, but only one is used at a time. Signals on the two cables are balanced and transformer-isolated at each end to minimize effects of ground level imbalance between the two computers. A trilevel signaling scheme is used: *no signal* (0 V), *positive voltage*, and *negative voltage*. The quiescent (not busy) condition of the two lines is the *no signal* state. When one D/R unit attempts to transmit a message to the other it supplies a 1-MHz carrier to its transmit cable (Fig. 1a). This turning on of the carrier by one D/R unit is a 1-bit message to the other that communication is desired. The alerted D/R unit acknowledges that message by returning a similar carrier (and later a 2-bit control message as explained below).

Digital information is conveyed by deleting one cycle of the carrier waveform, resulting in a bit time of 2 μs. The value of the bit is given by the phase of the last cycle of the carrier before deletion (Fig. 1b). Since the value of the first bit of a group of transmitted bits is given by the last cycle of the carrier that preceded the first gap, there is a possible 0.5-μs delay in its transmission while waiting for the carrier to come to the correct phase. The slight loss of time involved with this encoding scheme was felt worthwhile, considering the ease of converting the analog signal to the digital domain.

The analog line signals are converted to a digital waveform by a line receiver with hysteresis. The line receiver acts as a comparator and switches its output to a low level when the input voltage difference is greater than 2.5 V. Having switched to the low output state, however, the input voltage difference must return to less than 0.7 V before the output will switch again. This 1.8 V of hysteresis results in digital waveforms like those of Figs. 2a and 2b for the trilevel line signals of Figs. 1a and 1b respectively.

Reclocking of the waveform is accomplished by observing the transitions of the incoming digital waveform and the spacing between them. When only carrier is present, there is a waveform transition every 0.5 μs. Insertion of a data bit into the carrier stream causes a transition spacing of at least 1.5 μs and, in some cases, 2 μs. In either case a transition absence detector, which produces a pulse every microsecond unless reset by a transition, signifies the occurrence of a data bit (or end of carrier), and the level of the digital waveform (with hysteresis) gives its value. Similarly, the transition absence detector continuously delivers pulses every microsecond when the carrier is off.

III. Communications Example

There are many levels of handshaking involved in the communications of computers with D/R units. Initially, the calling computer (A) alerts its D/R unit by enabling the carrier enable flipflop (if not previously enabled) and also giving a *start transmitting* command. If there is no incoming carrier at this time, the D/R unit will go to the *busy transmitting* state and supply a carrier to the called computer (B). Simultaneously, a data interrupt will be generated into computer A to signify that the output buffer is empty. The generation of this interrupt is assured by the interface logic which holds the *buffer ready* signal in the not ready state when the D/R unit is not busy. The data interrupt can only occur when the D/R unit is busy.

After supplying the carrier to B, the D/R unit at the A end will wait for a return carrier. The D/R unit at B will return the carrier, provided its carrier enable flipflop has been enabled by B's program. Moreover, A will still wait for a 2-bit control package (normally 00) from B before sending the first byte. This control message will in turn only be sent after a data interrupt has been given to computer B and B has responded. This first data interrupt at the receiving end is similar to that at the transmit end; i.e., the interface logic holds the *buffer ready* signal in the not ready state when the D/R unit is not busy, a condition

which will generate this interrupt when the incoming carrier causes the unit to go *busy receiving*. This interrupt is superfluous in that it must be cleared by reading the contents of the buffer, which at this time is actually empty, but it alerts the software in B that it is being called and lessens the probability of a software race. The time between going *busy receiving* and the response of reading the buffer also serves to put a header of carrier in front of the first control message to A. This first control message is automatically sent as soon as the *buffer ready* signal goes true. The control message is otherwise artificially generated by the D/R unit, and its transmission is not reported to the byte counter in the interface logic. Thus computer A receives two indicators at the start of transmitting: (1) return carrier signifies active hardware at B's end; (2) return control message signifies an active program in B.

Thereafter, A sends the first byte from its buffer, B receives it and returns a control message, and the process is repeated at a rate dependent upon the computers involved. When the transmitting end has emptied its buffer, it interrupts computer A with a request for more data. Receipt of a control package request from B will not be obeyed until the buffer has been filled. Operation at the other end is similar in that a control message will not be sent unless there is buffer space available. A slight difference at the receiving end is that the D/R unit has storage for one byte on it. Since this byte is transferred in parallel to the buffer as the next control message is being sent, a *buffer not ready* signal cannot be generated until this time. Thus the next byte may arrive and be stored in this extra byte buffer before the word buffer has been cleared by the interrupt routine in B. This overlapping of byte transmission and interrupt routine helps to increase the average speed of transmission to the point where approximately 200K-bits/s transmission rates have been observed, including interrupt handling time.

When computer A has sent its last word (only complete words may be sent), an interrupt will be generated in that machine as a request to refill its buffer. The program at this time will decide to end transmission and will notify the status register by turning off the carrier enable flipflop. This will abruptly cut off the carrier to B and generate a status interrupt at that end. In the meantime, B has probably sent a control package to A, but this command will be ignored. The carrier off interrupt to B will notify that machine of the end of the message. Then B should check the status register to see if any error flags have been set or if there are some odd bytes in the buffer (A need not have sent a multiple of B's words). When the software in B is ready, it disables the carrier enable flipflop and resets its end-of-message indicator.

This final handshake of the transmission will generate an end-of-message interrupt at the A computer. This interrupt is the acknowledgment to the sender that transmission has terminated. Before this acknowledgment, the A end was in a slave state; i.e., an attempt to transmit again would result in its D/R unit going *busy receiving*.

IV. Status Register

The status register is the means by which software and hardware communicate. The software delivers commands to the D/R unit via the output status register and reads back various status data from the input status register. Despite this split input/output function, the status register is a single register of 16 bits. The split function was necessary to accommodate a 16-bit machine.

A list of input and output bit assignments is given in Table 1. Both XDS 900 series and PDP-11 bit names are given to avoid confusion. Following is a detailed description of each signal:

Bit 0 Out: *Start Transmitting.* A 1 transferred to the output status register in this bit position causes a *start transmitting* pulse to be delivered to the D/R unit. This will cause the D/R unit to go *busy transmitting* (see *bit 0 in*) if the carrier enable bit has been set (or is simultaneously set), provided that the D/R unit is not *busy receiving*. Since incoming carrier takes precedence over the *start transmitting pulse*, it is necessary to read the status register after giving this command to verify that it was obeyed.

Bit 2 Out: *Carrier Enable. Bit 2 out* and *in* controls the carrier enable flipflop. This flipflop when set allows the D/R unit to output its carrier either in response to input carrier or when starting to transmit. This flipflop must be reset to end the transmit mode. After receipt of an *end-of-message* (see *Bit 3 in*) signal in *receive* mode it must also be reset to cut off the return carrier.

Bit 3 Out: *Reset.* Transferring a 1 to this output status bit will cause the hardware to generate a 1-μs reset pulse. It resets signals as indicated in the input status register (see Table 1).

Bit 4 Out: *Control Message Flipflop.* The contents of this flipflop are copied twice to form the control message (request for next byte) in *receive* mode. The normal control message of 00 causes the D/R unit at the other end of the link to send the next byte without interrupting that computer. Receipt of an 11 control message will cause the next byte to be sent but will also set status *bit 10 in* and generate the status interrupt. The

current software convention is that control message 11 will be used to signal overflow of the receiving computer's message buffer.

Bits 5 and 6 Out. These bits are supplied in the PDP-11 computer to enable the two interrupts. Their implementation in any other machine is optional to the designer/programmer.

Input Status.

Bit 0 In: *Busy Transmitting.* A 1 in this bit position signifies that the D/R unit is busy transmitting, i.e., that carrier is being supplied to the opposite end of the communications link and that serial by bit transmission of bytes will occur upon receipt of a control message. To exit this mode, the programmer must reset the carrier enable flipflop (see *bit 2 out*).

Bit 1 In: *Busy Receiving.* A 1 in this position signifies that the D/R unit is busy receiving information from the opposite end of the link. This bit is set only when the carrier enable flipflop is set and a carrier is received from the other end. Once having entered this mode the D/R unit will remain *busy receiving* after the incoming carrier is cut off until the carrier enable flipflop is reset.

Bit 2 In. The same as *bit 2 out*, i.e., the contents of the carrier enable flipflop.

Bit 3 In: *End of Message.* This status information is supplied by the D/R unit and signifies the termination of incoming carrier. The setting of this bit is the normal exit from *receive* mode. It will remain set until shut off by the reset signal. It is also one term in the B interrupt (status interrupt).

Bit 4 In. The same as *bit 4 out*, i.e., the contents of the control message flipflop.

Bits 5 and 6 In. Optional enable interrupt flipflops.

Bit 7 In: *Interrupt A (Data Interrupt).* This bit in the status register is a copy of the data interrupt developed in the interface logic. It is to be the higher priority of the two interrupts associated with the D/R unit.

The next four bits (bits 8–11) in the output status register are error indicators and have the following properties in common:

(1) They are *or*-ed together (along with *end of message*) to form the B interrupt (status interrupt).

(2) They are all reset automatically when the D/R unit goes *not busy* or by the reset command (see *bit 3 out*).

Bit 8 In: *Error Number of Bits.* A binary 1 in this position indicates that the D/R unit received a message of incorrect length. If this bit is set, it is highly likely that the D/R unit is no longer active, since it is waiting for a byte packet or control message of correct length.

Bit 9 In: *Parity Error.* A binary 1 in this position indicates that the parity bit received with a byte was incorrect. The D/R unit remains active.

Bit 10 In: *Contrcl Message 11 Received.* A binary 1 in this position indicates that a double 1 control message was received. This may or may not be an error, depending upon the programming.

Bit 11 In: *Bit Received Transmitting.* A binary 1 in this position means that the D/R unit detected input data bit(s) while it was in the process of sending either a control message or a byte. Setting of this bit usually indicates a hardware malfunction or an extremely noisy environment.

Bit 12 In: *Incoming Carrier On.* This bit is set when the D/R unit detects incoming carrier and remains set until the end of message (*bit 3 in*) is set. When this bit is on, a transmit command will not be obeyed. Consideration was given to entering this term into the status interrupt under the control of another flipflop in the output status register, but it was deemed unnecessary at this time.

Bits 14 and 13 In: *First/Second Byte Here.* These bits are supplied from the byte counter in the computer interface logic. They are set to 1 as the first and then the second byte (in three-bytes/word machines) arrive. Since it is convenient to use the same counter in both the *transmit* and *receive* modes, these bits also signify first or second byte sent in *transmit* mode. These bits are usually set when the D/R unit is not busy.

Bit 15 In: *Interrupt B (Status Interrupt).* This bit is a copy of the lower-priority interrupt supplied to the computer. This interrupt is generated by the occurrence of any of the status bits footnoted in Table 1. It is cleared with the reset command (see *bit 3 out*).

V. Logical Races

Since both ends of the communication link act independently, there is a possibility of two machines trying to transmit at the same time. If the timing of transmission is such that the first carrier signals from each D/R unit cross in the cables, both D/R units will be in the *busy transmitting* mode. Once in this mode, neither D/R unit will do anything, as a first control message is needed to start the handshaking. For this reason, a software timeout on the link is necessary.

If the timing is slightly different so that one computer is delivering a *start transmitting* command just as the first cycles of carrier are arriving, this D/R unit will lose the race by going *busy receiving*. This fact can be determined by the software by having it read the status register at the time of the first data interrupt. As long as this program has a receive buffer available, it can proceed into *receive* mode when it loses a race in this manner. Thus the critical window for a race event which can hang up the link is the propagation time over the cable—a few microseconds at most.

VI. Interface Requirements

The D/R unit is intended to be a general-purpose communications unit demanding only a small interface between it and a computer. This interface will need to perform any level shifting into and out of a computer with different than TTL logic levels (0–5 V). In addition, it will have to buffer computer words on input and output. The output buffer is a shift register, while the input register is to accept parallel bytes from the D/R.

The status register is also part of the interface, although most of the input bits are developed and stored on the D/R unit. The output status register must be part of the interface, but only two bits of storage and two one-shots are needed.

Some logic has to be done by the interface. This interface logic must include a byte counter which counts a signal supplied by the D/R when bytes arrive or leave. On *receive* (*transmit*), when this byte counter indicates that the word buffer is full (empty), an interrupt is to be generated into the associated computer. This byte counter must also provide signals to indicate first/second byte arrived (gone).

Figure 3 shows a logic diagram of the byte counter and input data buffer for a two-byte machine. It is noticed that both bits of the counter (flipflops A and B) are held reset when *busy* is low, i.e., when the D/R unit is not busy. The *busy* term is also entered into the *and* gate C to cut off the data interrupt if the D/R is not busy. Thus when the D/R unit goes *busy* the reset side of flipflop B ($\overline{\textit{buffer ready}}$) will produce the interrupt.

The counter consisting of flipflops A and B thus starts a new cycle in the 0-0 state, and the set output of the B flipflop is sent to the D/R unit as the *buffer ready* signal. The set outputs of A and B also control the quad latches of the SN 7475's F, G, H, J. As long as A and B are 0, the contents of all the buffers are held constant, and when they go high, the inputs D_0 through D_7 are copied continuously onto the outputs.

When the interrupt is responded to, a signal must be developed somewhere in the interface to signify that either new data was loaded into the output buffer or that the contents of the input register have been read. A high-going signal of this type is shown entering one leg of NAND gate D and also into the set enable (J term) of flipflops A and B. After this signal leaves gate D, it enters gate E and results in producing a clock pulse to flipflops A and B. The master/slave characteristics of flipflops A and B assure that this simultaneous clock and enable will result in both flipflops setting.

VII. Development Status

The driver/receiver unit described in this article is one of the essential elements of the multicomputer communications system discussed in Ref. 1. The D/R units have been designed, fabricated, and successfully tested. Computer interfaces for word-by-word operation controlled by the central processing unit (CPU) have been designed and built for the PDP-11, Xerox 900-Series, and Xerox Sigma 5 computers. The interface at the Sigma 5 incorporates a multiplexer at the line driver and receiver interfaces of the D/R unit, which allows it to selectively communicate with one of several machines. Error-free transfers of over 10^9 bits have been obtained between these machines. Successful application of these links awaits only the completion of link-handling software and the embedding of the required software into the Sigma 5 operating system.

Reference

1. Layland, J. W., and Lushbaugh, W. A., "A Multicomputer Communication System," in *The Deep Space Network Progress Report*, Vol. XII, pp. 195–198, Technical Report 32-1526, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 15, 1972.

Table 1. Status register bit assignments

Output			Input		
930	PDP-11	Assignment	930	PDP-11	Assignment
23	0	Start transmitting	23	0	Busy transmitting
22	1		22	1	Busy receiving
21	2	Carrier enable	21	2	Carrier enable
20	3	Reset	20	3	End of message[a]
19	4	Control message flipflop	19	4	Control message flipflop
18	5	Enable B interrupt (PDP-11 only)	18	5	Enable B interrupt (PDP-11 only)
17	6	Enable A interrupt (PDP-11 only)	17	6	Enable A interrupt (PDP-11 only)
16	7		16	7	Interrupt A (data) (higher priority than interrupt B)
15	8		15	8	Error number bits received[a]
14	9		14	9	Parity error[a]
13	10		13	10	Control message 11 received[a]
12	11		12	11	Bit received transmitting (byte or control message)[a]
11	12		11	12	Incoming carrier on
10	13		10	13	Second byte here (930 only)
9	14		9	14	First byte here
8	15		8	15	Interrupt B (end of message or error)
			0	–	Interrupt A

[a]Signals that are reset by the reset command (bit 3 out) and *or*-ed together to form the B interrupt.

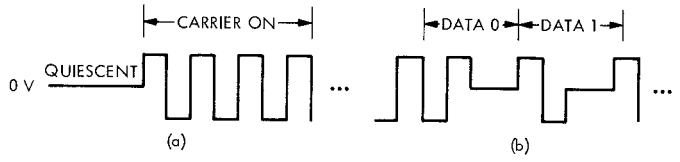

Fig. 1. Cable waveforms

Fig. 2. Restored waveforms

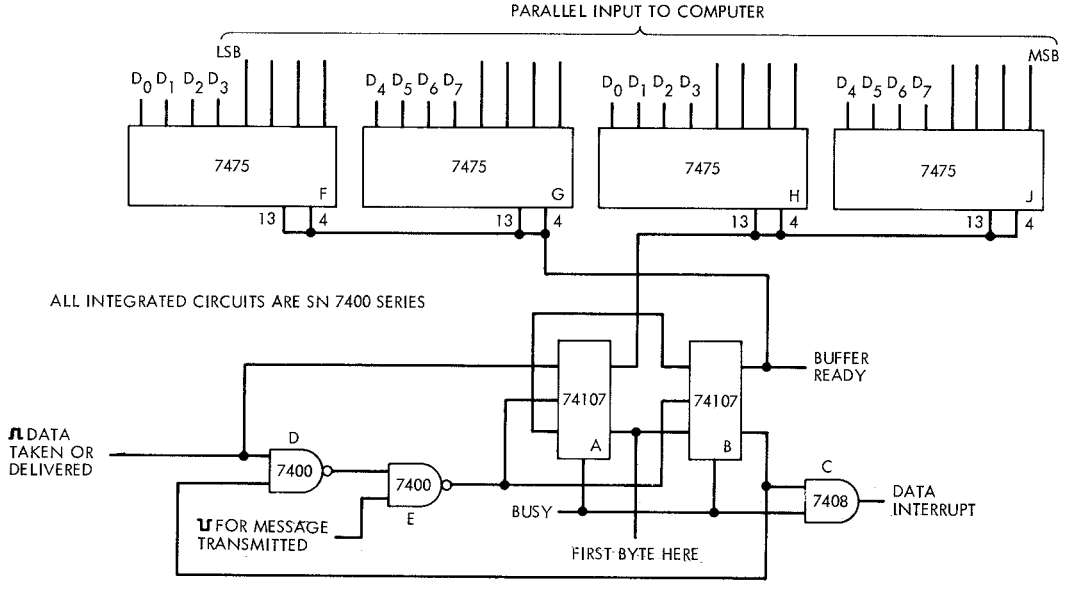

Fig. 3. Status register bit assignments