NASA TECHNICAL MEMORANDUM

NASA TM-73,242

NASA TM-73,242

A SERIAL DIGITAL DATA COMMUNICATIONS DEVICE

John L. Fetter

Ames Research Center
Moffett Field, Calif. 94035

September 1977

A SERIAL DIGITAL DATA COMMUNICATIONS DEVICE

John L. Fetter

Ames Research Center

SUMMARY

This report describes a general-purpose computer peripheral device, the Serial Communications Interface, which is used to provide a full-duplex, serial, digital data transmission link between a Xerox Sigma computer and a wide variety of external equipment, including computers, terminals, and special-purpose devices. The interface has an extensive set of user-defined options to assist the user in establishing the necessary data links that originate in the research and development environment of Ames Research Center. This report describes those options and other features of the Serial Communications Interface and its performance by discussing its application to a particular problem.

INTRODUCTION

The Serial Communications Interface (SCI) is designed to provide a serial, digital data communications system for a Xerox Sigma computer. It communicates with the Sigma computer via a parallel data path and with the external world via a serial link at data rates between 100 and 50,000 bits/sec. Although the SCI is designed specifically for a Sigma computer, its features can be applied to similar interfaces designed for other computers.

The particular application discussed in this report requires two SCIs as part of a computer system. The system, required for the Terminal Area Effectiveness (TAE) project, is designed to allow the merging of a manned, real-time aircraft simulation using the facilities of Ames Research Center (ARC), Moffett Field, CA, and an Air Traffic Control (ATC) environment simulation performed at the Federal Aviation Administration's National Aviation Facilities Experimental Center (FAA NAFEC), Atlantic City, NJ. The SCIs are part of the data communications system that transfers the simulated aircraft position data from ARC to NAFEC.

DESIGN REQUIREMENTS

One design requirement for the SCI was to produce a flexible serial input/output device that could be tailored to meet the various requirements of different experiments such as the TAE project. Each application may require unique parameters, such as data rates, transmission formats, and

external interface methods. For example, the data rate requirements can vary from 110 bits/sec, for interfacing to a computer terminal, up to 50,000 bits/sec, for computer-to-computer links. The electrical and mechanical connections from the SCI to the external device may be TTL-compatible (transistor-transistor logic) or may conform to the EIA standard RS-232-C (ref. 1) for separate applications.

In projected SCI applications, the complexity of the input/output (I/O) software drivers could vary significantly. This dictated a requirement that the SCI software I/O procedures be simple and fast. In addition, the speed requirement prohibits the use of the operating system I/O software drivers as the overhead of these programs is too great. For these reasons, the Xerox Character Oriented Communications (COC) system, the standard serial I/O system for Xerox Sigma computers, cannot be used for projected applications.

The ease of checkout and maintainability was also important. Therefore, the interface was constructed in a modular manner for fault isolation. Also, each register of the SCI was designed to be writeable and readable from the Sigma central processing unit (CPU) so that effective diagnostic programs could be designed.

CONFIGURATION

Figure 1 is a photograph of the SCI. The device is fabricated with standard, commercially available integrated circuit chips (ICs), which are almost exclusively from the TTL family. The ICs are mounted in individual sockets on logic panels; inter-IC connections are wire-wrapped. The logic panels and connectors are mounted to the chassis with front and rear panels, as shown in the photograph, and the entire unit mounts in a standard 19-inch rack. Figure 2, a block diagram of an SCI, shows the functional areas and interconnecting data paths. The remainder of this section discusses the functions of these areas. A brief overview precedes the discussion.

The SCI appears as three registers to the Sigma computer: the control/status register, the transmit data register, and the receive data register. Each register is accessible using a programmed I/O method, the Direct Input/Output (DIO) system. By properly controlling these three registers (as explained below), the user can initiate the operations of the SCI.

Sigma DIO System Interface

The first functional area discussed is the Sigma DIO system interface, shown as the connection to the Sigma computer in figure 2. The Xerox DIO system (refs. 2 and 3) is utilized by a user's computer program to access I/O devices with two instructions: read direct (RD) and write direct (WD). Each instruction contains an address field that instructs the DIO system to address one of a possible 2^{16} devices and to either input (for a RD instruction) or output (for a WD instruction) a 32-bit word from, or to, the addressed device. The SCI has two DIO device addresses, used in the

following manner. The control/status register has a unique DIO address so that a WD instruction with this register as the addressed device (a WDCONT instruction) will load the control function bits and a RD instruction to this register (a RDCONT instruction) will input the current state of the control/status register. The two data registers share the other DIO device address; a WDDATA instruction will load the transmit data register and a RDDATA instruction will input the current contents of either the transmit or receive data register (selected by the state of the RSDEL bit of the control/status register).

The decision to use the Sigma DIO system was based primarily on the design requirement of making the SCI I/O software procedures simple and fast. The DIO system is easy to program and is capable of high data rates. Additional advantages include the fact that programmed I/O systems, similar to the Xerox Sigma DIO, exist on nearly all manufacturer's computers so that a SCI-type device can be implemented on additional computers. Also, DIO devices are easily interfaced into the real-time simulation software systems at ARC. A possible drawback to a programmed I/O system is that it operates through the computer's CPU, thus requiring a potentially large amount of CPU service time. However, the SCI, as a medium-speed, serial device, does not make frequent requests for CPU time and therefore does not cause overhead problems.

The Sigma DIO interface section of the SCI interconnects the signals between the Xerox DIO system and the SCI, and consequently makes the SCI directly compatible with only a Xerox Sigma series computer. The logic is designed to monitor the DIO bus address lines and detect when the SCI is being accessed by a Sigma CPU instruction. During a computer access, the function of this logic is to synchronize with the Xerox DIO system and then determine the instruction type (RD or WD). The logic is then designed to generate the control signals to the other sections of the SCI, route the data to or from the proper SCI register, and finally release the DIO bus at the completion of the instruction.

Control/Status, Transmit Data, and Receive Data Registers

The control/status register, shown in the upper center in figure 2, contains user-controlled function bits and hardware-controlled status bits. Appendix A defines each bit of the control/status register. The functions of these bits are also described throughout this report.

The transmit data register, shown in the center of figure 2, serves as a temporary storage buffer between the Sigma computer and the SCI transmitter and thus contains the character(s) to be transmitted. A character is defined as the unit of transmission between the SCI and the external unit and, for this device, can be considered synonymous with a byte (8 bits).

The receive data register is the temporary storage buffer between the SCI receiver and the Sigma computer. Thus, in a fashion similar to the

transmit data register, the receive data register contains the character(s) last received by the SCI.

Since these two data registers are 32 bits wide and a character is 8 bits, it is obvious that up to 4 characters can be stored in a data register. These characters are left-justified within the data register as shown in figure 3. The number of characters buffered in the two data registers is dependent upon the application. This quantity is defined by the user in the byte count register (BREG) field of the control/status register.

In the design requirements, it was stated that each register of the SCI must be readable and writeable. No ambiguity exists for those operations with the control/status register, since this register has a unique DIO device address. However, certain status bits (e.g., the error flags) cannot be written into by the Sigma computer since the meaning of such an action is unclear and the incremental logic necessary was judged excessive.

The read/write operations for the data registers are handled as follows: To load the transmit data register, the user executes a WDDATA instruction; then, to read back the contents of the transmit data register, the user sets the RDSEL bit of the control/status register to logic 1 and executes a RDDATA instuction. To write into the receive data register, the user sets both of the RSEL bits of the control/status register to logic 1 and executes a WDDATA instruction (this action actually initiates an internal transmission from the transmit data register to the receive data register through the transmit and receive sections). At the completion of the transmission, the two data registers contain the same data. To read the receive data register, the user ensures the RDSEL bit is at the logic 0 state and executes a RDDATA instruction.

Transmit Section

The Transmit section, shown near the center of figure 2, contains the logic that fetches the proper character from the transmit data register, performs the parallel-to-serial conversion, adds the necessary control bits (start and stop bits as described in appendix B), outputs the serial bit stream, and controls the transmit busy and done flags. An interrupt (transmit complete interrupt) is generated and sent to the Sigma computer, indicating when the transmission of the characters in the transmit data register has been completed. This interrupt, and the analogous receive complete interrupt, are hardware-vectored, external interrupts input to the Sigma computer.

The basic component of the transmit section is a single MOS/LSI (metal oxide semiconductor/large scale integration) circuit, a programmable synchronous/asynchronous transmitter (manufactured by Western Digital Corporation, part number PT1482B). This circuit accepts a character, converts it to a serial bit stream, and adds the necessary control bits for either asynchronous or synchronous transmission (described in appendix B). The circuit is programmable in that the number of bits per character, the parity type (odd, even, or none), the number of stop bits (one or two), the clock rate,

and the synchronous fill-character are selectable. The number of bits per character is selectable as 5, 6, 7, or 8 bits, and if the character length selected is less than 8 bits, the character is right-justified in the 8-bit character positions of the data registers shown in figure 3. The number of bits per character and the other parameters are set by a series of miniature toggle switches in the SCI.

Receive Section

The next functional area, the receive section, shown just below the transmit section in figure 2, contains the logic that receives a serial bit stream, converts the data into parallel characters, strips off the control bits, checks for transmission errors, loads the received character in the proper character position of the receive data register, and controls the states of the receive busy, receive done, and error flags. The receive section generates an interrupt (the receive complete interrupt) when the proper number of characters has been received and is in the receive data register.

As with the transmit section, the receive section is designed around one basic component, a programmable synchronous/asynchronous receiver (manufactured by Western Digital Corporation, part number PR1472B). This MOS/LSI circuit accepts a serial bit stream, performs the serial-to-parallel conversions, strips off the control bits, and outputs a parallel character with error flags that pertain to this character. The programmable features of this circuit are similar to the programmable controls of the transmitter circuit and are also defined by microswitches in the receive section logic.

Device and Signal Select Section

The last functional area is the device and signal select section, shown in the far right of figure 2. This logic consists of multiplexers and demultiplexers that are programmed to select between maintenance modes and various external interface methods. The bits that control these multiplexers and demultiplexers are the transmit select (TSEL), receive select (RSEL) and device select (DSEL) bits of the control/status register (bits 4 to 9).

The maintenance modes include internal loop-back, in which the serial bit stream from the transmit section is routed back to the receive section as discussed previously; external loop-back, where the serial bit stream from the selected external device is routed back to the same device; and both internal and external loop-back concurrently. In addition, the transmit and receive sections can be selectively programmed to an idle state.

The device select bits (DSELO and DSELI) specify one of four external devices that will be the destination of the serial bit stream from the transmit section and will be the source of the serial data for the receive section. These four possible device connections differ in the method in which they interface to the external world and are described in the "Operations" section of this report. These multiple methods provide each SCI with the ability to

interface to different equipment requirements. In addition, one SCI can be connected to up to four devices and the user can select the device he desires by controlling the DSEL bits.

OPERATIONS

This section describes the operations of an SCI. Specifically, the control/status register bits are defined; an initialization process is described; the external interface methods are discussed; and the programming of the device is discussed.

The SCI contains many parameters that must be defined by a user for each application. This initialization involves setting the microswitches that control the programmable transmitter and receiver circuits, selecting the data rate, selecting the DIO device addresses of the SCI, and properly cabling the SCI to the Sigma computer and to the external device. This procedure is normally done once per application.

Before each daily use of the SCI, the user must initialize the control/status register function bits. The pertinent information includes the value of the BREG field as discussed previously, the state of the data terminal ready (DTR) function bit, and the device selected (DSEL). The DTR bit is defined only for devices 1, 2, and 3 and is a general-purpose logic signal that may be used externally such as a signal to a modem to indicate that the SCI (i.e., the data terminal) is ready to begin communications. The status bit carrier detect (CD, bit 18 of the control/status register) is the accompanying "handshaking" signal controlled by the external device. For example, a modem will set CD when the connection between itself and the external device is operative, thereby notifying the Sigma computer (through the SCI) that communications can proceed.

Before discussing the device select logic further, it is recommended that the reader familiarize himself with the asynchronous and isonchronous transmission formats and the function of a modem as discussed in Appendix B.

The SCI is designed to operate through one of three program-selected methods: TTL differential, EIA RA-232-C asynchronous, and EIA RS-232-C isonchronous. The nonselected external device connections are forced to an idle state so that power-on but inactive connections can be made between the SCI and the external devices. Therefore, it is possible to configure one SCI to meet several applications and select between applications under program control.

External Interface Methods

Device 0 (DSELO = 0, DSELI = 0) connections use a TTL-compatible differential transmission line; data are transmitted in the asynchronous format. The intended application for this connection is a direct computer-to-computer

intertie operating at medium-speed rates up to 50,000 bits/sec and at distances up to 1 km. Direct cables are used between sites so that data switching is not required and high reliability is achieved.

Device 1 (DSEL = 0, DSEL1 = 1) connections use the EAI RS-232-C standard signals; data are transmitted in the asynchronous format. This method is useful for interfacing a computer to data terminals within 20 m or to devices up to 5 km away when using asynchronous modems and dedicated circuits. Data rates are less than 10,000 bits/sec.

Devices 2 and 3 (DSEL0 = 1, DSEL1 = 0 or 1) connections also use the EIA RS-232-C standard signals, but transmit data in the isonchronous format. The intended application for these connections is for computer-to-computer interties at long distances (greater than 1 km). Synchronous modems and data quality circuits are necessary in this configuration. Data rates are dependent on the modems and the quality of the circuits; therefore, the slower data rates of 1200 to 4800 bits/sec are used.

Because of the modular design of the SCI, each external device connection can be modified. For instance, if another asynchronous transmission format connection is necessary, either device 2 or 3 is easily modified to use the internal clock. Another possible modification is to include the synchronous method of transmission in the SCI as the transmitter and receiver circuits (the MOS/LSI integrated circuits described earlier) support this transmission format.

Programming Procedures

The final topics in the Operations section are the actual programming procedures necessary to initiate and control a transfer by the SCI. These procedures govern how a transmission is activated and how data are received.

Before a transmission can take place, the user must execute a WDCONT instruction to define the function bits. In most cases, this instruction needs to be executed only once, but if problems occur it may be necessary to reset the SCI with additional WDCONT instructions. The discussions below assume that the control/status register has been properly loaded. In addition, the Sigma external interrupts (ref. 2) are assumed to be properly enabled.

A transmission from the SCI to the external device is initiated by the user when executing a WDDATA instruction. This instruction loads the transmit data register and activates the transmit section. The transmit section then fetches and transmits the number of characters defined in the BREG field of the control/status register. An exception in this operation occurs if a user tries to execute a WDDATA instruction and the transmit section is concurrently transmitting a character (as indicated by the BUSYT flag). In this case, the SCI inhibits the WDDATA operation and informs the Sigma computer of the fact via the condition codes (described below); the on-going transmission is not affected.

The completion of the transmit operation indicates to the Sigma computer that a new operation can be initiated and is signaled in three ways:

- The BUSY Transmitting bit of the control/status register is reset; (BUSYT was set by the WDDATA instruction).
- The DONE Transmitting bit of the control/status register is set; (DONET is reset by either an SCI reset operation or by a WDDATA instruction).
- The transmission complete interrupt is generated.

The receive section of the SCI responds to inputs from the external device and is therefore continuously monitoring the input serial bit stream. Each time a character is received, it is loaded into the receive data register. When the number of characters defined in BREG have been loaded into the receive data register, the receive complete interrupt is generated and sent to the Sigma computer to alert it that a RDDATA instruction should be initiated. The error flags in the control/status register (bits 24 to 26) contain the logical OR of each of the error flags of the received characters currently stored in the receive data register. Therefore, the procedure to detect errors on a per-character basis is to set the value of BREG to 1. (If the value of BREG is greater than 1 and an error occurs, it is not possible to determine which one or more of the received characters were in error.)

Because the external device controls the input data, new data could arrive at the SCI before the old data is read. Therefore, the Sigma CPU must be programmed to execute a RDDATA instruction before new data arrives. The minimum amount of time the computer has to prevent this data overrun is equal to the time required to transmit one character (at a 9600-baud rate, the character transmission time is approximately 1 ms). If a data overrun error does occur, the receive data register overrun error (RROR) flag of the control/status register is set.

The normal procedure upon receiving the receive complete interrupt is to program the Sigma CPU to execute an RDDATA instruction and then execute an instruction to inspect the condition codes and determine if a hardware detectable error has occurred. The condition codes (ref. 2) are two bits that are automatically read by the Sigma computer on each DIO instruction and are defined uniquely for each DIO device. The SCI uses these bits to report status and possibly save a computer access to the control/status register. For example, a condition code is set if the SCI is busy (as in the case described earlier when a WDDATA instruction was executed while a transmission was occurring) or if an error is detected (as in the RDDATA case above). The detectable errors include a parity error (PE), a framing error (FE), a hardware overrun error (OR), a software overrun error (RROR), and an invalid BREG value (INHIBIT).

In most applications, if the Sigma computer is informed of an error by the condition codes, the program can immediately execute an RDCONT

instruction, inspect the error flags, and take the appropriate action. If, however, a time-critical application exists, where the program cannot immediately execute a RDCONT instruction, the stored error flags can be used. On each execution of the RDDATA instruction, the current state of the PE, FE, OR, and RROR flags is loaded into bit positions 20 to 23 of the control/status register and these stored error flags will not change state if new data arrives (with new error flag values) until another RDDATA instruction is executed. Thus, the Sigma program does not need to input the control/status register immediately upon learning of an error, but can wait for a more opportune time.

The final programming aids are the Busy Receiving (BUSYR) and Done Receiving (DONER) bits of the control/status register. These flags can be inspected by the Sigma computer at any time (i.e., during a transmit or receive operation) to determine the status of the SCI receive operation. BUSYR is set by the start bit of a received character and is reset after receiving the number of characters defined in BREG; thus, BUSYR is set during the entire time it requires to receive BREG number of characters. DONER is set each time BUSYR is reset and is cleared by the execution of an RDDATA instruction.

SYSTEM PERFORMANCE

The Terminal Area Effectiveness (TAE) program (ref. 4) involves placing a manned, simulated aircraft into a simulated Air Traffic Control (ATC) environment. The aircraft simulation is performed at Ames Research Center, Moffett Field, CA, and the ATC simulation is conducted at NAFEC, Atlantic City, NJ. The broad scope of the TAE program is to evaluate ATC procedures. This system makes it possible for a researcher to determine the effect on pilots of meeting complex ATC procedural requirements.

The two SCIs are components of a computer system for the TAE experiments, as shown in figure 4. The task of this system is to transfer the position data of the ARC aircraft to NAFEC for display on a simulated ATC radar station (a computer-driven cathode ray tube). The ARC-simulated aircraft appears as a blip on the ATC display, which moves about corresponding to the movements of the simulated aircraft.

The ARC Sigma 7 computer, shown in the upper left of figure 4, is programmed to direct the data flow between the Systems Engineering Laboratories (SEL) 840 computer and the NAFEC Sigma 5 computer. The SEL 840 computer contains the aircraft model and the Xerox Sigma 5 computer simulates the ATC environment. The two SCIs shown in figure 4 serve as components in the links between the ARC Sigma 7 computer and each of the additional two computers.

SCI number 1 (see fig. 4) interfaces the Sigma 7 computer to the SEL 840 computer located approximately 1 km away. The interface device for the SEL 840 is a standard SEL module, the Asynchronous Serial Communications

Interface (ref. 5). This SEL device is similar to the SCI in that it communicates with its CPU using a direct programmed I/O bus, and it uses a full-duplex, asynchronous transmission format. The device differs from the SCI in that it does not contain any status-reporting or error-detecting circuitry and it can transmit or receive only one character between CPU accesses.

Because of the relatively long distance between computers, a TTL-differential transmission technique and a nonswitchable cable were used between sites. This implies using the device 0 connection on the SCI. A 9600-baud, asynchronous transmission format is used in this computer-to-computer link.

This link's operational procedure is a request-response process. The Sigma 7 computer requests the simulated aircraft's position data (longitude, latitude, altitude, and ground speed) by transmitting three characters to the SEL 840. These characters are pointers into an array in the SEL 840 computer, which contains the aircraft position data in a predefined format. The SEL 840 responds by transmitting the data bounded by the array pointers.

Because of the relative simplicity of the hardware, the Sigma 7 to SEL 840 computer-to-computer data link is nearly error free. In over 100 hr of operation (a typical daily run would last 4 hr) no errors have been traceable to this link of the transmission system.

The position data received from the SEL 840 is then transferred from the ARC Sigma 7 computer to the NAFEC Sigma 5 computer using SCI 2. For this computer-to-computer link, the isonchronous transmission method (device 3) is used. Also shown in figure 4 are the synchronous modems and the leased-line transcontinental telephone circuits connecting the two sites. The modems and telephone circuits are designed to operate at the 1200-baud rate.

The NAFEC serial interface device, part of the NAFEC Sigma computer system shown on the right half of figure 4, was designed specifically for this project. This device transmits and receives on a character-by-character basis, as does the SCI. However, 30-character buffers are used in transfers to and from the Sigma 5 computer (using the Sigma multiplexed input/output processor rather than the DIO system, see refs. 1 and 2).

Because of the great distance between the two sites, the request-response process described earlier for the Sigma 7 to SEL 840 data link is not used. The Sigma 7 computer continuously transmits 30-character buffers each 1/3 sec. These blocks of data contain synchronization characters (checked on each transmission to verify the block), aircraft position data, an aircraft identification number, ground speed data, and a transmission block number. In addition to the checking of the synchronization characters and the block number, a hardware parity check is performed on each character. No error correction is attempted; that is, if one or more errors are detected in a block, the entire block is thrown out and the computer waits for the next block. For example, if synchronization loss occurs, the receiving computer detects the error, starts a search for the start-of-block synchronization characters to define the next block, and ignores any data until

synchronization is reestablished. The fact that blocks of data are "thrown away" if an error is detected has not adversely affected system performance because the NAFEC computer receives new data three times a second, but updates the entire ATC graphics unit only once every 4 sec. This level of excess transmission allows for system expandability and ensures system reliability.

The transmission block number described above is for system performance evaluation as well as error detection. Each time the NAFEC Sigma 5 computer receives a block of data with no detectable errors, it turns the data around and retransmits it to ARC. The ARC Sigma computer receives the returned block of data and checks for an invalid block number and other errors. The number of blocks returned with no detectable errors versus the total number of blocks transmitted by the ARC Sigma computer provides a useful percentage to evaluate the system's performance.

This "good blocks received" percentage has been greater than 95% in tests already conducted. The errors that have occurred are generally traceable to transmission circuit "hits" when the circuit momentarily goes dead. Such circuit "hits" often cause several sequential bits to be in error (multibit errors) but usually only cause single block errors, which do not significantly affect the operation.

The rates at which the computer-to-computer data links operate must be adequate to ensure the appearance of a smooth-moving aircraft symbol on the ATC graphics display unit. This requirement is easily satisfied by both data links. The Sigma 7 to SEL 840 request-response operation takes, at worst case, 80 ms (the simulation loop time of the SEL 840 computer is 60 ms, so it could take that long to synchronize the two computers), and the Sigma 7 to Sigma 5 data link provides a new block of data approximately every 300 ms. However, as described above, the ATC graphics display unit is updated once every 4 sec; thus, the data link rate is sufficient.

The most important verification of the system performance is the visual inspection of the aircraft blip on the ATC graphics display unit. The blip is smooth-moving with no unrealistic discrete jumps between position updates. In addition, the voice link (shown in fig. 4) between the simulator pilot and the ATC controller is used to verify that the simulated aircraft reaches a checkpoint displayed on its radio navigation display at the same moment that the aircraft blip reaches the same checkpoint on the ATC graphics display unit.

CONCLUDING REMARKS

The SCI is an extremely flexible, general purpose, serial I/O device. Although the SCI is designed for a Xerox Sigma series computer, many of its features can be easily adapted to similar interfaces to provide an effective tool in various applications.

The design requirements of modular configuration, simple operating procedures, and simple maintenance have been realized. The modular and flexible construction of the SCI has enabled it to meet the requirements of various applications. In addition, because of the SCI's simple operation and programming features, the time to generate the necessary software has been relatively short. Finally, the SCI has been very reliable.

The maintenance and self-check features (e.g., the loop-back capabilities) of the SCI have been extremely valuable. They allow an extensive checkout, even when installed in a complete system (i.e., no cabling needs to be altered); thus, debugging is simplified. Included in the maintenance features is the ability to read or write any SCI register. This feature allows effective diagnostic programs to be developed to check out not only the SCI, but also the Sigma DIO system.

An especially noteworthy feature of the SCI is its ability to transmit and receive more than one character with only one computer access. That is, BREG number of characters are transmitted or received by one computer access. This buffering of data is very valuable in real-time programs. A programmer can control the SCI on a time-available basis and still maintain a high percentage of SCI utilization. If necessary, the SCI can be readily modified to increase the buffer size from its present 4 to 15 since BREG is a 4-bit field.

The use of this multiple character per computer access feature is limited in a high error probability environment since the SCI's error flags pertain to all of the BREG characters received. Thus, if error analysis is needed on each received character, the contents of BREG must equal 1 and no multiple character buffering is possible.

An interesting addition to the above capability, not included in the present SCI, is to include separate BREG values for the transmit and receive sections. For example in a request-response operation, the SCI could transmit 1 character and receive 4 characters back, thus making the operation more efficient. (A user suggested this procedure, but it could not be implemented in the SCI in time to meet program demands.)

The error detection capabilities of the SCI have proven essential, for example, in the Sigma 7 to Sigma 5 transcontinental computer-to-computer link. In addition, error recovery procedures, error correcting codes, or other special transmission controls can be developed in software with the complexity necessary for a particular application.

A final comment concerns an application in which many of the SCI's features could be applied. It involves controlling several concurrent transmissions with one hardware interface, for example, transmission to and from several slow speed (110, 300, 1200 baud) terminals. (Presently, it would take one SCI per terminal to fulfill this requirement.) In addition, it may be practical to include a microprocessor controller in the design to provide an extremely flexible device. Software could then be generated to provide special functions, such as receiving inputs from several terminals but

transmitting the same output to all of them. Such a device could be a valuable tool in an environment similar to the flight simulation laboratories of Ames Research Center.

In conclusion, the SCI has proved to be a useful, efficient and reliable device. Its use in several applications, primarily involving computer-to-computer data links, has confirmed its flexibility in meeting the demands of a research and development real-time flight simulation environment.

APPENDIX A

CONTROL/STATUS REGISTER

Bit	Name	Description
00	CLEAR	During a write to the control/status register instruction, the SCI is cleared. (The new control bits loaded into the control/status register are not affected.) On a control/status register read instruction, this bit indicates if a clear operation was performed by the last control/status register load operation.
01	DTII	Done Transmitting Interrupt Inhibit.
02	DRII	Done Receiving Interrupt Inhibit.
03	RDSEL	Read Select: selects the register to be read by a program executing a read data register instruction. =0: Receive data register. =1: Transmit data register.
04 05	TSEL0 TSEL1	Transmit Select: selects the serial bit stream that is routed to the device select demultiplexer. =00: serial data from transmitter IC. =01: constant one (or mark) level. =10: constant one (or mark) level. =11: serial data received from the external equipment.
06 07	RSEL0 RSEL1	Receive Select: selects the serial bit stream that is routed to the receiver IC. =00: serial data received from the external equipment (selected by state of the DSEL bits). =01: constant one (or mark) level. =10: constant one (or mark) level. =11: serial data from the transmitter IC.
08 09	DSEL0 DSEL1	Device Select: selects the external device to which the serial data stream (selected by TSEL) is transmitted and from which the external serial data stream is received. =00: TTL-compatible differential driver output and differential receiver input. Asynchronous transmission format. =01: EIA RS-232-C standard interface. Asynchronous format. =10: EIA RS-232-C standard interface. Isonchronous format. =11: EIA RS-232-C standard interface. Isonchronous format.
10	DTR	Data Terminal Ready Control Bit.
11	Spare	

Bit	Name	Description
12	BREG0	Byte count register: indicates the number of characters to be packed into the transmit and receive data registers. BREG0 is the most significant bit. This parameter is currently restricted to the values 1, 2, 3, or 4.
13	BREG1	
14	BREG2	
15	BREG3	
16	Spare	
17	Spare	
18	CD	Carrier Detect Status Bit.
19	Inhibit	Inhibit error flag: occurs if BREG is not equal to 1, 2, 3, or 4.
20	PES	The stored state of the corresponding error flags (bits 24-27). The state of the four error flags below is stored into these locations during each read data register instruction to ensure that the error flags do not change before the program has the chance to determine the error.
21	FES	
22	ORS	
23	RRORS	
24	PE	The error flags pertaining to the data currently in the receive data register.
25	FE	
26	OR	
27	RROR	PE = Parity Error. FE = Framing Error. OR = Overrun Error. RROR = Receive data Register Overrun Error.
28	BUSYT	Busy Transmitting flag.
29	DONET	Done Transmitting flag.
30	BUSYR	Busy Receiving flag.
31	DONER	Done Receiving flag.

APPENDIX B

MODEMS AND TRANSMISSION FORMATS

This appendix contains a brief description of modems and their options and of transmission formats. A further description is contained in reference 6.

A modem (an acronym for modulator-demodulator) converts between signals that use voltages to define logic levels (e.g., in TTL logic a ground voltage level is defined as a logic 0 and 5 volts is defined as logic 1) and signals that use frequencies to define logic levels. These frequency logic levels are transmitted over telephone or other transmission circuits. In addition, several frequencies representing transmit data, transmit clock, receive data, receive clock, etc., can be transmitted on one circuit.

Several types of modems are available, including those that are compatible with TTL-levels, or with EIA RS-232-C standard levels. Common operating speeds include 1200, 2400, 4800, or 9600 bauds. Such modems may operate using one of two methods: asynchronous or synchronous.

In an asynchronous format, each transmission site has its own internal clock, which is of approximately the same frequency, but the phase relation between the two clocks is arbitrary. Further, the time between transmissions is variable.

The transmission entity in an asynchronous format is a character formatted as shown in figure 5. As can be seen, the transmission idle state is the mark level. When a transmission is activated, the transmitter first transmits the start bit to alert the receiver. The character is then serially transmitted, least significant bit first. The receiver determines the approximate middle of each data bit (based on the mark-to-space transition of the start bit) and clocks in the data. If so programmed, the receiver will also clock in the parity bit and signal an error if the received parity bit does not match the parity bit generated from the received character. The stop bit (or bits) is transmitted at the end of each character and gives the receiver the opportunity to reset and thus be ready to accept the next character. The rate at which this operation proceeds is called the baud rate and equals one over the time to transmit one bit. Figure 6 shows the transmission of the data 01000011 (base 2) at 9600 bauds. Figure 7 shows the connections between sites, using an asynchronous transmission format and asynchronous modems.

A synchronous transmission format differs from the asynchronous format in that a clock is transmitted with the data. The connections between sites using this transmission format are shown in figure 8 with synchronous modems generating the clocks.

In the synchronous transmission format, no start/stop control bits accompany the data. Instead, an entire block of data is synchronized and received. Match characters precede each transmitted block and are used by the

receiver to achieve synchronization. Once synchronized, the receiver counts incoming bits to assemble the block of data. The primary advantages of this format are the common timing source used by both the transmitter and receiver for higher reliability and the more efficient use of the channel since no start/stop bits are transmitted. However, a more complicated receiver is necessary to reconstruct the block of data.

A third transmission format is the isonchronous format, which takes features from both asynchronous and synchronous formats. In the isonchronous transmission format, a character is the transmission entity (as in the asynchronous format) and is formatted as shown in figure 5. However, a synchronous link is used between sites as shown in figure 8.

The isonchronous operation proceeds as follows (refer to fig. 8). Modem A generates the clock for the data transmitted from site A to site B and sends the clock both to site A to inform the transmitter when to put data on the line and to modem B. Modem B passes the clock onto site B to inform the receiver when to sample the data. This transmission technique has the advantages that a false start bit is highly unlikely, the receiver samples at the exact middle of each bit (due to the synchronous transmission), and the receiver uses the simpler asynchronous control logic. The time between transmissions is variable but is an integer multiple of the bit time.

It should be noted that asynchronous, synchronous, and isonchronous transmissions can be implemented with or without modems. For example, a teletype (or similar device) is usually directly connected to a computer without the use of modems and the two devices communicate using the asynchronous transmission scheme.

REFERENCES

1. Electronic Industries Association Standard RS-232-C. Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange, August 1969.

2. Xerox Sigma 9 Computer Reference Manual, Xerox Corporation. Publication Number 901733C, June 1972.

3. Xerox Sigma Computer Systems/Interface Design Manual, Xerox Corporation. Publication Number 900973E, June 1973.

4. Tobias, Leonard; and O'Brien, Paul J.: Real-Time Manned Simulation of Advanced Terminal Area Guidance Concepts for Short-Haul Operations. NASA TN D-8499, 1977.

5. Supplement to Technical Manual, Digital Input/Output Controller. Systems Engineering Laboratories, Publication Number 306-095018-00S, August 1972.

6. Lucky, Robert W.: Common-Carrier Data Communication. In *Computer-Communication Networks*, Norman Abramson and Franklin F. Kuo, eds., Prentice-Hall, Inc., 1973.

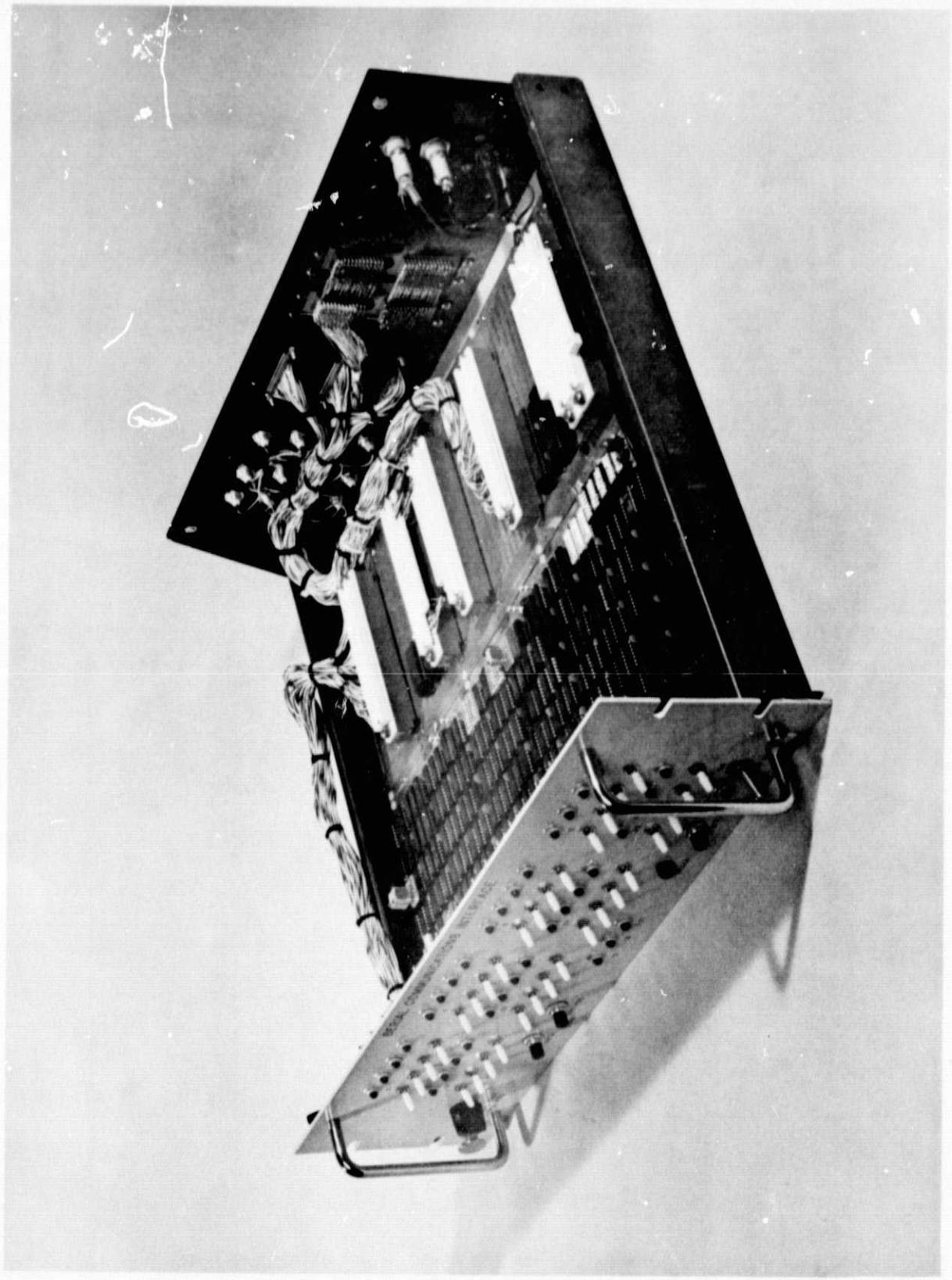

Figure 1.- Serial communications interface.

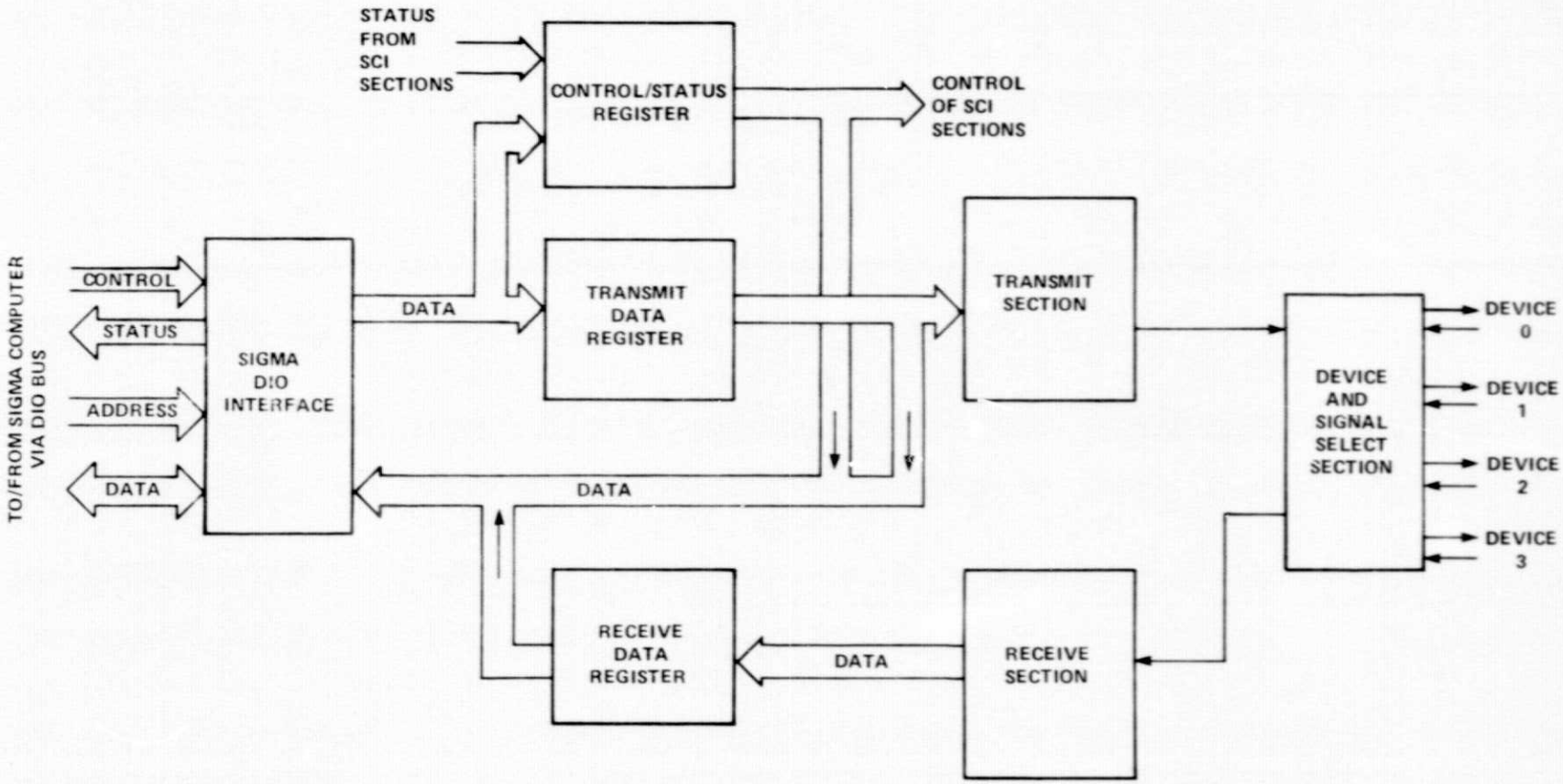

Figure 2.- Block diagram of serial communications interface.

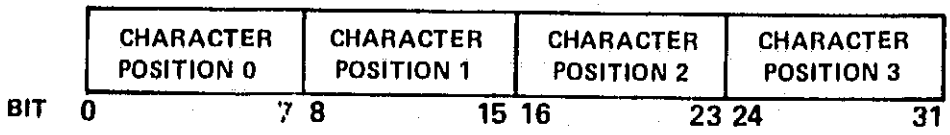

Figure 3.- Format of the SCI data registers.

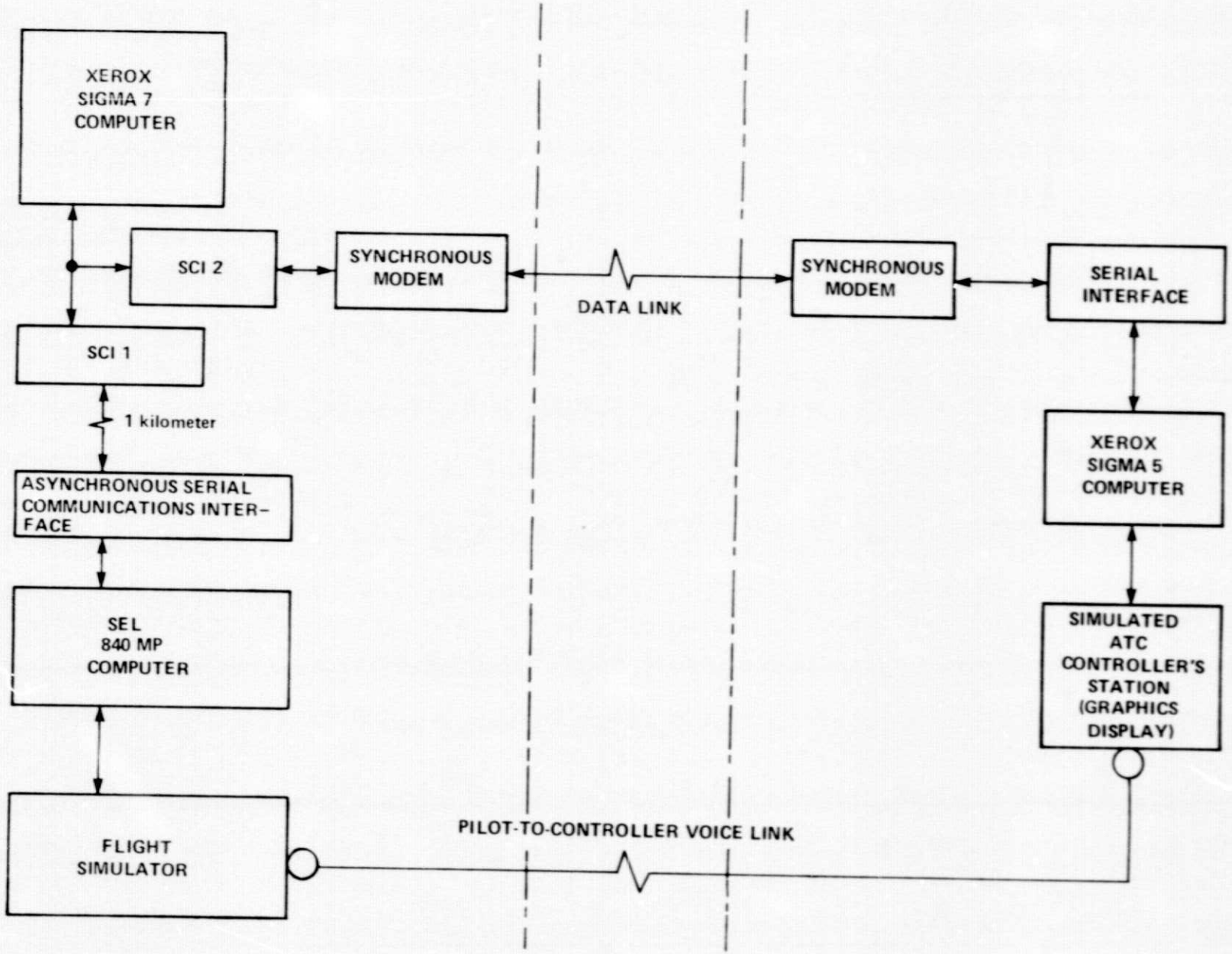

Figure 4.- TAE program computer system configuration.

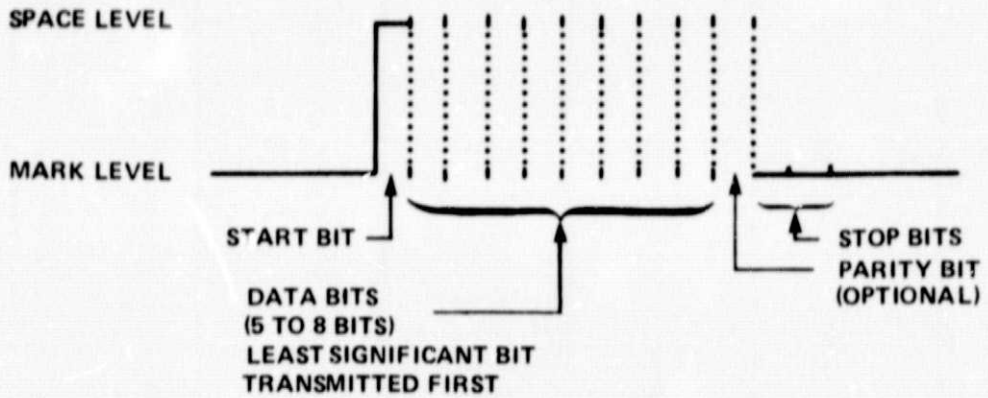

Figure 5.- Asynchronous transmission character format.

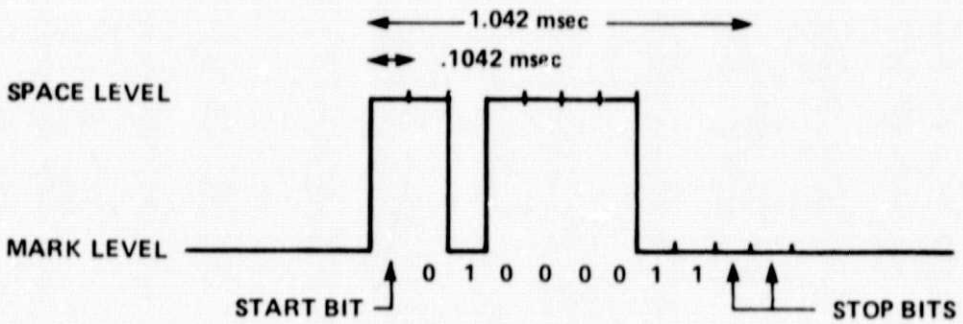

Figure 6.- Typical asynchronous data transmission.

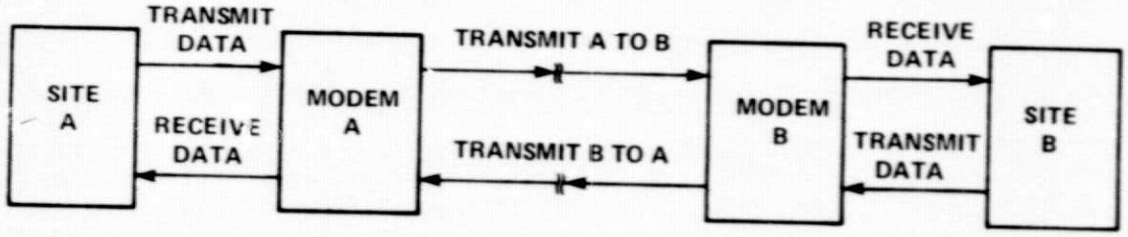

Figure 7.- Asynchronous intersite connections (asynchronous modems).

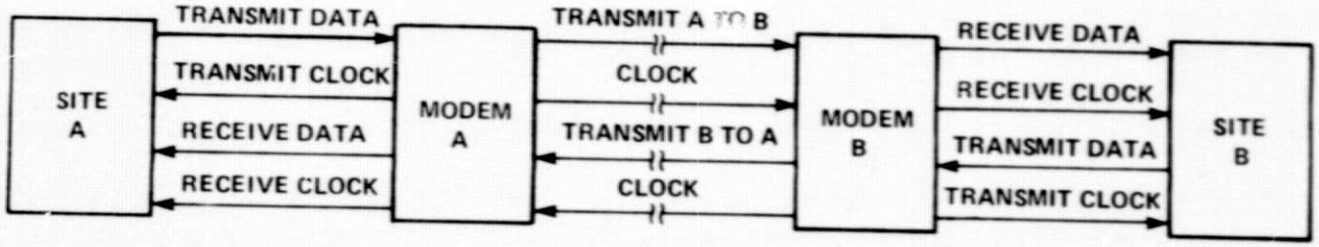

Figure 8.- Synchronous intersite connections (synchronous modems).

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-73,242		

4. Title and Subtitle	5. Report Date
A SERIAL DIGITAL DATA COMMUNICATIONS DEVICE	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
John L. Fetter	A-7041
	10. Work Unit No. 992-22-08-18-02

9. Performing Organization Name and Address	11. Contract or Grant No.
Ames Research Center Moffett Field, California 94035	

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, D. C. 20546	Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report describes a general-purpose computer peripheral device, the Serial Communications Interface, which is used to provide a full-duplex, serial, digital data transmission link between a Xerox Sigma computer and a wide variety of external equipment, including computers, terminals, and special-purpose devices. The interface has an extensive set of user-defined options to assist the user in establishing the necessary data links that originate in the research and development environment of Ames Research Center. This report describes those options and other features of the Serial Communications Interface and its performance by discussing its application to a particular problem.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Serial transmissions Data links Real-time, flight simulation	Unlimited STAR Category - 60

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	25	$3.50

*For sale by the National Technical Information Service, Springfield, Virginia 22161