PROCESSOR INTERFACE FRAME
DESCRIPTION
PERIPHERAL DATA STORAGE PROCESSOR
2-WIRE NO. 1 ELECTRONIC SWITCHING SYSTEM

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

Figures

CONTENTS PAGE

CONTENTS PAGE

1. GENERAL

1.01 This section describes the processor interface frame (PIF) and its operation as part of the peripheral data storage processor (PDSP) and the 2-Wire No. 1 Electronic Switching System (ESS). Figure 1 shows the overall functional location used with the No. 1 ESS PDSP configuration. Figure 2 shows the configuration used with Enhanced Private Switched Communication Service (EPSCS) and Expanded 911 service. The PIF is part of the PDSP and serves as a data buffer between No. 1 ESS, the 3A processor, and the analog transmission facilities. EPSCS and E911 are features provided by the PDSP.

1.02 This section is reissued to include the 2A generic changes. Since this reissue is a general revision, arrows ordinarily used to indicate changes have been omitted.

1.03 Abbreviations used in this section are listed in Part 7.

2. PHYSICAL DESCRIPTION

2.01 The PIF (Fig. 3, 4, 5, 6, 7, and 8) is a single bay frame 7 feet 0 inches high, by 2 feet 2 inches wide, by 1 foot 0 inches deep. The PIF is arranged to support one processor interface unit (PIU), one universal data link control circuit unit (UDLC), and from 1 to 16 data link interface (DLI) units consisting of the two plug-in circuit packs which are provided as necessary to meet office requirements.

2.02 The PIU contains two independent processor interface circuits; circuit 0 is located on the left side of the unit and circuit 1 on the right side. The same arrangement is used for the duplicated UDLC circuits. The DLI circuits are not duplicated but individually assigned to a data set. In general, the PIF layout follows the standard arrangement used for all No. 1 ESS frames. The power converters, filters, fuses, and relays are located at the bottom of frame. See Fig. 7.

3. PROCESSOR INTERFACE UNIT (PIU) FUNCTIONAL DESCRIPTION

A. General

3.01 The PIU serves as a data communications channel between the No. 1 ESS and the 3A

processor and, in addition, performs temporary buffering of the data. Figure 9 shows a detailed block diagram (marked to show data flow) of the PIU. The unit has the capability of storing 256 - 32-bit words from the No. 1 ESS which are passed to the 3A processor as 512 - 16-bit words, using the first-in first-out (FIFO) discipline. In the reverse direction, 512 - 16-bit 3A processor words can be stored and passed to the No. 1 ESS as 256 - 23-bit words, using the FIFO (not the same FIFO) discipline. The upper 9 bits of every other 3A processor word are dropped in making up one No. 1 ESS word. This buffering (called Q and R) is accomplished by using two 36-bit wide by 256-word long random access memories (RAMs) which allow for a 32-bit data word plus parity bits over each set of 8 data bits.

3.02 Two read only memory (ROM) microsequencers control the loading and unloading of the FIFO buffers. One sequencer, (termed Controller A) as directed by the No. 1 ESS, loads data into one of the two buffers (termed Q buffer) via the call store bus (part of the data comes from the address bus) and unloads the other buffer (termed R buffer) via the call store bus. The second sequencer (termed Controller B) is directed by the 3A processor and loads the R buffer and unloads the Q buffer.

3.03 Under direction of the No. 1 ESS and/or the 3A processor, status registers can be written or read. From the ESS two status registers (23 bits wide) and a buffer status register (buffer status gives the condition and the number of words in the buffer) can be controlled. From the 3A processor, six status registers (16 bits wide) can be controlled. Two of these registers give the status of the buffers. The sequencer, registers, and maintenance circuits which are controlled by the ESS are referred to as Controller A, while those equivalent functions associated with the 3A processor are referred to as Controller B (see Fig. 9).

B. Microsequencer

3.04 A diagram of the microsequencer is shown in Fig. 10. The sequencer consists of three circuit packs, FC470, FC471, and FC474. The ROM is 28 bits wide by 256 words long. Inputs I0-I24 are circuit state points (scan points). Two of these points can be sampled in each sequencer address state. The scan point states determine whether the microsequencer microprogram address is incremented or branches to a new address stored in the ROM. The FC470 combines the scan information (I0-I24) with the ROM feedback information (F0-F15) used to determine the next sequencer address.

3.05 In any given program address, two outputs from the sequencer (fire points) can be fired to control the flow of data, reading, or writing of status registers and various maintenance functions. One fire point from group G1-G7 and one from group H1-H15 can be used in each state.

C. Orders to PIU from No. 1 CC

3.06 Figure 11 shows the states of control bits (AG, AC, AH, A2, A1, A0, AW, AR) required from the ESS to Controller A for the various write, read, and interrupt functions which can be performed in normal maintenance and control modes. These control bits are part of the No. 1 ESS call store bus. In the normal and maintenance modes, three functions can be performed: (1) buffer write, (2) buffer read, and (3) buffer status read. In the control mode, four functions for each of the duplicated PIUs (circuit 0 and circuit 1) can be performed: (1) read X-status registers, (2) write X-status registers, (3) read Y-status registers, and (4) write Y-status registers.

D. Buffer Write

3.07 A buffer write transfers data from the call store bus to the Q buffer and is shown by the wide arrows in Fig. 12. Data, address and control signals enter the PIU through the cable receivers. This information arrives in three time-separated waves, each accompanied by a unique synch bit (ASIA, AS2, DSA). During the first two waves, address, data, and control information are registered. In the last wave, the data is registered in the input registers. Next, the data in the input register is gated onto the D-bus (DW13L), from the D-bus onto the M-bus (DDML), and into the Q buffer (WRQL).

E. Buffer Read

3.08 A buffer read transfers data from the memory register to the call store bus and from the R buffer to the memory register as shown in Fig. 13. To transfer data from the R buffer to the call store bus requires two buffer reads. During a buffer read, first the data in the memory register is gated onto the D-bus (DRDL) and then into the output register (GRWL). The data from the output register is then gated through the cable drivers onto the call store

bus. Second, the data in the R buffer is gated onto the M-bus (GMRL) and into the memory register (GMRL). Data going from the D-bus to the output register is reduced from 32 to 23 bits. Bits D14 through D22 are ANDed with bits D23 through 31, respectively, to form the shortened word. Twenty-three bit parity is added.

F. Buffer Status Register Read, and X- and Y-Status Reads (See Fig. 11)

3.09 A buffer status read transfers the data in the buffer status registers to the call store bus as shown in Fig. 14. Information concerning the individual bits of the X-status word, Y-status word, and the buffer status word are shown in Fig. 15. The buffer status read takes place as follows: The buffer status register is gated onto the D-bus (DBSL) and into the output registers (GRWL). The output registers are gated (DRWL) through the cable drivers to the call store bus. The X- and Y-status reads are similar. They have different control codes (see Fig. 11) and are driven onto the D-bus with DSRL—WXL and DSRL—WYL, respectively.

G. X- and Y-Status Register Write

3.10 An X- or Y-status write allows the ESS (over the call store bus) to set certain registers (see Fig. 11). Refer to the data flow in the X- or Y-status registers as shown in Fig. 16. First, the data to be written is registered in the input registers (W3CH). The input registers are driven onto the D-bus (DW13L) and into the X- (GSRL—WXL) or Y- (GSRL—WYL) status word. WXL and WYL are controlled by address bit A2 and complement each other.

H. 3A Auxiliary Processor Orders to PIU

3.11 Figure 17 shows a diagram of the control functions from the 3A processor and a table which defines the function of each of these signals (DP, DR, CP, SST, ADR0, and ADR1). The DP command writes data into the R buffer (see Fig. 9). A DR command reads data from the Q buffer. A CP command writes data into one of the status registers, but may be used to only select the status word select register (SWSR). An SST command reads the data in a status register and transfers it to the 3A processor.

I. Buffer Write (DP)

3.12 The DP command writes data into the R buffer as shown in Fig. 18. This is a 3-step operation. Registers J through O are 18-bit wide registers which allow for 16 data bits and 2 bits of parity, one over each set of 8 data bits. The M-bus is 36 bits wide and allows for the transfer of both the J- and K-registers to the R buffer in parallel. The J-register contains the low order bits which will go into the R buffer while the K-register contains the upper order bits. The 3A processor interfaces with the PIU through the duplex bus selector (DBS) and can write only one register at a time. Starting with WHAL pointing at the J-register, a DP command will cause data from the 3A processor to be written in the J-register; the WHAL pointer will then toggle and point to the K-register (step 1). A second DP command will cause data to be written into the K-register (step 2). At the termination of the second DP command, the contents of both the J- and K-registers will be gated onto the M-bus (step 3) activated by RMFL. From the M-bus, the data will be registered in the R buffer (WRRL). The WHAL pointer is toggled again and now points back at the J-register.

J. Buffer Read (DR)

3.13 The use of the DR order results in the data in the Q buffer being transferred to the 3A processor. Like the buffer write, this is a 3-step operation (see Fig. 19). Assuming that a word has been previously brought from the Q buffer into the N (low order bits) and O (high order bits) registers and that WHAL is pointing at the N-register, a DR command will move the contents of the N-register onto the J-bus (DFJL). From the J-bus, this data is gated into the L-register (RQXL) and also through the buffer (DJBL) to the 3A processor. The WHAL is toggled and now points at the O-register. A second DR command causes the O-register to be driven onto the J-bus (DFJL). This data is gated into the M-register (RQXL) and also through the buffer (DJBL) to the 3A processor. A new word is brought from the Q buffer to the N- and O-registers by gating the Q buffer (RQL) onto the M-bus through the buffer (RMFL) into the N-register (RMFL) and also into the O-register (RMFL). The WHAL pointer is toggled to point at the N-register.

K. Status Write (CP)

3.14 The data path for a 3A processor status write, a CP command, is shown in Fig. 20. There are six status registers associated with the B Controller (P, Q, R, S, T, and U). Only the S-, T- and U-registers can be written, and then only certain bits as shown

in Fig. 21. These registers are 16 bits wide. The upper 3 bits are called the SWSR and are used by the 3A processor to determine which status register is to be written or read. The P, Q, R, S, T, and U are selected, respectively, by the upper bits being 000, 001, 010, 011, 100, and 101. Bit 12 is used during a write to determine what part of the register is written. If bit 12 = 0, only the SWSR register is written. If bit 12 = 1, the total register can be written by a CP command. Data comes from the 3A processor through the buffer (DJBL) and is gated (GJSL) into the register specified by the SWSR.

L. Status Read (SST)

3.15 The data flow path for a 3A processor status read is shown in Fig. 22. When an SST command is issued by the 3A processor, the contents of the status register specified by the SWSR is driven onto the J-bus (DSJL). From the J-bus, the data is gated through the buffer (DJBL) to the 3A processor.

M. Buffer to Buffer Transfer

3.16 For maintenance purposes the PIU has the capability of transferring the contents of one buffer to the other. The contents of the R buffer can be transferred to the Q buffer and is controlled by the A Controller. The B Controller controls the transfer of data from the Q to the R buffer.

N. Q to R Buffer Transfer (LABL)

3.17 When the LAB (Loop-Around from B) bit is set (see the X-register of Fig. 15 and the U-register of Fig. 21), a 36-bit word from the Q buffer is gated onto the M-bus (WRQL) and from the M-bus into the R buffer (GMRL). This operation is under the direction of Controller B. See Fig. 23.

O. R to Q Buffer Transfer (LAAL)

3.18 When the LAA (Loop-Around from A) bit is set (see X- and U-registers) an R to Q buffer transfer takes place under the direction of Controller A as shown in Fig. 24. A 36-bit word from the R buffer is gated onto the M-bus (WRRL) and from the M-bus into the Q buffer (RQL).

P. Maintenance Loop-Around Functions

3.19 There are four maintenance functions which use two or more of the sequences. These are as follows: short loop-around from the ESS, short loop-around from the 3A processor, deep loop-around from the ESS, and deep loop-around from the 3A processor. These functions are performed in the No. 1 and 3A programs by first setting a special loop-around flip-flop (using an X-status register write from No. 1 CC) and then from issuing sequences of buffer read and buffer write orders.

Q. Short Loop-Around from ESS

3.20 A short loop-around from the ESS combines a buffer write (see Fig. 12) and a buffer read (see Fig. 13) as shown in Fig. 25. The LAA bit must be set. During the buffer write, data is registered in the input register, then passed via the D-bus to the output registers. The R buffer is not gated (DDML) to the M-bus. During the buffer read, the output register is driven onto the call store bus. Note the memory register is not activated (DRDL).

R. Short Loop-Around from 3A Processor

3.21 A short loop-around from the 3A processor combines a DP and DR command. However, when the LAB or RR (reread) bits are set, a mode of operation is forced which uses only the L- and M-registers as shown in Fig. 26. When WHAL points to the L-register, a DP command will load the L-register and toggle WHAL to point at the M-register. A second DP command loads the M-register and points WHAL to the L-register. By performing two consecutive DR orders, the contents of the L- and M-registers will be returned to the 3A processor.

S. Deep Loop-Around from ESS

3.22 A deep loop-around from the ESS combines a buffer write, a Q to R buffer-to-buffer transfer, and a buffer read as shown in Fig. 27. This operation involves both the A and B Controllers. Control bit LAB must be set.

T. Deep Loop-Around from 3A Processor

3.23 A deep loop-around from the 3A processor combines multiple DP commands, R to Q buffer-to-buffer transfer and multiple DR commands as shown in Fig. 28. It requires two DP orders to load the J- (step 1) and K- (step 2) registers. Before the second DP is complete, both the J- and K-registers are loaded into the R buffer (step 3). Immediately the data in the R buffer is transferred to the Q buffer

under direction of Controller A. This data will not be placed in the N- and O-registers (step 6) until the following two DR orders (steps 4 and 5). Now, two more DR orders (steps 7 and 8) will return the original words to the 3A processor if the Q and R buffers were originally zeroed.

U. PIU Parity Structure

3.24 Internal to the PIU all data has parity over each group of 8 bits (ie, D0-D7, D8-D15, D16-D25, and D24-D31). The parity bits are identified as DP0, DP8, DP16, and DP24. When data is written in from the 3A processor, the parity is included. These parity bits are passed through the R buffer to the Controller A side of the PIU. Parity sent to the ESS has an overall parity bit, that is over 18 address bits, and 23 data bits. All parity is odd. This parity is generated when the data to be sent to the ESS is gated onto the D-bus. When the ESS sends data to the PIU, the address and overall parity bits are checked. If parity passes, data flows to the Q buffer. If parity fails, no data flows and the all seems well (ASW) is blocked. Parity is also checked on a write from the 3A. If parity fails, data flow to the R buffer is blocked.

4. UNIVERSAL DATA LINK CONTROLLER (UDLC) FUNCTIONAL DESCRIPTION

A. General

4.01 The UDLC provides the No. 1 ESS and the 3A processor with a general purpose data communication channel capable of handling 16 DLI units independent of transmission protocol. Figure 2 shows a block diagram of the data communication channel.

4.02 The UDLC provides a general purpose distributed controller which accepts data and instruction from the 3A processor and issues required sequences of commands to the appropriately selected DLIs. This architecture requires minimal software intervention, as well as meeting the following criteria:

- Control from 1 to 16 data links
- Half-duplex or full-duplex operation
- Synchronous data transmission
- Asynchronous data transmission
- Independent of transmission protocol
- Growth.

4.03 The following are features of UDLC:

- Full Duplication
- Self-Checking
- Multiplexing-Demultiplexing
- Consolidated Status Registers and Routine Polling
- DLI Enabling (common enable pointer)
- Programmable Command Field
- Maintenance Loop-Arounds.

4.04 The UDLC interfaces with the 3A processor subparallel channel via the DBS. The DBS is a standard unit currently available. It contains both cable drivers and receivers in addition to control circuits that respond to specific types of commands. This permits the UDLC to communicate on either bus to either 3A processor or can be quarantined from both buses if a failure occurs. By utilizing the DBS, the UDLC can be located as much as 60 feet away from the 3A processor.

4.05 The UDLC system hardware basically consists of a device address decoder, parity and bus sequencer circuits, subparallel channel response leads, programmable registers and decoder, a number of DLI activity registers, multiplexer-demultiplexer, and a microcontroller, connected by a central data bus. Figure 29 contains a functional block diagram of the UDLC. Figure 30 contains a similar block diagram except drawn on a functional circuit pack level, and Fig. 31 lists the circuit packs and their functions.

4.06 Control and data communications between the UDLC and the individual DLIs are accomplished over a dc bus complex. The bus structure can be divided into three subbuses: the enable bus, command and reply bus, and a multiplexed bidirectional data access bus (DAB). Figure 32 shows the bus leads in more detail. The enable bus, reply and status leads consist of dedicated signal pairs for each DLI. The command bus and the DAB are parallel buses shared by all DLIs. For reliability purposes, the UDLC and its dc bus are duplicated. Each DLI has two sets of

bus drivers and receivers permitting access by either UDLC. Only one UDLC, however, will be active at any given moment. Figure 33 contains a significantly more detailed functional block diagram of the UDLC. The principal components are described in the following paragraphs.

B. Address Decoder

4.07 A single duplicated and matched address decoder circuit responds to two unique address combinations selected by address leads ADR(0) through ADR(4). An odd parity check is performed against bit ADR(5) to verify the integrity of the address received. If a parity failure on the address information is detected, no device action will occur. A failure is detected by the 3A processor as a result of an instruction timeout. Two unique addresses are required to identify data and commands intended specifically for the UDLC versus data transactions intended for a DLI unit. The DLI address permits the 3A processor to communicate with all 16 possible positions. The method used to select the appropriate DLI is described in the DLI part of this section in the line control circuit part.

C. Input Control Leads

4.08 Five control leads are employed to transmit commands from the 3A processor to either the UDLC or a data link using a one-out-of-six code. These leads are standard 3A processor commands: command present (CP), data present (DP), data request (DR), sense status (SST), and acknowledge interrupt (ACKI). The definition of these leads is as follows:

CP — The device selected is told that a command is present on the information leads while CP is set to "one." The information leads contain a 15-bit coded command on INF(1) through INF(15). Only the device with the proper address should respond to the command. INF(0) is reserved for a channel maintenance function while CP is set.

DP — When DP is set to "one," the device selected is told that data is present on the information leads.

DR — When DR is set to "one," the device selected is requested to gate data onto the information leads.

SST - When SST is set to "one," the device selected is requested to gate the contents of its status register onto the high ten bus information leads (INF bits 6 through 15), as well as its address (ADR) code onto the low six bus information leads (INF bits 0 through 5). Proper parity must be generated. An SST command must be accepted and properly responded to by the addressed device regardless of its busy status.

ACKI - This lead is used to identify the devices that are requesting interrupts. When ACKI is set to "one," each device requesting interrupt should set a predefined bit of the information lead to "one," and all other devices should reset their predefined bit to "zero." The processor can then identify the devices that are requesting interrupts by locating the "one" bits of the information leads. ACKI feature is not being used because the 3A processor polling method has been selected instead of using interrupts. Both methods cannot be simultaneously supported.

D. Response Leads

4.09 Four response leads are provided back to the 3A processor. These leads are used to transmit information that is developing within the peripheral device selected. The four leads are: synchronization (SYNC), BUSY, error (ER), and interrupt (INTP). They are defined as follows:

SYNC — After a control signal has been issued by the 3A processor and properly received by the device, the device sets SYNC to "one" to indicate that the control signal has been understood, all device check circuits have been settled, and information has been gated onto or from the information leads. In case the device is not ready, the device should hold SYNC to "zero" until it has responded properly. The control signal and the data on the information leads are held steady until the device has

properly responded to the control signal by setting SYNC. SYNC is released by the device (ie, set to "zero") when all control signals are reset to "zero."

BUSY — BUSY is a flag from the device selected by the ADR leads which is available to be tested directly by the processor. The BUSY signal is used by the device to inform the processor of its readiness to respond immediately to a subsequent DP or DR control signal during block data transfer. The BUSY lead is asserted at the same time and fundamentally by the same control circuit as SYNC.

ER — ER is set on or before the time that SYNC is set, whenever the device selected by the ADR leads has discovered an abnormal condition. ER should then continue to be held on as long as the device senses its address. All errors detected by the addressed device should be ORed and reported on the ER line.

INTP — INTP is set to one when any device on the bus (whether it is addressed or not) wishes to interrupt the processor. The interrupt is reset by a special UDLC CP command. Although this function is not being used as noted previously, the interrupt and interrupt response, ACKI, were originally designed into the circuit and could be added easily when required.

E. Select Register and Select Decoders

4.10 The Select Register and Select Decoders are the key to all UDLC operations by providing access to other internal registers, enabling maintenance tests, stimulating errors, and selecting the set of ROMs with the appropriate microcode. Figure 34 contains the definitions of all UDLC registers. Figure 35 shows the layout of the select register that drives the three select decoders. Figures 36 and 37 contain tables defining the input combinations to the decoder and their output functions.

4.11 The least six significant bits of the select register are not used for two reasons. First, on the PDSP parallel channel applications, information bit INFO is reserved for a channel maintenance function on all CP orders. Secondly, to design the UDLC with the ability of being used on either the parallel or serial channels, the least six bits are defined as the Sequencer Control (SCTL) field. The SCTL field provides a sequence code generator within the operation of the serial channel.

4.12 The UDLC CP instruction is dedicated to the function of reprogramming the select register by overwriting the previous contents with the new data.

F. Line Control Circuit

4.13 The line control circuit performs two functions: it selects the DLI to be enabled and provides a field from which DLI commands are generated. This circuit consists of a Line Control Register (LCR) and a DLI Enable Decoder driven by the LCR. The LCR is written into by the 3A processor, using a UDLC DP command, after first programming the select register for access as described previously. Figure 38 contains the layout of the LCR. The LCR is divided into three segments: enable address, command field, and block transfer length. The 4-bit binary enable address is decoded by the DLI Enable Decoder and identifies which of 16 possible data link units is to be accessed. For reliability, this circuit is duplicated and matched to prevent inadvertently enabling the incorrect DLI. A dedicated signal pair connects each DLI unit to the enable decoder outputs. The command field consists of four fixed command bits and seven programmable command bits that are defined as required according to the protocol or the DLI design.

4.14 The block transfer bits were originally intended to be used with the future block transfer feature of the 3A processor. The capability of expanding the command field by using the block transfer bits as additional programmable commands has also been provided for future applications. Figure 39 contains detailed information on specific bit functions.

G. SST Register

4.15 The Send Status (SST) register is used to trap errors in both the UDLC and the particular DLI that was being communicated with at the time the fault occurred. The register also returns the device address on information leads INF0 through INF5. The returned address is checked by the 3A processor to verify proper address decoder operation as well as to verify that the correct peripheral device

responded. Figure 40 shows the format of the SST register and defines the meaning of each individual error bit.

H. DLI Registers

4.16 The UDLC contains additional sets of registers used to indicate the general status of each DLI and if any immediate action is being requested by a DLI. One bit in each of the 16-bit registers is assigned on a permanent basis to each DLI. These bits are asynchronously set and reset under the control of individual DLIs with one exception. The interrupt register can only be reset by a special command. Figure 41 shows the layout of all the fixed assignment registers. The registers are defined as follows:

(a) TRANSMIT/BUSY register: This register is set as soon as data transmission in the DLI begins.

(b) Data available: A bit, set to a logical one, indicates that the receive FIFO pointer has reached the half-full threshold and data is available to be read from that DLI.

(c) Interrupt register: A programmable interrupt register is provided for those conditions in which immediate intervention is required. The interrupt register outputs are ORed together to generate the single interrupt request back to the 3A processor. The interrupt sequence has been described by the ACKI and INTP definitions. Once the proper DBS which is generating the interrupt has been identified, a UDLC DR command must be transmitted to interrogate the interrupt register and determine which DLI requested an interrupt. As already noted in the INTP definition, the contents of the interrupt register can only be reset under a special UDLC CP command.

(d) Interrupt mask register: Since it may not always be desirable to respond to a DLI requested interrupt due to a fault or scheduling priorities, a mask register is provided. This register is loaded with the appropriate information by a UDLC DP command. Functionally, the mask and interrupt registers are ANDed together for the proper control. To respond to a specific DLI interrupt request, the mask bit assigned to that device is set to a logical one. To ignore that interrupt request, the appropriate bit is reset.

4.17 All four sets of DLI activity registers can be independently interrogated by first issuing a UDLC CP command followed by a UDLC DR command. The CP command reprograms the select register to identify the register to be accessed, and the DR instruction gates the reply data onto the information leads.

I. Multiplexer/Demultiplexer

4.18 The data bus structure employed between UDLC and the DLI consists of the 9-bit parallel DAB. Each 16-bit word transmitted by the 3A processor to the UDLC is divided into two 8-bit words and demultiplexed out to the appropriate DLI. Similarly, two DLI words must be multiplexed to form a complete data word. As an integrity check on the data transmission between devices, parity is computed and accompanies the data bits. On the data transmitted out to the DLI, the parity bit accompanying the first 8-bit word is simply the low order parity bit from the 16-bit 3A processor data word. For the second 8-bit DLI word, the high order parity bit is used. The converse is true on data returned from the DLI to the 3A processor except parity is generated by the DLI. Figure 42 shows this data format. Data transmitted from the UDLC is checked by the DLI for proper received parity on each word. A positive acknowledgment reply must be generated back to the UDLC after receiving each word to indicate if parity has passed. If no positive reply is received, then either a parity failure or a device failure has occurred.

4.19 The UDLC may also be programmed into the half-word mode of operation. In this instance, only the lower eight data bits are used with parity for DLI DP and DR instructions. UDLC instructions utilize all bits.

J. Microcontroller

4.20 The microcontroller is a high speed system state controller consisting of a stand-alone ROM and a microprogram control circuit that sequences at a 320 nanoseconds (ns) execution time. It is used to control the repetitive processes of communicating with a DLI. The functions that the microcontroller performs are essentially the same as a wired logic system but offers exceptional reliability, space savings, plus flexibility. Figure 43 shows a functional block diagram of the microcontroller, which is duplicated and matched. The microcontroller consists of a programmable clock, ROM address counter, scan element, race resolving circuit, ROM with parity checker, and a matching circuit to verify that the duplicated circuits are in step

and no fault has occurred. No arithmetic unit is required.

4.21 ROM contains the microcoded instructions used in administering operations to a DLI including bus signaling protocol, maintenance tests, and normal data transactions. The ROM consists of 20 bits by 256 words. It may be expanded in increments of 256 words individually addressed by the Select Decoder A, with each set containing new microcoded sequences permitting communications with up to eight types of interface units. The microinstruction format, shown in Fig. 44, is divided into eight control fields. This wide structure provides savings in the depth of the microinstruction words necessary to accomplish a task and reduces the amount of time required to perform the particular operation. Figures 45, 46, and 47 contain detailed definitions and tables of the ROM output functions.

4.22 The microcontroller assumes control whenever a DLI addressed command is detected by the UDLC. The hardware selects the initial ROM location to begin execution based on the type of command received and loads the address into the ROM address counter circuit. Inputs to the microcontroller are monitored by the scanner module. Figure 48 contains a table identifying the scan inputs and their addresses. Based on the condition of the inputs, decisions are made to either increment the address counter or branch to a different location. In this manner, transfers are made to microcoded routines that determine the nature of the situation and take appropriate action.

4.23 The race resolving circuit is used to synchronize the two duplicated circuits when the scan element is enabled. Device speed differences and the asynchronous nature of the reply inputs could cause one circuit to respond immediately while the duplicated half requires an additional clock cycle. The programmable clock circuit determines the length of time the microcontroller will take to complete the microinstruction, particularly for scan operations. When a reply is received, however, the programmed delay will terminate immediately to save real time and permit the next microinstruction to begin.

5. DATA LINK INTERFACE (DLI) FUNCTIONAL DESCRIPTION

A. General

5.01 The DLI circuit provides an interface compatible with both the data line control requirements and the UDLC. Figure 49 is a block diagram of the DLI showing the interconnections with the FIFO data buffers and the universal synchronous asynchronous receiver transmitter (USART). The USART performs the parallel-to-serial and serial-to-parallel conversions.

5.02 The DLI consists of two circuit boards, FG4 and FG5. FG4 contains the write first-in first-out (FIFO) buffer with FIFO pointers, parity checker/generator, bus 0 drivers and receivers, and the internal bus control circuit. Circuit board FG5 contains the USART, electronic industry association (EIA) interface, USART control with digital loop-around features, status register, read FIFO buffer with FIFO pointer, and bus 1 driver and receivers.

B. DLI Features

5.03 The following is a list of features of DLI:

- EIA Interface
- USART
- Data Buffering—FIFO
- Parity Check
- Line Status Register
- Digital Loop-Around (Level Two).

C. Bus Circuit

5.04 The DLI contains two bus circuits, 0 and 1, from the two respective UDLCs permitting communications from either controller. Each bus is divided into a 9-bit command field, four reply bits, and a 9-bit bidirectional DAB. Table A lists the command bus bits and definitions with respect to this DLI implementation.

D. Parity Check and Generation

5.05 The parity checker tests the integrity of the data being received on the DAB. Parity is recomputed over the eight data bits and matched against the parity bit received. After passing parity, the data is loaded into the input buffer as well as any other device intended, and a positive acknowledgment is returned to the UDLC. For all data reads, the parity bit is generated by the DLI and accompanies the data to the 3A processor.

E. Input Buffer and First Maintenance Loop-Around

5.06 The input buffer is a single 8-bit register. It is loaded with the data presently being written from the UDLC to the DLI. The data remains in the

input buffer until overwritten by a new word. On normal write operations the data word would also be loaded into the transmit FIFO or USART depending on type of command. This facilitates the first-level maintenance loop-around. If the write command is identified as a first-level maintenance loop-around operation, the data is loaded into the input buffer only. By issuing a first-level maintenance loop-around read command, the data is read out of the input buffer onto the common internal DLI bus, parity is generated, and the data is returned to the 3A processor. This permits testing of the bus between the UDLC and DLI, checks the bus receivers and driver, verifies proper parity operation, and partially checks the DLI control circuit. Additionally, this test can be performed on an active DLI without interfering with normal data transmission.

F. FIFO and FIFO Pointers

5.07 The DLI contains two FIFOs, one designated WRITE FIFO or transmit, the other READ FIFO or receive. Each is 8 bits by 64 words. A read and write FIFO pointer circuit, comprised of two separate sets of up/down counters, is used to indicate the number of words contained within the FIFOs. Both pointers can be interrogated by a single 3A processor command.

G. Control

5.08 The DLI control is implemented with wired logic. The control portion of the DLI administers access to the common D-bus in response to data and/or commands received from the UDLC. It provides administrative control timing over the USART for programming of the USART and to synchronize the transfer of data from the WRITE FIFO into the USART for transmission over the data set. Conversely, the transfer of incoming data from the USART to the READ FIFO is similarly administered. Since USART and USART-to-FIFO operation is asynchronous in nature with respect to 3A processor activities, the control also employs temporary lockouts to prevent simultaneous use of the D-bus.

H. USART

5.09 The Universal Synchronous/Asynchronous Receiver/Transmitter is a programmable microprocessor that provides the parallel-to-serial and serial-to-parallel conversions required for the serial data link communications. Once programmed into the transmit mode, the USART raises the transmitter-ready output lead. The control circuit transfers one data word from the WRITE FIFO into the USART where it is serialized. The USART's data set control section generates a request-to-send data command to the data set. In response to a clear-to-send data signal reply, transmission begins.

5.10 In the reverse direction, serial data is received from the data set if previously programmed into the receive mode of operation. After assembling an entire character, the receiver-ready output is raised and the completed character is fetched by the control circuit and stored in the READ FIFO.

I. EIA Interface and Digital Loop-Around

5.11 The EIA (RS232C) interface converts the TTL signal level to ±12 volt levels. The EIA connections and definitions are shown in Table B. Inserted between the EIA interface and the USART is a loop-around circuit activated by a second-level maintenance loop-around command. This inhibits data from being transmitted to the data set and permits the USART, operated in the full duplex mode, to loop around its transmitted data back into its receiver. This enables a full operational test when the data contained in the READ FIFO is matched against the data transmitted.

6. POWER

A. General

6.01 Power is supplied to the processor interface frame from the power distributing frame over both power buses A and B. A total of two −48 volt, two +24 volt, and two ground return feeders is required for each processor interface frame. Power and ground feeders are terminated at the top rear of the frame to cables provided in the frame uprights that are connected to the power filter circuit located in the base of the frame.

6.02 Only one filter unit is required for the frame. The filter is equipped with two +24 volt and two −48 volt filter circuits. The power from the filter unit is distributed to the power connector circuits located directly along the fuse panel unit.

6.03 A total of six +5 volt and two −12 volt power converters is required per PIF. Two of the +5 volt converters are required for each processor interface circuit. One +5 volt converter is required for each UDLC circuit. The two −12 volt converters supply power to the DLI circuits only. One 12V converter supplies DLI 0-7 while the other converter supplies DLI 8-15.

6.04 Power from the converters is routed to the fuse panel unit before being routed to the respective circuit packs in the frame. Also located on the fuse panel units are the power control and power sequence relays.

6.05 A power sequence technique is used to prevent bus faults which can occur if either the call store bus or SPCH bus drivers accidentally outpulse onto the bus during power-up or power-down operations. In order to prevent false outputs during the power-up operation, the power to the bus driver's circuit packs is applied only after power is up on all other circuit packs. For the power-down operations the reverse is true, and power is first removed from the bus driver packs.

B. Power Controls

6.06 A manual power control unit is provided for each processor interface circuit (see Fig. 2, 3, and 4) and permits power to be removed from either bus 0 or bus 1 during circuit pack replacement or other maintenance operations.

6.07 This power controls ***only*** affect the UDLCs and PIUs. Power will remain on the DLI units. The removal of power from DLIs is achieved by individual switches located on each circuit board. Both DLI circuit packs (FG4 and FG5) must be powered down before they are removed.

7. ABBREVIATIONS

7.01 Abbreviations used in this section are as follows:

ACKI—Acknowledge Interrupt

CC—Central Control

CNCC—Customer Network Control Center

CP—Command Present

CSACC—Customer Service Administrative Control Center

DAB—Data Access Bus

DBS—Duplex Bus Selector

DLI—Data Link Interface

DP—Data Present

DR—Data Request

EIA—Electronic Industry Association

ER—Error

EPSCS—Enhanced Private Switched Communication Service

ESS—Electronic Switched System

E911—Expanded 911 Service

FIFO—First-In First-Out

INTP—Interrupt

LAA—Loop Around A

LAB—Loop Around B

LCR—Line Control Register

NS—Nanoseconds

PCH—Parallel Channel

PDSP—Peripheral Data Storage Processor

PIF—Processor Interface Frame

PIU—Processor Interface Unit

RAM—Random Access Memory

ROM—Read Only Memory

RR—Reread

SCTL—Sequence Control

SST—Sense Status

SWSR—Status Word Select Register

SYNC—Synchronization

UDLC—Universal Data Link Control

USART—Univeral Synchronous Asynchronous Receiver Transmitter.

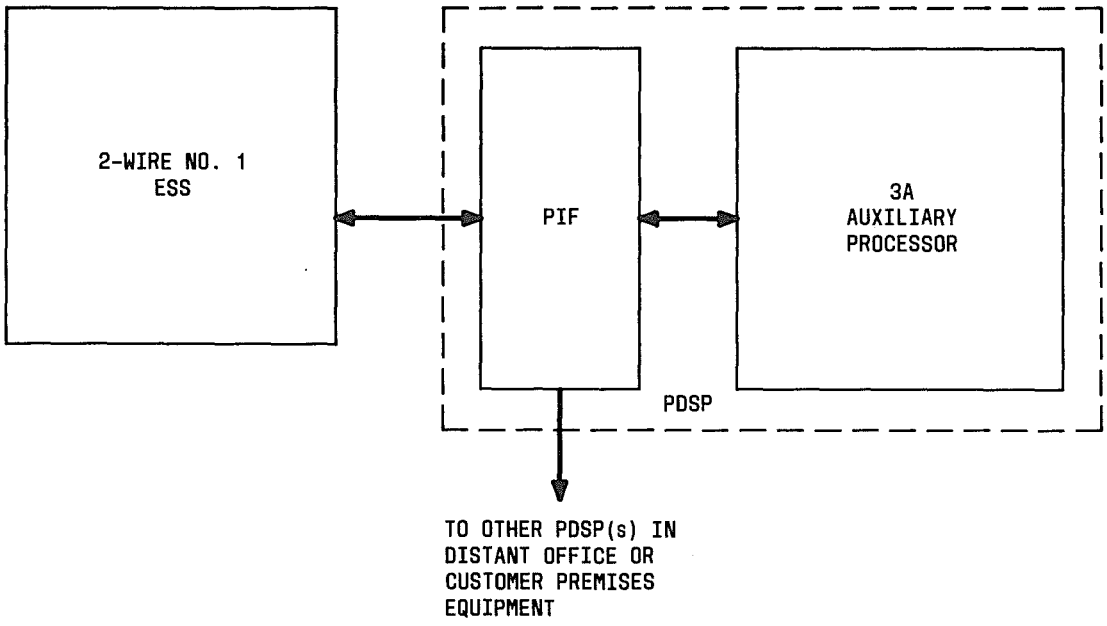


Fig. 1—PIF, ESS, and PDSP Configuration

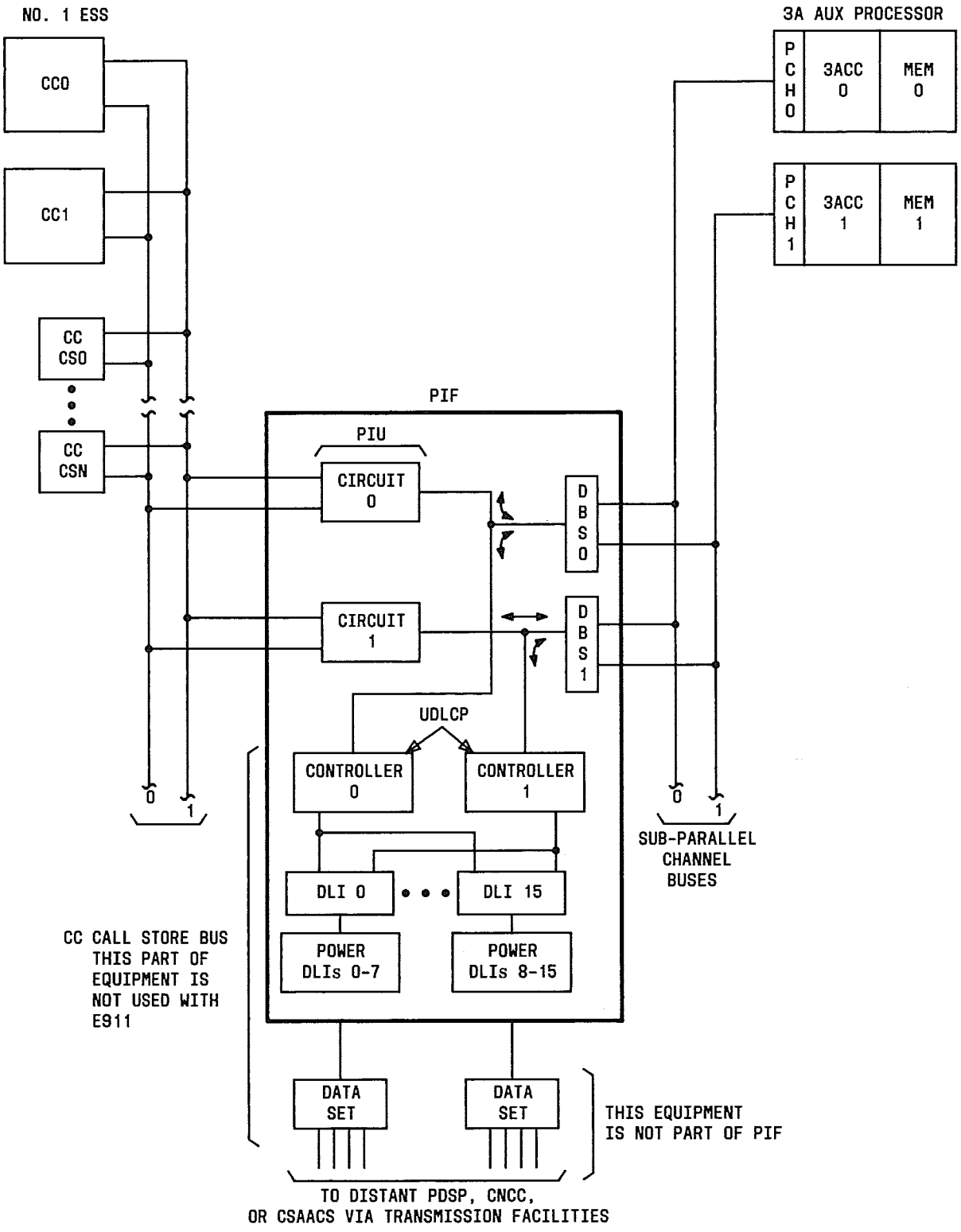
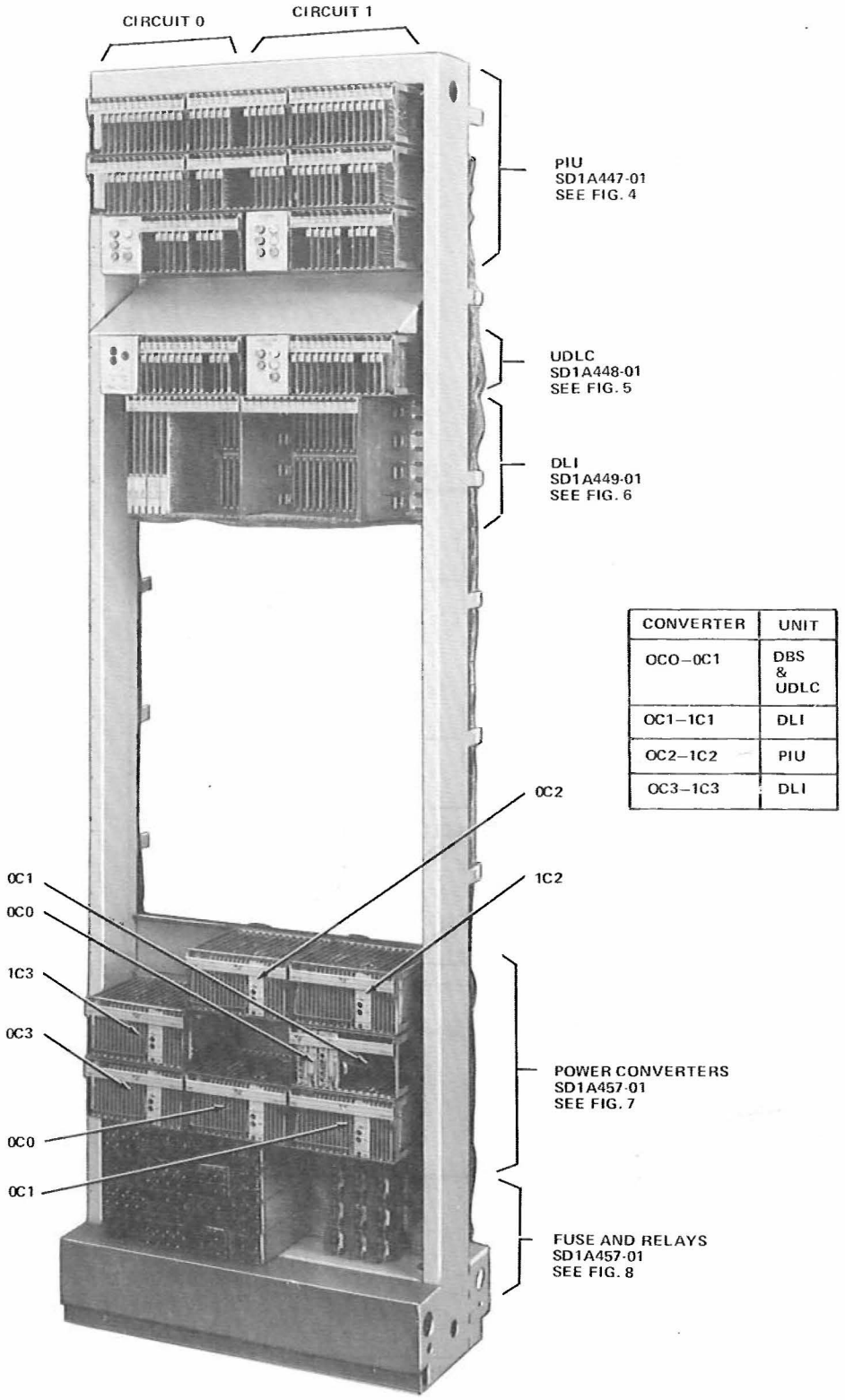


Fig. 2—No. 1 ESS PDSP Configuration for EPSCS and E-911



CONVERTER	UNIT
OC0–0C1	DBS & UDLC
OC1–1C1	DLI
OC2–1C2	PIU
OC3–1C3	DLI

Fig. 3—Processor Interface Frame

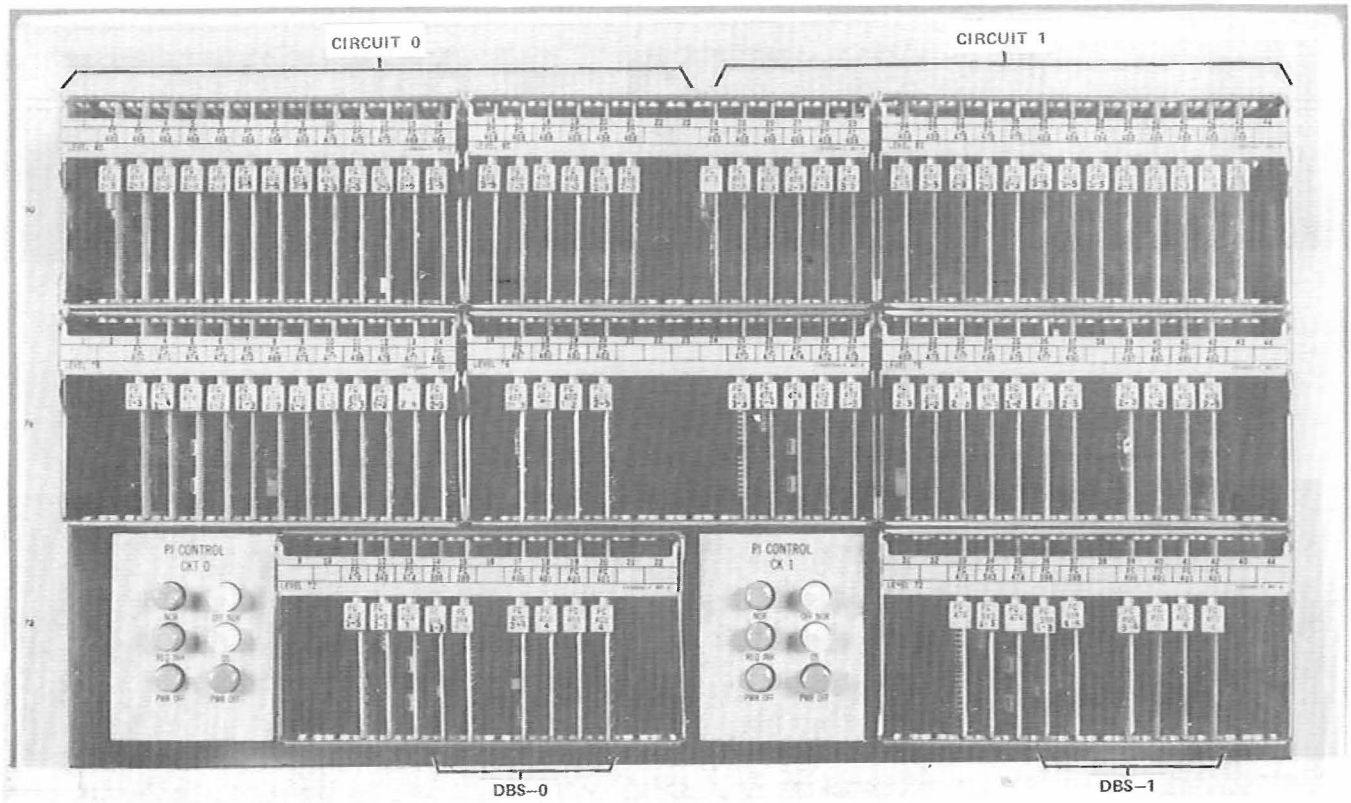


Fig. 4—Processor Interface Unit

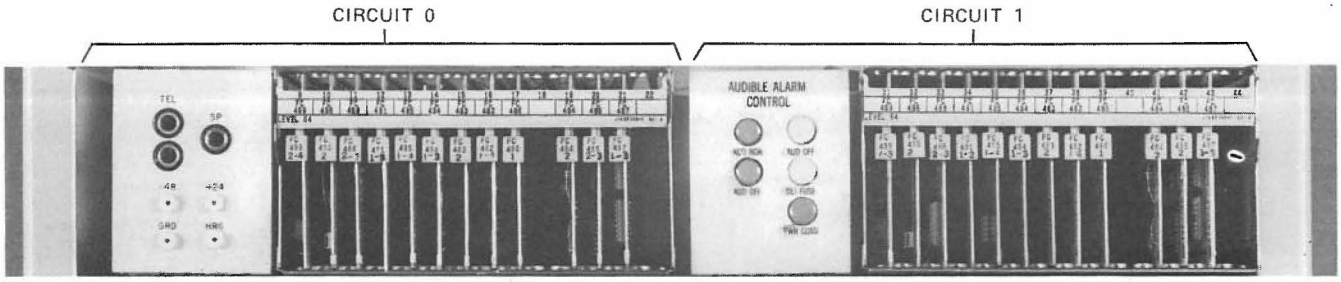


Fig. 5—Universal Data Link Control

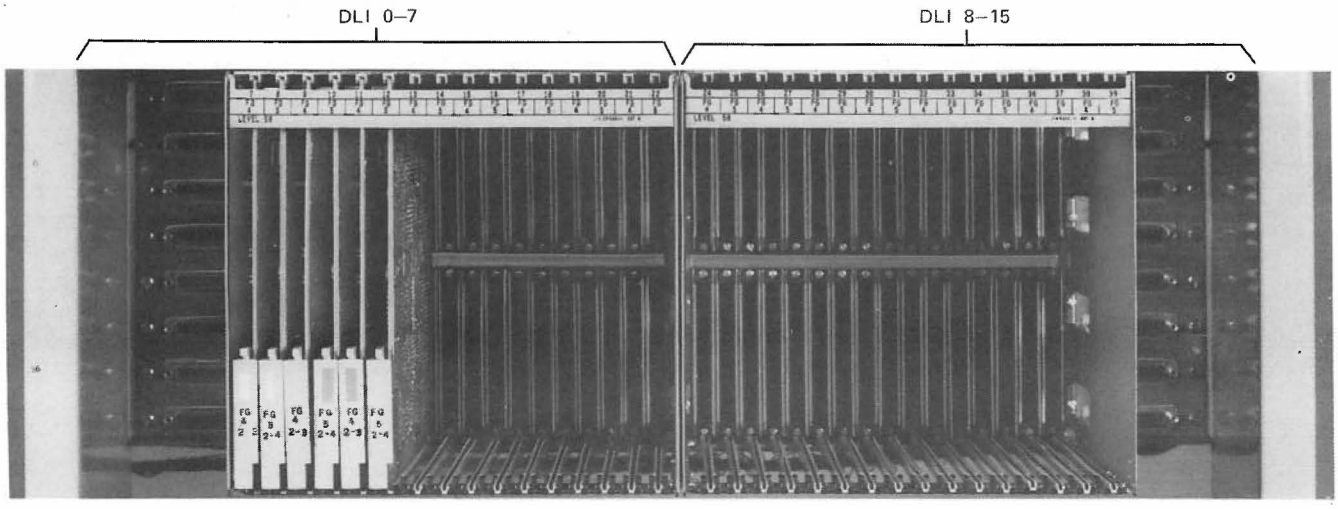


Fig. 6—Data Link Interface

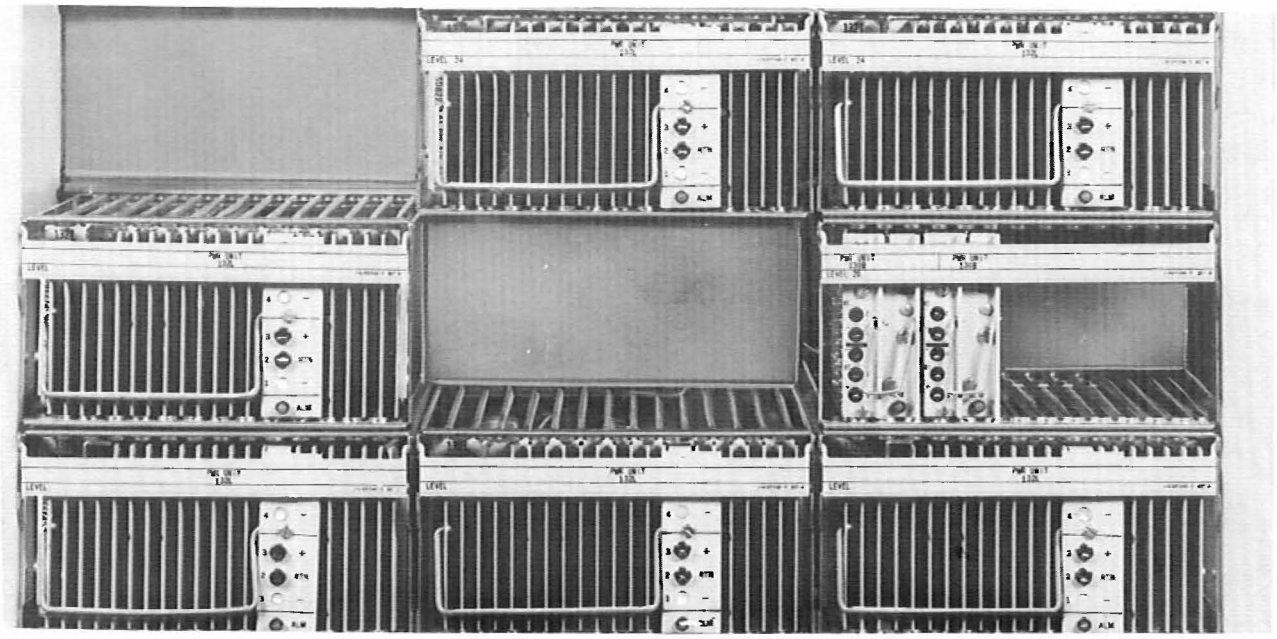


Fig. 7—Power Converters

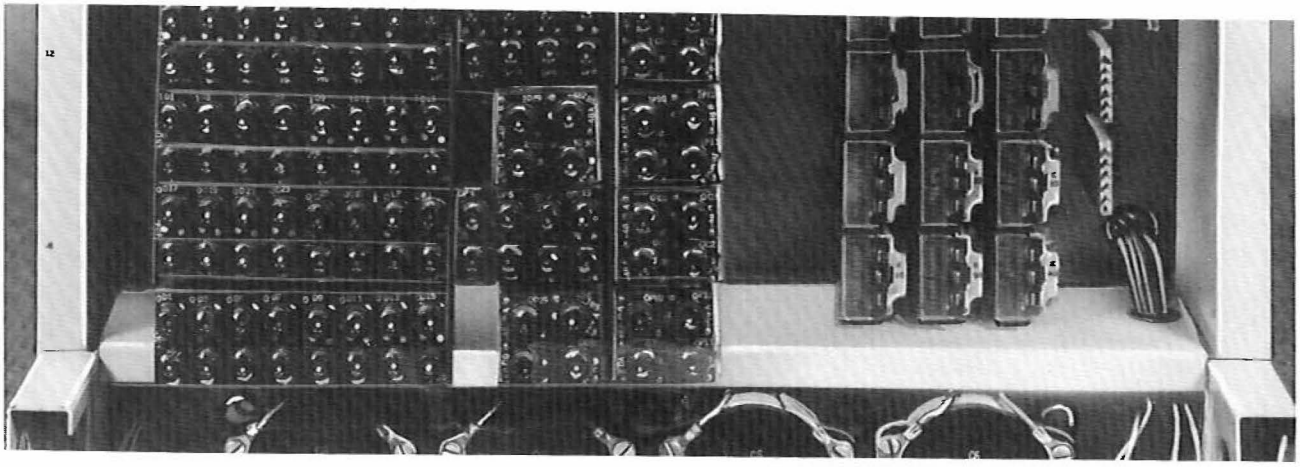
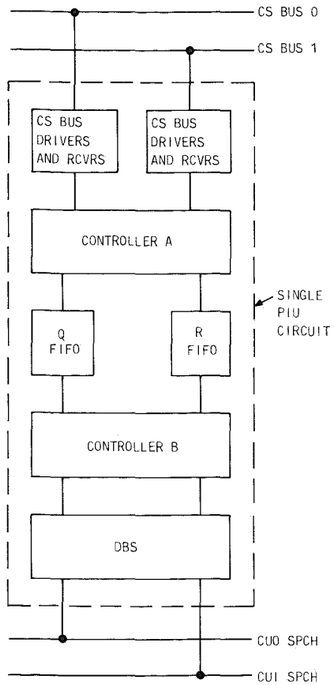


Fig. 8—Fuse and Relays



SIMPLIFIED BLOCK DIAGRAM

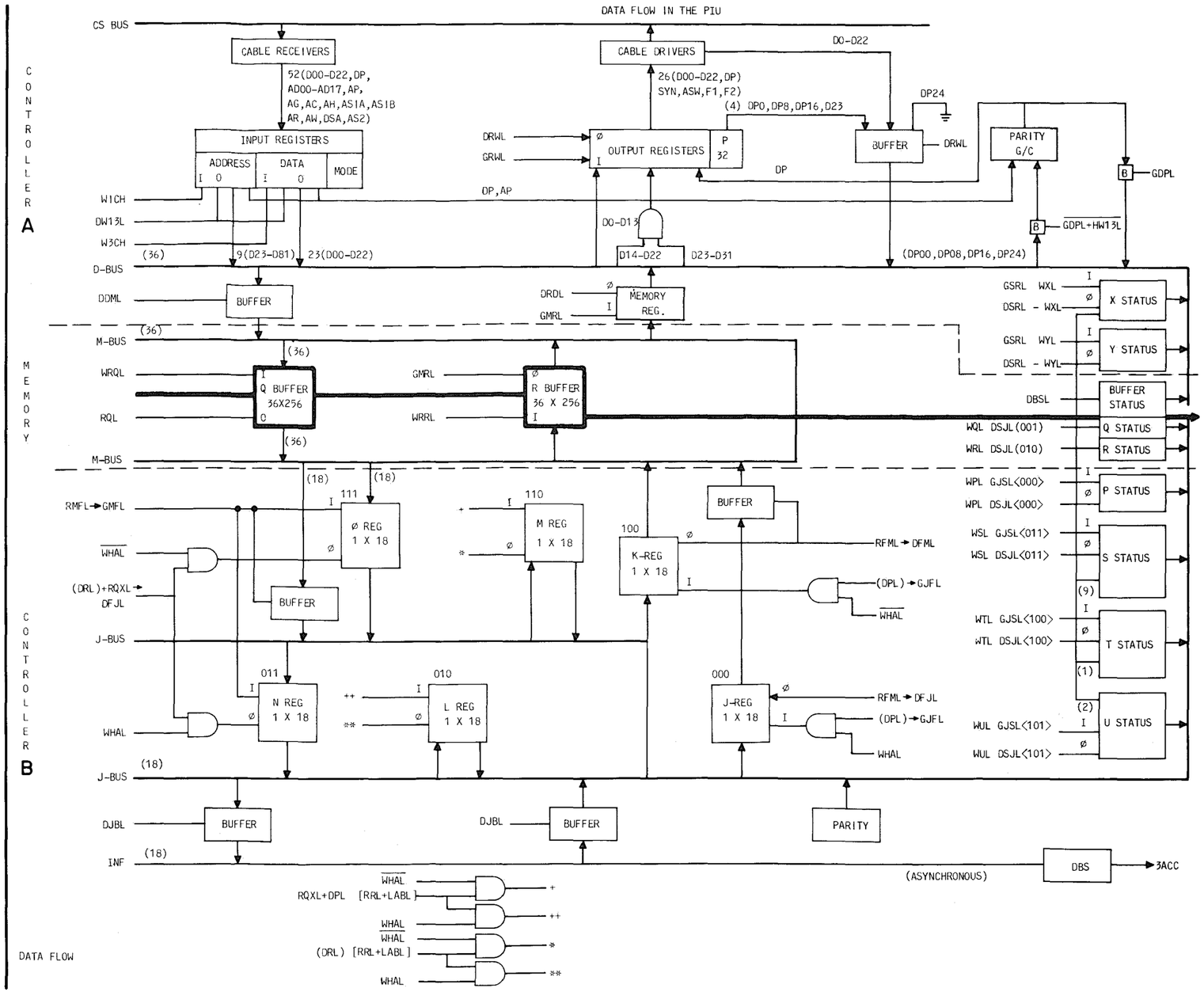


Fig. 9—PIU Simplified Block Diagram and Data Flow

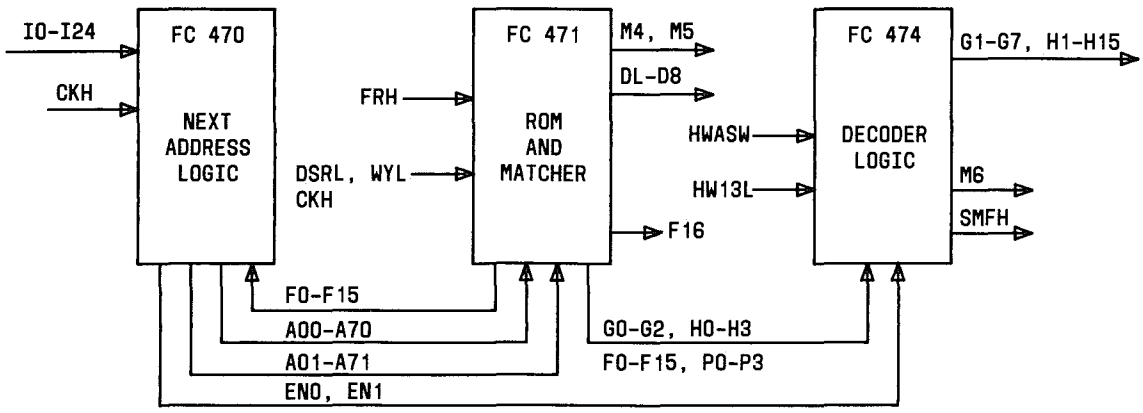
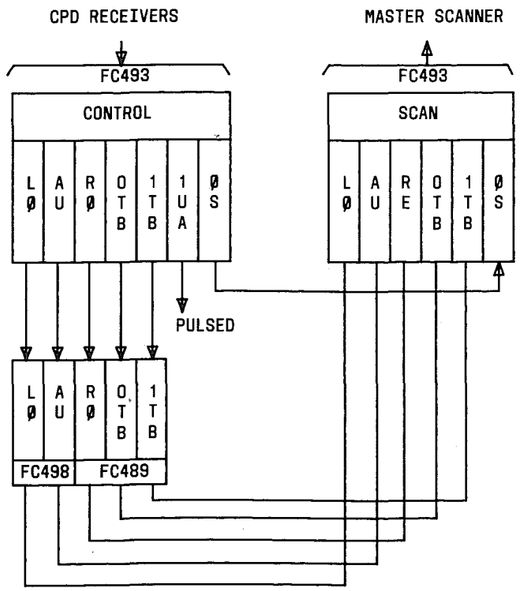
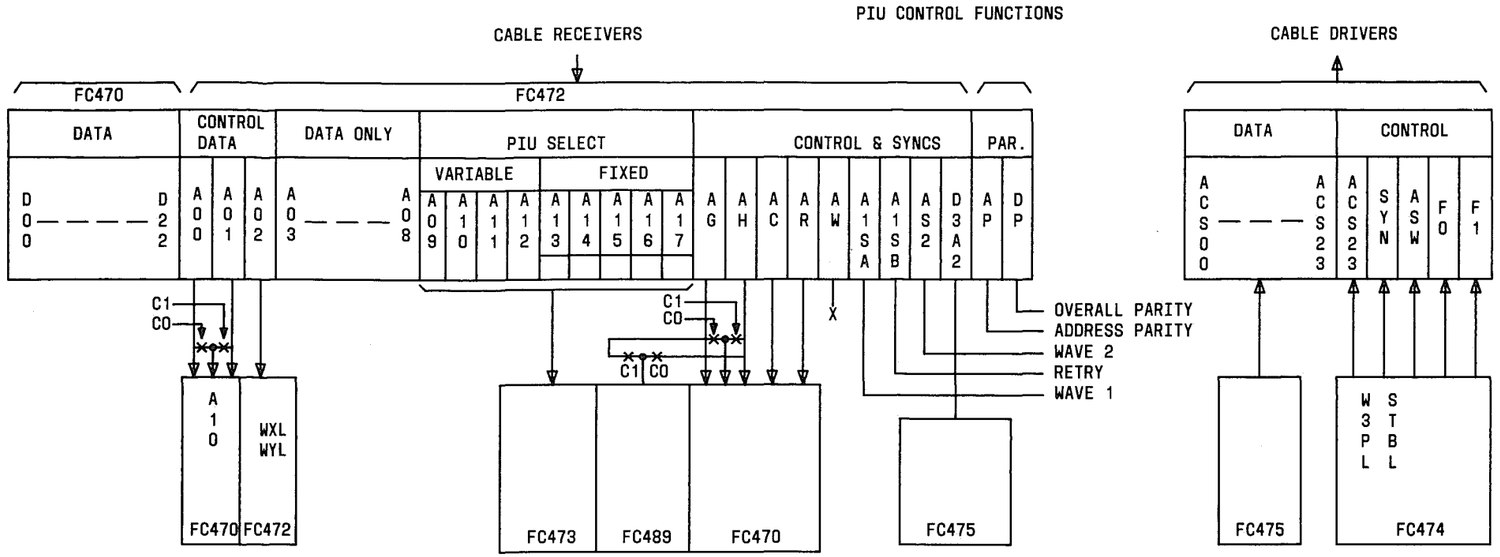


Fig. 10—Read Only Memory Microsequencer





AG	AC	AH	A2	A1	AO	AR	AW	FUNCTION	
0	0	0	0	0	0	1	0	NORMAL	FIFO READ
0	0	0	*	*	*	0	1	NORMAL	FIFO WRITE
0	0	0	1	0	1	1	0	NORMAL	BUFFER STATUS READ
1	1	0	0	0	0	1	0	0 MAINTENANCE	FIFO READ
1	1	0	*	*	*	0	1	0 MAINTENANCE	FIFO WRITE
1	1	0	1	0	1	1	0	0 MAINTENANCE	BUFFER STATUS READ
0	1	1	0	0	0	1	0	1 MAINTENANCE	FIFO READ
0	1	1	*	*	*	0	1	1 MAINTENANCE	FIFO WRITE
0	1	1	1	0	1	1	0	1 MAINTENANCE	BUFFER STATUS READ
1	0	1	1	0	1	1	0	CONTROL	CIRCUIT 0 READ X - STATUS WORD
1	0	1	1	0	1	0	1	CONTROL	CIRCUIT 0 WRITE X - STATUS WORD
1	0	1	0	0	1	1	0	CONTROL	CIRCUIT 0 READ Y - STATUS WORD
1	0	1	0	0	1	0	1	CONTROL	CIRCUIT 0 WRITE Y - STATUS WORD
1	0	1	1	1	0	1	0	CONTROL	CIRCUIT 1 READ X - STATUS WORD
1	0	1	1	1	0	0	1	CONTROL	CIRCUIT 1 WRITE X - STATUS WORD
1	0	1	0	1	0	1	0	CONTROL	CIRCUIT 1 READ Y - STATUS WORD
1	0	1	0	1	0	0	1	CONTROL	CIRCUIT 1 WRITE Y - STATUS WORD

* DO NOT CARE STATE

CO - CONTROLLER 0
C1 - CONTROLLER 1

Fig. 11—ESS-PIU Control Functions

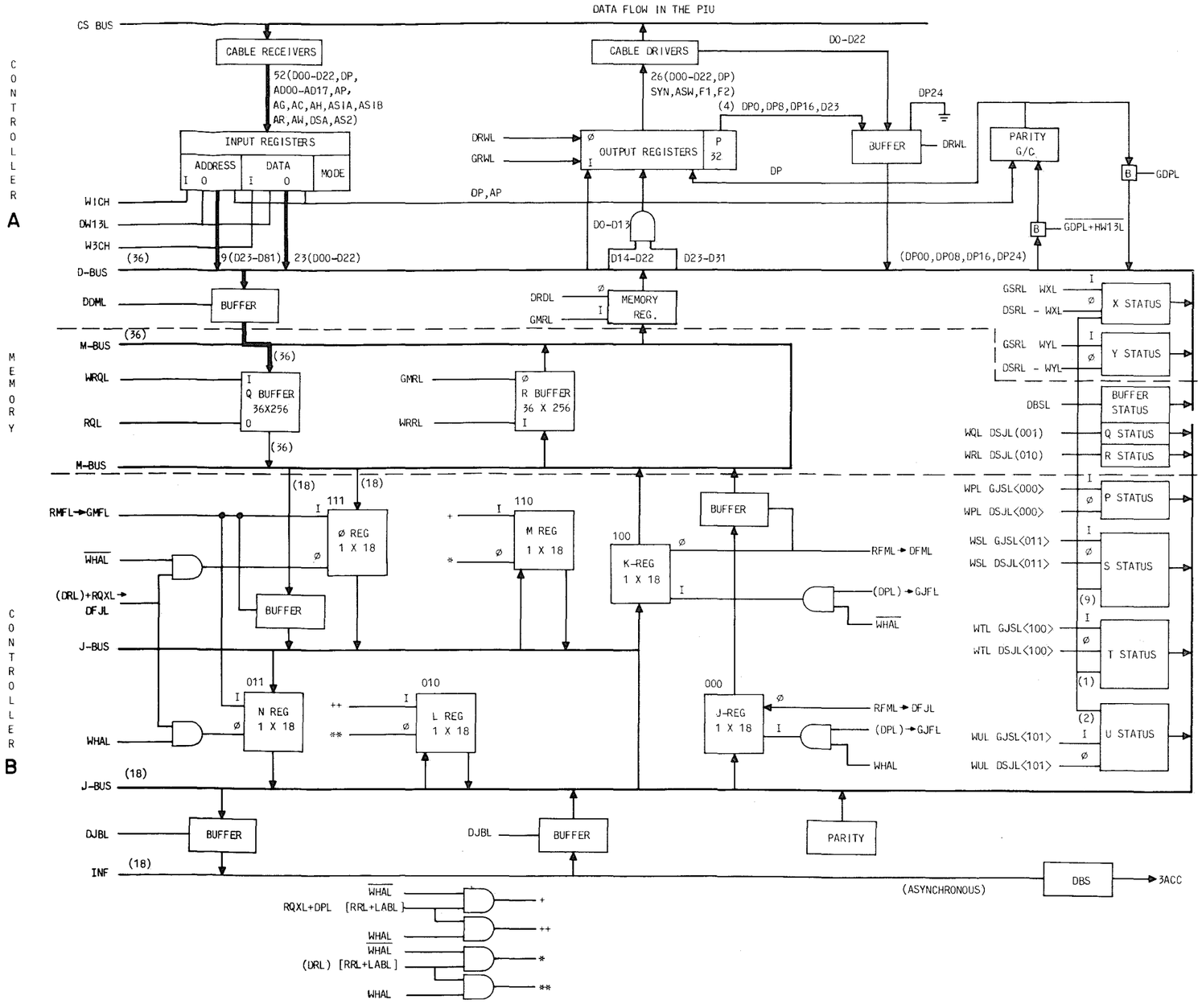


Fig. 12—Buffer Write

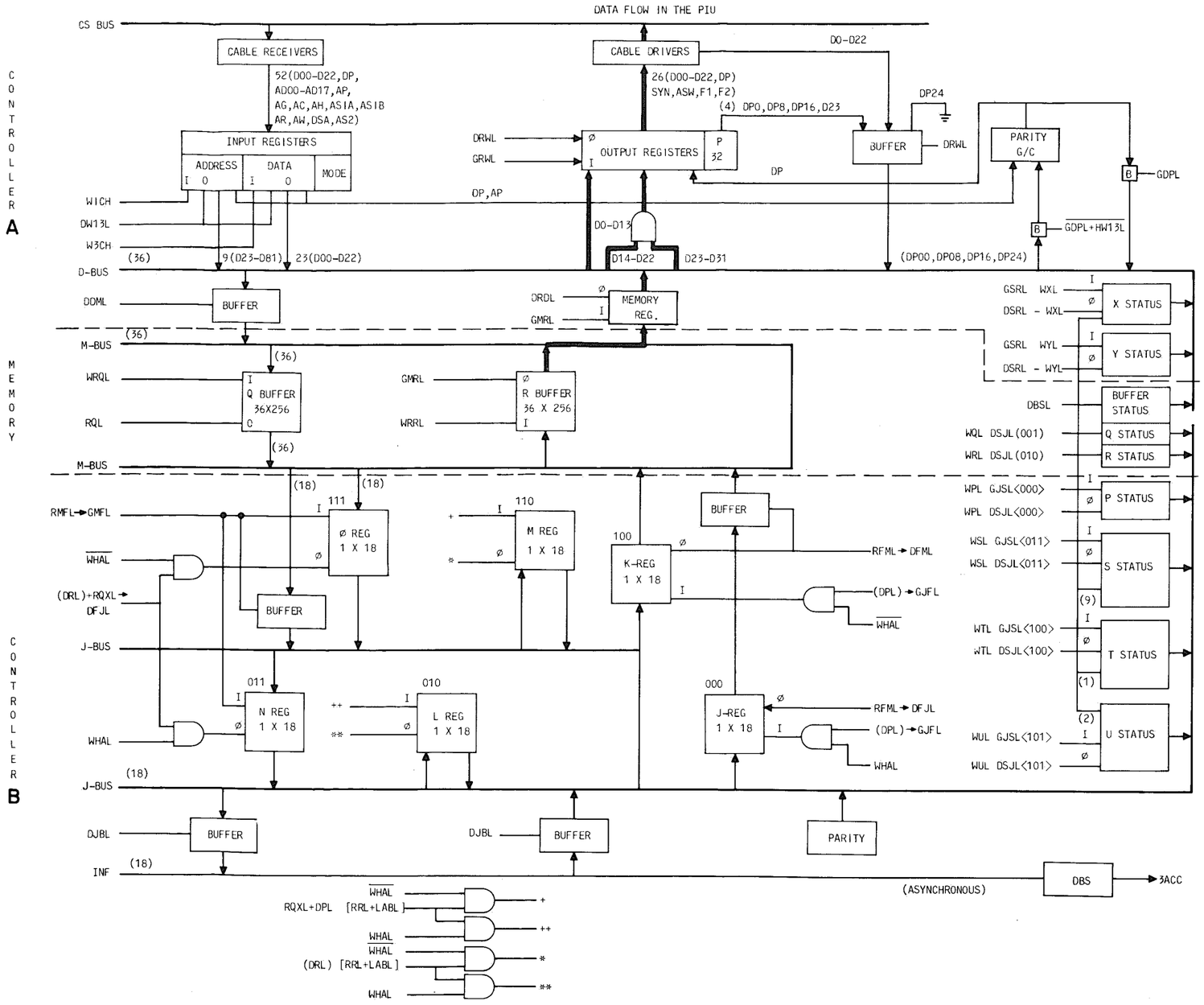


Fig. 13—Buffer Read

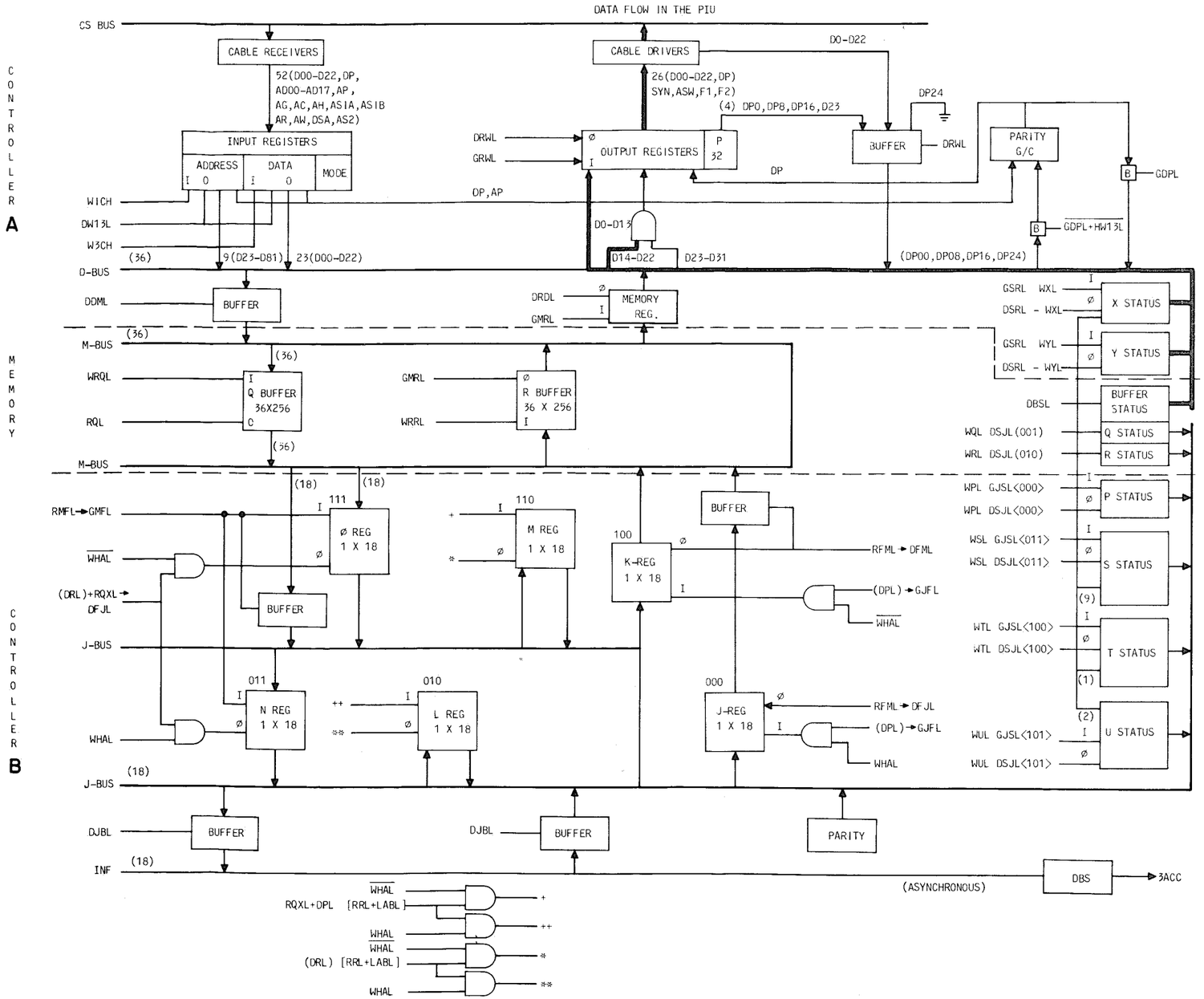


Fig. 14—Status Read from ESS

BIT	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X REGISTER	1					ROQ	IHER R3S	D6 R W3S	D5 R W3S	D4 R W3S	D3 R W3S	D2 R W3S	D1 R W3S	489 FPERA RZ	490 3TB R W3S	477 476 ZBC W *	489 S1 R W	489 S0 R W	489 IF3A R WS	489 IF PIU R W	498 LAB R WS	489 LAA R WS	489 IUA R WS
Y REGISTER	0										490 SF3A R W	490 SF2A R W	490 SF1A R W	490 SF0A R W	471 PERA 7 R	471 PERA 6 R	471 PERA 5 R	471 PERA 4 R	471 PERA 3 R	471 PERA 2 R	471 PERA 1 R	471 PERA 0 R	489 ISMA R Z
BUFFER STATUS WORD	477 BUR B C		477 64SPR B	477 BOR B D	477 NWBR 7 B	477 NWBR 6 B	477 NWBR 5 B	477 NWBR 4 B	477 NWBR 3 B	477 NWBR 2 B	477 NWBR 1 B	477 NWBR 0 B	477 64SPQ B	477 BOQ B C	477 BUQ B D	477 NWBQ 7 B	477 NWBQ 6 B	477 NWBQ 5 B	477 NWBQ 4 B	477 NWBQ 3 B	477 NWBQ 2 B	477 NWBQ 1 B	477 NWBQ 0 B

- 470 → FC CIRCUIT PACK CODE ON WHICH THE FUNCTION IS LOCATED
- XX → FUNCTION
- R → NON-MEMORY READ CAPABILITY
- W → NON-MEMORY WRITE CAPABILITY
- 3 → 3A CC HAS WRITE ACCESS
- S → 3A CC HAS READ ACCESS
- Z → WRITING A ZERO INTO BIT ZEROS THE BIT, A ONE DOES NOTHING
- * → WRITING A ONE INTO ZBC ZEROS ALL THE BITS IN THE BUFFER STATUS WORD. ZBC READS AS A ZERO.
- B → MEMORY READ CAPABILITY
- C → ESS BUFFER STATUS WORD READ ZEROS THIS BIT
- D → 3ACC CAN ZERO
- ▨ → NOT USED BITS

Fig. 15—Controller A—Buffer Status and Status Words

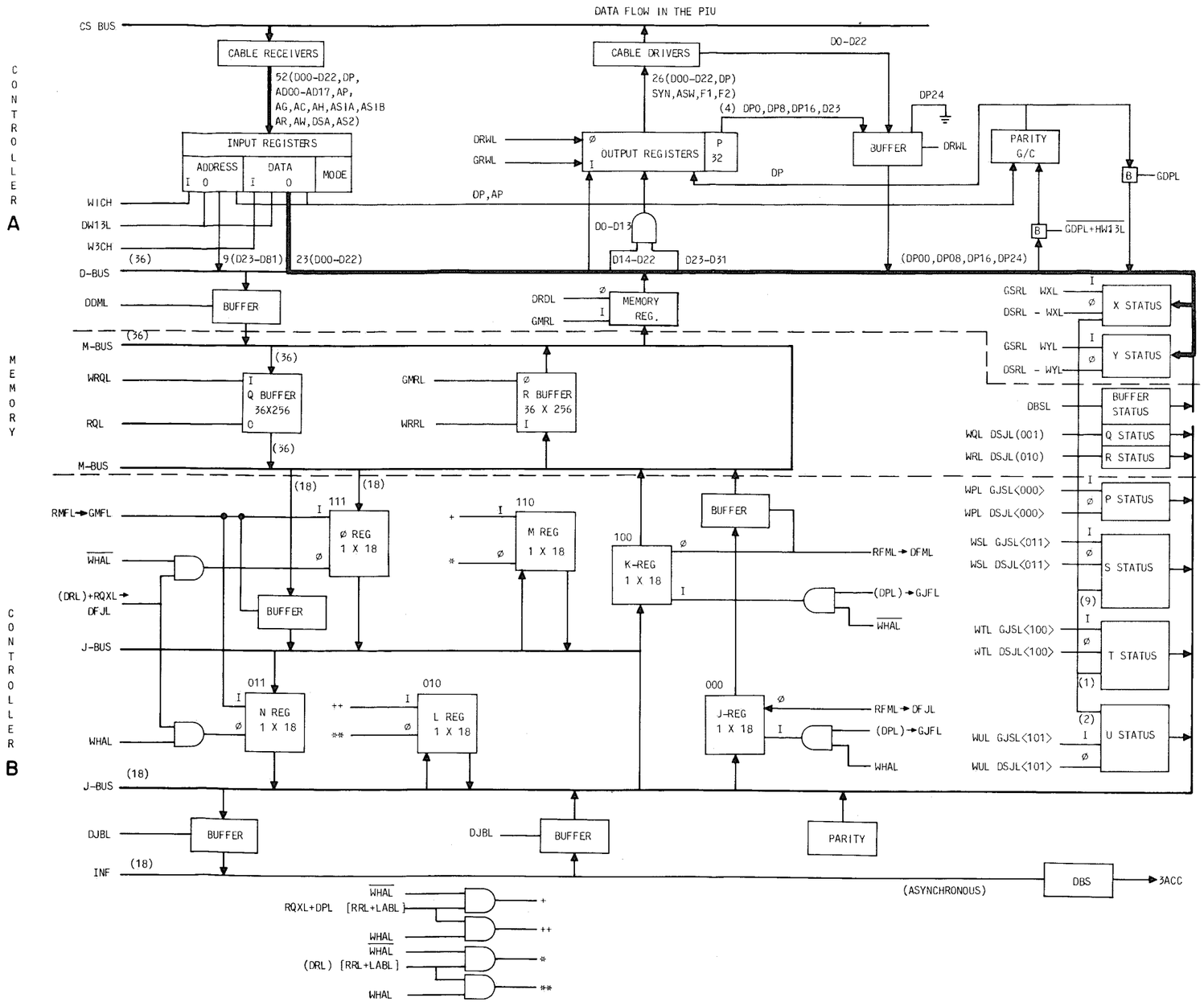
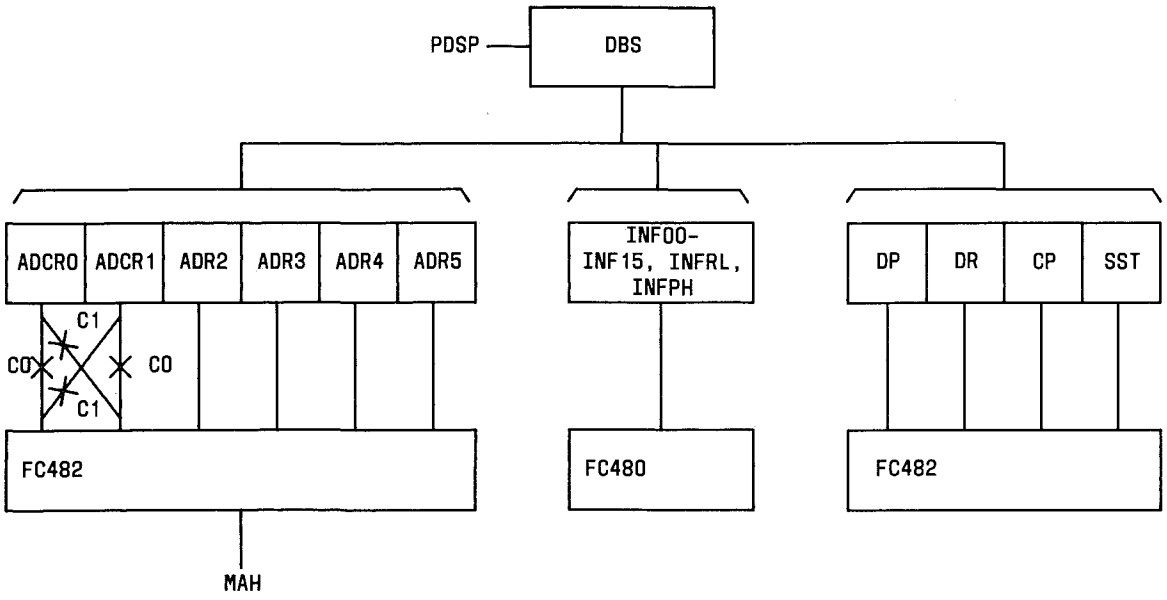


Fig. 16—Status Write from ESS



DP	DR	CP	SST	ADR0	ADR1	FUNCTION	
1	0	0	0	1	1	NORMAL	FIFO WRITE
0	1	0	0	1	1	NORMAL	FIFO READ
0	0	1	0	1	1	NORMAL	STATUS WRITE
0	0	0	1	1	1	NORMAL	STATUS READ
1	0	0	0	1	0	MAINTENANCE 0	FIFO WRITE
0	1	0	0	1	0	MAINTENANCE 0	FIFO READ
0	0	1	0	1	0	MAINTENANCE 0	STATUS WRITE
0	0	0	1	1	0	MAINTENANCE 0	STATUS READ
1	0	0	0	0	1	MAINTENANCE 1	FIFO WRITE
0	1	0	0	0	1	MAINTENANCE 1	FIFO READ
0	0	1	0	0	1	MAINTENANCE 1	STATUS WRITE
0	0	0	1	0	1	MAINTENANCE 1	STATUS READ

Fig. 17—PDSP-PIU Control Functions

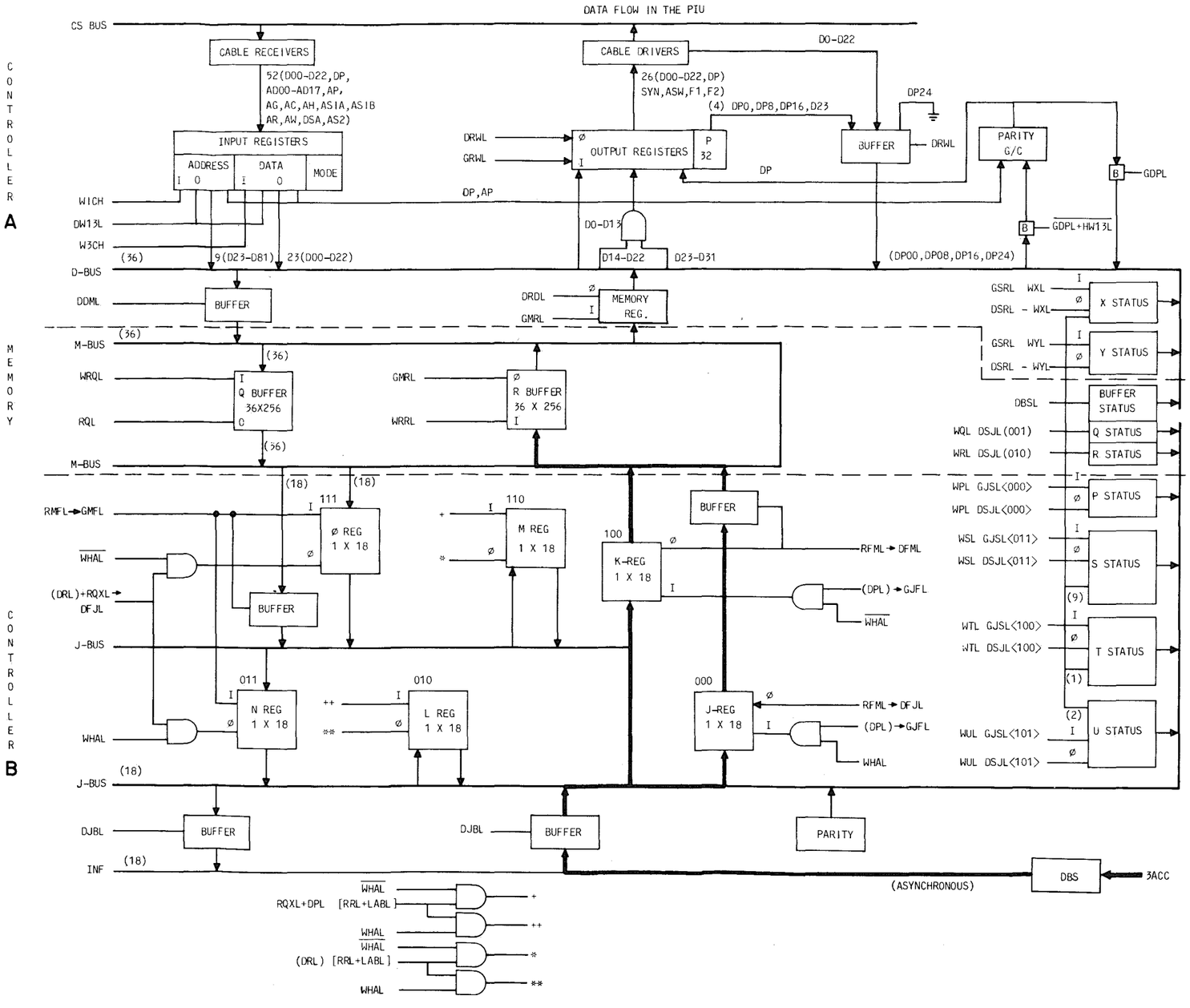


Fig. 18—PDSP Buffer Write (DP)

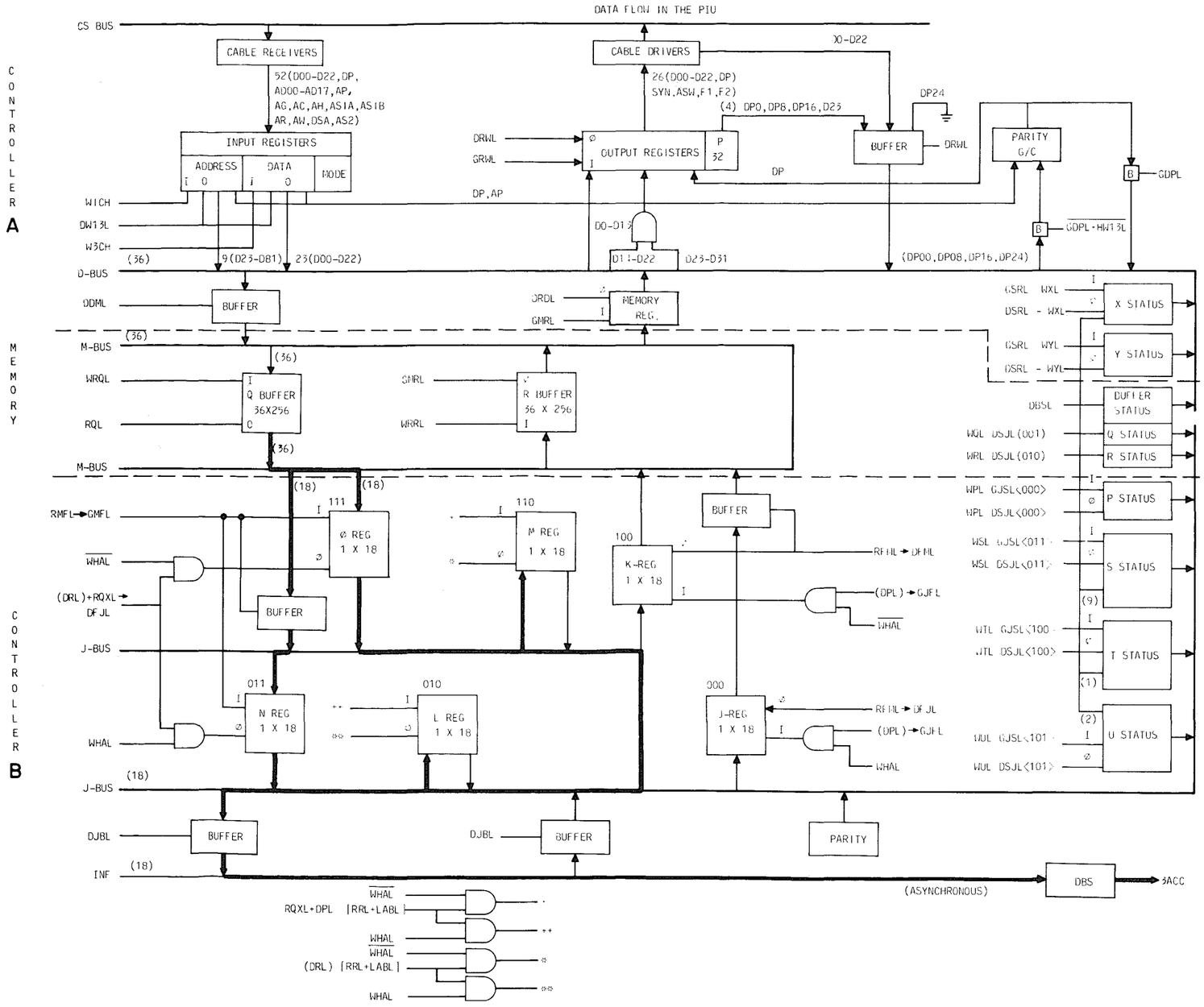


Fig. 19—PDSP Buffer Read (DR)

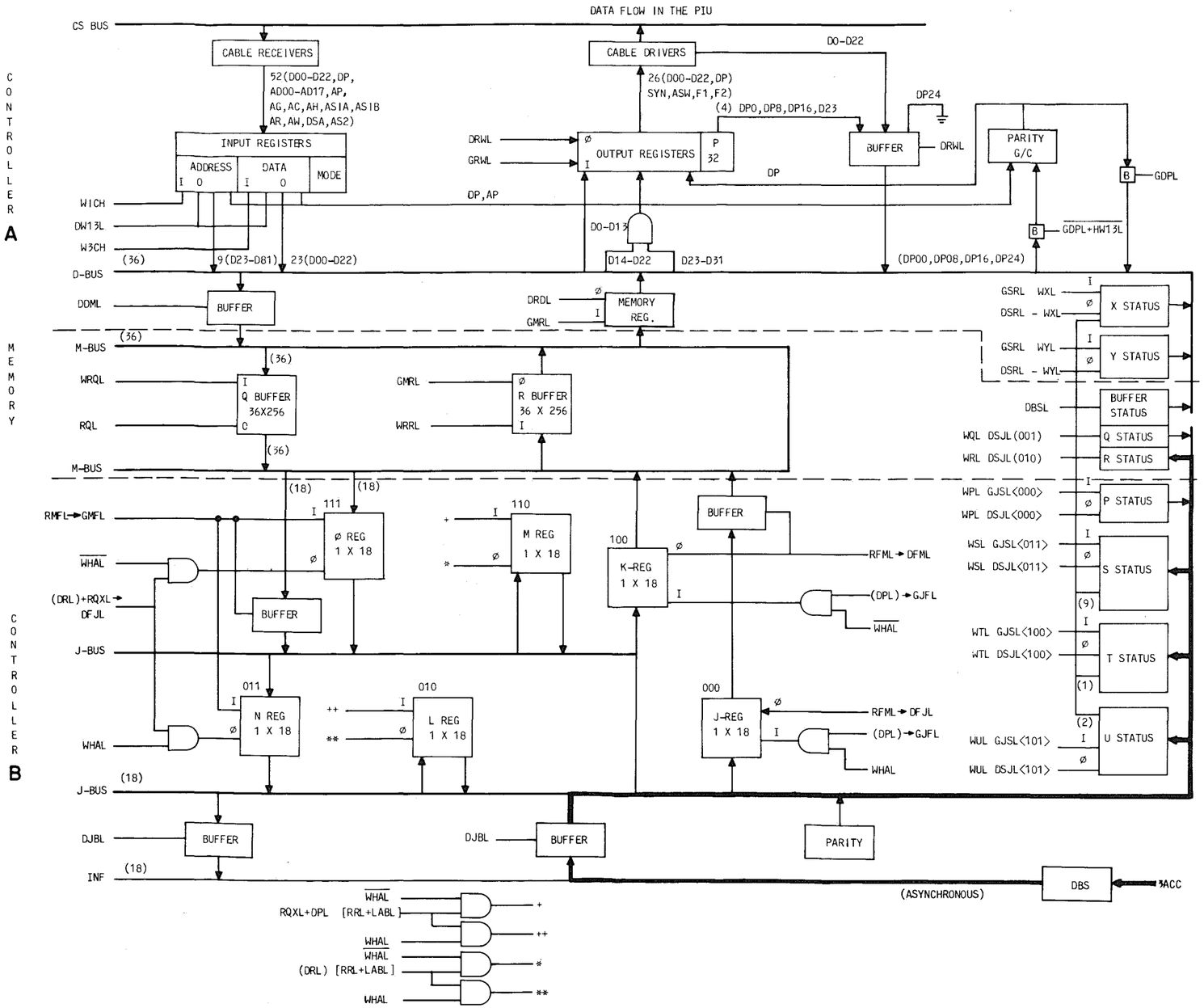


Fig. 20—PDSP Status Write (CP)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P	0	0	0	X 496 ES & AU	489 1TB	489 OTB										477 IOI
Q	0	0	1	X		477 64SPQ	477 BØQ	477 BUQ	477 NWBQ 7	477 NWBQ 6	477 NWBQ 5	477 NWBQ 4	477 NWBQ 3	477 NWBQ 2	477 NWBQ 1	477 NWBQ 0
R	0	1	0	X		477 64SPR	477 BØR	477 BUR	477 NWBR 7	477 NWBR 6	477 NWBR 5	477 NWBR 4	477 NWBR 3	477 NWBR 2	477 NWBR 1	477 NWBR 0
S	0	1	1	X	498 WHAL	490 IHER	490 3TB	490 D1	490 D2	490 D3	490 D4	490 D5	490 D6	489 1UA	498 LØ	493 ØS
T	1	0	0	X	489 IF3	498 RR	490 SF 3	490 SF 2	490 SF 1	490 SF 0	490 ADR 5	490 ADR 4	490 ADR 3	490 ADR 2	490 ADR 1	490 ADR 0
U	1	0	1	X	498 LAB	498 ISMB +	489 LAA	498 (CBFH) FPERB	471 PERB 7	471 PERB 6	471 PERB 5	471 PERB 4	471 PERB 3	471 PERB 2	471 PERB 1	471 PERB 0

2 1 0

SWSR
STATUS WORD

- (blank) → NOT AN OVERWRITABLE BIT
- \+ → OVERWRITING A ONE RESETS THIS BIT, IT CANNOT BE SET BY A CP ORDER
- X → NOT AN F/F, IT AFFECTS EXECUTION OF A SINGLE ORDER ONLY
- (hatched) → UNUSED BIT, ALWAY READS ONE
- * → INTERRUPT/DMA BITS NO LONGER PROVIDED
- 482 → FC CIRCUIT PACK CODE WHERE FUNCTION IS LOCATED

SWSR - STATUS WORD SELECT REGISTER
* CONTENTS DETERMINE WHICH STATUS WORD IS SELECTED
* SWSR IS OVERWRITTEN BY EVERY CP COMMAND.

ESØ - ENABLE STATUS OVERWRITE
* ESØ = 0, ONLY SWSR IS OVERWRITTEN.
* ESØ = 1, TOTAL STATUS WORD IS OVERWRITTEN.

NOTE:
CONTENTS OF THE SWSR ARE RETURNED EVERY SST ON BITS 13-15 TO IDENTIFY THE REGISTER BEING RETURNED.

Fig. 21—Controller B—Status Registers

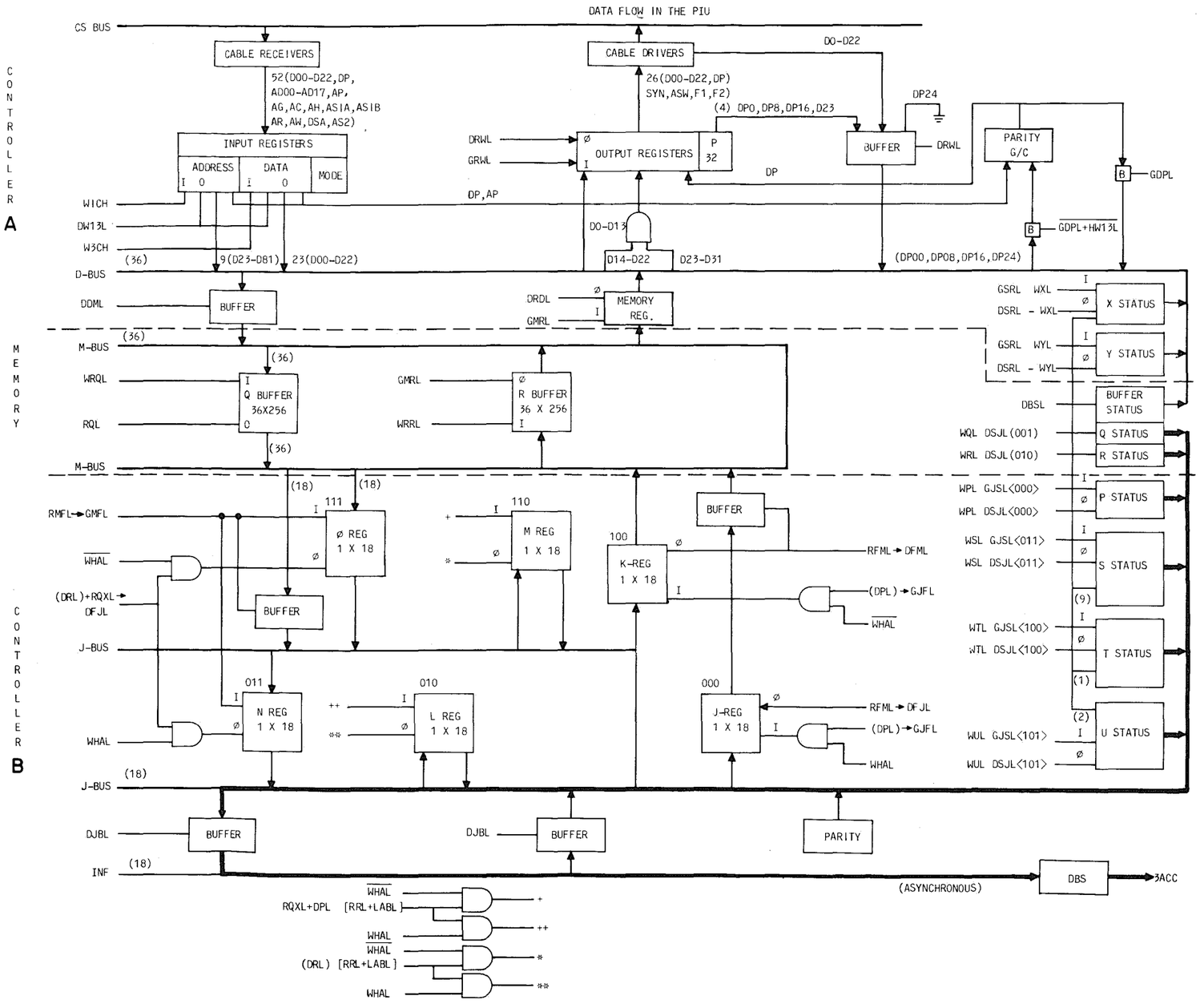


Fig. 22—PDSP Status Read (SST)

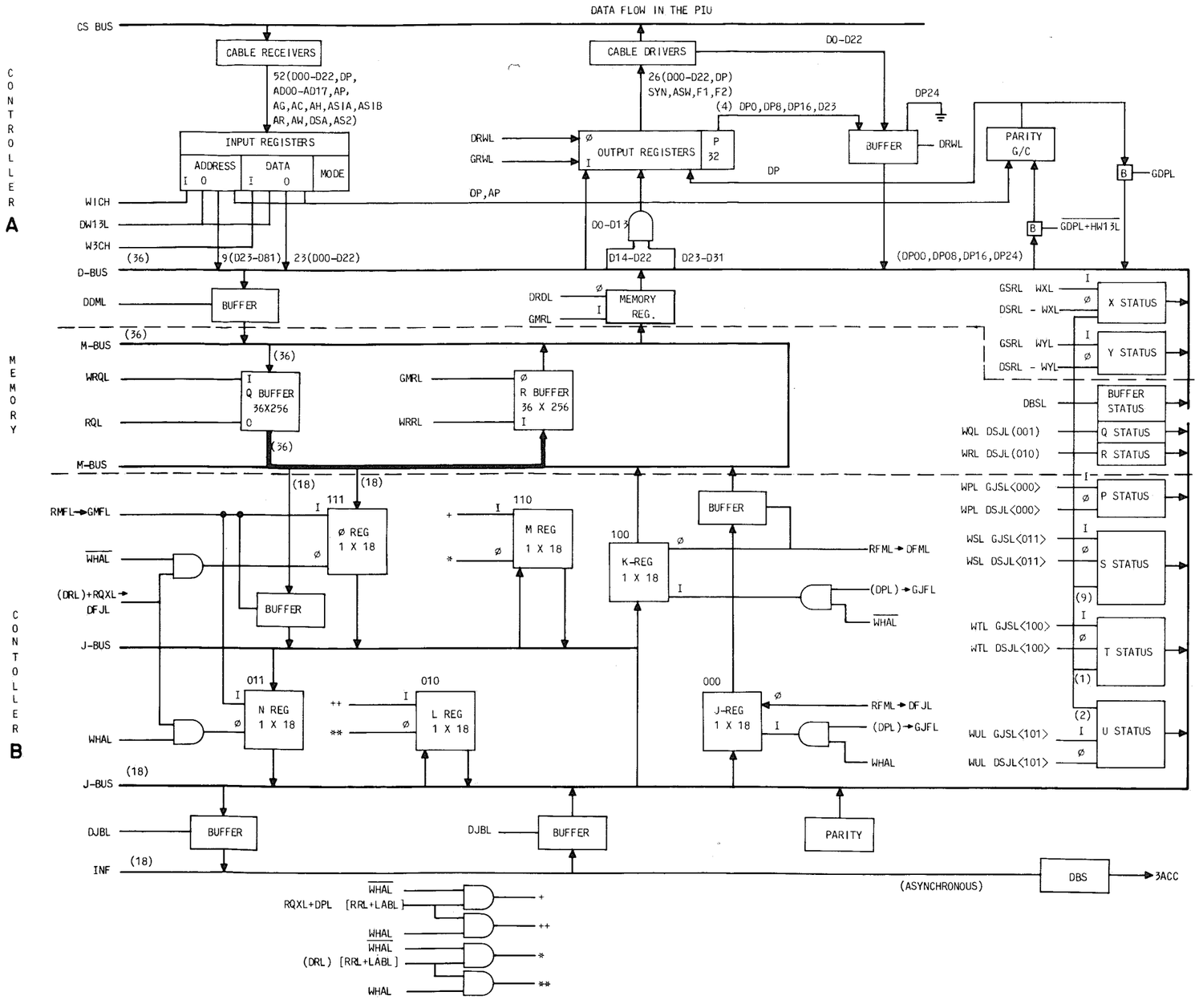


Fig. 23—Q to R Buffer Transfer

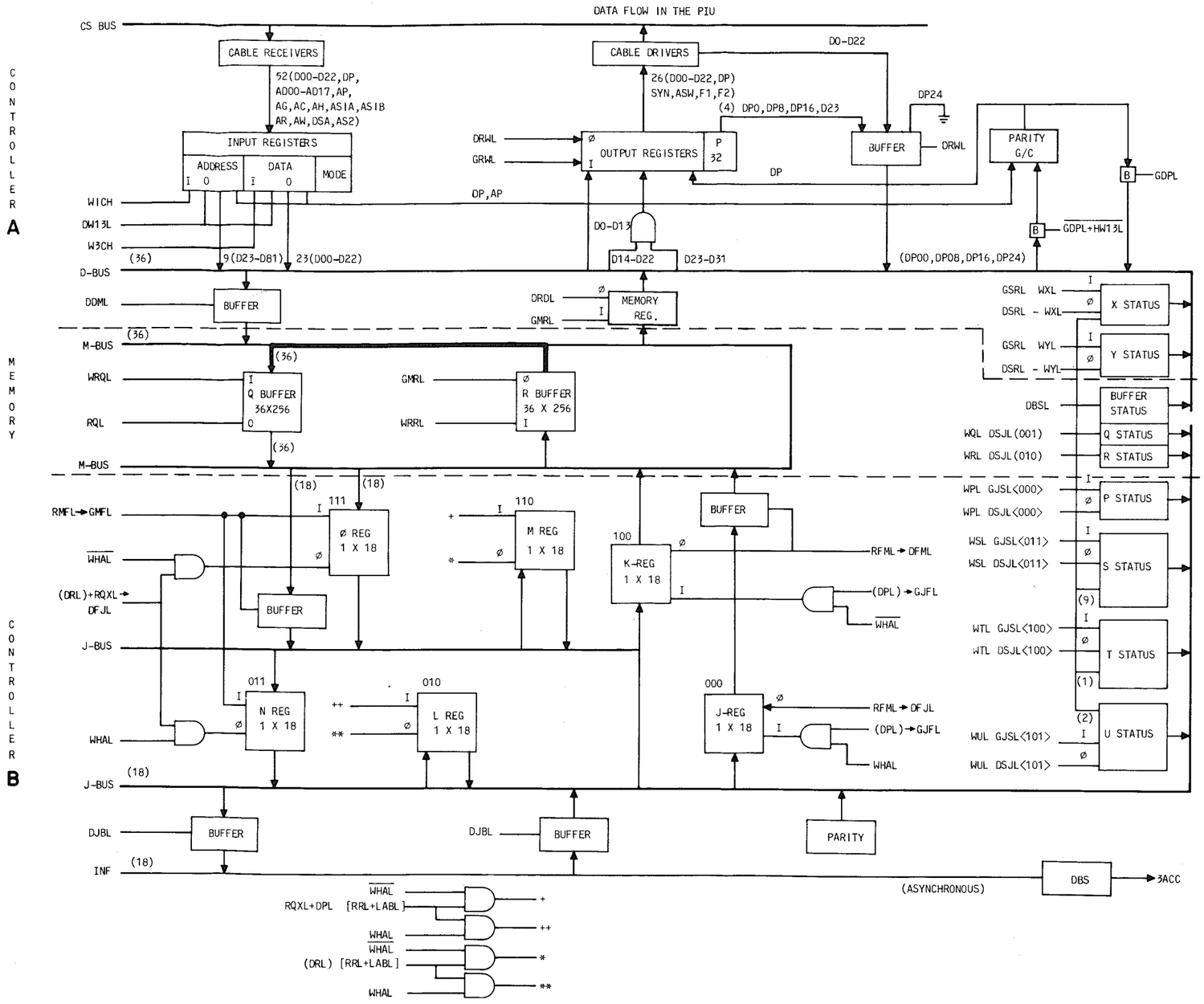


Fig. 24—R to Q Buffer Transfer

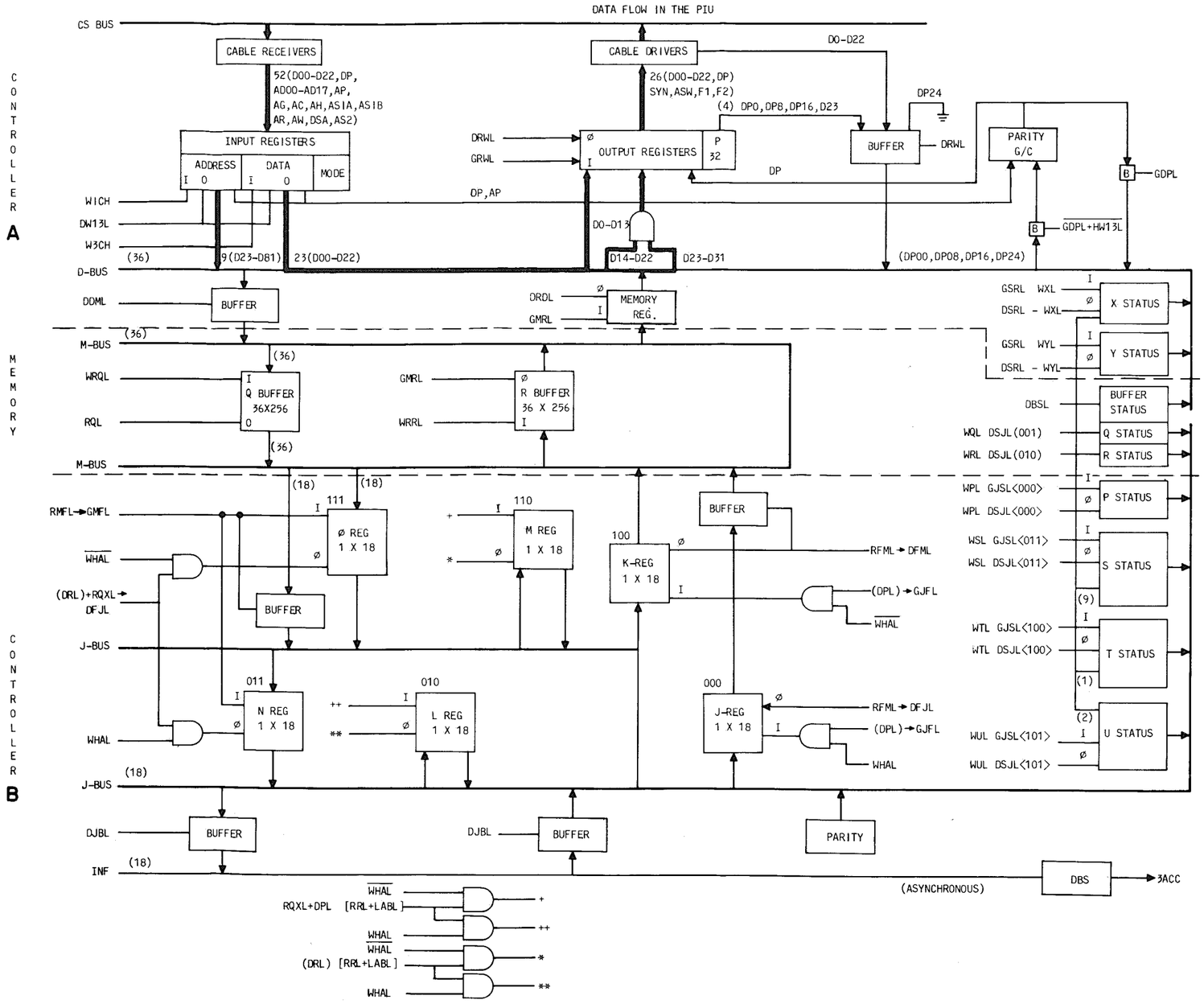


Fig. 25—Short Loop-Around from No. 1 ESS

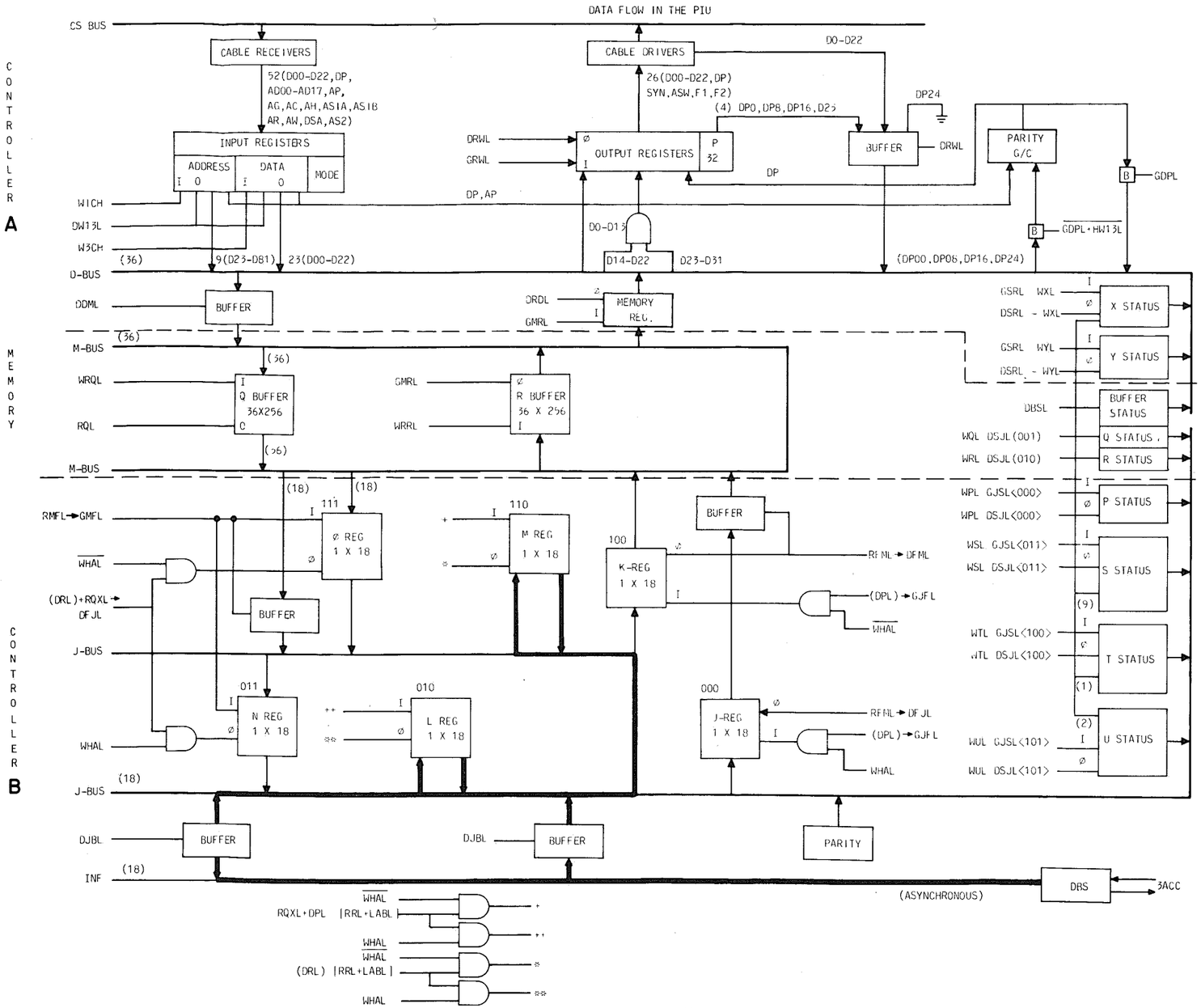


Fig. 26—Short Loop-Around from 3A Processor

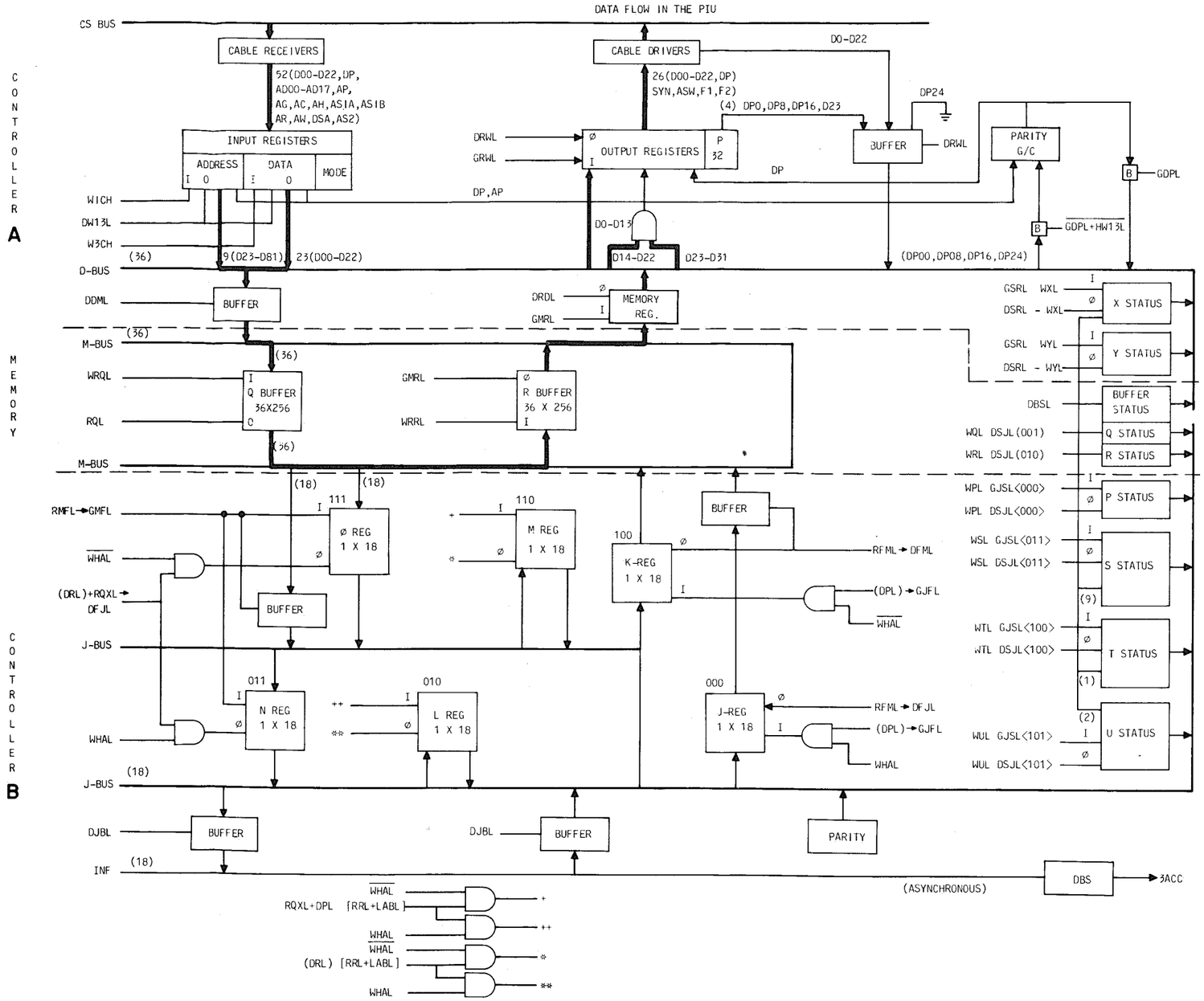


Fig. 27—Deep Loop-Around from No. 1 ESS

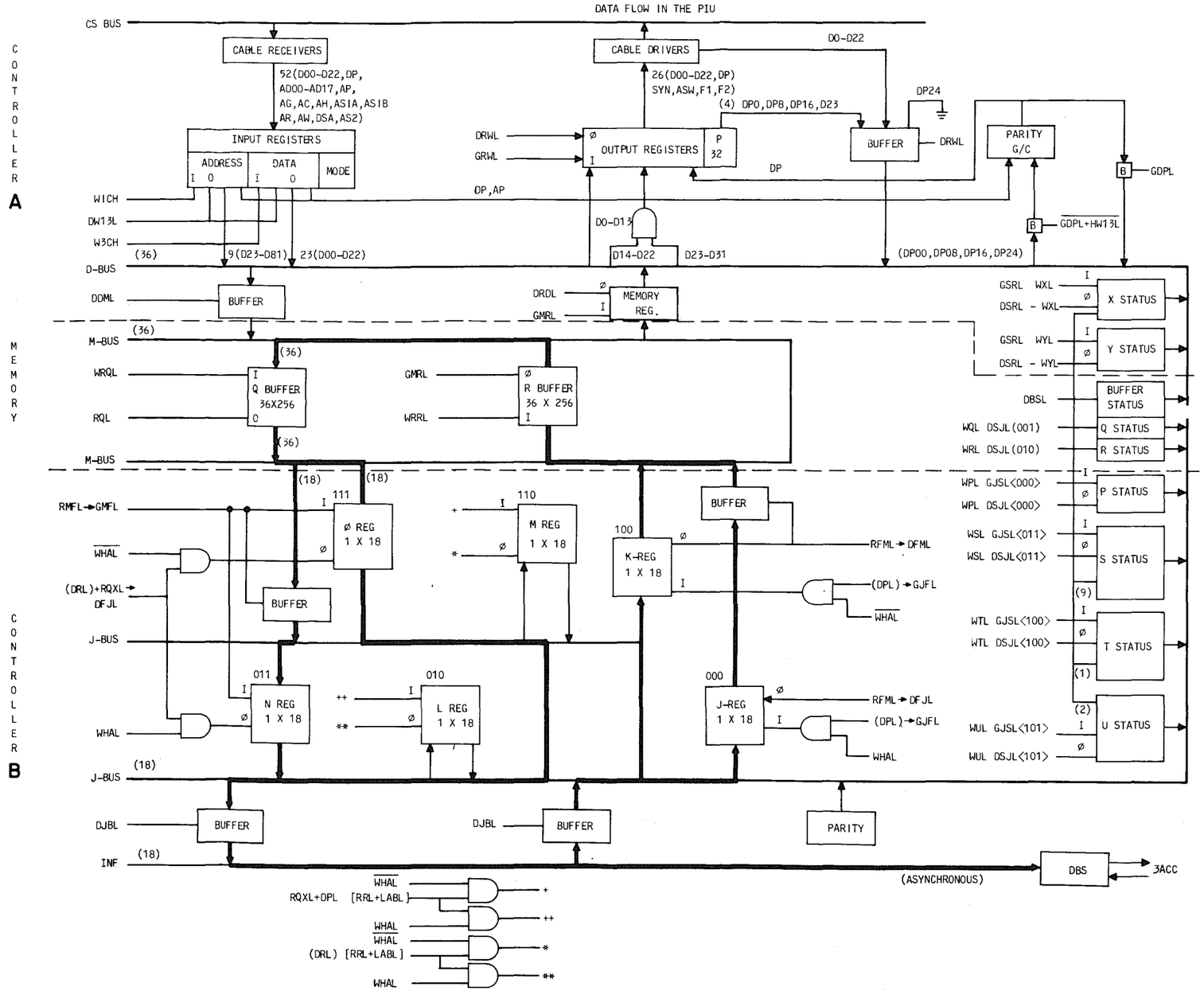


Fig. 28—Deep Loop-Around from 3A Processor

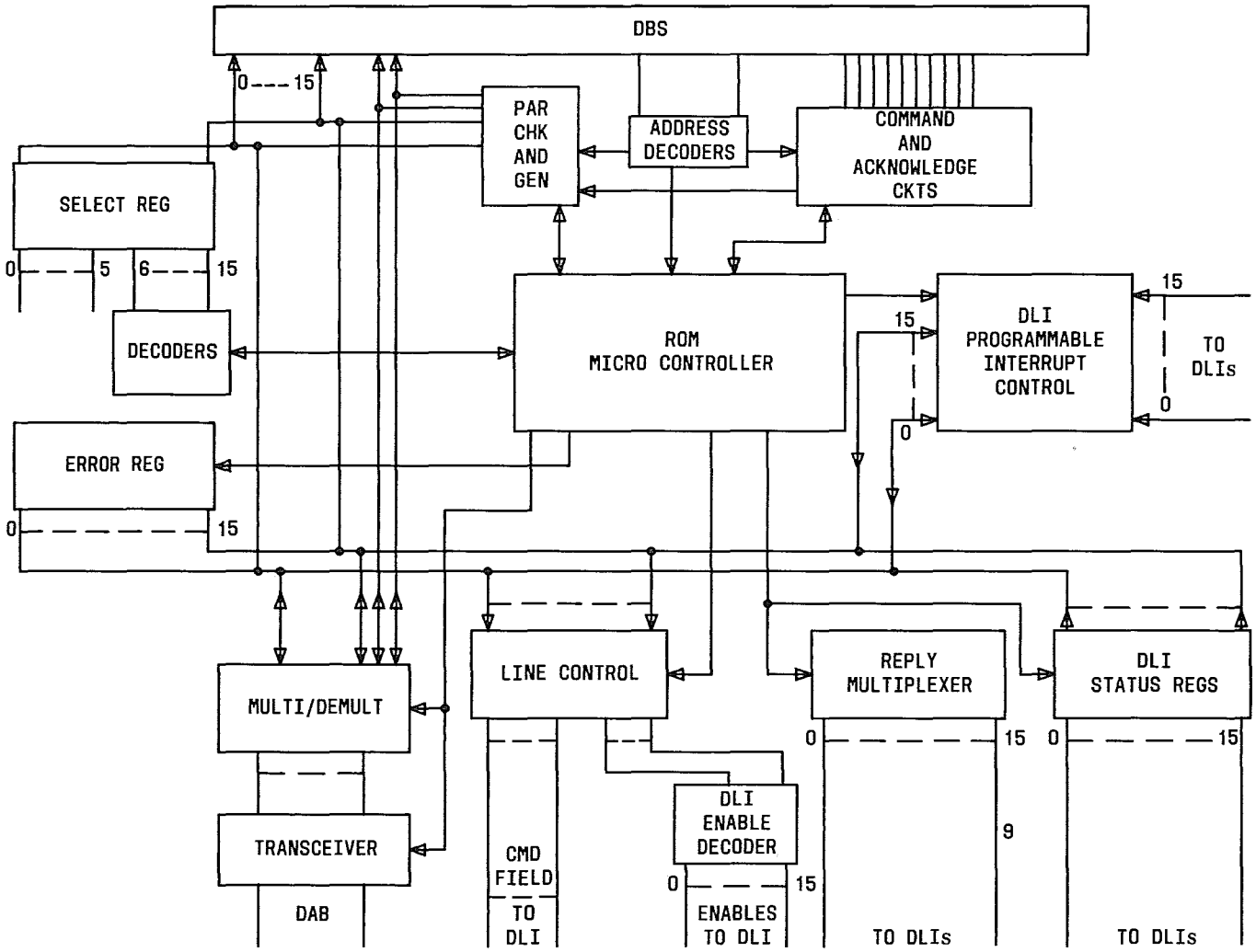


Fig. 29—Universal Data Link Control Block Diagram

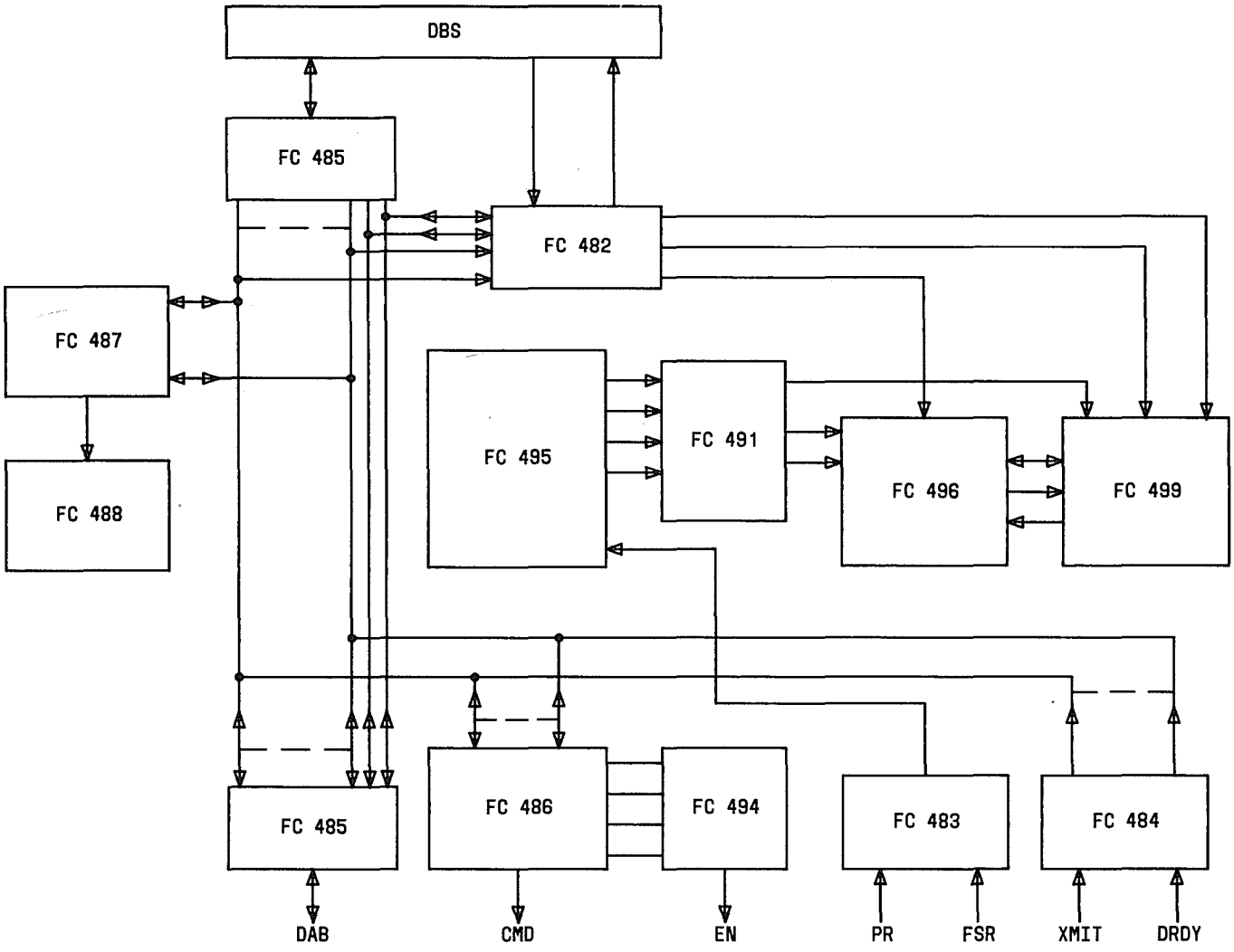


Fig. 30—Universal Data Link Control Circuit Pack Block Diagram

CP CODE	UDLC FUNCTION
FC482	ADDRESS, PARITY AND BUS SEQUENCER CIRCUIT
FC483	REPLY MULTIPLEXERS
FC484	STATUS MULTIPLEXERS
FC485	BIDIRECTIONAL BUS, DATA MULTIPLEXER/DEMULTIPLEXER AND TRANSCEIVERS
FC486	LINE CONTROL REGISTER
FC487	SELECT AND STATUS REGISTER
FC488	SELECT DECODERS
FC491	READ-ONLY-MEMORY AND PARITY CHECK
FC494	LINE CONTROL DECODER
FC495	MICROCONTROLLER
FC496	DATA LINK READ AND WRITE CONTROL FUNCTIONS
FC499	CLOCK AND I/O COMMAND DECODING
	DLI FUNCTION
FG4	INPUT BUFFER AND PARITY CHECK, TRANSMIT FIFO AND POINTER, I/O CONTROL AND BUS 0 CIRCUIT
FG5	USART & EIA INTERFACE, RECEIVE FIFO WITH POINTER, USART CONTROL & STATUS REGISTER, BUS 1 CIRCUIT

Fig. 31—Univeral Data Link Interface and Data Link Circuit Pack Codes

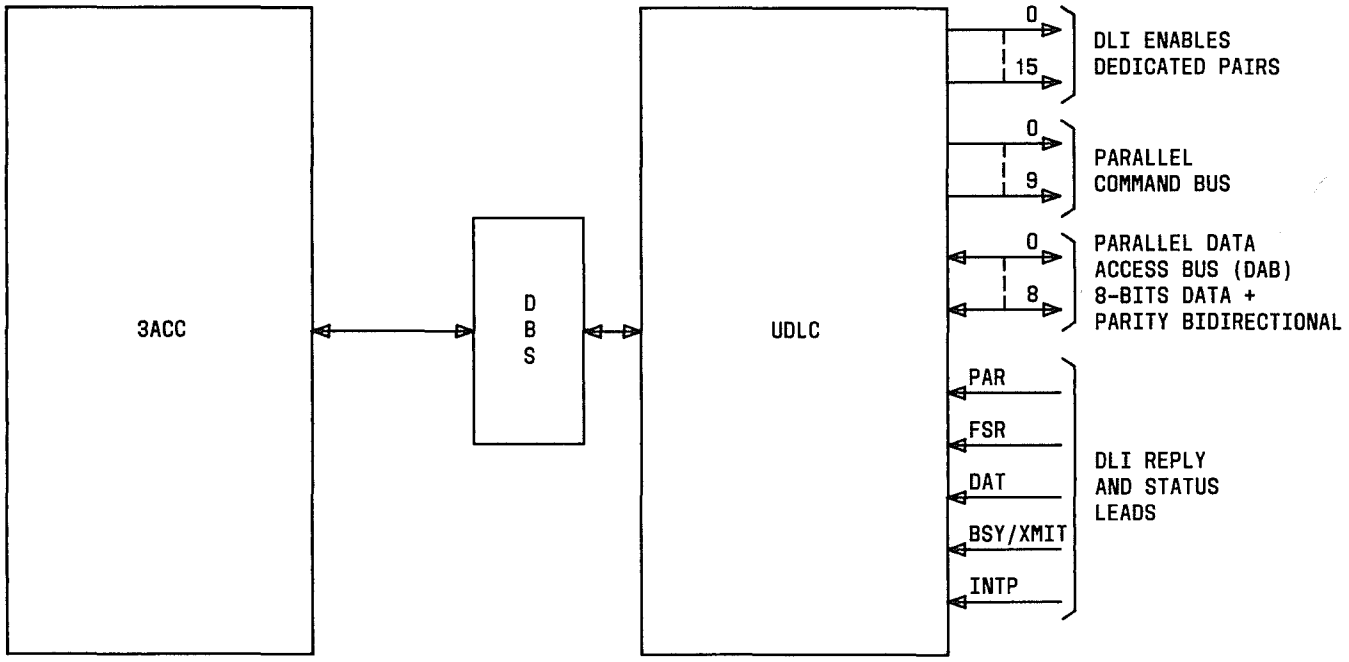


Fig. 32—Enables, Reply and Status Bus

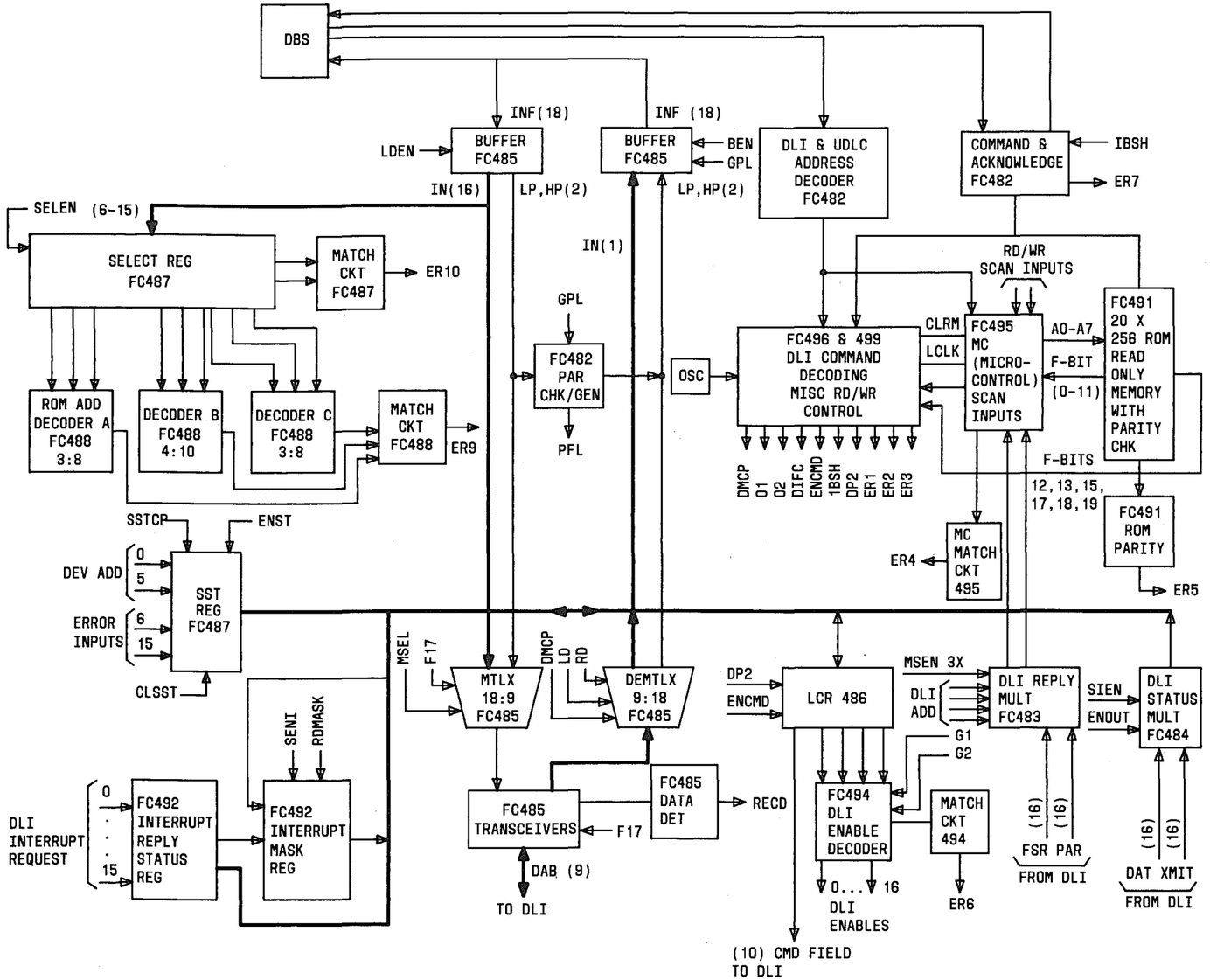


Fig. 33—Detailed Block Diagram of Universal Data Link Control

UDLC REGISTER DEFINITIONS

- SELECT REGISTER: INF BITS 6-15 USED BITS 5-0 ARE SPARE
  (1) SELECTS OR ENABLES ACCESS TO OTHER INTERNAL REGISTERS
  (2) ENABLES MAINTENANCE TESTS AND STIMULATES ERRORS
  (3) SELECTS APPROPRIATE ROM SEQUENCE

- STATUS REGISTER: BITS 0-5 CONTAIN DEVICE ADDRESS
  BITS 6-15 ERROR FLAGS IDENTIFYING WHAT GENERATED AN ERROR RESPONSE TO THE 3A

- LINE CONTROL REGISTER: LCR PEFORMS THREE FUNCTIONS USED IN COMMUNICATING WITH DATA LINK INTERFACE UNITS:
  (1) ENABLES APPROPRIATE DLI
  (2) PERMITS BLOCK TRANSFERS OF DATA
  (3) PROVIDES A PROGRAMMABLE COMMAND FIELD DEFINING WHAT OPERATION THE DLI IS TO PERFORM

- DATA AVAILABLE: DLI STATUS REGISTER INDICATES WHEN A PARTICULAR DLI HAS REACHED ITS FIFO BUFFER COUNT 32 THRESHOLD

- TRANSMIT/BUSY: INDICATES IF A PARTICULAR DLI IS BUSY TRANSMITTING DATA

- INTERRUPT MASK: PROGRAMMABLE REGISTER WHICH ENABLES OR INHIBITS 3A CC INTERRUPT REQUESTS FROM DLIs

- INTERRUPT REGISTER: DLI CONTROLLED INTERRUPT REQUESTS

Fig. 34—Definitions of Universal Data Link Control Registers

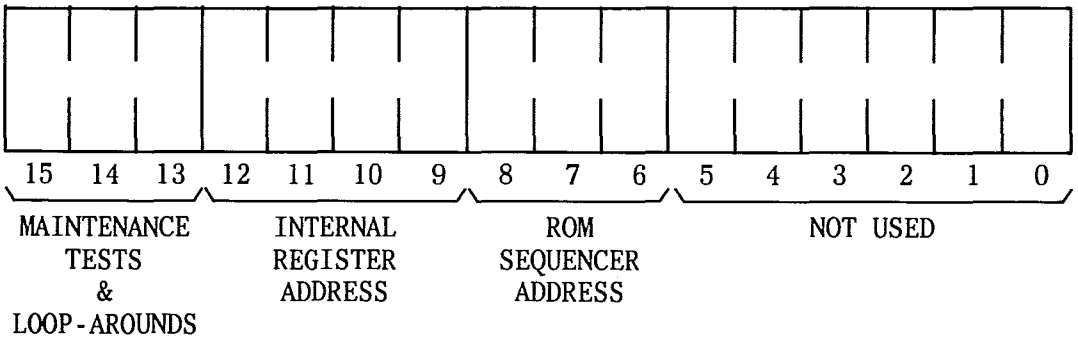


Fig. 35—Select Register

DECODER A INPUTS

FUNCTION	$\overline{\text{INF 8}}$	$\overline{\text{INF 7}}$	$\overline{\text{INF 6}}$	LEAD NAME
EPSCS MICROCODE	0	0	0	ROM 0
SPARE	0	0	1	ROM 1
SPARE	0	1	0	ROM 2
SPARE	0	1	1	ROM 3
SPARE	1	0	0	ROM 4
SPARE	1	0	1	ROM 5
SPARE	1	1	0	ROM 6
SPARE	1	1	1	ROM 7

DECODER C INPUTS

FUNCTION	$\overline{\text{INF 15}}$	$\overline{\text{INF 14}}$	$\overline{\text{INF 13}}$	LEAD NAME
SPARE	0	0	0	MSEN 0
DECODER A&B MISMATCH ER A-HALF	0	0	1	MSEN 1
XMIT BAD PAR ON DAB	0	1	0	MSEN 2
SPARE	0	1	1	MSEN 3
1/2-WORD MODE	1	0	0	MSEN 4
DECODER A&B MISMATCH B-HALF	1	0	1	MSEN 5
INHIBIT DLI ENABLES	1	1	0	MSEN 6
SPARE	1	1	1	MSEN 7

Fig. 36—Select Decoders A and C

DECODER B INPUTS

FUNCTION	$\overline{\text{INF 12}}$	$\overline{\text{INF 11}}$	$\overline{\text{INF 10}}$	$\overline{\text{INF 9}}$	LEAD NAME
READ/WRITE LCR	0	0	0	0	SEN 0
READ/WRITE INTERRUPT MASK	0	0	0	1	SEN 1
STIM. A-HALF EN DECODER ER	0	0	1	0	SEN 2
READ DLI INTERRUPT REQUESTS	0	0	1	1	SEN 3
N.C.	0	1	0	0	SEN 4
READ TRANSMIT/BUSY STATUS	0	1	0	1	SEN 5
READ DATA READY STATUS	0	1	1	0	SEN 6
MAIN. READ DEMULTIPLEXER	0	1	1	1	SEN 7
UDLC RESET	1	0	0	0	SEN 8
STIM. A-HALF DECODER C ER	1	0	0	1	SEN 9
STIM. B-HALF DECODER C ER	1	0	1	0	SEN 10
STIM. B-HALF EN DECODER ER	1	0	1	1	ASEN 11
CLR. DUP. HALF SELECT REG.	1	1	0	0	ASEN 12
STIM. MICROCONTROLLER ER.	1	1	0	1	ASEN 13
STIM. ROM PARITY FAILURE	1	1	1	0	BSEN 14
UDLC MAIN. LOOP AROUND	1	1	1	1	BSEN 15

Fig. 37—Select Decoder B

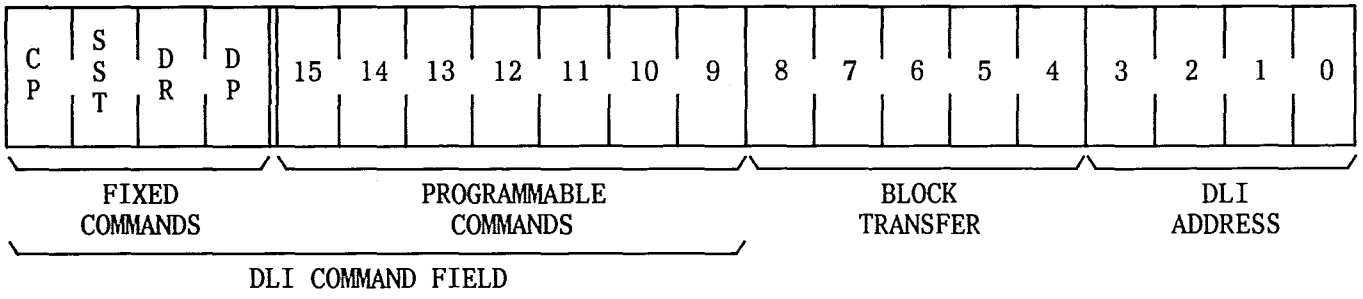


Fig. 38—Line Control Register Format

LINE CONTROL REGISTER

- INF BITS 3, 2, 1, 0: BINARY CODE DEFINING WHICH DLI IS TO BE ENABLED
- INF BITS 8-4: BLOCK TRANSFER COUNT. INDICATES HOW MANY CONSECUTIVE READS OR WRITES ARE TO OCCUR (NOT USED FOR EPSCS)
- INF BITS 15-9 (PART OF) COMMAND FIELD BITS CMD6-CMD0
  BITS CMD10-CMD7 ARE REGENERATED 3A COMMANDS AND HAVE FIXED DEFINITIONS

  CMD6-CMD0 ARE USER DEFINED.

FOR EPSCS THE FOLLOWING DEFINITIONS ARE ASSOCIATED WITH THE DLI:

- BIT 15 – CMD0: ENABLE DLI STATUS REGISTER READ
- BIT 14 – CMD1: END WRITE SEQUENCE
- BIT 13 – CMD2: CONTROL BIT (CONTROL READS OR WRITES)
- BIT 12 – CMD3: MASTER RESET
- BIT 11 – CMD4: READ FIFO POINTER
- BIT 10 – CMD5: SECOND LEVEL DLI MAINTENANCE LOOP AROUND
  DIGITAL LOOP AROUND AT OUTPUT SIDE OF USART
- BIT 9 – CMD6: FIRST LEVEL MAINTENANCE LOOP AROUND

  CMD7: DP – DATA PRESENT (WRITE)

  CMD8: DR – DATA REQUEST (READ)

  CDM9: SST – SPARE

  CMD10: CP – COMMAND PRESENT

  [PERMITS TRANSMITTAL OF A COMMAND TO A DLI WITHOUT USING MICROCONTROLLER
  DOES NOT INVOLVE ANY DATA TRANSACTION]

Fig. 39—Line Control Register

SST REGISTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ER10	ER9	ER8	ER7	ER6	ER5	ER4	ER3	ER2	ER1	DEVICE ADDRESS					

ER1: DLI READ ATTEMPT FAILURE (RECD)

ER2: FIFO STATUS REPLY (EMPTY READ OR FULL WRITE FIFO)

ER3: DLI WRITE PARITY FAILURE (PAR)

ER4: MICROCONTROLLER MISMATCH

ER5: ROM PARITY FAILURE

ER6: DLI ENABLE DECODER MISMATCH

ER7: 3A COMMAND ERROR

ER8: SECOND HALF OF MULTIPLEXED/DEMULTIPLEXED WORD FAILURE

ER9: SELECT DECODERS MISMATCH

ER10: SELECT REGISTER MISMATCH

Fig. 40—SST Register

DLI 15	DLI 14	DLI 13	DLI 12	DLI 11	DLI 10	DLI 9	DLI 8	DLI 7	DLI 6	DLI 5	DLI 4	DLI 3	DLI 2	DLI 1	DLI 0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FIXED ASSIGNMENT OF BITS TO DATA LINK INTERFACE UNITS.

- DATA READY
- TRANSMIT-BUSY
- INTERRUPT REGISTER
- INTERRUPT MASK REGISTER
- BOTH REPLY MULTIPLEXERS

Fig. 41—Fixed Assignment Registers

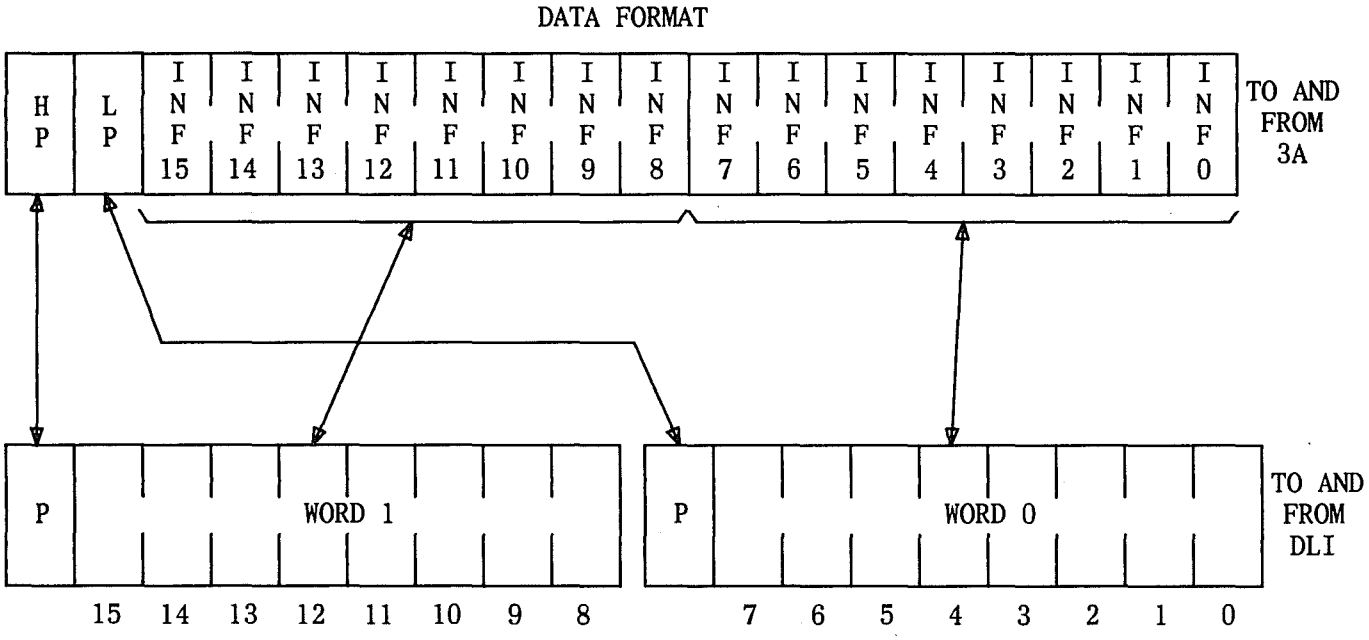


Fig. 42—Data Format

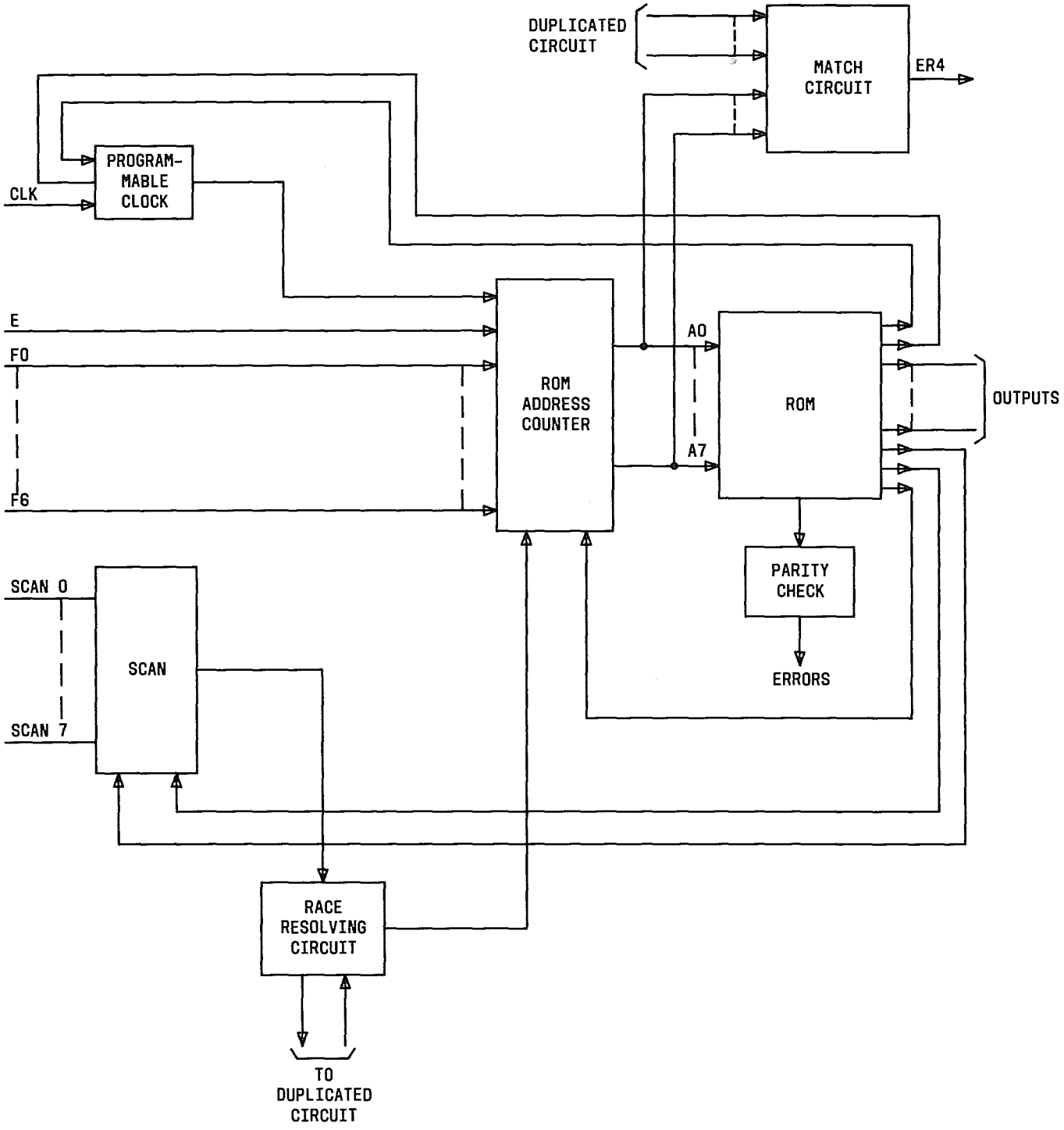


Fig. 43—Microcontroller Block Diagram

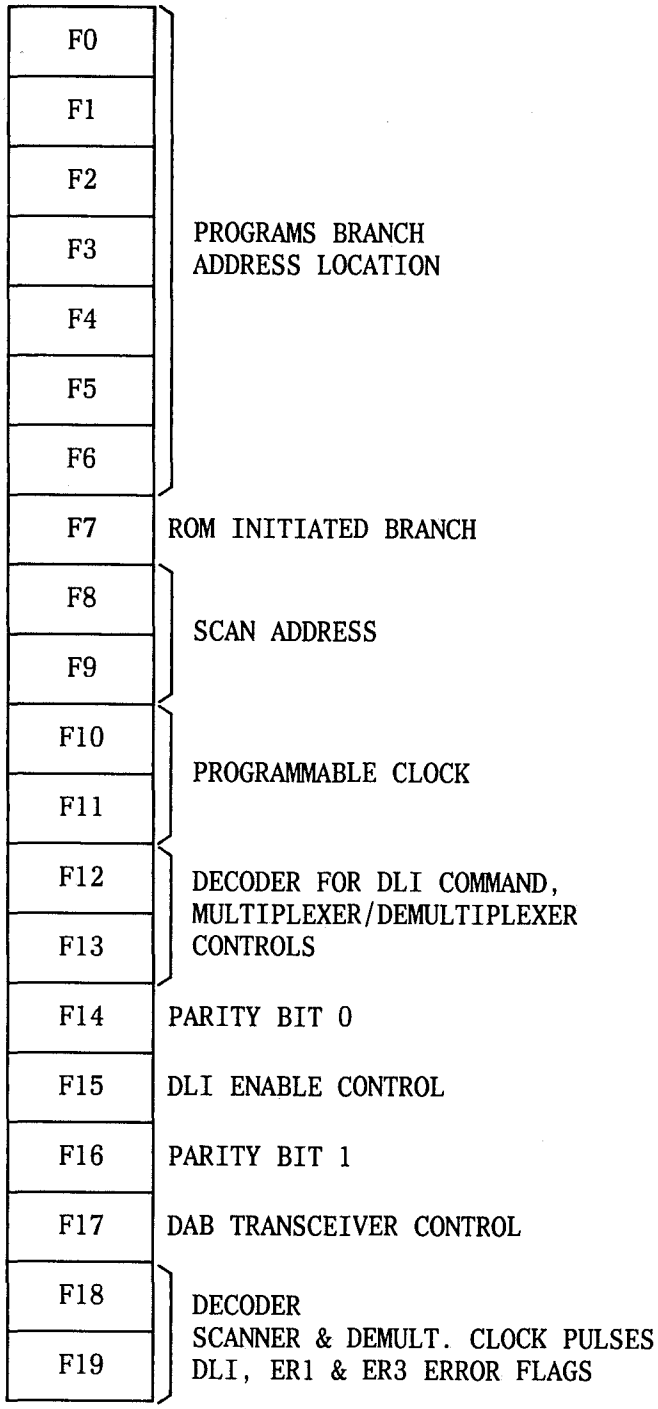


Fig. 44—Microcode Format

FC 491

ROM OUTPUT BIT FUNCTIONS (F0-F9)

F0-F3: BRANCH ADDRESS, ADDRESS BITS 0-3
F4-F6: BRANCH ADDRESS, ADDRESS BITS 5-7

NOTE: ADDRESS BIT 4 IS A FIXED HARD WIRED BIT CONTROLLED BY XDRL AND IS NOT UNDER ROM CONTROL

F7 : ROM INITIALTED BRANCH COMMAND

F8,F9: MICROCONTROLLER SCANNER ADDRESS COMBINED WITH A HARD WIRED ADDRESS BIT, XDRL, FORMS A 3-BIT SCANNER ADDRESS FIELD.

SCANNER ADDRESS			FUNCTIONAL NAME	NET NAME	DEFININITION
XDRL	F9	F8			
C	B	A			
0	0	0	SCAN 1A	BSEN15	WRITE MAINTENANCE MODE
0	0	1	SCAN 2A	PAR	DLI PARITY REPLY
0	1	0	SCAN 3A	FSR	DLI FIFO STATUS REPLY
0	1	1	SCAN 4A	MSEN1	WORD MODE SELECTION
1	0	0	SCAN 1B	BSEN15	READ MAINTENANCE MODE
1	0	1	SCAN 2B	RECD	RECEIVED DATA FROM DLI
1	1	0	SCAN 3B	FSR	DLI FIFO STATUS REPLY
1	1	1	SCAN 4B	MSEN1	WORD MODE SELECTION (HALF-FULL)

XDRL = 0 = DP (3A WRITE COMMAND)
XDRL = 1 = DR (3A READ COMMAND)

Fig. 45—Read Only Memory Output Bit Function (F0-F9)

FC 491

ROM OUTPUT BIT FUNCTIONS (F10-F16)

F10 & F11: MICROCONTROLLER PROGRAMMABLE CLOCK CONTROL

F12 & F13: CONTROLS DUAL 2-BIT BINARY DECODER ON FC496 WHICH, WHEN COMBINED WITH A 3A READ OR WRITE COMMAND (DRL and XDPL respectively), INITIATES FUNCTIONS WITHIN THE UDLC FOR UDLC/DLI READ/ WRITE OPERATIONS.

DECODER FC496				FUNCTION
XDPL	DRL	F13	F12	
0	0	0	0	SPARE
0	0	0	1	SET MSEL FLIP-FLOP
0	0	1	0	SET CMD FLIP-FLOP
0	0	1	1	RESET CMD F/F SET SYNC F/F
1	1	0	0	SPARE
1	1	0	1	SET MSEL F/F
1	1	1	0	SET CMD F/F
1	1	1	1	RESET CMD F/F SET SYNC F/F

F14 : * PARITY BIT 0 (ODD PARITY) OVER FOLLOWING ROM OUTPUT BITS: F0, F1, F2, F4, F8, F15, F17, F18, F19

F15 : DEDICATED DLI ENABLE COMMAND

F16 : * PARITY BIT 1 (ODD PARITY) OVER FOLLOWING ROM BITS: F3, F5, F6, F7, F9, F10, F11, F12, F13

*NOTE: ROM PARITY BITS ARE SPLINTERED IN SUCH A MANNER THAT A FAILURE OF AN ROM DIP CONTAINING 4 BITS (EVEN PARITY) WILL BE DETECTED. ALSO, IF A DOUBLE FAULT SHOULD OCCUR SUCH THAT ONE PARITY CIRCUIT ALSO FAILS, IT WOULD BE DETECTED DURING THE ROM SEQUENCE. THIS GUARANTEES RELIABILITY WITHOUT THE NEED OF CIRCUIT DUPLICATION.

Fig. 46—Read Only Memory Output Bit Function (F10-F16)

FC 491

ROM OUTPUT BIT FUNCTIONS (F17-F19)

F17 : TRANSCEIVER STROBE FOR DLI DATA TRANSMISSION AND MUTIPLEXER CLOCK.

F18 & F19: CONTROLS DUAL 2-BIT BINARY DECODER ON FC499 AND IS USED DURING UDLC/DLI COMMUNICATIONS TO GENERATE CLOCK PULSES AND RAISE DLI ERROR FLAGS.

DECODER FC499				FUNCTION
XDPL	DRL	F19	F18	
0	0	0	0	SPARE
0	0	0	1	GEN. DEMULT. LOAD PULSE
0	0	1	0	GEN. ER 1 RECD FAILURE
0	0	1	1	MICROCONTROLLER SCANNER ENABLE
1	1	0	0	SPARE
1	1	0	1	SPARE
1	1	1	0	GEN. ER 3 PAR FAILURE
1	1	1	1	MICROCONTROLLER SCANNER ENABLE

Fig. 47—Read Only Memory Output Bit Function (F17-F19)

MICROCONTROLLER SCAN POINTS

NET NAME	FUNCTIONAL NAME	MEANING
BSEN 15	SCAN 1A	ENTER WRITE MAINTENANCE MODE
PAR	SCAN 2A	PARITY REPLY FROM DLI
FSR	SCAN 3A	FIFO STATUS REPLY FROM DLI
MSEN 1	SCAN 4A	WORD MODE SELECTION (HALF/FULL)
BSEN 15	SCAN 1B	ENTER READ MAINTENANCE MODE
RECD	SCAN 2B	RECEIVED DATA
FSR	SCAN 3B	FIFO STATUS REPLY FROM DLI
MSEN 1	SCAN 4B	WORD MODE SELECTION (HALF/FULL)

THE A-SCAN POINTS ARE USED BY THE WRITE MICRCODE.

THE B-SCAN POINTS ARE USED BY THE READ MICRCODE.

Fig. 48—Microcontroller Scan Points

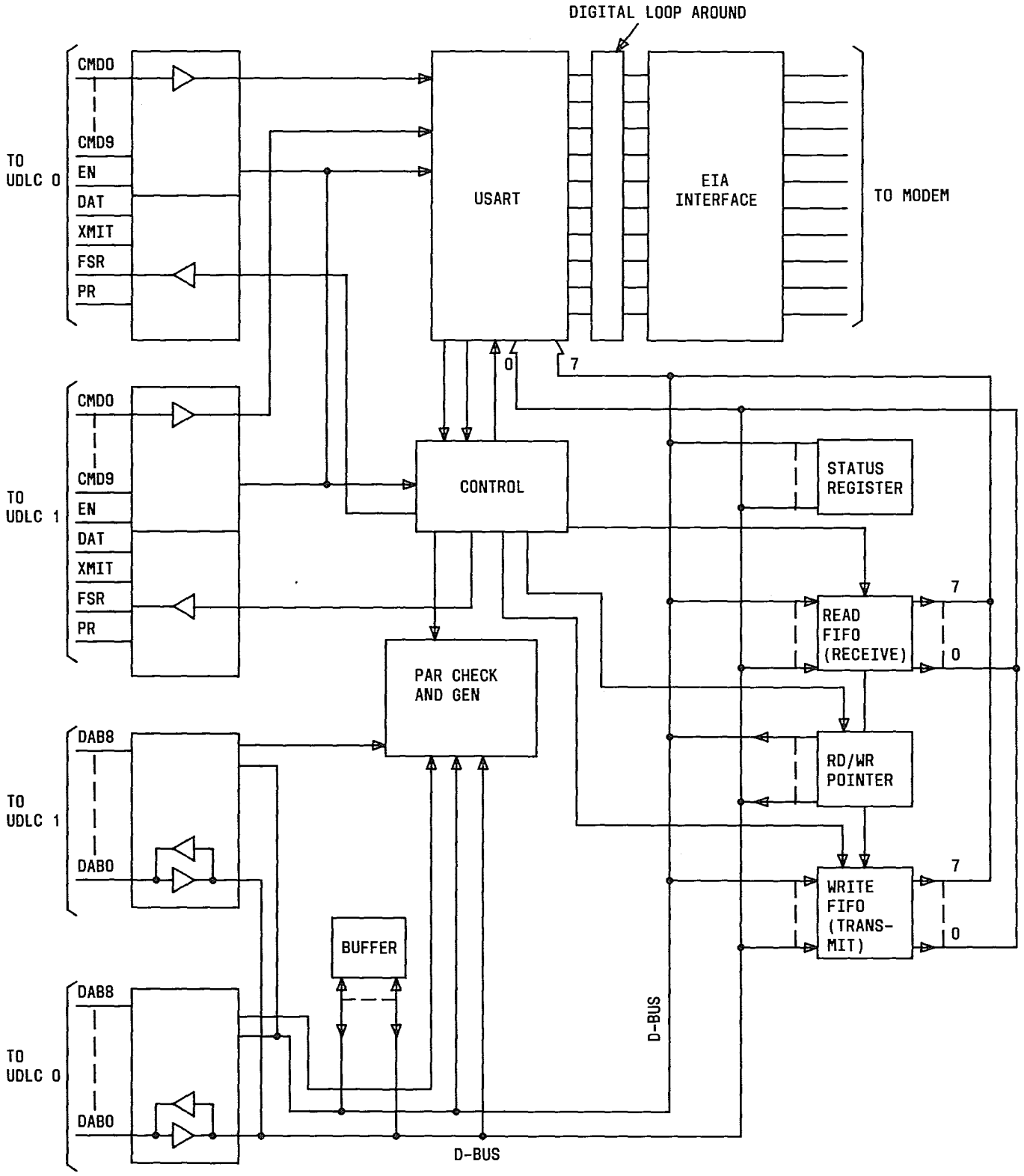


Fig. 49—Data Link Interface Block Diagram

TABLE A

DATA LINK INTERFACE COMMANDS

DLI COMMAND	NET NAME	FUNCTION
DLI Status Register	CMD 0	Read of DLI Status Register.
End Write	CMD 1	Sets End Write F/F enabling a DLI control feature that autonomously reprograms the USART to terminate transmission when write FIFO becomes empty.
Control Bit	CMD 2	When set to a "one" with Write Command, reprograms USART. When set with read, does USART Status Register Read.
Master Reset	CMD 3	Resets everything in the DLI.
FIFO Pointer	CMD 4	Returns the contents of both the transmit and receive FIFO pointers.
2nd Main, Loop-Around	CMD 5	DLI second-level maintenance loop around. Digital loop-around at output side of USART.
1st Main, Loop-Around	CMD 6	First-level maintenance loop around at front end of DLI.
Write	CMD 7	Write command, used to write data into transmit FIFO or reprogram USART.
Read	CMD 8	Read command, used to read receive FIFO, DLI status register, USART status register and front-end input buffer.

TABLE B

DLI
EIA INTERFACE CABLE CONNECTIONS
DB TYPE CONNECTOR FOR RS232C INTERFACE
NOTE

CONNECTOR PIN NUMBER	EIA STD NAME	201C NAME	FG5 PIN NUMBER	DIRECTION	DESCRIPTION
1	AA	FG	319T	Both	Frame Ground
2	BA	SD	201B	To Data Set	Transmitted Data
3	BB	RD	105B	To DLI	Received Data
4	CA	RS	301B	To Data Set	Request to Send
5	CB	CS	103B	To DLI	Clear to Send
6	CC	DSR	004B	To DLI	Data Set Ready
7	AB	SG	319T	Both	Signal Ground
8	CF	CO	005B	To DLI	Carrier On
15	DB	SCT	104B	To DLI	Serial Clock Transmit
17	DD	SCR	006B	To DLI	Serial Clock Receive
19	SCA	RR	—	To Data Set	Remote Release (Strap to Pin 21)
20	CD	DTR	002B	To Data Set	Data Terminal Ready
21	CG	RDY	—	To DLI	Ready (Strap to 19)
22	CE	RI	—	To DLI	Ring Indicator 1 (Strap to Pin 23)
23	CH/CI	RG2	—	To Data Set	Ring Indicator 2 (Strap to Pin 22)

Note: Terminals not listed are not used.