Interfacing the DP8344 to Twinax

National Semiconductor
Application Note 516
Thomas J. Quigley
March 1988

OVERVIEW

The DP8344, or **Biphase Communications Processor** from National Semiconductor's Advanced Peripherals group brings a new level of system integration and simplicity to the IBM® connectivity world. Combining a 20 MHz RISC architecture CPU with a flexible multi-protocol transceiver and remote interface, the BCP is well suited for IBM 3270, 3299 and 5250 protocol interfaces. This Application Note will show how to interface the BCP to twinax, as well as provide some basics about the IBM 5250 environment.

5250 ENVIRONMENT

The IBM 5250 environment encompasses a family of devices that attach to the IBM System/34, 36 and 38 mid-size computer systems. System unit model numbers include the 5360, 5362, 5364, 5381, and 5382, and remote controller models 5294 and 5251 model 12. The system units have integral work station controllers and some may support up to 256 native mode twinax devices locally. Native mode twinax devices are ones that connect to one of these host computers or their remote control units via a multi-drop, high speed serial link utilizing the 5250 data stream. This serial link is primarily implemented with twinaxial cable but may be also found using telephone grade twisted pair. Native mode 5250 devices include mono-chrome, color and graphics terminals, as well as a wide range of printers and personal computer emulation devices.

TWINAX AS A TRANSMISSION MEDIA

The 5250 environment utilizes twinax in a multi-drop configuration, where eight devices can be ''daisy-chained'' over a total distance of 5000 ft. and eleven splices, (each physical device is considered a splice) see *Figure 1*. Twinax can be routed in plenums or conduits, and can be hung from poles between buildings (lightning arrestors are recommended for this). Twinax connectors are bulky and expensive, but are very sturdy. Different sorts may be purchased from IBM or a variety of third party vendors, including Amphenol. Twinax should not be spliced; to connect cables together both cables should be equipped with male connectors and a quick-disconnect adapter should be used to join them (Amphenol #82-5588).

Twinaxial cable is a shielded twisted pair that is nearly 1/3 of an inch thick. This hefty cable can be either vinyl or teflon jacketed and has two internal conductors encased in a stiff polethylene core. The cable is available from BELDEN (type #9307) and other vendors, and is significantly more expensive than coax.

The cable shield must be continuous throughout the transmission system, and be grounded at the system unit and each station. Since twinax connectors have exposed metal connected to their shield grounds, care must be taken not to expose them to noise sources. The polarity of the two inner conductors must also be maintained throughout the transmission system.

The transmission system is implemented in a balanced current mode; every receiver/transmitter pair is directly coupled to the twinax at all times. Data is impressed on the transmission line by unbalancing the line voltage with the driver current. The system requires passive termination at both ends of the transmission line. The termination resistance value is given by:

$R_t = Z_{O/2}$; where

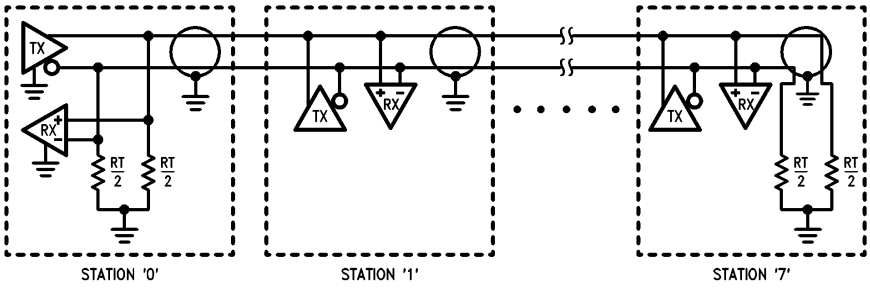
R_t: Termination Resistance

Z_o: Characteristic Impedance

In practice, termination is accomplished by connecting both conductors to the shield via 54.9Ω, 1% resistors; hence the characteristic impedance of the twinax cable of 107Ω ±5% at 1.0 MHz. Intermediate stations must not terminate the line; each is configured for ''pass-through'' instead of ''terminate'' mode. Stations do not have to be powered on to pass twinax signals on to other stations; all of the receiver/transmitter pairs are DC coupled. Consequently, devices must never output any signals on the twinax line during power-up or down that could be construed as data, or interfere with valid data transmission between other devices.

WAVEFORMS

The bit rate utilized in the 5250 protocol is 1 MHz ±2% for most terminals, printers and controllers. The IBM 3196 dis-



TL/F/9635–1

FIGURE 1. Multi-Drop Transmission Lines

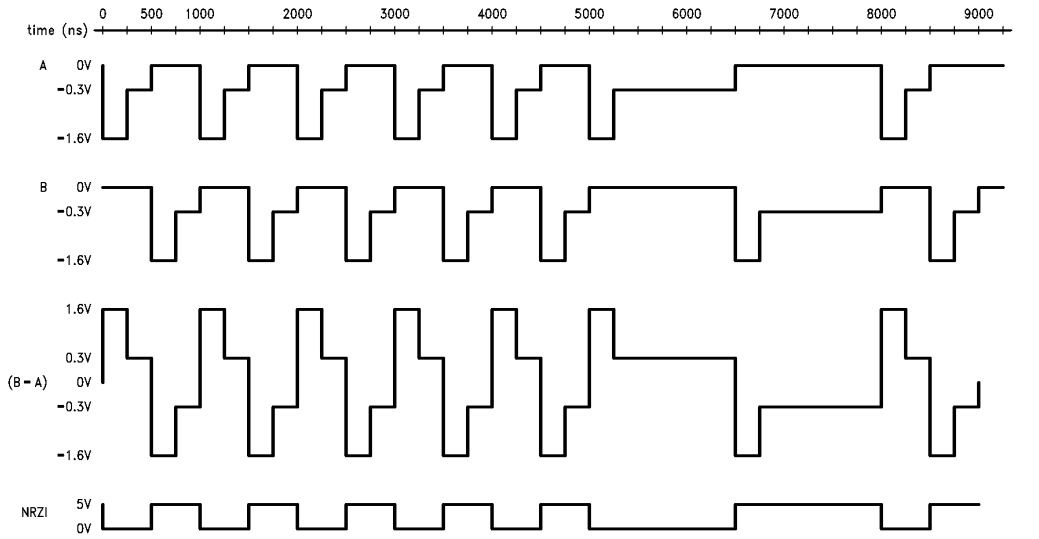
The eight stations shown include the host device as a station. The first and last stations are terminated while intermediate stations are not.

IBM® is a registered trademark of International Business Machines Corporation.

RRD-B30M115/Printed in U. S. A.

play station has a bit rate of 1.0368 MHz ±0.01%. The data are encoded in biphase, NRZI (non-return to zero inverted) manner; a ''1'' bit is represented by a positive to negative transition, a ''0'' is a negative to positive transition in the center of a bit cell. This is opposite from the somewhat more familiar 3270 coax method. The biphase NRZI data is encoded in a pseudo-differential manner; i.e. the signal on the ''A'' conductor is subtracted from the signal on ''B'' to form the waveform shown in *Figure 2*. Signals A and B are not differentially driven; one phase lags the other in time by 180°. *Figures 3* and *4* show actual signals taken at the driver and receiver after 5000 ft. of twinax, respectively.

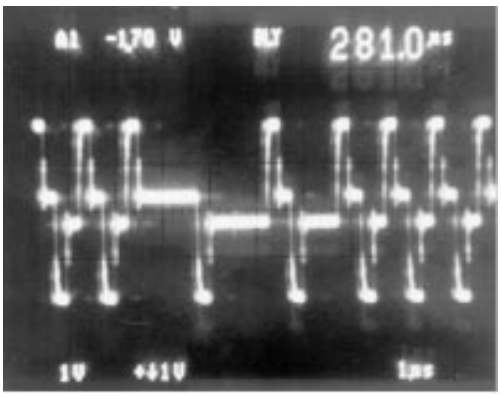
The signal on either the A or B phase is a negative going pulse with an amplitude of −0.32V ±20% and duration of 500 ±20 ns. During the first 250 ±20 ns, a predistortion or pre-emphasis pulse is added to the waveform yielding an amplitude of −1.6V ±20%. When a signal on the A phase is considered together with its B phase counterpart, the resultant waveform represents a bit cell or bit time, comprised of two half-bit times. A bit cell is 1 μs ±20 ns in duration and must have a mid bit transition. The mid bit transition is the synchronizing element of the waveform and is key to maintaining transmission integrity throughout the system.



TL/F/9635–2

FIGURE 2. Twinax Waveforms

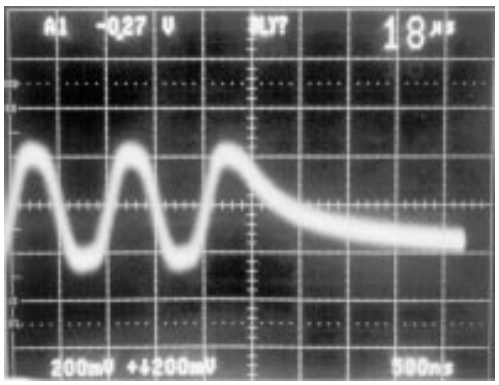
The signal on phase A is shown at the initiation of the line quiesce/line violation sequence.
Phase B is shown for that sequence, delayed in time by 500 ns.
The NRZI data recovered from the transmission.



TL/F/9635–3

FIGURE 3. Signal at the Driver

The signal shown was taken with channel 1 of an oscilloscope connected to phase B, channel 2 connected to A, and then channel 2 inverted and added to channel 1.



TL/F/9635–4

FIGURE 4. Signal at the Receiver

The signal shown was viewed in the same manner as *Figure 3*. The severe attenuation is due to the filtering effects of 5000 ft. of twinax cable.

As previously mentioned, the maximum length of a twinax line is 5000 ft. and the maximum number of splices in the line is eleven. Devices count as splices, so that with eight devices on line, there can be four other splices. The signal 5000 ft. and eleven splices from the controller has a minimum amplitude of 100 mV and a slower edge rate. The bit cell transitions have a period of 1 μs $\pm$30 ns.

5250 BIT STREAM

The 5250 Bit stream used between the host control units and stations on the twinax line consists of three separate parts; a bit synchronization pattern, a frame synchronization pattern, and one or more command or data frames. The bit sync pattern is typically five one bit cells. This pattern serves to charge the distributed capacitance of the transmission line in preparation for data transmission and to synchronize receivers on the line to the bit stream. Following the bit sync or line quiesce pattern is the frame sync or line violation. This is a violation of the biphase, NRZI data mid bit transition rule. A positive going half bit, 1.5 times normal duration, followed by a negative going signal, again 1.5 times normal width, allows the receiving circuitry to establish frame sync.

Frames are 16 bits in length and begin with a sync or start bit that is always a 1. The next 8 bits comprise the command or data frame, followed by the station address field of three bits, a parity bit establishing even parity over the start, data and address fields, and ending with a minimum of three fill bits (fill bits are always zero). A message consists of a bit sync, frame sync, and some number of frames up to 256 in total. A variable amount of inter-frame fill bits may be used to control the pacing of the data flow. The SET MODE command from the host controller sets the number of bytes of zero fill sent by attached devices between data frames. The zero fill count is usually set to zero. The number of zero fill bits injected between frames by the BCP is set by the Fill Bit select register {FBR}. This register contains the one's complement of the number of BITS sent, not bytes.

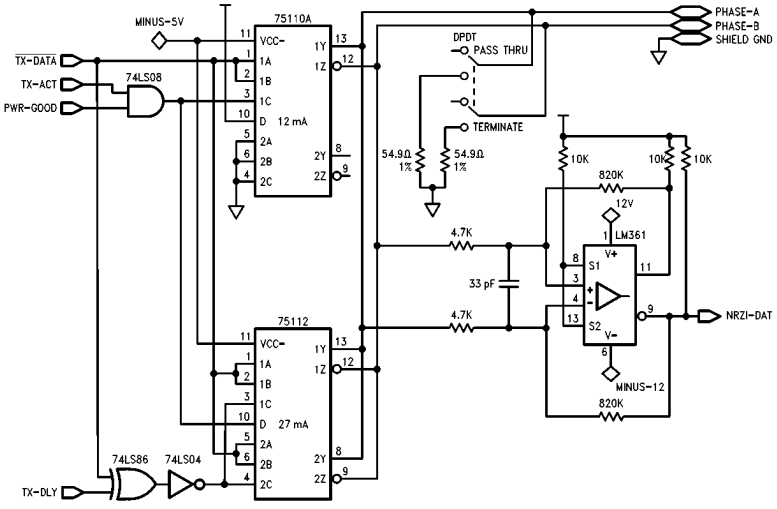
Message routing is accomplished through use of the three-bit address field and some basic protocol rules. As mentioned above, there is a maximum of eight devices on a given twinax line. One device is designated the controller or host and the remaining seven are slave devices. All communication on the twinax line is host initiated and half duplex. Each of the seven devices is assigned a unique station address from zero to six. Address seven is used for an End Of Message delimiter, or EOM. The first or only frame of a message from controller to device must contain the address of the device. Succeeding frames do not have to contain the same address for the original device to remain selected, although they usually do.

The last frame in a sequence must contain the EOM delimiter. For responses from the device to the controller, the responding device places its own address in the address field in frames 1 to (n $-$ 1),where n $\leq$ 256, and places the EOM delimiter in the address field of frame n. However, if the response to the controller is only one frame, the EOM delimiter is used. The controller assumes that the responding devices was the one addressed in the initiating command.

Responses to the host must begin in 60 $\pm$20 μs, although some specifications state a 45 $\pm$15 μs response time. In practice, controllers do not change their time out values per device type so that anywhere from 30 μs to 80 μs response times are appropriate.

DRIVER CIRCUITS FOR THE DP8344

The transmitter interface on the DP8344 is sufficiently general to allow use in 3270, 5250, and 8-bit transmission systems. Because of this generality, some external hardware is needed to adapt the outputs to form the signals necessary to drive the twinax line. The chip provides three signals: $\overline{\text{DATA-OUT}}$, DATA-DLY, and TX-ACT. $\overline{\text{DATA-OUT}}$ is biphase serial data (inverted). DATA-DLY is the biphase serial data output (non-inverted) delayed one-quarter bit-time. TX-ACT, or transmitter active, signals that serial data is being transmitted when asserted. $\overline{\text{DATA-OUT}}$ and DATA-DLY can be used to form the A and B phase signals with their three levels by the circuit shown in *Figure 5*. TX-ACT is used as an external transmitter enable. The BCP can invert the sense of the $\overline{\text{DATA-OUT}}$ and DATA-DLY signals by asserting TIN {TMR[3]}. This feature allows both 3270 and 5250 type biphase data to be generated, and/or utilization of inverting or non-inverting transmitter stages.



TL/F/9635–5

FIGURE 5. Schematic

The current mode drive method used by native twinax devices has both distinct advantages and disadvantages. Current mode drivers require less power to drive properly terminated, low-impedance lines than voltage mode drivers. Large output current surges associated with voltage mode drivers during pulse transition are also avoided. Unwanted current surges can contribute to both crosstalk and radiated emission problems. When data rate is increased, the surge time (representing the energy required to charge the distributed capacitance of the transmission line) represents a larger percentage of the driver's duty cycle and results in increased total power dissipation and performance degradation.

A disadvantage of current mode drive is that DC coupling is required. This implies that system grounds are tied together from station to station. Ground potential differences result in ground currents that can be significant. AC coupling removes the DC component and allows stations to float with respect to the host ground potential. AC coupling can also be more expensive to implement.

Drivers for the 5250 environment may not place any signals on the transmission system when not activated. The power-on and off conditions of drivers must be prevented from causing noise on the system since other devices may be in operation. *Figure 5* shows a ''DC power good'' signal enabling the driver circuit. This signal will lock out conduction in the drivers if the supply voltage is out of tolerance.

Twinax signals can be viewed as consisting of two distinct phases, phase A and phase B, each with three levels, off, high and low. The off level corresponds with 0 mA current being driven, the high level is nominally 62.5 mA, +20%−30%, and the low level is nominally 12.5 mA, +20%−30%. When these currents are applied to a properly terminated transmission line the resultant voltages impressed at the driver are: off level is 0V, low level is 0.32V ±20%, high level is 1.6V ±20%. The interface must provide for switching of the A and B phases and the three levels. A bi-modal constant current source for each phase can be built that has a TTL level interface for the BCP.

An integrated solution can be constructed with a few current mode driver parts available from National and Texas Instruments. The 75110A and 75112 can be combined to provide both the A and B phases and the bi-modal current drive required as in *Figure 5*. The external logic used adapts the coax oriented BCP outputs to the twinax interface circuit, and prevents spurious transmissions during power-up or down. The serial NRZ data is inverted prior to being output by the BCP by setting TIN, {TMR[3]}.

RECEIVER CIRCUITS

The pseudo-differential mode of the twinax signals make receiver design requirements somewhat different than the coax 3270 world. Hence, the analog receiver on the BCP is not well suited to receiving twinax data. The BCP provides both analog inputs to an on-board comparator circuit as well as a TTL level serial data input, TTL-IN. The sense of this serial data can be inverted by the BCP by asserting RIN, {TMR[4]}.

The external receiver circuit must be designed with care to ensure reliable decoding of the bit-stream in the worst environments. Signals as small as 100 mV must be detected. In order to receive the worst case signals, the input level switching threshold or hysteresis for the receiver should be nominally 29 mV ±20%. This value allows the steady state, worst case signal level of 100 mV 66% of its amplitude before transitioning.

To achieve this, a differential comparator with complementary outputs can be applied, such as the National LM361. The complementary outputs are useful in setting the hysteresis or switching threshold to the appropriate levels. The LM361 also provides excellent common mode noise rejection and a low input offset voltage. Low input leakage current allows the design of an extremely sensitive receiver, without loading the transmission line excessively.

In addition to good analog design techniques, a low pass filter with a roll-off of approximately 1 MHz should be applied to both the A and B phases. This filter essentially conducts high frequency noise to the opposite phase, effectively making the noise common mode and easily rejectable.

Layout considerations for the LM361 include proper bypassing of the ±12V supplies at the chip itself, with as short as possible traces from the pins to 0.1 μF ceramic capacitors. Using surface mount chip capacitors reduces lead inductance and is therefore preferable in this case. Keeping the input traces as short and even in length is also important. The intent is to minimize inductance effects as well as standardize those effects on both inputs. The LM361 should have as much ground plane under and around it as possible. Trace widths for the input signals especially should be as wide as possible; 0.1 inch is usually sufficient. Finally, keep all associated discrete components nearby with short routing and good ground/supply connections.

Design equations for the LM361 in a 5250 application are shown here for example. The hysteresis voltage, V_h, can be expressed the following way:

$$V_h = V_{rio} + ((R_{in}/(R_{in} + R_f) \times V_{ol}) - (R_{in}/(R_{in} + R_f) \times V_{ol}))$$

where

V_h — Hysteresis Voltages, Volts

R_{in} — Series Input Resistance, Ohms

R_f — Feedback Resistance, Ohms

C_{in} — Input Capacitance, Farads

V_{rio} — Receiver Input Offset Voltage, Volts

V_{oh} — Output Voltage High, Volts

V_{ol} — Output Voltage Low, Volts

The input filter values can be found through this relationship:

$$V_{cin} = V_{in1} - V_{in2/1} + jwC_{in}(R_{in1} + R_{in2})$$

where $R_{in1} = R_{in2} = R_{in}$:

$$Fro = w/2\pi$$

$$Fro = 1/(2\pi \times R_{in} \times C_{in})$$

$$C_{in} = 1/(2\pi \times R_{in} \times Fro)$$

where

V_{in1}, V_{in2} — Phase A and B signal voltages, Volts

V_{cin} — Voltage across C_{in}, or the output of the filter, Volts

R_{in1}, R_{in2} — Input resistor values, $R_{in1} = R_{in2}$, Ohms

Fro — Roll-Off Frequency, Hz

W — Frequency, Radians

The roll-off frequency, Fro, should be set nominally to 1 MHz to allow for transitions at the transmission bit rate. The transition rate when both phases are taken together is 2 MHz, but then R_{in1} and R_{in2} must be considered, so:

$Fro2 = 1/(2\pi \times (R_{in1} + R_{in2}) \times C_{in})$

or,

$Fro2 = 1/(2\pi \times 2 \times R_{in} \times C_{in})$

where $Fro2 = 2 \times Fro$, yielding the same results.

The following table shows the range of values expected:

TABLE I

Value	Maximum	Minimum	Nominal	Units	Tolerance
R_{IN}	4.935E+03	4.465E+03	4.700E+03	Ω	0.05
R_F	8.295E+05	7.505E+05	7.900E+05	Ω	0.05
C_{IN}	4.4556E−11	2.6875E−11	3.3863E−11	F	
V_{OH}	5.250E+00	4.750E+00	5.000E+00	V	
V_{OL}	4.000E−01	2.000E−01	3.000E−01	V	
V_{IN+}	1.920E+00	1.000E−01		V	
V_{IN-}	1.920E+00	1.000E−01		V	
V_{RIO}	5.000E−03	0.000E+00	1.000E−03	V	
R	6.533E−03	5.354E−03	5.914E−03	Ω	
Fro	1.200E+06	8.000E+05	1.000E+06	Hz	0.2
V_H	3.368E−02	2.691E−02	2.880E−02	V	
Xc	7.4025E+03	2.9767E+03	4.7000E+03	Ω	

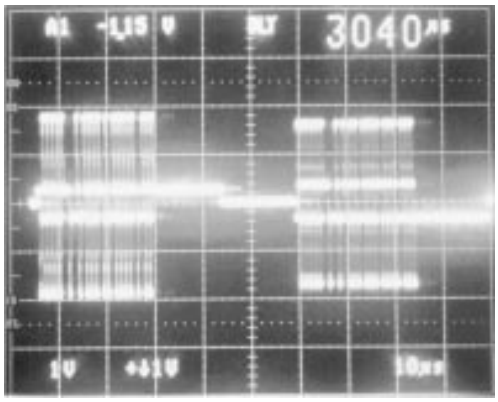
The BCP has a number of advanced features that give designers much flexibility to adapt products to a wide range of IBM environments. Besides the basic multi-protocol capability of the BCP, the designer may select the inbound and outbound serial data polarity, the number of received and transmitted line quiesces, and in 5250 modes, a programmable extension of the TX-ACT signal after transmission.

The polarity selection on the serial data stream is useful in building single products that handle both 3270 and 5250 protocols. The 3270 biphase data is inverted with respect to 5250.

Selecting the number of line quiesces on the inbound serial data changes the number of line quiesce bits that the receiver requires before a line violation to form a valid start sequence. This flexibility allows the BCP to operate in extremely noisy environments, allowing more time for the transmission line to charge at the beginning of a transmission. The selection of the transmitted line quiesce pattern is not generally used in the 5250 arena, but has applications in 3270. Changing the number of line quiesces at the start of a line quiesce pattern may be used by some equipment to implement additional repeater functions, or for certain inflexible receivers to sync up.

The most important advanced feature of the BCP for 5250 applications is the programmable TX-ACT extension. This feature allows the designer to vary the length of time that the TX-ACT signal from the BCP is active after the end of a transmission. This can be used to drive one phase of the

twinax line in the low state for up to 15.5 μs. Holding the line low is useful in certain environments where ringing and reflections are a problem, such as twisted pair applications. Driving the line after transmitting assures that receivers see no transitions on the twinax line for the specified duration. The transmitter circuit shown in *Figure 5* can be used to hold either the A or B phase by using the serial inversion capability of the BCP in addition to swapping the A and B phases. Choosing which phase to hold active is up to the designer; 5250 devices use both. Some products hold the A phase, which means that another transition is added after the last half bit time including the high and low states, with the low state helf for the duration, see *Figure 6*. Alternatively, some products hold the B phase. Holding the B phase does not require an extra transition and hence is inherently quieter.



TL/F/9635–6

FIGURE 6. Line Hold Options

The signal was viewed in the same manner as *Figures 3* and *4*. The lefthand portion of the signal is a transmitting device utilizing line hold on phase A. The right hand side shows the IBM style (phase B) line hold.

To set the TX-ACT hold feature, the upper five bits of the Auxilliary Transceiver Register, {ATR [3–7]}, are loaded with one of thirty-two possible values. The values loaded select a TX-ACT hold time between 0 μs and 15.5 μs in 500 ns increments.

SOFTWARE INTERFACE

The BCP was designed to simplify designing IBM communications interfaces by providing the specific hardware necessary in a highly integrated fashion. The power and flexibility of the BCP, though, is most evident in the software that is written for it. Software design for the BCP deserves careful attention.

When designing a software architecture for 5250 terminal emulation, for example, one concern the designer faces is how to assure timely responses to the controller's commands. The BCP offers two general schemes for handling the real time response requirements of the 5250 data stream: interrupt driven transceiver interface mode, and polled transceiver interface mode. Both modes have strengths that make them desirable. The excellent interrupt response and latency times of the BCP make interrupts very useful in most 5250 applications.

Although factors such as data and instruction memory wait states and remote processors waiting BCP data memory accesses can degrade interrupt response times, the minimum latency is 2.5 T-states. The BCP samples all interrupt sources by the falling edge of the CPU clock; the last falling edge prior to the start of the next instruction determines whether an interrupt will be processed. When an interrupt is recognized, the next instruction in the present stream is not executed, but its address is pushed on the address stack. A two T-state call to the vector generated by the interrupt type and the contents of {IBR} is executed and [GIE] (Global Interrupt Enable) is cleared. If the clock edge is missed by the interrupt request or if the current instruction is longer than 2 bytes, the interrupt latency is extended.

Running in an interrupt driven environment can be complex when multiple sessions are maintained by the same piece of code. The software has the added overhead of determining the appropriate thread or session and handling the interrupt accordingly. For a multi-session 5250 product, the transceiver interrupt service routines must determine which session is currently selected through protocol inferences and internal semaphores to keep the threads separate and intact.

In a polled environment, the biggest difficulty in designing software is maintaining appropriate polling intervals. Polling too often wastes CPU bandwidth, not polling frequently enough loses data and jeopardizes communication integrity. Standard practice in servicing polled devices is to count CPU clock cycles in the program flow to keep track of when to poll. A program change can result in lengthy recalculations of polling intervals and requalifications of program functionality. Using the programmable timer on board the BCP to set the polling interval alleviates the need to count instructions when code is changed or added. In both polled or interrupt environments, the latency effects of remote processors waiting memory accesses must be limited to a known length of time and figured into both polling intervals and worst case interrupt latency calculations. Using the programmable timer on the BCP makes both writing and maintaining polled software easier.

SOFTWARE ARCHITECTURE FOR 5250 EMULATION

The 5250 data rate is much lower than that of the 3270 data stream, hence it is possible for the BCP to emulate all seven 5250 sessions with a CPU frequency of 8 MHz. Choosing a 16 MHz crystal allows the transceiver to share the CPU clock at OCLK/2, eliminating an extra oscillator circuit. The 8 MHz rate yields a 125 ns T-state, or 250 ns for most instructions. Interrupt latency is typically one instruction (assuming no wait states or remote accesses) which is suitable for 5250 operation. If more speed is desired, the CPU could be switched to 16 MHz operation.

A MULTI-MODE TRANSCEIVER

The BCP provides two 5250 protocol modes, promiscuous and non-promiscuous. These two modes afford the designer a real option only when the end product will attach to one 5250 address at a time. The non-promiscuous mode is configured with an address in the {ATR} register and only re-

ceives messages whose first frame address matches that address, or an error occurs in the first frame of the message. Filtering out unwanted transmissions to other addresses leaves more CPU time for other non-protocol related tasks, but limits the device to one address at a time. The promiscuous mode allows messages to any and all addresses to be received. Resetting the transceiver during a message destined to another device forces the transceiver to begin looking for a start sequence again, effectively discarding the entire unwanted message. Because of its flexibility, the promiscuous mode is used in this illustration.

REAL TIME CONSIDERATIONS

Choosing a scheme for servicing the transceiver is basic to the design of any emulation device. The BCP provides both polled and interrupt driven modes to handle the real time demands of the chosen protocol. In this example, the interrupt driven approach is used. This implies the extra overhead of setting up interrupt vectors and initializing the interrupt masks appropriately. This approach eliminates the need to figure polling intervals within the context of other CPU tasks.

5250 CONFIGURATION

Configuring a complex device like the BCP can be difficult until a level of familiarity with the device is reached. To help the 5250 product designer through an initial configuration, a register by register description follows, along with the reasons for each configuration choice. Certainly, most applications will use different configurations than the one shown here. The purpose is to illustrate one possible setup for a 5250 emulation device.

There are two major divisions in the BCP's configuration registers: CPU specific and transceiver specific ones.

CPU SPECIFIC CONFIGURATION REGISTERS:

{**DCR**}**—Device Control Register—**This register controls the clocks and wait states for instruction and data memory. Using a value of H#A0 sets the CPU clock to the OCLK/2 rate, the transceiver to OCLK/2, and no wait states for either memory bank. As described above, the choice of a 16 MHz crystal and configuring this way allows 8 MHz operation now, with a simple software change for straight 16 MHz operation in the future.

{**ACR**}**—Auxiliary Control Register—**Loading this register with H#20 sets the timer clock source to CPU-CLK/2, sets [BIC], the Bidirectional Interrupt Control to configure BIRQ as an input, allows remote accesses with [LOR] cleared, and disables all maskable interrupts through [GIE] low. When interrupts are unmasked in {ICR}, [GIE] must be set high to allow interrupts to operate. [GIE] can be set and cleared by writing to it, or through a number of instructions including RET and EXX.

{**IBR**}**—Interrupt Base Register**—This register must be set to the appropriate base of the interrupt vector table located in data RAM. The DP8344 development card and monitor software expect [IBR] to be at H#1F, making the table begin at H#1F00. The monitor software can be used without the interrupt table at H#1F00, but doing so is simplest for this illustration.

{**ICR**}**—Interrupt Control Register**—This register contains both CPU and transceiver specific controls. From the CPU point of view, the interrupt masks are located here. In this illustration, the system requires receiver, transmitter, BIRQ, and timer interrupts, so that in operation those interrupt bits should be unmasked. For initialization purposes, though, interrupts should be masked until their vectors are installed and the interrupt task is ready to be started. Therefore, loading [ICR] with H#7F is prudent. This also sets the receiver interrupt source, but that will be discussed in the next section.

TRANSCEIVER CONFIGURATION REGISTERS:

{**TMR**}**—Transceiver Mode Register**—This register controls the protocol selection, transceiver reset, loopback, and bit stream inversion. Loading this register with H#0D sets up the receiver in 5250 promiscuous mode, inverts serial data out, does not invert incoming serial data, does not allow the transmitter and receiver to be active at the same time, disables loopback, and does not reset the transceiver. Choosing to set [RIN] low assumes that serial data will be presented to the chip in NRZI form. Not allowing the receiver and transmitter to operate concurrently is not an issue in 5250 emulation, since there is no defined repeater function in the protocol as in 3270 (3299). Bits [B5, 6], [RPEN] and [LOOP] are primarily useful in self testing, where [LOOP] routes the transmitted data stream into the receiver and simultaneous operation is desirable. Please note that for loopback operation, [RIN] must equal [TIN]. [TRES] is used regularly in operation, but should be left off when not specifically needed.

{**TCR**}**—Transceiver Command Register**—This register has both configuration and operation orientated bits, including the transmitted address and parity bits. For this configuration, the register should be set to H#00 and the specific address needed summed into the three LSBs, as appropriate. The [SEC] or Select Error Codes bit is used to enable the {ECR} register through the {RTR} transceiver FIFO port, and should be asserted only when an error has been detected and needs to be read. [SLR], or Select Line Receiver is set low to enable the TTL-IN pin as the serial data in source. The BCP's on chip comparator is best suited to transformer coupled environments, and National's LM361 high speed differential comparator works very well for the twinax line interface. [ATA], or Advance Transmitter Active is normally used in the 3270 modes to change the form of the first line quiesce bit for transmission. Some twinax products use a long first line quiesce bit, although it is not necessary. The lower four bits in {TCR} are used to form the frame transmitted when data is written into {RTR}, the transceiver FIFO port. Writing into {RTR} starts the transmitter and/or loads the transmit FIFO. The least significant three bits in {TCR} form the address field in that transmitted frame, and B3, [OWP] controls the type of parity that is calculated and sent with that frame. [OWP] set to zero calculates even parity over the eight data bits, address and sync bit as defined in the IBM 5250 PAI.

{**ATR**}**—Auxilliary Transceiver Register**—Since this application is configured for promiscuous mode, the {ATR} register serves only to set the line hold function time. In non-promiscuous mode, the three least significant bits of this register are the selected address. Setting this register to H#50 allows a 5 μs hold time and clears the address field to 0, since promiscuous mode is used.

{**FBR**}**—Fill Bit Register**—This register controls the number of biphase zeros inserted between concatenated frames when transmitting. This register should be set upon reception of the SET MODE instruction from the host. {FBR} contains the one's complement of the number of inter-frame fill bits so that H#FF sends no extra fill bits.

{**ICR**}**—Interrupt Control Register**—As discussed in the CPU configuration section, this register sets [RIS] or Receiver Interrupt Select as well as the interrupt mask. Setting the register to H#7F selects [DA + ERR], Data Available or transceiver ERRor, as the interrupt source. This interrupt is asserted when either a valid frame has been clocked into the receive FIFO or an error has occurred. Other interrupt options are available including: [RA], Receiver Active; and [RFF + ERR], Receive FIFO Full or transceiver ERRor. For 5250 protocols the [DA + ERR] is most efficient. The [RFF + ERR] interrupt will not assert until the FIFO is full . . . regardless of whether the incoming message is single or multi-frame. [RA] provides plenty of notice that a frame is incoming, but due to the speed of the BCP, this advanced warning is not generally needed. [DA + ERR] provides a notification just after the parity bit has been decoded from the incoming frame which is almost 3 μs prior to the end of the frame. With the CPU running at 8 MHz, that allows typically nine instructions ([(4 * 3) − 3)]) for interrupt latency, trap and bank switch after interrupting.

MULTI-SESSION POWER

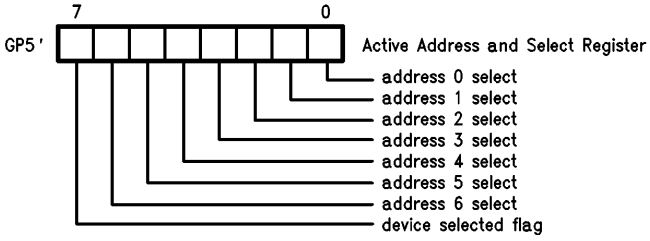
Handling multiple sessions in software is not trivial, and making the receiver service routines interrupt driven complicates the task further. The BCP is so fast, that at 8 MHz handling a multi-frame message by interrupting on the first frame and polling for succeeding frames is very inefficient. To maximize bandwidth for non-protocol related tasks, the CPU should handle each frame separately on interrupt and exit. To do this, a number of global state variables must be maintained. Since the alternate B register bank is primarily used for transceiver functions anyway, dedicating the other registers in that bank permanently as state variables is acceptable in most cases; doing so speeds and simplifies access to them. Defining the following registers as:

enables the software to keep track of the various states the protocol must handle.

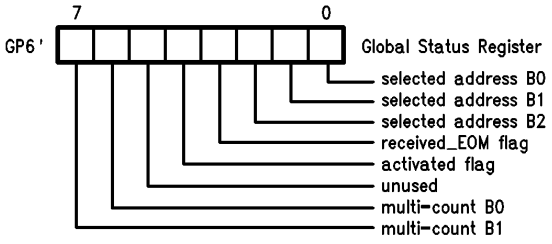
The active address bits in GP5′ allow individual addresses to be active, or any combination of addresses. When interrupted by a message to a non-selected address, [TRES] is toggled to reset the receiver until the beginning of the next message is detected. [B7] is used to determine if any particular address is "selected" and in the process of receiving data. The selected flag is set and cleared according to specific protocol rules set up in the IBM PAI.

Register GP6′ contains the selected address storage [B0–2], where the address of the device expecting at least one other frame is stored when exiting the interrupt service routine, so that upon interruption caused by the reception of that frame, the address is still available. The received__EOM flag, [B3] is set when a message is decoded that contains B#111 or EOM delimiter. It is stored in this global status register to allow the protocol to determine the end of a transmission. In most multi-byte transmissions, the number of data frames expected is dictated by the protocol. However, ACTIVATE WRITE commands to printers can have any number of data frames associated with them up to 256. In this situation, the activated flag, [B4] is set to signal a variable length stream. Certain host devices also concatenate commands within messages, obscuring the determination of end of message. This scheme allows the software to keep track during such scenarios. The multi-count bits, [B6–7] are used in addition to the EOM delimiter to determine the end of a transmission. The number of additional frames expected in a given multi-byte command is written into these bits (note that a maximum of three bytes can be planned for in this way). When the count is terminated and no EOM delimiter is present, the algorithm then assumes a multi-command message is in progress. [B5] is unused.

Register GP7′ is used to store the received data or error code for passage to other routines. The data can be passed on the stack, but dedicating a register to this function simplifies transactions in this case. Keeping track of received data is of utmost importance to communications devices.



TL/F/9635–7



TL/F/9635–8

GP7′—Bits [0–7] Received Data or Error Register

RECEIVER INTERRUPT

The receiver interrupt algorithm handles any or all seven addresses possible on the twinax line. The same code is used for each address by utilizing a page oriented memory scheme. Session specific variables are stored in memory pages of 256 bytes each. All session control pages, or SCPs are on 256-byte even boundaries. By setting the high order byte of a BCP index register to point to a particular page or SCP, the low order byte then references an offset within that page. Setting up data memory in such a way that the first SCP begins at an address of B#xxxx x000 0000 0000 further enhances the usefulness of this construct. In this scheme, the high byte of the SCP base pointer can be used to set the particular SCP merely by summing the received or selected address into the lower three bits of the base register.

NORMAL OPERATION

In normal operation, the configuration described thus far is used in the following manner: After initializing the registers, data structures are initialized, and interrupt routines should be activated. This application utilizes the receiver, transmitter, timer, and bi-directional interrupts. Since {IBR} is set to H#1F, the interrupt table is located at H#1F00. A LJMP to the receiver interrupt routine should be installed at location H#1F104, the transmitter interrupt vector at H#1F08, the BIRQ interrupt vector at H#1F10, and the Timer interrupt vector at H#1F14. Un-masking the receiver interrupt and BIRQ at start up allows the device to come on-line.

When interrupt by the receiver, the receiver interrupt service routine first checks the [ERR] flag in {TSR [B5]}. If no errors have been flagged, the received__EOM flag is either set or cleared. This is accomplished by comparing {TSR [B0–2]} with the B#111 EOM delimiter. A test of the selected flag, {GP5′ [B7]} determines if any of the active addresses are selected. Assuming that the system is just coming on line, none of the devices would be selected. If the frame is addressed to an active device, the SCP for that device is set, and the command is parsed. Parsing the command sets the appropriate state flags, so that upon exiting, the interrupt routine will be prepared for the next frame. Once parsed, the command can be further decoded and handled. If the command is queue-able, the command is pushed on the internal command queue, and the receiver interrupt routine exits. If the command requires an immediate response, then the response is formulated, the timer interrupt is setup, and the routine is exited.

The timer interrupt is used in responding to the host by waiting an appropriate time to invoke the transmit routine. The typical response delay is 45 $\pm$15 μs after the last valid fill bit received in the command frame. Some printers and terminals are allowed a full 60 $\pm$20 μs to respond. In either case, simply looping is very inefficient. The immediate response routine simply sets the timer for the appropriate delay and unmasks the timer.

In the transmit routine, the data to be sent is referenced by a pointer and an associated count. The routine loads the appropriate address in the three LSBs of {TCR}, and writes the data to be sent into {RTR}. This starts the transmitter. If the data count is greater than the transmit FIFO depth (three bytes), the Transmit FIFO Empty interrupt [TFE] is setup. This vectors to code that refills the FIFO and re-enables that interrupt again, if needed. This operation must be carried out before the transmitter is finished the last frame in the FIFO or the message will end prematurely.

The last frame transmitted must contain the EOM delimiter. It can be loaded into {TCR} and data into {RTR} while the transmitter is running without affecting the current frame. In other words, the transmit FIFO is 12 bits wide, including address and parity with data; the address field is clocked along with the data field. In this way, multi-byte response may be made in efficient manner.

ERROR HANDLING

In 5250 environments, the time immediately after the end of message is most susceptible to transmission errors. The BCP's receiver does not detect an error after the end of a message unless transitions on the line continue for a complete frame time or resemble a valid sync bit of a multiframe transmission. If the twinax line is still active at the end of what could be an error frame, the receiver posts the LMBT error. For example, if noise on the twinax line continues for up to 11 μs after the three required fill bits, the receiver will reset without flagging an error. If noise resembles a start bit, the receiver now expects a new frame and will post an error if a loss of synchronization occurs. If the noisy environment is such that transitions on the receiver's input continue for 11 μs, or the receiver really has lost sync on a real frame, the error is posted.

Basically, the receiver samples [LA] in addition to the loss of synchronization indication to determine when to reset or to post an error. After a loss of synchronization in the fill bit portion of a frame, if the [LA] flag's time-out of 2 μs is reached prior to the end of what could be the next frame, the receiver will reset. If the transitions prevent [LA] from timing out for an entire 11 μs frame time, a LMBT error is posted. This method for resetting the receiver is superior in that not only are the spurious loss of mid bit errors eliminated, the receiver performs better in noisy environments than other designs.

SUMMARY

The IBM 5250 twinax environment is less understood and in some ways more complex than the 3270 environment to many developers. This application note has attempted to explain some basics about twinax as a transmission medium, the hardware necessary to interface the DP8344 to that medium, and some of the features of the BCP that make that task easier. Schematics are included in this document to illustrate possible designs. Details of the twinax waveforms were discussed and figures included to illustrate some of the more relevant features. Also, some different software approaches to handling the transceiver interface were discussed.

REFERENCES

5250 Information Display to System/36 and System/38 System Units Product Attachment Information, IBM, November 1986.

Transmission Line Characteristics, Bill Fowler, National Semiconductor Application Note AN-108.

Basic Electromagnetic Theory, D.T. Paris, F.K. Hurd McGraw-Hill Inc., 1969.

APPENDIX A: EXAMPLE CODE

The following code was assembled with the HILEVEL assembler. Table II shows the correlation between HILEVEL mnemonics and the mnemonics used in National data sheets for the DP8344V.

TABLE II

HILEVEL	National Semiconductor
MOVE Rs,Rd	MOVE Rs,Rd
LD Ptr,Rd{,Mde}	MOVE [mIr],Rd
ST Rs,Ptr{,Mde}	MOVE Rs,[mIr]
LDAX Ptr,Rd	MOVE [Ir + A],Rd
STAX Rs,Ptr	MOVE Rs,[Ir + A]
LDNZ n,Rd	MOVE [IZ + n],rd
STNZ Rs,n	MOVE rs,[IZ + n]
LDI n,Rd	MOVE n,rd
STI n,Ptr	MOVE n,[Ir]
ADD Rs,Rd	ADDA Rs,Rd
ADDRI Rs,Ptr{,Mde}	ADDA Rs,[mIr]
ADDI n,Rsd	ADD n,rsd
ADC Rs,Rd	ADCA Rs,Rd
ADCRI Rs,Ptr{,Mde}	ADCA Rs,[mIr]
SUBT Rs,Rd	SUBA Rs,Rd
SUBRI Rs,Ptr{,Mde}	SUBA Rs,[mIr]
SUBI n,Rsd	SUB n,rsd
SBC Rs,Rd	SBCA Rs,Rd
SBCRI Rs,Ptr{,Mde}	SBCA Rs,[mIr]
AND Rs,Rd	ANDA Rs,Rd
ANDRI Rs,Ptr{,Mde}	ANDA Rs,[mIr]
ANDI n,Rsd	AND n,rsd
OR Rs,Rd	ORA Rs,Rd
ORRI Rs,Ptr{,Mde}	ORA Rs,[mIr]
ORI n,Rsd	OR n,rsd
XOR Rs,Rd	XORA Rs,Rd
XORRI Rs,Ptr{,Mde}	XORA Rs,[mIr]
XORI n,Rsd	XOR n,rsd
CMP Rs,n	CMP rs,n
CPL Rsd	CPL Rsd
BIT Rs,n	BIT rs,n
SRL Rsd,n	SHR Rsd,b
SLA Rsd,n	SHL Rsd,b
ROT Rsd,n	ROT Rsd,b

TL/F/9635–9

```
JMP   n                 JMP   n
LJMP  n                 LJMP  nn
JMPR  Rs                JMP   Rs
JMPI  Ptr               LJMP  [Ir]
JRMK  Rs,n,m            JRMK  Rs,b,m
JMPB  Rs,s,p,n          LJMP  Rs,p,s,nn
JMPF  s,f,n             JMP   f,s,n
                        Jcc   n    - opt. syntax for JMP f-

CALL  n                 CALL  n
LCALL n                 LCALL       nn
CALLB Rs,s,p,n          LCALL       Rs,p,s,nn

RET   {g{,rf}}          RET   {g {,rf}}
RETF  s,f{,g{,rf}}      RETF  f,s,{,{g} {,rf}}
                        Rcc   {g {,rf}}    - opt. syntax -

EXX   a,b{,g}           EXX   ba,bb,{,g}

TRAP  n{,g}             TRAP  n {,gU}
```

Table 2.

```
Addr      Line

            1           .REL
            2   TAB     8
            3   WIDTH   132
            4   LIST    S,F
            5   TITLE   RXINT
            6   ;-------------------------------------------------------------------
            7   ;       RXINT - 9/21/87
            8   ;
            9   ;       pseudo code
           10   ;
           11   ;bool          selected;                       /* station is selected
           12   ;byte          seladdr;                        /* address of selected station
           13   ;byte          multicount;                     /* number of frames in this multi
           14   ;bool          activated;                      /* command has been activated
           15   ;
           16   ;rxint()
           17   ;byte          data;                           /* data storage
           18   ;bool          rx_eom;                         /* received EOM
           19   ;bool          lta;                            /* line turn around flag
           20   ;{
           21   ;       if (error) {
           22   ;               if (logerror()== true) return;      /* receiver errors
           23   ;               }
           24   ;       else {
           25   ;               if (TSR == EOM) rx_eom = true;      /* set received EOM flag
           26   ;               else rx_eom = false;
           27   ;
           28   ;               if (!selected) {
```

TL/F/9635–10

```
29    ;                       if (active) {
30    ;                               if (!rx_eom) {
31    ;                                       seladdr = (T5R * EOM);
32    ;                                       IZ = (SCPBASE + seladdr); /* set SCP to appropriate session */
33    ;                                       data = rtr;
34    ;                               else {
35    ;                                       proto_error(); /* should not get here
36    ;                                       reset_xcvr();
37    ;                                       return();
38    ;                                       }
39    ;                               }
40    ;                       else {
41    ;                               reset_xcvr();           /* not of interest
42    ;                               return();
43    ;                               }
44    ;                       if (multiframe) {               /* activate write, etc...
45    ;                               multicount = parse(data); /* set number of frames */
46    ;                               selected = true;        /* only way to select   */
47    ;                               queue(data);
48    ;                               }
49    ;                       else {                          /* not multi
50    ;                               if ((var = single_decode(data)) == queable)
51    ;                                       queue(data);
52    ;                               else if (var == immed) immediate(data);
53    ;                       }
54    ;               else {                                  /* selected        */
55    ;                       IZ = (SCPBASE + seladdr);

Addr      Line RXINT

56    ;                       data = rtr
57    ;               if (activated) {                        /* in the middle of transmission
58    ;                    act_data(data);
59    ;                    if (rx_eom) {                      /* end of message
60    ;                      selected = false;
61    ;                      activated = false;
62    ;                      }
63    ;                    return();
64    ;                    }
65    ;                      if (multicount > 0) {
66    ;                               queue(data);
67    ;                               if (multicount-= 0) {
68    ;                                       if (rx_eom) selected = false;
69    ;                                       }
70    ;                               }
71    ;                      else {
72    ;                               if (multiframe) {
73    ;                                       multicount = parse(data);
74    ;                                       queue(data);
75    ;                                       }
76    ;                               else {
77    ;                                       if ((var = single_decode(data)) == queable)
78    ;                                               queue(data);
```

TL/F/9635–11

```
 79   ;                                        else if (var == immed) immediate(data);
 80   ;                                        if (rx_eom) selected = false;
 81   ;                                        }
 82   ;                                  }
 83   ;                            }
 84   ;                      }
 85   ;            }
 86   ;            return();
 87   ;}
 88   ;logerror()
 89   ;{
 90   ; bool result;
 91   ;     switch (error_type) {
 92   ;     case RDIS:
 93   ;          result = err_rdis();                    /* receiver diabled while active
 94   ;          break;
 95   ;     case LMBT:
 96   ;          result = err_lmbt();                    /* loss of midbit error
 97   ;          break;
 98   ;     case PARR:
 99   ;          result = err_parr();                    /* parity error
100   ;          break;
101   ;     case OVF:
102   ;          result = err_ovf();                     /* receiver FIFO overrun
103   ;          break;
104   ;     default:
105   ;          result = err_unknown();                 /* strange error handler
106   ;          break;
107   ;     }
108   ;     return(result);
109   ;   }
110   ;

Addr      Line RXINT

111   ;err_lmbt()
112   ;{
113   ;     if (!DA && !selected && !delay(LA)) return(false); /* delay of 6 usec
114   ;     else {
115   ;          log();                                  /* bump error counters
116   ;          return(true);                           /* admit defeat
117   ;          }
118   ; }
119   ;-------------------------------------------------------------------------
120   ;       name:          RXINT
121   ;       description:   receiver interrupt handler
122   ;
123   ;              received datum is sent to other routines thru gp7'
124   ;              SCP is set appropriately in IZ
125   ;              GP5P - active addresses:bits 0-6
126   ;                     selected flag:   bit 7
127   ;              GP6P - multicount:      bit 7-6
128   ;                     unused:          bit 5
```

TL/F/9635–12

```
                129   ;                       activated:      bit 4
                130   ;                       rx_eom flag:    bit 3
                131   ;                       seladdr:        bits 2-0
                132   ;               GP7P - received data
                133   ;
                134   ;       entry:          DA interrupt, GP5', GP6'
                135   ;       exit:           ACC',GP7' ARE DESTROYED
                136   ;       history:        tjq  9/16/87   create
                137   ;-------------------------------------------------------------------
                138           PUBLIC  RCVRINT
                139
                140           EXTRN   PARSE,QUEUE,IMMEDECODE,RESXCVR
                141           EXTRN   MIDERRL,MIDERRH,OVFERRL,OVFERRH,PARERRL,PARERRH
                142           EXTRN   RXERRL,RXERRH,RSPCTL,RSPCTH,BASESCP,IESERRL,IESERRH
                143
                144
                145   SELERR: EQU     B#01000000      ; select the error register
                146   RXEOM:  EQU     B#00001000      ; rxeom flag
                147   EOM:    EQU     B#00000111      ; EOM delimeter
                148   MULTI:  EQU     B#11000000      ; multicount
                149   SELECT: EQU     B#10000000      ; selected flag
                150   LTA:    EQU     B#101           ;     "
                151   CFLAG:  EQU     B#00000010      ; CARRY FLAG
                152
00000           153   RCVRINT:
                154           EXX     MA,AB,DI        ; SET APPROPRIATE BANK
00000 AEE8      154
00001 D500      155           JMPF    NS,RERR,NOERROR
00002 CC00      156           CALL    RXERROR         ; ERROR IN FRAME
00003 D900      157           JMPF    S,C,EXIT        ; ABORT
00004 D900      158   NOERROR:
00004 B07B      159           LDI     EOM,ACC         ; LOAD MASK
                160           AND     TSR,GP7         ; FORM ADDRESS
00005 F165      160
                161           CMP     GP7,EOM         ; TEST
00006 307B      161
00007 D000      162           JMPF    NS,Z,C1RXINT    ; IF NOT EQUAL, JUMP

 Addr       Line RXINT

0000B 508A      163           ORI     RXEOM,GP6       ; ELSE SET EOM FLAG
00009 CB00      164           JMP     C2RXINT         ;
0000A CB00      165   C1RXINT:
0000A 4F7A      166           ANDI    RXEOM*,GP6      ; CLEAR IT
                167   ;
                168   ; DECIDE IF WE'RE ALREADY SELECTED
                169   ;
0000B           170   C2RXINT:
                171           JMPB    GP5,S,B7,DEVSELECT ; IF ALREADY SELECTED
0000B BDE9      171
0000C 0000      171
                172   ;
                173   ; NOT SELECTED...DECIDE IF ADDRESS IS ACTIVE, IE; VALID FOR US
```

TL/F/9635–13

```
                174   ;
0000D           175   DEVTABLE:                     ; ELSE, SEE IF ACTIVE
                176           JRMK    TSR,ROT6,MSK3   ; JUMP BASED ON THE ADDRESS FIELD*4
0000D B3C5      176
                177           JMPB    GP5,NS,B0,RSTRX ; ADDR 0 - IF NOT ACTIVE, RESET RX
0000E 8C09      177
0000F 0000      177
                178           LJMP    LOADSCP         ; ACTIVE DEVICE, SET scp
00010 CE00      178
00011 0000      178
                179           JMPB    GP5,NS,B1,RSTRX ; ADDR 1 - IF NOT ACTIVE, RESET RX
00012 8C29      179
00013 0000      179
                180           LJMP    LOADSCP         ; ACTIVE DEVICE, SET scp
00014 CE00      180
00015 0000      180
                181           JMPB    GP5,NS,B2,RSTRX ; ADDR 2 - IF NOT ACTIVE, RESET RX
00016 8C49      181
00017 0000      181
                182           LJMP    LOADSCP         ; ACTIVE DEVICE,
00018 CE00      182
00019 0000      182
                183           JMPB    GP5,NS,B3,RSTRX ; ADDR 3 - IF NOT ACTIVE,
0001A 8C69      183
0001B 0000      183
                184           LJMP    LOADSCP         ; ACTIVE DEVICE,
0001C CE00      184
0001D 0000      184
                185           JMPB    GP5,NS,B4,RSTRX ; ADDR 4 - IF NOT ACTIVE,
0001E 8C89      185
0001F 0000      185
                186           LJMP    LOADSCP         ; ACTIVE DEVICE,
00020 CE00      186
00021 0000      186
                187           JMPB    GP5,NS,B5,RSTRX ; ADDR 5 - IF NOT ACTIVE,
00022 8CA9      187
00023 0000      187
                188           LJMP    LOADSCP         ; ACTIVE DEVICE,
00024 CE00      188
00025 0000      188
                189           JMPB    GP5,NS,B6,RSTRX ; ADDR 6 - IF NOT ACTIVE,
00026 8CC9      189

 Addr       Line RXINT

00027 0000      189
                190           LJMP    LOADSCP         ; ACTIVE DEVICE,
00028 CE00      190
00029 0000      190
                191           LCALL   RESXCVR         ; ADDR 7 - RECEIVED EOM ...WE'RE NOT INTERESTED
0002A CE80      191
0002B 0000      191
0002C CB00      192           JMP     EXIT            ; QUIT
```

TL/F/9635–14

```
                193   ;
                194   ; LOAD THE SCP POINTER, IZ
                195   ;
0002D           196   LOADSCP:
                197           XOR     ACC,ACC         ; CLEAR
0002D F908      197
                198           MOVE    ACC,ZLO         ; LOW BYTE
0002E FE48      198
0002F B008      199           LDI     BASESCP,ACC     ; SET UP UPPER BYTE OF SCP POINTER
                200           MOVE    ACC,ZHI         ;
00030 FE68      200
00031 B07B      201           LDI     EOM,ACC         ; EOM MASK
                202           AND     TSR,ACC         ; LEAVE IN ACC
00032 F105      202
                203           ADD     ZHI,ZHI         ; ADD INTO Z POINTER
00033 E273      203
                204   ;
                205   ; DECODE THE COMMAND FRAME
                206   ;
00034           207   DECODE:
                208           MOVE    RTR,GP7         ; GET RX DATA
00034 FD64      208
                209           JMPB    GP7,S,B0,MULTIFRM; IF MULTIFRAME
00035 BD0B      209
00036 0000      209
                210           LCALL   IMMEDECODE      ; ELSE, IMMEDIATE ACTION REQUIRED
00037 CE80      210
00038 0000      210
00039 CB00      211           JMP     EXIT
0003A CB00      212   MULTIFRM:
                213           LCALL   PARSE           ; SET MULTI COUNT
0003A CE80      213
0003B 0000      213
0003C 5B09      214           ORI     H'80,GP5        ; SELECTED = TRUE
0003D 4F8A      215           ANDI    EOM*,GP6        ; CLEAR SELECTED ADDRESS
0003E B07B      216           LDI     EOM,ACC         ; MASK ADDRESS
                217           AND     TSR,ACC         ; LEAVE IN ACC
0003F F105      217
                218           OR      GP6,GP6         ; SET NEW ADDRESS
00040 F54A      218
                219           LCALL   QUEUE           ; PLACE ON QUEUE
00041 CE80      219
00042 0000      219
00043 CB00      220           JMP     EXIT            ;
                221   ;
                222   ; THIS CODE IS BRANCHED TO IF THE DEVICE IS SELECTED
                223   ;       FIRST, SET SCP BASED ON SELECTED ADDRESS

 Addr       Line RXINT

                224   ;
00044           225   DEVSELECT:
                226           XOR     ACC,ACC         ; CLEAR ACC
```

TL/F/9635–15

```
00044 F908   226
             227          MOVE    ACC,ZLO        ; CLEAR LOW BYTE OF POINTER
00045 FE48   227
00046 B008   228          LDI     BASESCP,ACC    ; BASE OF SESSION CONTROL PAGE
             229          MOVE    ACC,ZHI        ; UPPER BYTE
00047 FE68   229
00048 B07B   230          LDI     EOM,ACC        ; MASK ADDRESS
             231          AND     GP6,ACC        ; LEAVE IN ACC
00049 F10A   231
             232          ADD     ZHI,ZHI        ; FORM SCP POINTER
0004A E273   232
             233     ;
             234     ; NOW DECIDE ABOUT MULTIFRAME POSSIBILITIES
             235     ;
             236          MOVE    RTR,GP7        ; GET DATA
0004B FD64   236
0004C BC08   237          LDI     MULTI,ACC      ; MULTI MASK
             238          AND     GP6,ACC        ; COUNT IN UPPER NIBBLE
0004D F10A   238
             239          SRL     ACC,ROT6       ; POSITION IN LOWER NIBBLE
0004E C8C8   239
0004F D800   240          JMPF    S,Z,NEWCOMM    ; NOT in A MULTIBYTE
             241          LCALL   QUEUE          ; MULTI, SO PUSH ON QUEUE
00050 CEB0   241
00051 0000   241
00052 2018   242          SUBI    H#01,ACC       ; DECREMENT MULTICOUNT
00053 D800   243          JMPF    S,Z,TERMULTI   ; IF ZERO, MULTI HAS TERMINATED
             244     ;
             245     ; MULTI STILL IN PROGRESS
             246     ;
00054 43FA   247          ANDI    MULTI*,GP6     ; CLEAR OUT OLD COUNT
             248          SLA     ACC,ROT6       ; REPOSITION COUNT
00055 C948   248
             249          OR      GP6,GP6        ; SUM INTO STATUS
00056 F54A   249
00057 CB00   250          JMP     EXIT
             251     ;
             252     ; MULTICOUNT HAS REACHED ZERO, SO TERMINATE
             253     ;
00058        254     TERMULTI:
00058 43FA   255          ANDI    MULTI*,GP6     ; CLEAR OLD COUNT TO ZERO
             256          JMPB    GP6,NS,B3,C1TERM; IF NOT EOM,
00059 8C6A   256
0005A 0000   256
0005B 47F9   257          ANDI    SELECT*,GP5    ; ELSE, SELECT = FALSE
0005C CB00   258          JMP     RSTRX          ; RESET THE TRANSCEIVER
0005D CB00   259     C1TERM:
0005D CB00   260          JMP     EXIT
             261     ;
             262     ; NEW COMMAND; MULTI OR SINGLE
             263     ;
0005E        264     NEWCOMM:
```

TL/F/9635–16

```
Addr      Line RXINT

          265          JMPB    GP7,NS,B0,SINGLE; IF NEW COMMAND IS NOT MULTI,
0005E BC0B 265
0005F 0000 265
          266          LCALL   PARSE           ; IS MULTI, SET COUNT
00060 CEBO 266
00061 0000 266
          267          LCALL   QUEUE           ; PUSH ON QUEUE
00062 CEB0 267
00063 0000 267
00064 CB00 268          JMP     EXIT            ; QUIT, TIL NEXT FRAME
          269    ;
          270    ; NEW COMMAND IS SINGLE AND/OR NEEDS IMMEDIATE RESPONSE
          271    ;
00065     272    SINGLE:
          273          LCALL   IMMEDECODE      ; SINGLE...GO DO IT
00065 CEB0 273
00066 0000 273
          274          JMPB    GP6,NS,B3,EXIT  ; IF NOT EOM...
00067 BC6A 274
00068 0000 274
00069 47F9 275          ANDI    SELECT*,GP5     ; CLEAR SELECTED BIT
0006A 47F9 276    RSTRX:
          277          LCALL   RESXCVR         ; RESET, CLEAR DATA OUT
0006A CEB0 277
0006B 0000 277
0006C 0000 278    EXIT:
0006C AFB0 279          RET     RI,RF           ; RETURN GRACEFULLY
          280
          281    ;-------------------------------------------------------------------
          282    ;       name:           RXERROR
          283    ;       description:    receiver ERROR handler
          284    ;
          285    ;       entry:          DA + ERR interrupt, GP5', GP6'
          286    ;       exit:           ACC',GP7' ARE DESTROYED
          287    ;       history:        tjq  9/16/87   create
          288    ;-------------------------------------------------------------------
          289    ;
          290    ; RECEIVER ERROR HANDLER
          291    ;
0006D     292    RXERROR:
0006D 5406 293          ORI     SELERR,TCR      ; SET ECR BIT
          294          MOVE    RTR,GP7         ; GET ERROR TYPE
0006E FD64 294
0006F 4BF6 295          ANDI    SELERR*,TCR     ; RESET TCR
          296          JMPB    GP7,S,B1,LMBTERR; LOSS OF MIDBIT
00070 BD2B 296
00071 0000 296
          297          JMPB    GP7,S,B3,PARERR ; PARITY
00072 BD6B 297
00073 0000 297
          298          JMPB    GP7,S,B4,OVFERR ; OVERFLOW
```

TL/F/9635–17

```
00074 BDBB   298
00075 0000   298
00076 0000   299    ILLEGAL:
00076 B008   300            LDI     ILLEGAL,ACC     ; WHAT ERROR IS THIS?

 Addr       Line RXINT

00077 CB00   301            JMP     BUMPERR         ; SHOULD NOT GET HERE!!
00078 CB00   302    LMBTERR:
00078 DE00   303            JMPF    S,DA,CLEARC     ; if DA, THEN NO ERROR
             304            JMPB    GP5,S,B7,LOGIT  ; IF SELECTED, POST
00079 BDE9   304
0007A 0000   304
0007B CC00   305            CALL    SDLY            ; DELAY FOR 6 USEC
             306            JMPB    NCF,NS,B5,CLEARC; IF NOT ACTIVE - DISCARD, ELSE POST
0007C BCA1   306
0007D 0000   306
0007E 0000   307    LOGIT:
0007E B008   308            LDI     MIDERRL,ACC     ; LOSS OF MIDBIT
0007F CB00   309            JMP     BUMPERR         ; INCREMENT COUNTER
00080 CB00   310    PARERR:
00080 B008   311            LDI     PARERRL,ACC     ; PARITY
00081 CB00   312            JMP     BUMPERR
00082 CB00   313    OVFERR:
00082 B008   314            LDI     OVFERRL,ACC     ; OVERFLOW...VERY BAD!
00083 B008   315    BUMPERR:
             316            ADD     ZLO,YLO         ; FORM NEW POINTER
00083 E212   316
00084 B018   317            LDI     H#01,ACC        ; INCREMENT
             318            LD      PTRY,GP6        ; FETCH OLD COUNT
00085 C0CA   318
             319            ADDRI   GP6,PTRY,POSTD  ; WRITE OUT NEW
00086 A04A   319
00087 D100   320            JMPF    NS,C,RXEXIT     ; GET OUT
             321            LD      PTRY,GP6        ; FETCH UPPER BYTE
00088 C0CA   321
             322            ADDRI   GP6,PTRY        ;
00089 A0CA   322
0008A 5020   323            ORI     CFLAG,CCR       ; SET CARRY
0008B 5020   324    RXEXIT:
0008B AFB0   325            RET                     ; DO NOT restore flags
0008C AFB0   326    CLEARC:
0008C 4FD0   327            ANDI    CFLAG#,CCR      ; CLEAR CARRY
0008D CB00   328            JMP     RXEXIT
             329    ;---------------------------------------------------------------------
             330    ;       name:           SDLY
             331    ;       description:    delay routine, MULTIPLES OF 4.8usec,
             332    ;                       1.4 usec OVERHEAD, MAX OF 410usec
             333    ;       entry:          delay count on stack
             334    ;       exit:           acc destroyed
             335    ;       WARNING:        DONT CALL THIS WITH COUNT = 0!
             336    ;       history:        tjq  9/16/87   create
             337    ;---------------------------------------------------------------------
```

TL/F/9635–18

```
            338
0008E       339    SDLY:
            340           EXX     MA,MB,NA1     ; BANK, ALLOW INTERRUPTS
0008E AE80  340
            341           MOVE    DS,ACC        ; GET COUNT
0008F FD1F  341
            342           MOVE    GP7,DS        ; PUSH GP7 REGISTERS USED
00090 FFEB  342
            343           MOVE    GP6,DS
 Addr      Line RXINT

00091 FFEA  343
            344           MOVE    ACC,GP7       ; USE GP7 FOR COUNT ALSO
00092 FD68  344
00093 FD68  345    SDLYLP1:
00093 B03A  346           LDI     H#03,GP6      ; LOAD FOR 4.8usec COUNTS
00094 B03A  347    SDLYLP2:
00094 201A  348           SUBI    H#01,GP6      ; DECREMENT COUNT
00095 D000  349           JMPF    NS,Z,SDLYLP2  ; CONTINUE UNTIL EXHAUSTED
00096 201B  350           SUBI    H#01,GP7      ; DECREMENT OUTER COUNT
00097 D000  351           JMPF    NS,Z,SDLYLP1  ; CONTINUE IF NOT ZERO
            352           MOVE    DS,GP6        ; POP REG
00098 FD5F  352
            353           MOVE    DS,GP7        ;
00099 FD7F  353
0009A AFB0  354           RET     RI,RF         ; RETURN, RESTORE FLAGS
            355
            356
            357           END

Assembly Phase complete.
    0 error(s) detected.
```

TL/F/9635–19

Lit. # 100516

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408