ADMINISTRATIVE DATA LINK FACILITY DESCRIPTION NO. 1 ELECTRONIC SWITCHING SYSTEM

CONTENTS PAGE

**Reprinted to comply with modified final judgment.

CONTENTS PAGE

1. GENERAL

1.01 This section describes the administrative data link facilities used with the No. 1 Electronic Switching System (ESS).

1.02 This section is reissued for the following reasons:

(1) The data set 201A has been manufacture discontinued and superseded by the data set 201C. The change has the following effects:

(a) The end office must tell the regional office by voice communications if a 201C or 201A data set is being used. The regional office can work with either but must be notified.

(b) Offices using the 201C data set no longer require an 804A data auxiliary set but must use a type 565 phone set or equivalent. Offices continuing to use the 201A data set still must be equipped with the 804A data auxiliary set.

(2) To make minor changes.

1.03 The administrative data link facility provides a medium speed data communication connection

between a No. 1 ESS central processor and a remote facility equipped with the proper data terminal equipment. Such remote facilities include Western Electric Regional Data Centers (RDC) equipped with commercial computers used for ESS translation analysis and repack, and other ESS offices equipped with data link equipment. Utilizing appropriate procedures, the contents of ESS memory may be directly transmitted back and forth between the ESS central office and the remote facility via the data link. Furthermore, the use of the data link facility requires no special lines interconnecting the ESS office and the remote facility, since a standard switched voice-grade (DDD) telephone line is used for data transmission.

1.04 The data link facility is used primarily to transmit No. 1 ESS translation data to the RDC for repacking. After the translation information has been repacked, the new data is sent back to the ESS office where program store (PS) cards are written. This will eliminate the logistical problem of shipping mods of cards back and forth between the ESS central office and RDC whenever translations are repacked.

1.05 Trasmission via the data link facility involves reading binary information out of ESS (or commercial computer) memory in parallel form, converting the parallel data to serial form, and using the serial binary data to modulate a carrier that can be transmitted over the DDD network. Reception of data is essentially the reverse of the transmission process. The block diagram in Fig. 1 shows the equipment composing the data link facility, as well as the signal flow throughout the circuit. A buffer unit performs the function of parallel-to-serial conversion, and a data set utilizes the binary dc signals to phase-modulate a carrier for transmission via the DDD network. ▶The data set may be either a type 201C or 201A3 (manufacture discontinued).

(a) The 201C data set is used with a type 565 phone set, or equivalent, which allows for voice and data transmission.

(b) The 201A3 data set is equipped with a type 804A data auxiliary set, which is basically a standard telephone station set equipped with facilities for manual selection of voice or data transmission modes. At an RDC, similar data sets and commercial computer equipment are used.◀

1.06 The program controlling the exchange of data via the data link (data transfer program or XDLT) and the data link diagnostic program (XDLD) are located in program store module 05 (MOD 5) auxiliary test program module (described in PA-1A500). This module must be inserted into PS prior to use of the data link facilities.

1.07 When information is to be exchanged over the data link, a talking connection is first established using the data auxiliary set between personnel at the No. 1 ESS office and a remote location ▶by means of the type 565 phone set or data auxiliary set.◀ The operators verbally identify their locations and decide on data to be transmitted and the direction of transmission. ▶Personnel at both ends must use the same type of data set. The regional offices are equipped with both the 201C and 201A data sets but must be informed which to use.◀ After preparatory procedures utilizing the appropriate TTY messages, data is transmitted over the link using binary synchronous transmission, employing American National Standard Code for Information Interchange (ASCII) control characters as described in Part 3.

1.08 Abbreviation used in this section are explained in Part 7.

2. DATA LINK APPLICATIONS

2.01 Primary applications for the No. 1 ESS administrative data link facility include the following:

(a) Transmitting the contents of the PS translation area from an ESS central office to the RDC for translations area analysis (TAA) and translations retrofit repack (TRR).

(b) Transmitting PS data from the RDC to the ESS central office for card writing of translation changes.

Note: The MOD 5 program controlling ESS memory data transfer via the data link facility is not limited to the transmission of program store data alone. The contents of any program store ***or*** call store address may be transmitted, if necessary, by using the appropriate procedure.

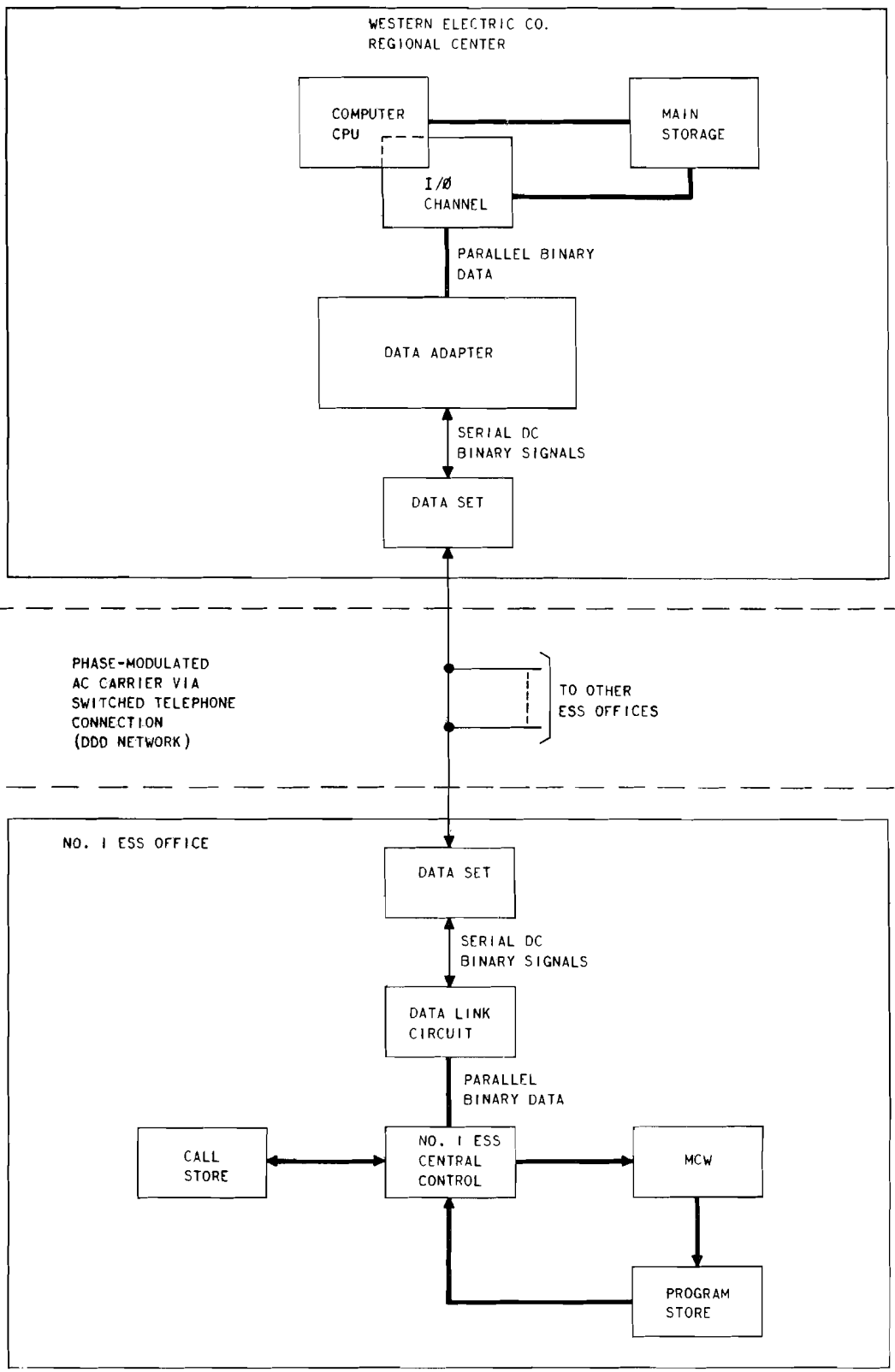


Fig. 1—Block Diagram of No. 1 ESS Administration Data Link Facility

3. DATA LINK COMMUNICATIONS

METHOD OF TRANSMISSION

A. Binary Synchronous Transmission

3.01 Communication is accomplished over No. 1 ESS data link by means of ***binary synchronous transmission.*** Using this form of transmission, information is transmitted between data link stations in the form of marks and spaces (Fig. 2). The basic unit of intelligence is in the form of an 8-bit binary synchronous character (Fig. 2A). The characters that comprise a data transmission are formulated by the administrative data link program.

3.02 ***Synchronous Transmission:*** Binary synchronous transmission means that transmission synchronization between data link stations occurs on a group basis rather than trying to synchronize with each character transmitted. Using binary synchronous transmission, a group or block of characters can be transmitted as a continuous stream of data. The block is preceded and followed by special ASCII control characters (Table A). Binary synchronous transmission allows more efficient transmission, ease of implementation, and greater message length flexibility than nonsynchronous transmission.

3.03 ***Asynchronous Transmission:*** Non-synchronous or asynchronous transmission utilizes an 8-bit character (Fig. 2B) that is preceded by a start bit (a mark), and followed by a stop bit (a space). This technique of data transmission requires that the receiving station pause and digest these synchronizing signals with each character. Overall transmission time is increased because of character synchronization. Asynchronous is the normal mode of transmission used in TTY loops.

B. Transparent Mode

3.04 The No. 1 ESS data link operates in the ***transparent mode*** of binary synchronous transmission. This mode is initiated by sending (or receiving) DLE STX. From this point on, the only control character recognized and acted upon is the data link escape (DLE) character. Operation in the transparent mode permits transmission of binary data in non-ASCII form. Thus, raw program store (PS) or call store (CS) binary words can be transmitted directly to a regional center without first converting them to ASCII characters.

3.05 When data is transmitted, character synchronization is maintained, but all 8-bit positions of each character are used for data. The No. 1 ESS memory is transmitted in variable length blocks with lengths up to 64 PS words (one PS

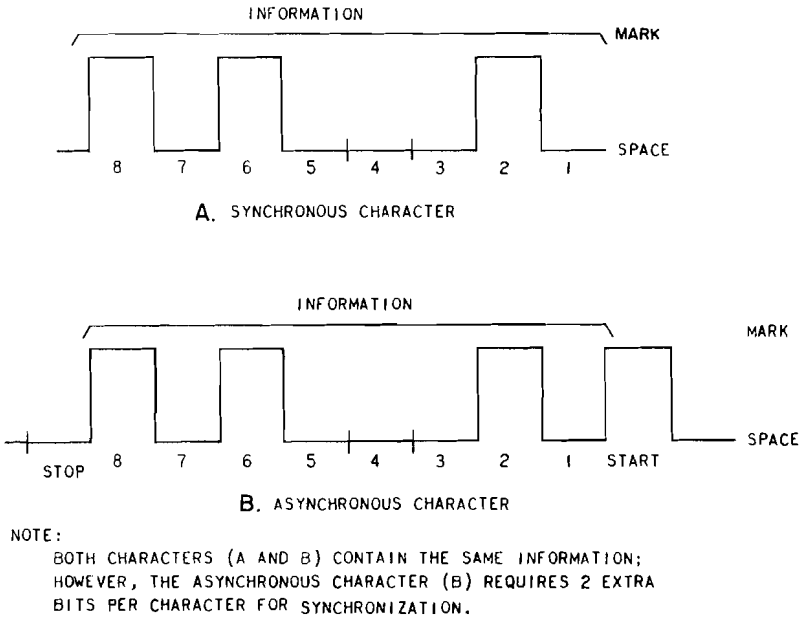


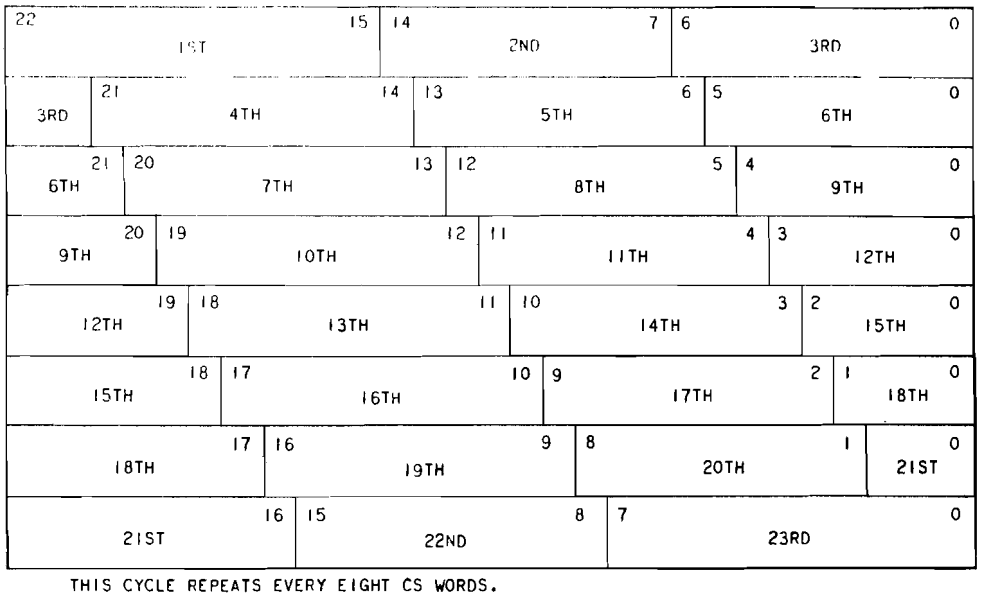
Fig. 2—Binary Characters

TABLE A

ASCII CONTROL CHARACTERS

CHARACTER	BINARY BITS PARITY 7 6 5 4 3 2 1	MEANING
SYN	0 0 0 1 0 1 1 0	Synchronous Idle
STX	0 0 0 0 0 0 1 0	Start of Text
ETB	1 0 0 1 0 1 1 1	End of Transmission Block
ETX	1 0 0 0 0 0 1 1	End Of Text
EOT	0 0 0 0 0 1 0 0	End of Transmission
ENQ	1 0 0 0 0 1 0 1	Enquiry
ACK 0	1 0 1 1 0 0 0 0	Affirmative Reply
ACK 1	0 0 1 1 0 0 0 1	Affirmative Reply
NACK	0 0 0 1 0 1 0 1	Negative Acknowledgment
DLE	0 0 0 1 0 0 0 0	Data Link Escape
WACK	0 0 1 1 1 0 1 1	Wait: Positive Acknowledgment
PAD	1 1 1 1 1 1 1 1 or 0 1 1 1 1 1 1 1	Begin and End Characters No Meaning

card) or 128 CS words. The sequence for sending the data from a call store buffer area is shown in Fig. 3. This sequence repeats once every eight call store words. The data is sent in a data stream format; that is, bit 22 of the first word is sent first, etc. Since the data link circuit shifts data out of the right side of registers, all characters must be revised before transmission. Figure 4 shows how the PS data is buffered in the call store before transmission. Each data block is framed with beginning and ending control characters.



THIS CYCLE REPEATS EVERY EIGHT CS WORDS.

Fig. 3—Data Sequence for Transmission

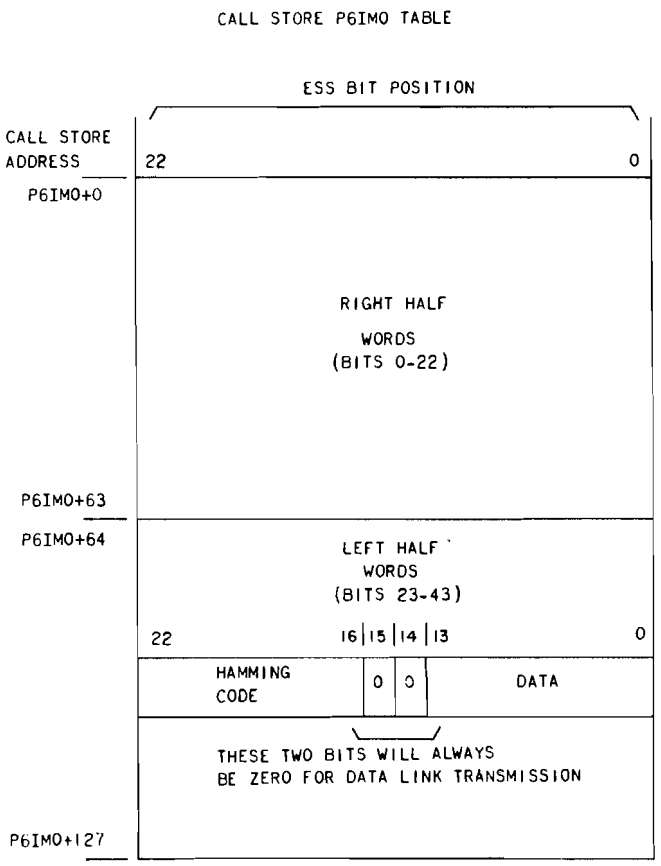


Fig. 4—Program Store Data as Organized in Call

3.06 The following is a general format for data block transmission.

Start of Block Sequence	PADPAD	(Bit synchronizing characters)
	SYNSYN	(Character synchronizing characters)
	DLESTX	(Start of text control characters)
	TEXT 1 TEXT	(Data words)
End of Block Sequence	DLEETB (OR ETX)	(End of block control characters)
	CRC	
	PADPAD	(Cyclic error checking code)

Start and ***end of block*** sequences consist of 16-bit control words. A 16-bit control word is formed with the help of two 8-bit control characters. The SYN and STX characters prepare the data link remote station to receive the data text. The ***end of block*** control characters inform the remote station that the block is complete. The ***error checking code*** or cyclic redundancy check (CRC) characters are compared with similar characters generated at the remote station at the end of each data block. When the WE regional center is the master station (transmitter) sending data to the No. 1 ESS terminal, the data stream must always contain an ***even*** number of data characters per block. If a block contains an odd number of text characters, a PAD character (eight 1s) should be inserted as a fill character just before the DLE ETB or ETX sequence. Also, in all cases, the PAD character is used to assure complete transmission of the last error check character. This is because a 16-bit input/output register is used in the data link circuit and the software must look at 16 bits at a time.

3.07 When transmitting in the transparent mode, the possibility exists that raw binary data may take the form of an ASCII control character. To counter this possibility, the data link escape (DLE) character is used in every transparent mode transmission. The DLE character precedes the transmission of every ASCII control character, identifying it to the remote station. While in the transparent mode, only control characters that are preceded by a DLE character are recognized. If a DLE character appears within a raw data bit configuration, the circuit will recognize it and go out of the transparent mode (paragraph 3.04). If, however, the next character is another DLE, the circuit will go back into the transparent mode. Thus, whenever a DLE is encountered in the raw binary data, an extra DLE is inserted following the first DLE. As previously explained, no other ASCII character is recognized during transparent mode transmission.

TRANSMISSION SEQUENCE

3.08 The following paragraphs describe the sequence of events that take place during data link transmission. Formats used for data transmission are also described. A general flowchart showing the sequence of events during a typical transmission is shown in Fig. 5.

A. Establishment Phase

3.09 The establishment phase consists of voice communication between operators at each end of the data link to provide terminal identification and establish the ***master*** and ***slave*** station status.

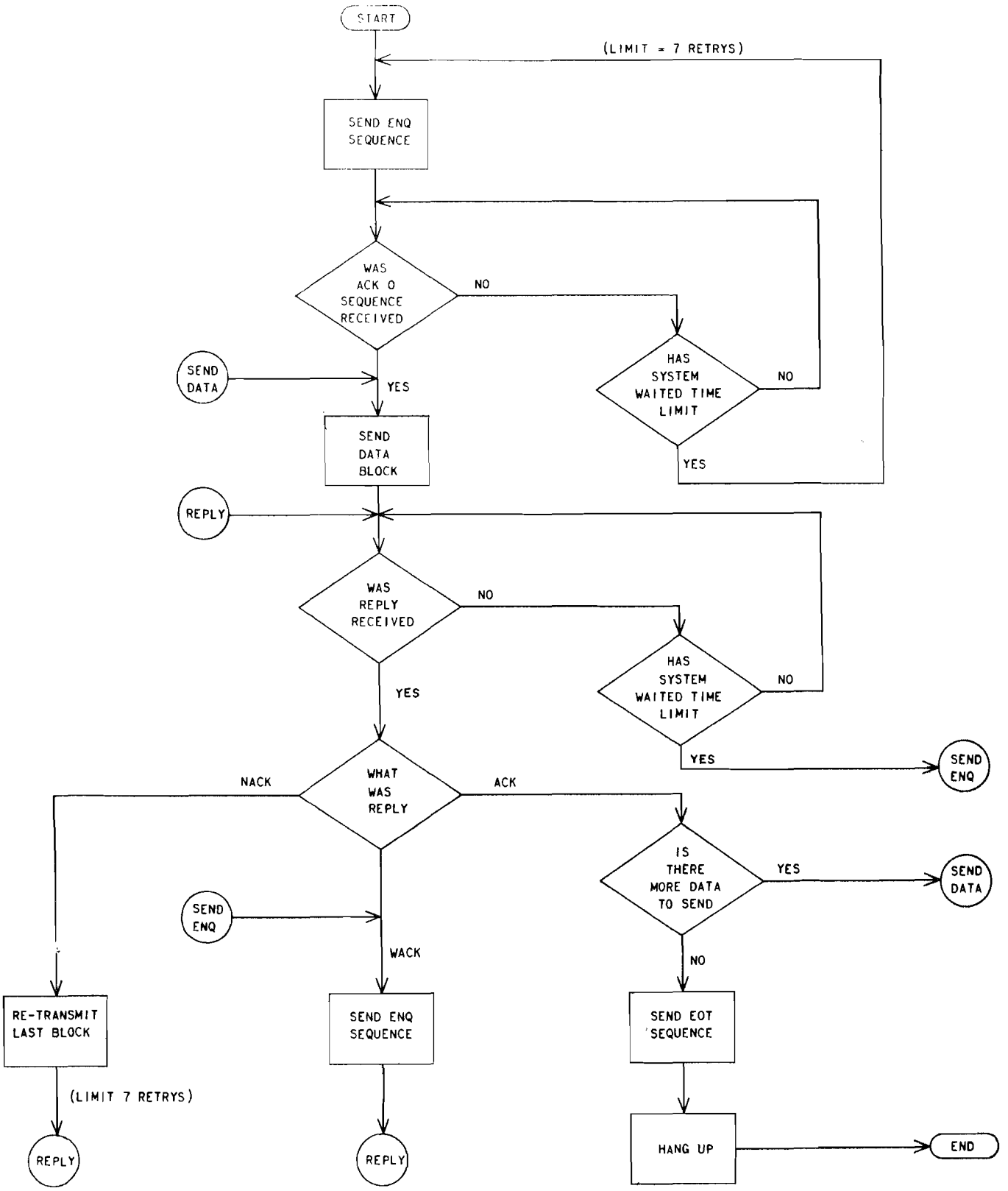


Fig. 5—Data Link Transmission Sequence

A. INQUIRY (ENQ) SEQUENCE	B. POSITIVE ACKNOWLEDGMENT SEQUENCES ACK0	ACK1	C. NEGATIVE ACKNOWLEDGMENT (NACK) SEQUENCE	D. WAIT BEFORE ACKNOWLEDGMENT (WACK) SEQUENCE	E. END OF TRANSMISSION (EOT) SEQUENCE
PAD	PAD	PAD	PAD	PAD	PAD
PAD	PAD	PAD	PAD	PAD	PAD
SYN	SYN	SYN	SYN	SYN	SYN
SYN	SYN	SYN	SYN	SYN	SYN
ENQ	DLE	DLE	NAK	DLE } 16-bit WACK	DLE
PAD	0	1	PAD	SEMICOLON } character	EOT
PAD	PAD	PAD	PAD	PAD	PAD
PAD	PAD	PAD	PAD	PAD	PAD

Note 1: All characters that make up each sequence are standard 8-bit ASCII control characters (See Table A).

Note 2: All initial PADs may be replaced by a non-zero number of SYNs.

Fig. 6—Data Link Transmission Control Sequences

The station transmitting data will act as the ***master*** and place a call to the receiving station, which then must act as a ***slave.*** ***Master-slave*** designations are determined at the start of a run and do not change. That is, if a No. 1 ESS is transmitting translation data to a WE regional center, the ESS is the ***master*** and the regional center is the ***slave.*** The ***slave*** will, however, sometimes be receiving from the ***master*** and sometimes be transmitting to the ***master.*** ***Slave*** responses are replies to previous ***master*** transmissions and never contain anything other than control characters.

3.10 To begin a transmission over the data link, the operator of the ***master*** station establishes voice communication with the ***slave*** operator via standard switched telephone lines using the data auxiliary set. The two station operators verbally confirm their identities and describe the transmission that is to take place. Upon completion of verbal confirmation, the ***slave*** station operator will place his data set in the data mode by pressing the data button on the data set. A short time later the ***master*** station operator will activate his data set and also activate the data link transfer program by typing in the appropriate TTY messages. The ***master*** will then send the inquiring (ENQ) sequence as shown in Fig. 6A.

3.11 The ***slave*** must reply to this inquiry with the correct acknowledgment (ACK) sequence. There are two types of ACKs: the ACK0 and ACK1 (Fig. 6B). These ACK sequences are alternated throughout the entire transmission starting with ACK0. Thus, the ***slave*** will reply with the ACK0 sequence. This is the only permitted response to the ENQ. If the ENQ is not received or is received incorrectly, the ***slave*** will not reply. After three seconds without a reply, the ***master*** will again transmit the ENQ sequence. After seven retries without receiving the ACK0 from the ***slave,*** the ***master*** will abort.

Start of Block Sequence		*PADPAD *SYNSYN *DLESTX
Text	10 CS words sent 16 bits at a time	START ADDRESS (FOR MAJOR BLOCK) END ADDRESS (0 FOR ESS RECEIVING) MOD NUMBER (E (22) + MOD IN OCTAL IF RECV MODS) CARD NUMBER (E (22) + MOD, PASS AND CARD IF RECV SINGLE CARDS) RIGHT HASH SUM (0 UNLESS ESS RECV MODS) LEFT HASH SUM (0 UNLESS ESS RECV MODS) OFFICE ID1 (RIGHT HALF K2NAME) OFFICE ID2 (LEFT HALF K2NAME) SPARE SPARE
End of Block Sequence		*DLEETB *CRCCRC *PADPAD

*16-bit control words

Note: A HEADING DATA BLOCK consists of 10 call store words sent in 16-bit words preceded and followed by standard control characters.

Fig. 7—Heading Data Block

B. Transmission Phase

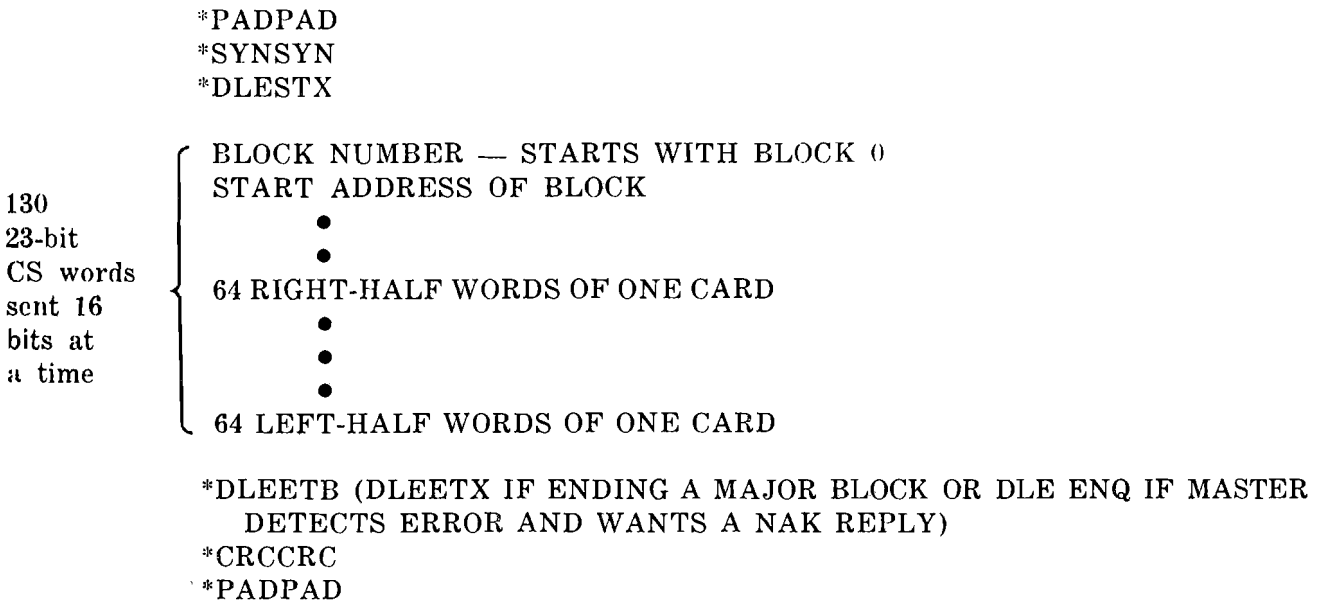
Heading Block

3.12 After the ***master*** has received the ACK0 sequence from the ***slave,*** it will then transmit the heading data block shown in Fig. 7. This block contains information concerning succeeding data blocks. It identifies the ESS office transmitting (or receiving) and the starting address as it appears in ESS memory (or CS) of the PS data being transmitted. If ESS is transmitting, this data block will also indicate the end address of the transmission. When the ESS is receiving, this block will indicate if the next major data block is an entire mod of PS cards or a single card. In the case when mods are being transmitted, the right and left hash sums are also included.

Data Blocks

3.13 Following the heading block, PS or CS data will be transmitted in one or several blocks, as required. There are two basic types of data blocks which are transmitted. The first type, called a minor block, terminates the data portion with the DLE ETB (end of block) character. (The heading data block is considered a minor block.) The second type is called the major block and is terminated by the DLE ETX (end of text) character. When transmitting single cards, the card image is a major data block. When transmitting mods, each card except the last card in the mod is considered a minor data block. The last card completes the mod and is considered a major block and is terminated with the DLE ETX.

3.14 The ***slave*** may acknowledge the transmission of a major or minor data block in one of three ways: with the proper ACK sequence; with the NACK sequence; or with the WACK sequence. The NACK sequence (negative acknowledgment), shown in Fig. 6C, is sent if the transmission is felt to be incorrect. The acceptability of the transmission is determined by comparing the cyclic redundant checking (CRC) result as computed by the ***slave*** with the CRC of the ***master*** which has been transmitted at the end of the data block,



*16-bit control words

Note: A PS DATA BLOCK consists of 130 call store words preceded and followed by control characters.

Fig. 8—Program Store Data Block

and/or by any other safety check which the ***slave*** software may wish to make on the data. When the ***master*** receives the NACK reply, it will retransmit the previous data block. A maximum of seven consecutive retries will be attempted before the ***master*** aborts.

3.15 If the data block is acceptable as transmitted, but the ***slave*** is not ready to receive more data, the slave replies with the WACK (wait before acknowledgment) sequence as shown in Fig. 6D. The slave may not be ready to receive for reasons such as the WRITE button has not been pressed to start the card writer or card writing is in progress. When the ***master*** receives the WACK sequence, it will respond with an ENQ sequence. If the ***slave*** is still not ready for more data, it will again reply with a WACK. There is no limit to the number of WACK sequences. When the ***slave*** is ready for additional data, it replies with the proper ACK sequence.

3.16 When the ***master*** receives the ACK sequence, it transmits the data block in the format shown in Fig. 8. All data blocks except the heading data blocks are in this format. Minor blocks will be followed by DLE ETB and major blocks by DLE ETX. Each of these blocks consists of 130 23-bit call store words, preceded and followed by data link control character. The first 23-bit word gives the block number being transmitted. The block number is incremented with each block, beginning with 0 for the first nonheading data block. The next word is the start address of the following PS data. The next 64 words are the right halves of the PS words being transmitted. The last 64 words are the left halves. Thus, each data block consists of a complete PS twistor card image. The ***slave*** will respond to this transmission with a NACK, ACK, or WACK sequence as described in paragraph 3.14.

3.17 The final data block in the major block ends with the DLE ETX. After the ***slave*** responds with an ACK, the ***master*** will transmit a heading data block and further minor and/or major data blocks.

C. Termination Phase

3.18 If the ***master*** has no further data to be transmitted, it will send the EOT (end of transmission) sequence, as shown in Fig. 6E, and go on-hook. The ***slave*** station will go on-hook without generating a reply.

D. Delays, Retrys and Aborts

3.19 In general, if the No. 1 ESS is the ***master*** station, it will not permit more than three seconds to elapse without receiving a reply of some type from the ***slave.*** If no reply is received within this time in response to any sequence, an ENQ sequence will be sent. If no reply is received in response to the seventh transmission of the ENQ, the ***master*** will assume the transmission path has been lost and will abort.

3.20 If the No. 1 ESS is acting as the ***slave*** station and is not ready for more data, it will delay sending the WACK sequence for two seconds. If during the two seconds the ***slave*** becomes ready for more data, the ACK sequence will be sent without delay. This reduces the number of WACK transmissions and also reduces the probability that an error will occur during a WACK sequence.

3.21 All data blocks, including the heading data block, contain a cyclic error checking code. In the event that a transmission error occurs, it most likely will be detected by a CRC check failure. The data block can then be retransmitted. There is no such check character on ENQ, NACK, WACK, or ACK sequences. If an initial ENQ sequence is to be received and does not appear correct, the ESS will make no reply. This will cause a retry of the ENQ by the ***master*** within three seconds. If the ***master*** is expecting a NACK, WACK, or ACK sequence and receives a reply which is none of these, it will assume the reply was a NACK and will retransmit the previous block. Note that the ***slave*** will be aware that a block is being retransmitted because it will contain a block number identical to the previous block or will be in heading format. The ***slave*** may ignore a retransmission of a data block in this case, but must correct his ACK0-ACK1 alternation. For example, if the ***slave*** responded with an ACK0 to a given data block but the ***master*** interpreted this reply as a NACK and retransmitted, the ***slave*** must again respond with ACK0 and not ACK1.

E. Single Card Transmission

3.22 When PS cards are installed, the hash sums of the newly written mods will be verified. It is possible that the hash sums will not agree with the expected sums. This may be due to a card writer failure or some other reason. Hash sums of individual cards will then be taken and the card or cards in error will be determined. If an individual card is in error, a retransmission of that single card must be made to the ESS. It is not practical to require rewriting an entire mod of 128 twistor cards because one or two cards are incorrect. Therefore, the operator of the ***master*** must be able to selectively retransmit single cards. The heading data block for this type of transmission will not be required: that is, the ***master*** will send the initial ENQ sequences. Upon receiving an ACK0, it will send the data block corresponding to the card to be transmitted. When the ACK1 is received for this data, the ***master*** will send the EOT and terminate. The missing heading data will be supplied to the ESS by the ESS operator via a TTY input message. The block count for this type of retransmission will be ignored.

4. DATA LINK EQUIPMENT

4.01 The No. 1 ESS data link circuit consists of a buffer unit and a data set ▶201C◀ (or 201A3 equipped with a data auxiliary set 804A6) connected to a voice-grade telephone line. The data link circuit shares a miscellaneous trunk frame with other trunk circuits.

A. Data Set

4.02 The serial binary bit stream is generated by the No. 1 ESS data link buffer unit (described in 4.06) or by the RDC computer data adapter. The data is in the form of mark and space signals, varying between two specified dc levels (Fig. 2A). The data is transmitted over the standard DDD network. Data sets are interposed between the ESS data link circuit and the telephone line, and between the telephone line and the RDC computer data adapter, respectively.

Data Set 201C

4.03 ▶The 201C data set is a synchronous, serial binary transmitter receiver that operates at 2400 bits per second over 2- or 4-wire private lines and the switched network. It has status indicators,

self-test operation, and a self-contained line control circuit. The line control circuit makes it unnecessary to have an 804A data auxiliary set. It is necessary to have a type 565 phone set or equivalent. The 201C data set and operation are described in Section 592-029-100.

Data Set 201A

4.04 Data sets 201A3 used in this circuit operate at a data rate of 2000 bits per second, each bit having a 500-microsecond duration. Each is equipped with an internal clock for timing the buffer unit as well as the data set. The data sets operate in either the transmit or receive mode and are controlled by the No. 1 ESS program. For a complete description of data set 201A3, refer to Section 592-011-101.

4.05 Essentially, a serial binary transceiver, data set 201A3 is comprised of a phase modulator and demodulator. The modulator utilizes the serial binary dc signals shifted out of the buffer unit shift register to phase-modulate an audio frequency tone which is transmitted over the telephone line. The demodulator detects the phase-modulated tone coming in over the telephone line and converts this tone back into serial binary dc signals which are shifted into the buffer unit.

4.06 Each 201A3 data set is equipped with a data auxiliary set 804A6. The data auxiliary set provides a means for establishing voice communication between receiving and transmitting locations plus facilities for manually selecting either voice or data transmission modes. For a complete description of data auxiliary set 804A6, refer to Section 598-030-100.

B. Buffer Unit

4.07 The purpose of the data link circuit buffer unit is to provide a data interface between the data link and ESS central processor. All data transmission via the data link is serial, whereas data transfer within the ESS central processor is handled in parallel. The function of converting serial binary data to parallel binary data, or vice versa (depending upon direction of transmission), is provided by this unit, as well as buffering data for central processor input-output. Cyclic error checking is also performed within the buffer unit. A more complete discussion of the operation of the buffer unit and its associated circuits is given in Part 5. Figure 9 gives a simplified block diagram of the data link circuit.

5. DATA LINK CIRCUIT OPERATION

TRANSFER OF DATA

A. Transmit Mode

5.01 When an area of PS memory is to be sent, TTY messages are typed into the system specifying the area to be transmitted. PS data, as explained in paragraph 3.16, is sent in blocks of 64 words (one PS card), corresponding to 128 CS words. Preceding the PS data, two 23-bit words are sent which contain the block count (starting with 1 for the first data block) and the start address in PS of the data contained within the block. The image of each PS card within the area of PS to be transmitted is assembled in the P6IMO table of call store, which serves as a storage or buffer area for the data transfer program. This table is 128 CS words long, starting at address P6IMO (see Fig. 4). The first 64 CS words are the right halves of the 64 PS words being transmitted and the last 64 CS words are the left halves. The block count and start address words are assembled by the program in the call store J-level client work table.

5.02 The program computes the block count and start address of the data block and loads it into the J-level client work table. The image of the PS card being sent is then assembled in the P6IMO call store table. A pointer is used by the program to load the 44-bit word from the translation area of PS into the P6IMO table. Two data readings are made at each of the 64 locations in PS. The first data reading obtains bits 0 to 22 of the PS word and the second reading obtains bits 23 to 43. When the data in the client table is ready and all 64 words have been loaded into the P6IMO table, the program requests the system to transmit an ENQ sequence to the RDC. If an ACK is received, the program directs the system to transmit the proper control sequences. Every five milliseconds (J-Level interrupt), a check is made to see if 16 bits of data may be shifted out of the tables via the peripheral unit address bus (PUAB) into the data link input buffer register (IBR). This check determines if the IBR has been read. The contents of the IBR are shifted into the data link shift register (SR) once every eight milliseconds. If the RDY (ready) flip-flop indicates

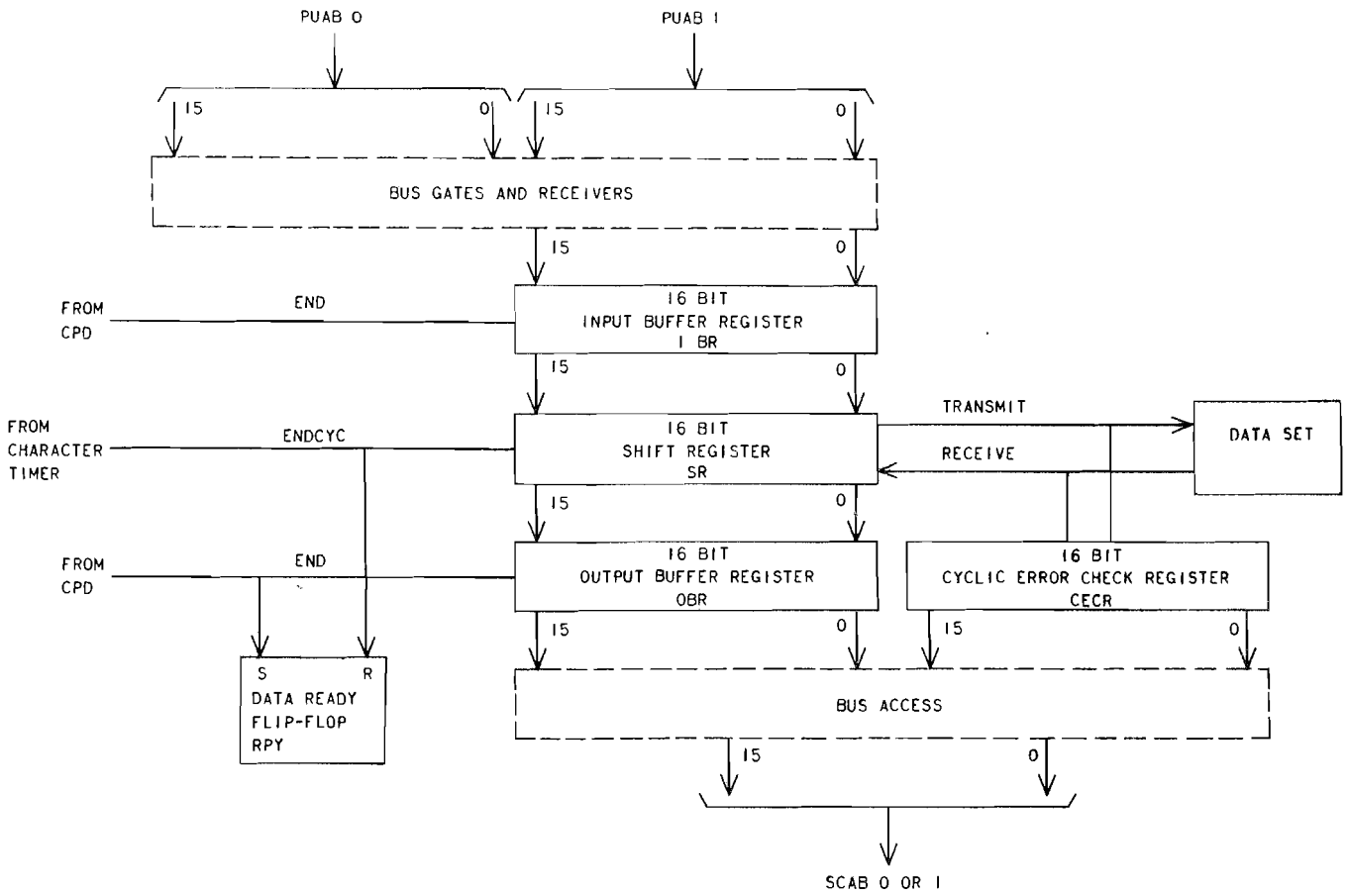


Fig. 9—Simplified Block Diagram of No. 1 ESS Data Link Circuit

the IBR is ready for more data (RDY-1), the next 16 bits of the data are loaded into the IBR. If there is more data to be sent, the tables will again be loaded and the contents transmitted as previously explained. This sequence is repeated until all PS memory designated for transmission has been sent.

5.03 CS data is handled by the program using a sequence similar to that described for PS data. TTY messages are entered which specify the area of CS to be transmitted. Data is read directly from the specified area of CS in 23-bit words and loaded into the P6IMO table, where it is stored until called up for transmission by the data transfer program. The P6IMO table holds a maximum of 128 call-store words.

5.04 The data transfer program supplies the IBR with 16 bits of parallel data from the CS buffer area via the PUAB. This action can occur only after the program empties the IBR of any data previously transferred from CS. The circuit continually takes data from the IBR, gates it into the SR, and then shifts it out serially into a data set. The data set phase-modulates the data and transmits it over the DDD network to a receiving data set.

5.05 The functional part of the No. 1 ESS data link circuit necessary for data transmission is shown by the simplified block diagram in Fig. 10A. The program reads data from the call store buffer area and sends it to the buffer circuit over the PUAB, together with the CPD enable (EN0) function. The program uses the EN0 function to gate the data from the PUAB into the IBR and set the ready (RDY) flip-flop. When the shift register (SR) shifts its contents out to the data set (16 bits in series) and is ready for new data, the ENDCYC function is generated which loads the SR from the

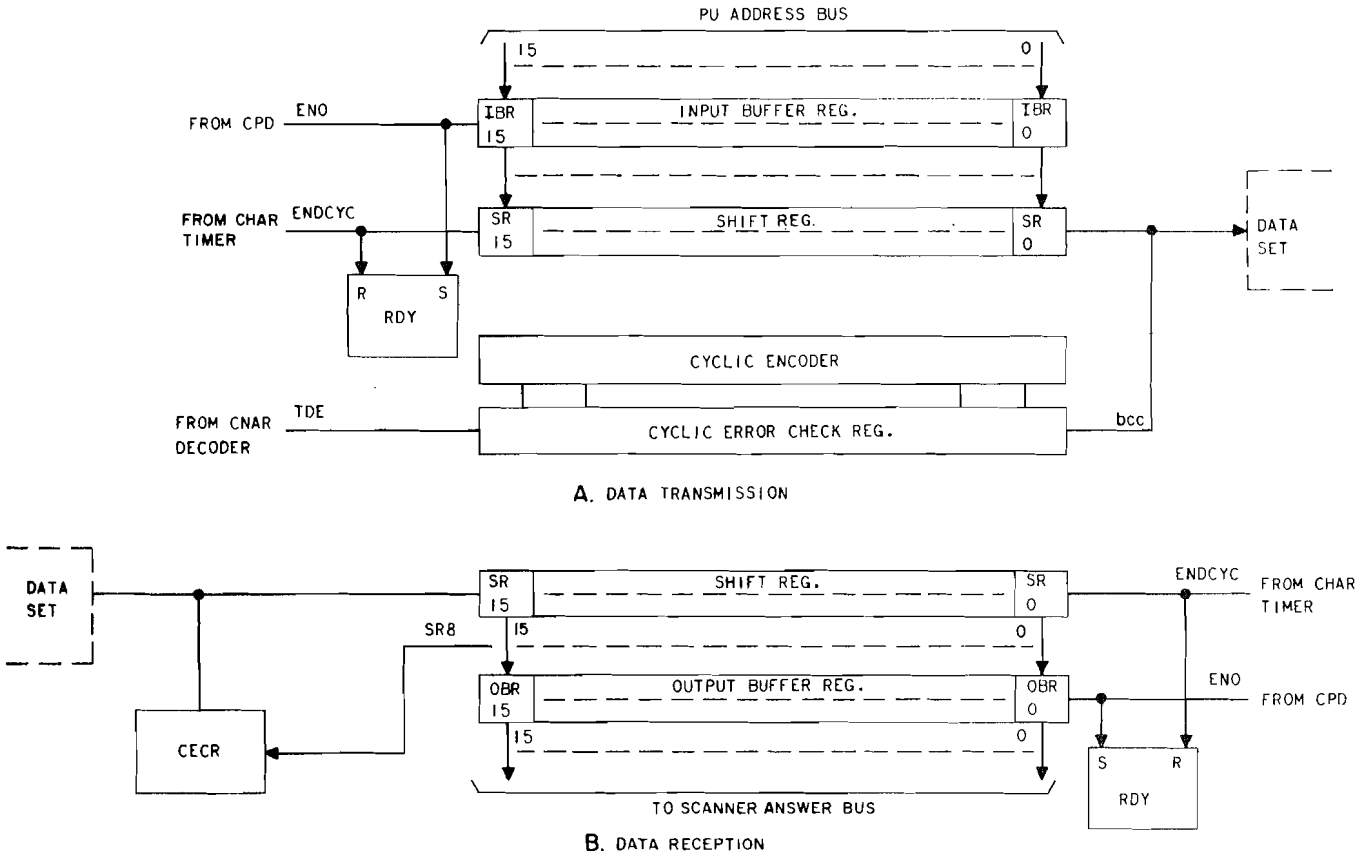


Fig. 10—Functional Parts of Data Link Terminal Circuit Used for Transmission and Reception

IBR with two data link characters (16 bits in parallel), and sets the RDY flip-flop. The SR then shifts one bit each 1/2 millisecond to the data set until all 16 bits have been sent, after which the SR is again loaded from the IBR. This process is repeated until all data has been sent to the data set.

5.06 As each bit is shifted out to the data set, it is used to form a block check character in the cyclic error check register (CECR). At the end of data transmission, the contents of the CECR are shifted, one bit each 1/2 millisecond, out to the data set. The cyclic error checking character (CRC) is sent to the regional center and compared with the CRC computed for the data received at that end of the link. When the two CRCs fail to compare, the negative acknowledgment (NAK) character is sent back to the ESS terminal and the block is retransmitted automatically. The receipt of two successive NAKs will cause termination of the transmission and require manual corrective action before transmission can be resumed.

B. Receive Mode

5.07 In the receive mode, the data is demodulated by the data set and continually loaded into the buffer unit SR. The data is read, 16 bits at a time, by parallel insertion units an output buffer register (OBR) that is accessible by the No. 1 ESS program via the scanner answer bus (SCAB).

5.08 The functional part of the data link circuit necessary for data reception is shown by the simplified block diagram in Fig. 10B. The binary pulse train is shifted into the SR from the data set. After each shift, bits 15 through 8 are decoded until a SYNC character is found. When two successive SYNC characters are detected, character synchronism has been achieved. The ENDCYC function is now generated which gates the contents of the SR to the OBR and resets the RDY flip-flop. With the RDY flip-flop in the set state (equal to one), the program reads the contents of the OBR via the scanner answer bus with the ENO function and stores the data in the call store.

The RDY flip-flop is set with the EN0 function so that the program will know when new data is ready in the OBR.

5.09 Once the ENDCYC function gates the SR to the OBR, the next two characters (16 bits) from the data set are shifted into the SR. As each bit is shifted through cell 8 of the SR, it is combined with the CECR to generate the CRC. When the last of the data is shifted into the SR, the CECR is shifted into the SR for program access via the scanner answer bus (SCAB).

5.10 When the program shifts incoming data from the SCAB to call store, the call store P6IMO table is again utilized to store the incoming data for card writing. Control sequences are first sent which tell the program to prepare to receive data. If a heading block is received, the program stores the information concerning the starting address of the PS data being transmitted and determines whether the following data block is an entire mod of PS cards or a single card.

5.11 Upon receipt of the following data block, the data transfer program loads the P6IMO table with the contents of the OBR, 16 bits at a time, until P6IMO contains a complete PS card or 64 PS words. At this time, the card writing control program (PCCW) takes control, and via appropriate TTY printouts, indicates that a complete mod (or single card) is to be written. TTY printouts indicate to office personnel when to initiate the appropriate procedures to activate the memory card writer (MCW).

5.12 (PCCW) scans the MCW for a WRITE (WR) signal. If a WR signal is detected, the data transfer program extracts the proper PS word from the P6IMO table and transmits it to the MCW to be card written. If there are additional data blocks to be transmitted to ESS, the RDC continually queries the ESS with ENQ sequences. If card writing is in progress, the ESS transmits a WACK sequence in response to each ENQ sequence from the RDC. After all the words contained in P6IMO have been sent to the MCW and the card has been written, control is returned to the XDLT program which directs the ESS to respond to the next ENQ sequence with an ACK sequence, indicating the ESS is ready to receive another block of data. If a single card is being transmitted, a DLE ETX character follows the first complete data block causing the above polling sequence to be initiated. This continues until another card is made ready for writing or data link transmission is terminated by the end of transmission (DLE EOT) sequence.

5.13 When a complete mod is being transmitted, the MCW continues to write cards. XDLT, in conjunction with PCCW, must keep up with the MCW as it writes each card. If not, it will abort. This sequence continues until all cards within a complete mod are written, as indicated by the transmission of the DLE ETX sequence from the RDC.

5.14 If no more mods are to be transmitted, as indicated by the reception of the end of DLE EOT sequence, all data link transmission ceases. If additional mods are to be transmitted, the first data block is sent and held in the P6IMO table until the MCW is loaded with a new mod and ready for card writing. As previously explained, the transmission of each succeeding data block is delayed until the previous block is written.

CENTRAL PROCESSOR ACCESS

5.15 The data link circuit can accept data from the No. 1 ESS central processor (CP) via either of the two PUABs and, depending on which of the two enabler signals is received, send data to the CP on either of the two SCABs.

TIMING

5.16 Timing for the data link circuit is derived from the ◆201C (or 201A3) data set◆ clock. In the transmit mode, the data set connected to the data link circuit supplies the timing from its oscillator. In the receive mode, the far-end or remote data set supplies the timing via the carrier signal.

ERROR CHECKING

5.17 Unlike other ESS circuits, no maintenance interrupt is used for error checking in the data link circuit. As previously described, a cyclic redundancy checking code (CRC) is computed in the CECR based on data shifted through the SR. Since the CRC is computed by the data link circuit and not by the ESS program, there exists a gap in the error check of transparent data (Fig. 11). When data is received by the ESS from the regional data center, errors in transmitting the data from the SR to the central control circuit via the SCAB

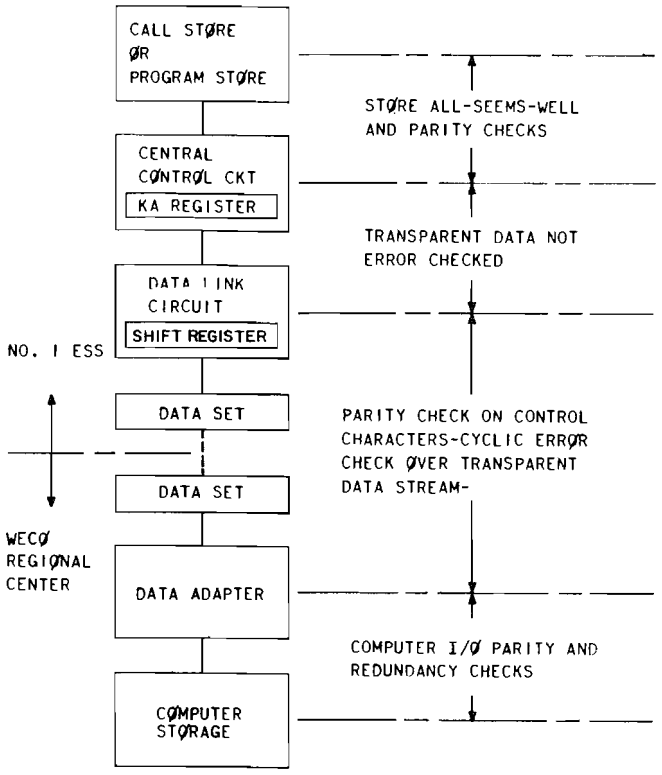


Fig. 11—Error Checking

cannot be detected. If the data is encoded with Hamming check bits (as is the case with translation words to be written into the program store), error checking can be performed in the ESS by the program. The nontransparent data characters consist of a 7-bit ASCII code with parity on the eight bit. This parity bit is generated by the ESS program in the central control circuit so that no error checking gap exists for nontransparent data.

5.18 When transparent data is transmitted to the RDC from ESS, the CRC is computed over the data as it passes through the SR as before. Errors in transmitting the data from central control to the SR will be undetected and the possibility exists that the CRC will be computed over erroneous data. To protect against this possibility, the data in the SR must be returned to the central control for verification with the original data.

5.19 To provide a return path for the transmitted data to be verified, use is made of the 2-way characteristics of the data link circuit. By combining the data link transmitting and receiving modes of operation, a complete round trip for the data is possible. The data is sent to the data link circuit via PUAB and gated to the IBR by means of the EN0 enable. The EN0 enable also gates the OBR to the central control via the scanner answer bus. The 16 bits of data from the OBR are verified by program with the data sent to the IBR two SR cycles ago. In the event that this check fails, XDLT will end the block being transmitted with DLE ENQ. This causes the slave to reply with NACK. This must be done since it is possible that the data became mutilated prior to entering the CECR and, therefore, the two CRCs would match.

5.20 The cycle time of the SR is eight milliseconds and the data transfer program operates on segments five milliseconds apart. There are many segments of the program during which data is not sent to the IBR from the SR because previous data is still in the IBR. The RDY flip-flop provides the mechanism for synchronizing the program to the SR cycle time. This flip-flop is set by the EN0 enable to indicate that the IBR has new data for the shift register. When the SR takes the data from the IBR at the end of its cycle (ENDCYC),

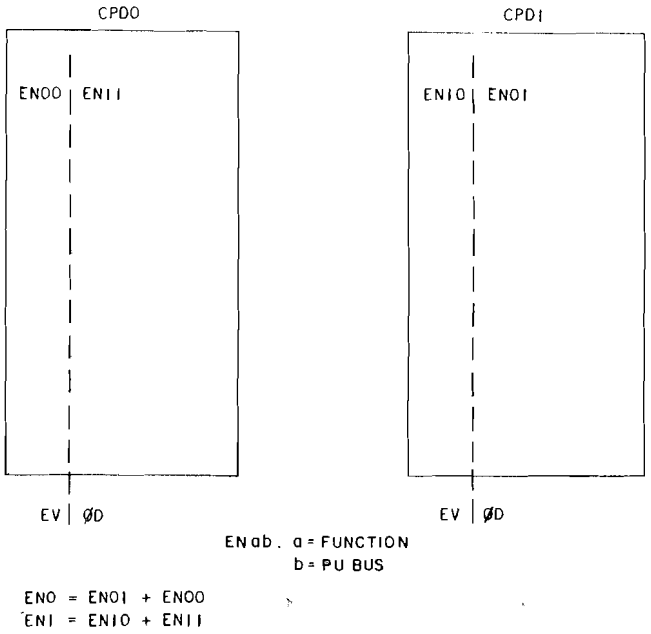


Fig. 12—Enable Assignments and Lead Functions

the RDY flip-flop is reset. The program first checks the RDY flip-flop each five milliseconds and sends new data if the RDY flip-flop is reset. A CPD enable (EN1) is used to interrogate the RDY flip-flop via the scanner answer bus.

5.21 To ensure maximum hardware operations during the "round trip" of data, the data in the SR is shifted serially out to the data set and back to the other end of the SR. In this way, the data is rotated through the SR before being gated to the OBR.

5.22 Since the circuit is unduplicated, only two out of the usual four CPD assignments are used for the enabling (EN0) function. The two assignments are dictated by the requirement that the circuit be operable with either CPD or PUAB. For this reason, the assignments are made in separate CPDs and in different columns (corresponding to different PUABs). The remaining two assignments are used for the EN1 enable. Figure 12 illustrates the CPD assignments.

A. Fault Recognition

5.23 The error checking capability thus far described should be sufficient to determine whether a block of data is transmitted or received properly. Errors in the transmitted data may be divided into three classes: random errors to be expected in any data transmission facility, asynchronism due to the program falling out of synchronism with the data link circuit, and malfunctions in the hardware.

(a) ***Random Errors:*** Errors in the first class may be overcome by simply repeating the data transmission if the expected error rate is small compared to the length of the transmission. It may be that a bad network connection is causing excessive errors. In this case, the data transmission should be successful over a new network connection. If this fails, the alternate PUAB and CPD can be used.

(b) ***Asynchronism:*** Errors of the second class are a result of the critical timing required of the program to synchronize with the data link circuit. Normally, the data may be retransmitted successfully unless the system is experiencing a high occurrence of maintenance interrupts. In this event, data link transmission should be terminated and the trouble corrected before attempting retransmission.

(c) ***Hardware Malfunctions:*** Errors of the third class are the result of a hardware malfunction. If the retransmission of the data is successful, a marginal operation of the hardware may be indicated. When these errors occur frequently, a diagnosis is necessary. However, data obtained when the errors occurred may be the only means of locating the marginal trouble.

6. REFERENCES

PA-1A500—Auxiliary Programs

Section 231-149-301—Administrative Data Link Facility, Transmit and Receive Procedures

Section 592-011-101—Data Set 201A3, A4 and 201B3, B4 Transmitter-Receiver, Description and Operation

Section 592-029-100—Data Set 201C Transmitter-Receiver, Description and Operation

Section 598-030-100—Data Auxiliary Set 804A, Description and Operation

7. ABBREVIATIONS

ASCII	American National Code for Information Interchange
CECR	Cyclic Error Check Register
CP	Central Processor
CPD	Central Pulse Distributor
CRC	Cyclic Redundancy Check (Error Checking Code)
CS	Call Store
DDD	Direct Distance Dialing
ESS	Electronic Switching System
IBR	Input Buffer Register
MCW	Memory Card Writer
OBR	Output Buffer Register
PCCW	Card Writing Control Program
PS	Program Store
PUAB	Peripheral Unit Address Bus
RDC	Regional Data Center
RDY	Ready
SCAB	Scanner Answer Bus
SR	Shift Register
TAA	Translations Area Analysis
TRR	Translations Retrofit Repack
TTY	Teletypewriter
WE	Western Electric Company, Inc.
XDLD	Data Link Diagnostic Program
XDLT	Data Link Data Transfer Program