MDB

MLSI-DUVII

SYNCHRONOUS SERIAL LINE INTERFACE

INSTRUCTION MANUAL

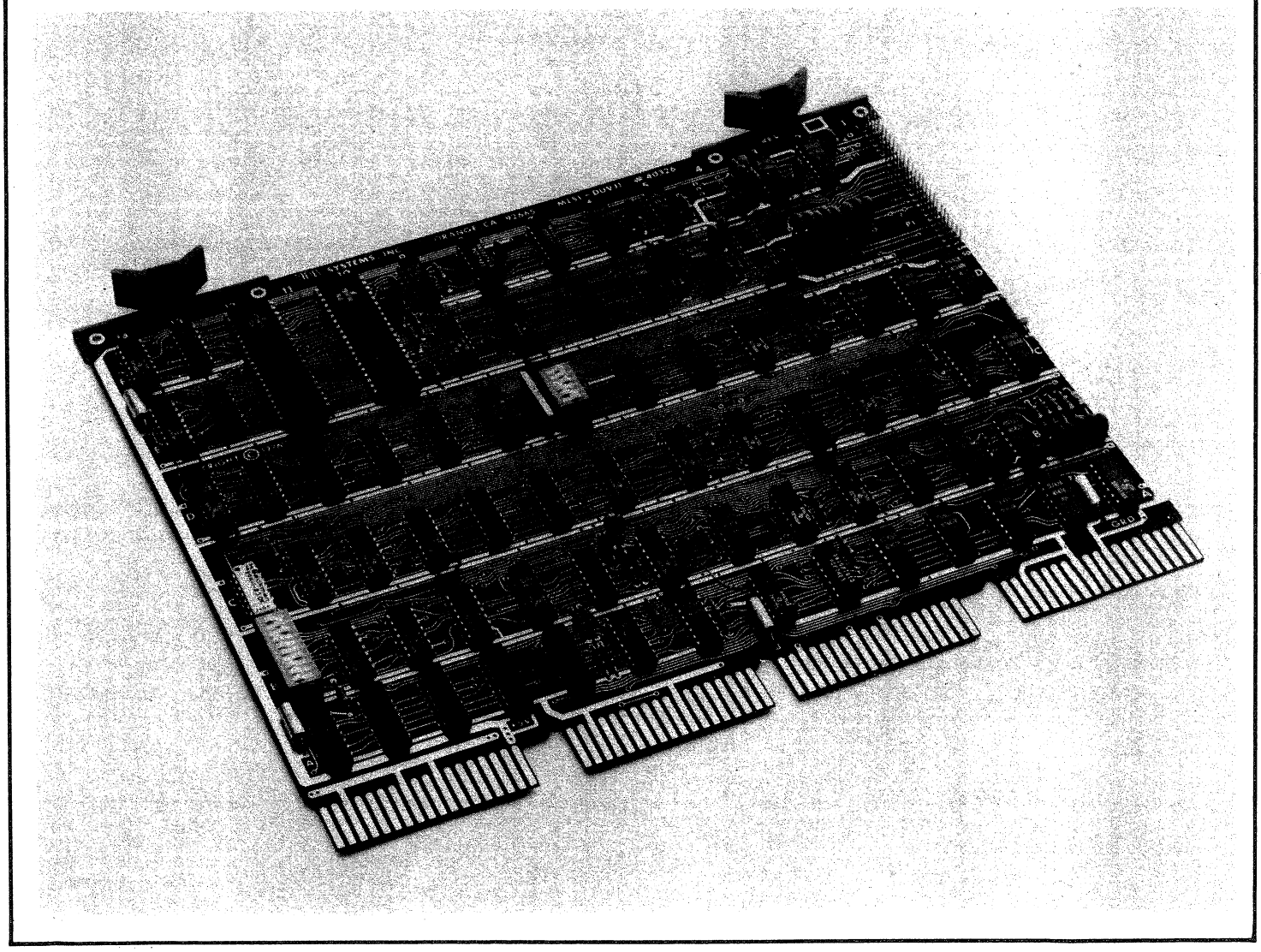


TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

MDB SYSTEMS INC.
1995 N. Batavia Street
Orange, California 92665
714-998-6900
TWX:910-593-1339

MLSI-DUV11
SYNCHRONOUS SERIAL LINE INTERFACE

INTRODUCTION

The MLSI-DUV11 Synchronous Serial Line Interface permits a DEC KD11-type processor to communicate through a modem or dataset, working in synchronous or isochronous communication modes, either half-duplex or full duplex. The DUV11 is program-controlled, and is completely compatible with existing DEC LSI-11 operating and diagnostic software.

The DUV11 performs double-buffered communications, converting received serial data to parallel form, and parallel data to serial form for transmission. The module provides level conversion between on-board TTL levels and EIA levels at the modem interface, and accommodates transmission rates up to 40k baud. Compatible modems include Bell 303/313, and 201/202 types. Jumpers permit adapting the DUV11 to a variety of modem operating configurations.

Device addresses, and interrupt vectors, may be preset using DIP switches on the module.

PHYSICAL DESCRIPTION

The MLSI-DUV11 Synchronous Serial Line Interface is a quad module which occupies one quad slot in the MLSI-BPA-84 backplane/cardguide assembly. The module connects to the MLSI bus through the backplane, and connects to a 303/313- or 201/202 type modem through a standard 50-pin EIA connector. The mating connector is a 50-pin Berg, or equivalent.

The module requires DC power from the backplane as follows:

- +5VDC at 1.5 A
- +12VDC at 0.1 A
- -12VDC at 0.04 A

INSTALLATION

Setting Addresses

Switches on the module must be set to encode the required device address, and the required interrupt vector address.

Setting Device Address - The device address is preset using switches on the DIP device in location 14B. The least-significant bit (DATA03) is set by switch No. 1, and the most-significant bit (DATA12) is set with switch No. 10. Compatible DEC software requires that the device address be within the floating address range from 760010 to 763776. Modified software will permit using an address beyond that range.

To set the device address, set a bit to "0" by setting the related switch to the "off" position. Set the switch to the opposite (ON) position to set the bit to a "1". (See table 1.)

Table 1 Device Address Selection

Digit W		Digit X		Digit Y		Digit Z	
Sw. No. 10	Value	Sw. No. 9 8 7	Value	Sw. No. 6 5 4	Value	Sw. No. 3 2 1	Value
o	6	o o o	0	o o o	0	o o o	0
●	7	o o ●	1	o o ●	1	o o ●	1
		o ● o	2	o ● o	2	o ● o	2
		o ● ●	3	o ● ●	3	o ● ●	3
		● o o	4	● o o	4	● o o	4
		● o ●	5	● o ●	5	● o ●	5
		● ● o	6	● ● o	6	● ● o	6
		● ● ●	7	● ● ●	7	● ● ●	7
Address = 7WXYZ0		o = switch open		● = switch closed (ON)			

Setting Interrupt Vector Address – The standard vector addresses are XX4 (transmit interrupt) and XX0 (receive interrupt). The vector address is set using switches on the DIP module in location 8D. Switch No. 1 sets the least-significant changeable bit (DATI03), and switch No. 6 sets the most-significant changeable bit (DATI08). It is necessary only to set the receive interrupt vector.

Although the Interface permits a vector address range from 000 to 774, DEC software limits the floating vector address to the range from 300 to 774. Modified software will permit operating outside of the floating vector address range.

To set the vector address, set a bit to "0" by setting the related switch to the ON position. Set the switch to the opposite ("off") position to set the bit to a "1". (See table 2.)

Table 2 Interrupt Vector Address Selection

Digit X		Digit Y	
Sw. No. 6 5 4	Value	Sw. No. 3 2 1	Value
● ● ●	0	● ● ●	0
● ● o	1	● ● o	1
● o ●	2	● o ●	2
● o o	3	● o o	3
o ● ●	4	o ● ●	4
o ● o	5	o ● o	5
o o ●	6	o o ●	6
o o o	7	o o o	7
Address = XY0			
o = switch open		● = switch closed (ON)	

Connecting Jumpers

There are five jumpers on the module to be connected, or disconnected, to adapt the DUV11 for specific modem configurations. Jumpers are as follows:

Jumper	Location	Comments
1	6C	Furnished open. Connect to permit synchronization on a single sync character, instead of on two consecutive sync characters (Internal Synchronous mode).
2	1E	Furnished closed. Open to disconnect secondary receive channel between modem and DUV11.
3	1F	Furnished closed. Open to disconnect secondary transmit channel between modem and DUV11.
4	1F	Furnished open. Connect jumper to ground, as required, to connect modem chassis ground to DUV11 signal ground.
5	9F	Furnished closed. Open to prevent clearing of RTS, DTR, and SXD bits in the Receiver Status Register.

Installing MLSI-DUV11 Module

Install the module in any quad I/O slot in the backplane/cardguide assembly. Device priority is determined by the position of the module in the backplane, with highest priority given the module nearest the KD11 processor module, with decreasing priority in successively lower slots.

Connect the serial interface cable between the Berg connector on the DUV11 module and modem. This cable should not be longer than 25 feet. Table 3 lists, and gives pin numbers for, the different signals carried on the serial interface cable.

Table 3 Interface/Modem Connector Pins

Pin	Source	Signal	Pin	Source	Signal
1		Chassis Gnd (jumpered)	17		not used
2		Gnd	18		Gnd
3	DUV11	Transmit Data	19		not used
4	Gnd		20		Gnd
5	Modem	Receive Data	21		not used
6		Gnd	22		Gnd
7	DUV11	Request to Send	23		not used
8		Gnd	24		Gnd
9	Modem	Clear to Send	25		not used
10		Gnd	26		Gnd
11	Modem	Data Set Ready	27	DUV11	Sec. Transmit Data (jumpered)
12		Gnd	28		Gnd
13		Signal Ground	29	Modem	Transmit Clock
14		Gnd	30		Gnd
15	Modem	Carrier	31	Modem	Secondary Receive Data
16		Gnd	32		Gnd

Table 3 Interface/Modem Connector Pins (cont'd)

Pin	Source	Signal	Pin	Source	Signal
33	Modem	Receive Clock	42		Gnd
34		Gnd	43	Modem	Ring
35		not used	44		Gnd
36		Gnd	45		not used
37		not used	46		Gnd
38		Gnd	47	DUV11	External Clock
39	DUV11	Data Terminal Ready	48		Gnd
40		Gnd	49		not used
41		not used	50		Gnd

BACKPLANE INTERFACE TERMS

Table 4 lists and defines signals at the backplane connector of the DUV11 module.

Table 4 Backplane Interface Terms

Bus Pin	Term	Description
CL2	BIRQL	**Interrupt request.** Asserted by processor-enabled interrupt enable and interrupt request logic. Indicates to processor that data may be input or output. Program status word bit 7 must be "0" in order for processor to acknowledge an interrupt request.
CH2	BDINL	**Data input.** When BSYNCL is asserted, indicates an input transfer from the active bus master. When BSYNCL is *not* asserted, implies that an interrupt operation is in process.
CE2	BDOUTL	**Data output.** Implies that valid data is available on lines BDAL0L through DBAL15L and, with reference to the bus master device, that an output transfer is in process. The slave device responding to BDOUTL must assert BRPLYL to complete the data transfer.
CP2	BBS7L	**Bank 7 select.** Indicates that address on the bus is from the upper 4k bank (28k-32k). With BSYNCL asserted, BBS7L remains active until addressing of bus cycle is completed.
CT2	BINITL	**Initialize.** Generated by processor during a power-up operation. Clears all devices on the I/O bus.
CF2	BRPLYL	**Reply.** Generated by DUV11 in response to either BDINL or BDOUTL. Indicates that input data is available on the BDAL bus, or that output data has been accepted from the bus.
CM2	BIAKIL	**Interrupt acknowledge.** Asserted by processor in response to BIRQL.

Table 4 Backplane Interface Terms (cont'd)

Bus Pin	Term	Description
CN2	BIAKOL	**Interrupt acknowledge out.** Normally asserted to device having next-lower priority on interrupt chain, and appears at BIAKIL input to that device. If DUV11 stores an interrupt request, BIAKOL is negated at the next device.
CJ2	BSYNCL	**Synchronize.** Asserted by bus master device when it has placed an address on lines BDAL0L through BDAL15L.
CK2	BWTBTL	**Byte Control.** Asserted during address time to indicate that an output sequence is to follow (DATO or DATOB). Also asserted during data time for byte addressing in DATOB cycle.
	BDALnL	**Data/address.** One of the 16 data/address bus lines used to transfer all address and data information.
CU2	BDAL0L	Bit 0.
CV2	BDAL1L	Bit 1.
DE2	BDAL2L	Bit 2.
DF2	BDAL3L	Bit 3.
DH2	BDAL4L	Bit 4.
DJ2	BDAL5L	Bit 5.
DK2	BDAL6L	Bit 6.
DL2	BDAL7L	Bit 7.
DM2	BDAL8L	Bit 8.
DN2	BDAL9L	Bit 9.
DP2	BDAL10L	Bit 10.
DR2	BDAL11L	Bit 11.
DS2	BDAL12L	Bit 12.
DT2	BDAL13L	Bit 13.
DU2	BDAL14L	Bit 14.
DV2	BDAL15L	Bit 15.

THEORY OF OPERATION

GENERAL

This section describes the theory of operation of the DUV11 in each of the principal modes of operation, and describes each functional area of logic. Refer to logic diagrams included in this manual.

COMMUNICATION MODES

The DUV11 may be program-controlled to operate in any one of three modes of operation: isochronous, internal synchronous, and external synchronous.

Isochronous Serial Transmission

In isochronous operation, each data character is framed by a Start bit, and followed by one, one-and-a-half, or two Stop bits. The Start bit causes the receiver to prepare to count data bits (5, 6, 7, or 8), and the Stop bit causes the receiver to check synchronization following each character. The receiver and transmitter operate with a common clock furnished by the modem.

Internal Synchronous Serial Transmission

In this mode of operation, Start and Stop bits are not used. Instead, the entire message is preceded on the line by a synchronizing code (sync characters). When the receiver recognizes the codes it locks on the bit stream and, using a counter, assembles characters from subsequent bits. Note that data transmission must be continuous if synchronization is to be maintained.

External Synchronous Serial Transmission

This mode of operation is similar to Internal Synchronous operation, except that synchronization is performed by the program as it examines parallel-bit words assembled from the incoming serial data. The program asserts a flag bit to the DUV11 when synchronization is achieved.

MAINTENANCE MODES

The DUV11 hardware offers use of three program-controlled maintenance modes. The user may write diagnostic software utilizing these valuable facilities.

System Test

The System Test mode exercises the entire DUV11. An internally generated clock is used to clock the receiver and transmitter, as a character is loaded from the program and then transmitted. The transmitted output is routed to the receiver, and the program monitors the receiver output and control functions for errors.

Internal Loop Test

This test is run to isolate a fault in the DUV11 to one of several functional areas of logic. Clock and data channels between the DUV11 and the modem are inhibited.

The program loads a test character for transmission, and then furnishes a single-step clock which transmits the test character to the receiver. The receiver output and control functions are monitored for errors.

External Loop Test

This test detects faults in the serial interface circuits and cable. The test is performed with the modem disconnected from the interface cable, and with a test connector installed on the modem end of the cable.

A programmed single-step clock is routed to simulate receive and transmit clocks normally received from the modem. A programmed test character is loaded for transmission, and then transmitted to the receiver through the modem interface circuits, interface cable, and the special test connector. The program monitors the receiver output and control functions for errors.

This test also checks-out modem control lines.

BLOCK DIAGRAM

Figure 1 shows the general organization of logic in the MLSI-DUV11.

Information that programs the operation of the unit is taken from the data/address bus and loaded (by commands decoded along with the device address) into transmitter and receiver status registers, and into the parameter status register in transmitter logic.

Transmitter and receiver status registers are read/write registers, and logic and modem status may be read from these registers onto the bus.

Bits loaded into the parameter status register determine the communications mo[illegible] (isochronous operation, or internal or external synchronous operation), select parity generation and sensing, select serial word length, and specify the synchronization code to be matched for internal synchronization.

Data is multiplexed onto the bus by commands decoded along with the device address, and may be the contents of either the receiver or transmitter status register, the interrupt vector address, or parallel-bit characters assembled from the received serial data.

Bits loaded into the transmitter status register control the routing of serial data and clocks when a maintenance mode is selected.

The following paragraphs describe the contents of each programmed register, outline the general sequence of operation, and describe the operation of logic in each block shown in the block diagram.

REGISTER DEFINITIONS

The following paragraphs define the meaning and function of each bit or character that may be contained in the Receiver Status Register, the Receiver Data Buffer Register, the Parameter Status Register, and the Transmitter Status Register.

Receiver Status Register (RXCSR (16XXX0)

The read/write register RXCSR is programmed, or monitored, to communicate status and control information between the processor and the modem interface.

The RXCSR register may be loaded by word or by byte. When the register is to be loaded in two bytes, the bus signal BWTBTL is asserted by the processor along with the data on the bus. Data bit BDAL00 (A00) is used, along with BWTBTL (BYTE), to select the portion of the register into which the data is to be loaded. That is, when A00 is true, the lower half of the register is loaded and, when $\overline{A00}$ is true, the upper half is loaded. The load logic is strobed by RXCSR0, decoded from the device address.

The contents of the RXCSR register are as follows. (The letter R indicates a modem status bit which is only read by the processor. Other bits may be either loaded from the bus, or read to the bus. A "not used" bit is read as 0.)

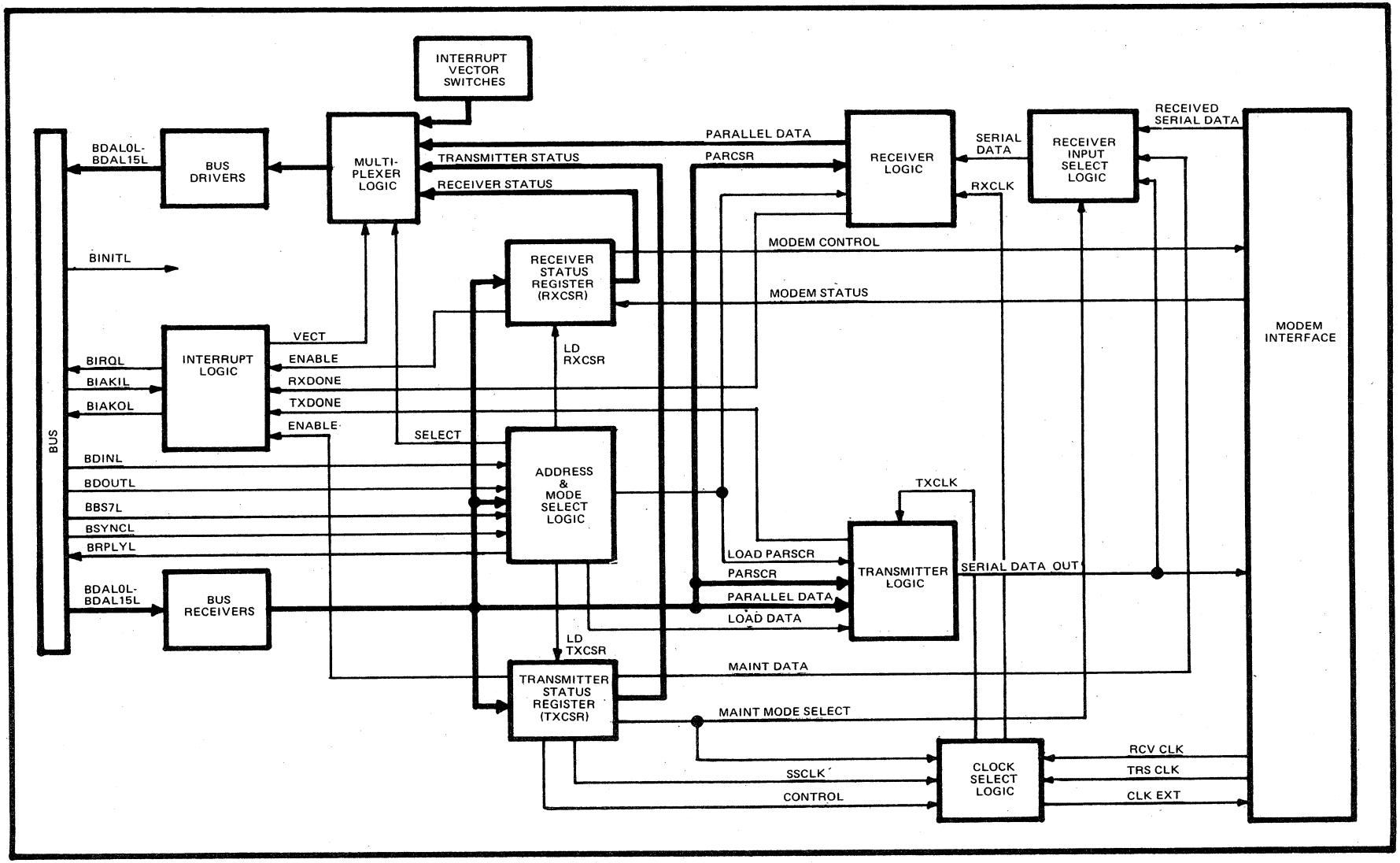


Figure 1 MLSI-DUV11 Synchronous Serial Line Interface, Simplified Block Diagram

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DATSET CH	RING	CTS	CARR	RCVR ACT	SECRV	DSR	STRP SYNC	RX DONE	RCVR INB	DSET INB	SCH SYNC	SXD	RTS	DTR	NOT USED
R	R	R	R	R	R	R		R							

Table 5 lists and defines the functions of each bit in the read/write RXCSR Register.

Table 5 Receiver Status Register Bits

Bit	Bit Name	Description
15	DATSETCH (read only)	**Data Set Change.** Set to indicate a change in modem status. Set by a transition of any of the following lines: - Ring - Clear To Send - Carrier - Secondary Received Data - Data Set Ready A receiver interrupt will be generated if this bit is set while DSETINB is set. DATSETCH is cleared by BINITL or a programmed MRST signal, or by the RXCSR read strobe.
14	RING (read only)	**Ring.** When set, indicates that a Ring signal is being received from the modem.
13	CTS (read only)	**Clear to Send.** When set, indicates that the modem is on and ready to accept data for transmission.
12	CARR (read only)	**Carrier.** When set, indicates that the modem carrier is on.
11	RCVRACT (read only)	**Receiver Active.** When the Controller is operating in the internal synchronous mode, this bit is set when the correct number of contiguous SYNC characters (one or two) has been received. If operating in the external synchronous or isochronous mode, this bit follows the SCH SYNC bit. RCVRACT is cleared by BINITL or a programmed MRST signal, or by a transition (from 1 to 0) of SCH SYNC.
10	SECRV (read only)	**Secondary Receive Data.** Set when the Secondary Receive Data line from the modem is asserted.
09	DSR (read/write)	**Data Set Ready.** When set, indicates that the modem is on and ready to transmit and receive data.
08	STRPSYNC (read/write)	**Strip Sync.** If bit is set, a received character that matches the contents of the Sync register does *not* cause a receiver interrupt unless a receiver error has been detected (RXDBUF bit 15 is 1). STRIP SYNC is cleared by BINITL or a programmed MRST signal.

Table 5 Receiver Status Register Bits (cont'd)

Bit	Bit Name	Description
07	RXDONE (read only)	**Receiver Done.** Set when synchronization has been achieved and a character has been loaded into the RXDBUF register, unless STRIP SYNC is set. If STRIP SYNC is set and the character is a sync character received without errors, RX DONE is *not* set. When RXDONE is set (and if RVRINB has been set), a receive interrupt request is generated. RXDONE is cleared by BINTL or a programmed MRST signal, or by the RXDBUF read strobe.
06	RVRINB (read/write)	**Receiver Interrupt Enable.** When set, permits a receiver interrupt request to be generated when the RX DONE bit is set.
05	DSETINB (read/write)	**Data Set Interrupt Enable.** When set, permits a receiver interrupt request to be generated when the DATSETCH bit is set. Cleared by BINTL or a programmed MRST signal.
04	SCHSYNC (read/write)	**Search Sync.** When the Controller is operating in the internal synchronous mode, this bit enables receiver synchronization logic and causes the receiver to compare incoming bits with the contents of the Sync register, in search for a Sync character. In isochronous operation, SCHSYNC enables logic that generates the RXDONE flag. In external synchronous operation, SCHSYNC enables logic that generates the RXDONE flag and causes the receiver to frame incoming characters. SCHSYNC is cleared by BINTL or a programmed MRST signal.
03	SXD (read/write)	**Secondary Transmit Data.** When set, asserts STD line to the modem. Bit controls channel for transmitting supervisory information from the modem to the processor. Optionally cleared by BINITL or programmed MRST signal (selected by jumper 5).
02	RTS (read/write)	**Request to Send.** When set, causes Controller to assert RTS line to the modem. Indicates to modem that data is to be transmitted. Cleared by BINITL or MRST.

Table 5 Receiver Status Register Bits (cont'd)

Bit	Bit Name	Description
01	DTR (read/write)	**Data Terminal Ready.** When set, indicates that the Controller is on, programmed, and ready to receive data from the modem. The DTR line asserted to the modem permits the Controller to be connected to the channel. Optionally cleared by BINITL or programmed MRST signal (selected by jumper 5).

Receiver Data Buffer Register RXDBUF (16XXX2)

This read-only register (located in the LSI receiver device) stores the parallel-bit character accumulated from the serial input. The stored data appears on lines IDATA00 through IDATA07, simultaneously with receiver status bits.

The RXDBUF register is controlled by signals generated within the receiver device. The $\overline{\text{RXDBUFI}}$ strobe derived from the device address is used to clear the RXDONE flip-flop after the data has been accepted by the processor.

The contents of the RXDBUF register are as follows. (A "not used" bit is read as a 0.)

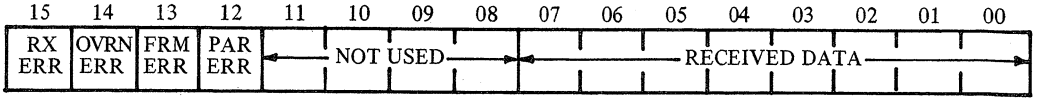


Table 6 lists and defines the functions of each bit in a read-only RXDBUF Register.

Parameter Status Register PARSCR (16XXX2)

The Parameter Status Register (located in the LSI transmitter device) is loaded, by strobe $\overline{\text{PARSCR0}}$, with bits specifying the operating parameters of the Controller: That is, synchronous or isochronous operation, word length, parity, parity sense, and sync character configuration. A "not used" bit is loaded with a 0. The contents of the PARSCR register are as follows.

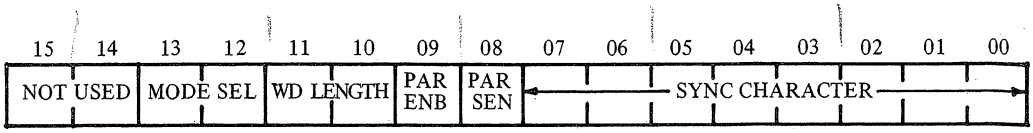


Table 6 Receiver Data Buffer Register Bits

Bit	Bit Name	Description
15	RXERR	**Receiver Error.** Set whenever OVRNERR, FRMERR, OR PARERR is set. Cleared only when all receive error bits are cleared.
14	OVRNERR	**Overrun Error.** When set, indicates that a character has been loaded into the RXDBUF register before the previously loaded character was serviced by the processor, and the previously loaded character has been lost. The time period (in seconds) during which the RXDONE flag may be serviced is: 1/baud rate X bits per character. Cleared by BINITL or programmed MRST signal, or by the RXDBUF read strobe.
13	FRMERR	**Framing Error.** When set, indicates that the received character was not terminated by a valid Stop bit. This error will occur only in the isochronous mode of operation. Cleared by BINITL or a programmed MRST signal, or by the RXDBUF read strobe.
12	PARERR	**Parity Error.** When set, indicates that the parity of received character does not agree with the programmed odd or even parity. PARERR is always 0 if parity is not programmed. Cleared by BINITL or a programmed MRST signal, or by the RXDBUF read strobe.
07-00	Rcvr Data	**Receiver Data.** The data character to be transferred to the processor. The buffer is right-justified for 5, 6, 7, or 8 bits. The parity bit (if used) is loaded into the next vacant higher-order bit position. For example, a 5-bit character plus parity bit is loaded with the parity bit in position 05 in the buffer. The parity bit is not loaded for an 8-bit character. Rcvr Data is not cleared. BINITL or a programmed MRST signal sets the buffer to *all 1's.*

Table 7 lists and defines the functions of the bits in the write-only PARSCR Register.

Table 7 Parameter Status Register Bits

Bit	Bit Name	Description
13, 12	MODE SEL	**Mode Select.** These bits encode the required mode of operation, as follows: Mode — Bit 13 — Bit 12 Internal Synchronous — 1 — 1 External Synchronous — 1 — 0 Isochronous — 0 — 0 The unused code 01 will produce errors in the Controller.
11, 10	WD LENGTH	**Word Length.** These bits encode the number of bits (excluding parity) in the characters received and transmitted. Word length is encoded as follows: No. of Bits — Bit 11 — Bit 10 5 — 0 — 0 6 — 0 — 1 7 — 1 — 0 8 — 1 — 1
09	PAR ENB	**Parity Enable.** When this bit is set, parity is generated by the transmitter and checked by the receiver.
08	PAR SEN	**Parity Sense.** This bit controls the sense of the parity bit (odd or even). When PAR SEN is set, *even* parity is generated and checked by the Controller. When this bit is 0, *odd* parity is generated and checked.
07-00	SYNC CHAR	**Synchronization Character.** This 8-bit register specifies the sync character to be used in achieving synchronization with received serial information. In synchronous operation, the transmitter uses the sync character as a "fill" character when the program's output rate is too low to maintain continuous transmission. The period (in seconds) at the required output rate is equal to: $(\frac{1}{\text{baud rate}} \times \text{bits per character}) - 1/2 \text{ bit period.}$

Transmitter Status Register TXSCR (16XXX4)

The Transmitter Status Register is a read/write register for programming and monitoring the Controller's transmitter. The TXCSR Register can be loaded by word or by byte. When the register is to be loaded in two bytes, the bus signal BWTBTL is asserted by the processor along with the data on the bus. Data bit BDAL00 (A00) is used, along with BWTBTL. (BYTE), to select the portion of the register into which the data is to be loaded. That is, when A00 is true, the lower half of the register is loaded and, when $\overline{A00}$ is true, the upper half is loaded. The load logic is strobed by TXCSR0, decoded from the device address.

The contents of the register are as follows. (The letter R indicates a bit which is only read by the processor, and the letter W indicates a bit which is only written. Other bits may be either loaded from the bus, or read onto the bus. A "not used" bit is read as a "0".

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DNA	MIT DAT	SS CLK	MS 01	MS 00	RX INP	NOT USED	MRST	TX DONE	TX INB	DNA INB	SEND	HLF DUP	NOT USED		BREAK
R							W	R							

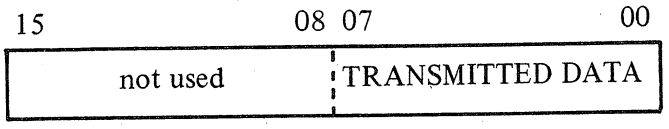
Table 8 lists and defines the functions of each bit in the read/write TXCSR Register.

Transmitter Data Buffer Register TXDBUFR (16XXX6)

The Transmitter Data Buffer Register is a write-only register which stores one character as it waits for transfer to the transmitter register. The XMTR DATA character may contain 5, 6, 7, or 8 bits and is right-justified (with the least-significant bit in position 00).

Bit positions 08 through 15 are never used.

XMTR DATA is not cleared. Instead, BINITL or a programmed MRST signal sets *all 1's* in this register.



OPERATING SEQUENCE

Initialization and Program Loading

Before the DUV11 can handle data, it must be initialized and programmed. As the unit is initialized (by BINITL), control flip-flops are cleared, and the receiver and transmitter are set to the idle state. With the receiver idle, RXDBUF is set to all 1's. With the transmitter idle, its output is a Mark (high) level.

The DUV11 is programmed by putting the device address on the bus, and storing address recognition and a register load bit. The control/status word is then put on the bus and the stored load bit is strobed to the appropriate register to load that register.

The RXCSR, TXCSR, and PARSCR registers must all be programmed, in any sequence, before data transfer can begin.

Table 8 Transmitter Status Register Bits

Bit	Bit Name	Description
15	DNA (read only)	**Data Not Available.** Set (in synchronous mode operation only) by the transmitter when a "fill" character is transmitted. A fill character is transmitted when the program responds to a TXDONE interrupt request too late to maintain continuous data transmission. If DNA is set while bit 05 is set, a transmitter interrupt request is generated. DNA is cleared by BINITL or a programmed MRST signal (bit 08), or by the TXCSR read strobe.
14	MITDAT (read/write)	**Maintenance Data.** Used by the diagnostic program (in the internal and external loop maintenance modes) to simulate an input to the receiver. Cleared by BINITL or a programmed MRST signal (bit 08).
13	SSCLK (read/write)	**Single-Step Maintenance Clock.** Used by the diagnostic program (in the internal and external loop maintenance modes) to simulate transmitter and receiver clocks. Cleared by BINITL or a programmed MRST signal.
12, 11	MS01/MS00 (read/write)	**Maintenance Mode Select 01 and 00.** Used to select either normal operation, or one of three maintenance modes. Modes are selected as follows: Mode — Bit 12 — Bit 11 Normal — 0 — 0 Internal Maintenance Loop — 0 — 1 External Maintenance Loop — 1 — 0 System Test — 1 — 1
10	RX INP (read only)	Indicates receiver input in internal and external loop maintenance modes.
08	MRST (write only)	**Master Reset.** Set to generate a pulse which initializes the registers, the receiver, and the transmitter, and inhibits the DURPLY acknowledgement to the processor.
07	TXDONE (read only)	**Transmitter Done.** Set by BINITL or programmed MRST signal after the word to be transmitted has been transferred from the TXDBUF register to the transmitter register, and the first bit of the parallel character is put on the serial transmitter data line. Indicates that the next character may be transferred to the TXDBUF register. If TXDONE is set while TXINB (bit 06) is set, a transmitter interrupt request is generated.

Table 8 Transmitter Status Register Bits (cont'd)

Bit	Bit Name	Description
		TXDONE is cleared only by the TXDBUF load strobe.
06	TXINB (read/write)	**Transmitter Interrupt Enable.** This bit is set to allow the TX DONE bit to generate a transmitter interrupt request. Cleared by BINITL or a programmed MRST signal.
05	DNA INTEB (read/write)	**Data Not Available Interrupt Enable.** When set, allows a set DNA bit to generate a transmit interrupt request.
04	SEND (read/write)	Send. When set, enables the transmitter. Transmission will begin as a character is loaded into the TXBUF register. SEND must remain set until the entire message has been transmitted. If SEND is cleared, the transmitter completes sending the character in the transmitter register and then enters the idle state. SEND is cleared by BINITL or a programmed MRST signal.
03	HLFDUP (read/write)	**Half Duplex.** When set, causes communications in the half-duplex mode. That is, the receiver is disabled whenever SEND (bit 04) is set. Cleared by BINITL or a programmed MRST signal.
00	BREAK (read/write)	When BREAK is set, the serial transmitter output line is held in the Space (low) state. Transmitter operation otherwise proceeds normally. BREAK is used in diagnostic operation to inhibit the transmitter output while data is input to the receiver (using MITDAT bit 14 of this register). Cleared by BINITL or a programmed MRST signal.

Communications Interface Handshaking

In a typical communications sequence, handshaking between two DUV11's through a modem attached to each (figure 2) is as follows:

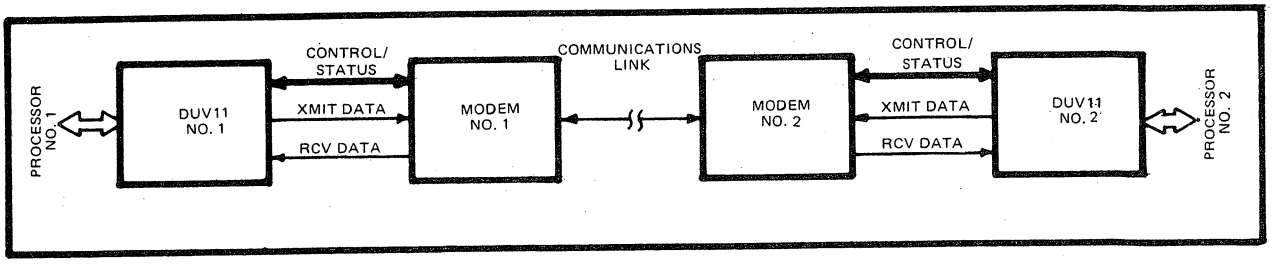


Figure 2 Simplified Communications Interface Diagram

Assume that Modem No. 2 calls Modem No. 1.

a. Modem No. 1 asserts RING at DUV11 No. 1.

b. DUV11 No. 1 assets DTR to Modem No. 1.

c. Modem No. 1 sends CARRIER to Modem No. 2.

d. Modem No. 2 asserts CARRIER to DUV11 No. 2, and to Modem No. 1.

e. Modem No. 1 asserts CARRIER to DUV11 No. 1, and DUV11 No. 2 sets Search Sync bit (SCHSYNC) to enable its receiver.

f. DUV11 No. 1 asserts RTS to Modem No. 1.

g. Modem No. 1 asserts CTS to DUV11 No. 1.

h. DUV11 No. 1 sets SEND and TXINB bits to enable its transmitter.

Note that a modem must assert DSR before any data communications can begin.

Supervisory data may be transmitted (through SECXMIT and SECREC lines) simultaneously with normal communications.

All communications can be terminated by the program clearing the DTR (Data Terminal Ready) bit in the Receiver Status Register.

Basic Transmitter Sequence

Data communications can be initiated either by the program, or by the modem operator.

The program initiates communication by loading a data character into the TXDBUF register. If the transmitter is enabled (SEND is asserted), the transmitter sends the character to the remote modem and then asserts TXDONE to request the next character. If the program has another character to send, it responds by transferring the next character to the TXDBUF register. If the last character of the message has been transferred and transmitted, the program clears SEND.

To initiate communication, the modem operator dials a remote station. The remote modem detects the call and responds by asserting RING to its DUV11, which generates a receiver interrupt. The processor at the remote station then branches to an interrupt service subroutine which directs the handshaking sequence and loads and transmits a character to the calling modem.

If, during synchronous transmission, the program fails to load the TXDBUF register before the last bit of the current character is transmitted, a SYNC character (stored in the PARSCR register) is inserted in the TXDBUF register to maintain synchronization at the receiving station. If isochronous operation is programmed, the transmitter simply holds a Mark level on the line until the program loads the next character.

Basic Receiver Sequence

Data reception is initiated when serial data is received at the receiver, and if the SCHSYNC bit has been set in the Receiver Status Register.

In the Internal Synchronous mode of operation, SCHSYNC enables the receiver to compare incoming bits with the stored Sync code, in order to establish receiver synchronization. When synchronization has been achieved, RCRVACT is asserted to enable the receiver to frame incoming data bits. Each framed character is transferred to the RXDBUF register, RXDONE is asserted, and a receiver interrupt request is asserted.

The processor branches to an interrupt service subroutine and reads the contents of the RXDBUF register from the bus. If the program fails to read the RXDBUF register before the next character is transferrred to the register, the character previously in the register is lost and OVRNERR is set in the Receiver Status Register.

In External Synchronous operation, SCHSYNC enables generation of the RXDONE flag, and RCVRACT follows SCHSYNC. Subsequent operation is generally the same as for Internal Synchronous operation.

In Isochronous operation, synchronization is obtained using Start and Stop bits in the serial format, and SCHSYNC only enables generation of RXDONE. RCVRACT follows SCHSYNC.

LOGIC DESCRIPTION

The following paragraphs describe the operation of each major functional area of MLSI-DUV11 logic. Refer to the logic diagram (Dwg. No. 40326) in this manual.

Master Reset Logic

The DUV11 logic may be initialized by either BINITL asserted by the processor, or by bit BDAL08L (DATA08) asserted along with a TXCSR0 command. In the latter case, logic is re-initialized after the upper byte of an output word has been stored in the output register (by UTXCS0).

The pulse MRST is triggered by DATA08•UTXCS0, and is ORed with the INIT signal derived from BINITL. The resulting signal MR, and its complement, clear registers and control flip-flops throughout the DUV11.

If jumper No. 5 is connected to the MR line, the term OPTMR will reset the output register device storing DTR, RTS, and SXD commands. This device is otherwise not reset but only updated by a new output word.

Logic needed to acquire communications with the bus; that is, device address logic; is initialized by PWRON as the +5V dc voltage supply reaches a useable level following application of power.

Device Address Logic

The device address appears on the data/address bus (BDAL3L-BDAL12L) and is compared with the 10-bit address preset by switches on a DIP module (14B). The comparison is performed by exclusive-OR gates which output to a common point.

When the address on the bus matches the preset address, flip-flop 7B/5 is enabled and then set by the rise of SYNC (BSYNCL). This asserts ADR at the decoder strobe gate 7A/11.

The decode strobe is controlled by logic which holds the stobe true throughout the duration of a DATIN command, but turns off the strobe as soon as RPLY is asserted following a DATOUT command.

RPLY (DURPLY) is asserted when any of the following is asserted: DATIN, DATOUT, VECT, or MR. The negation of RPLY indicates that an operation has been completed by the DUV11.

The stobed command decoder (4A) decodes stored DAT01 and DAT02 bits, and DATOUT, to assert (low) one of seven control signals as follows:

- $\overline{\text{RXCSRI}}$ .... Clears DATA SET CH bit in the Receiver Status Register.
- $\overline{\text{RXCSRO}}$ .... Loads the Receiver Status Register.
- $\overline{\text{RXDBUFI}}$ .... Sets RXDONE, if the receiver holds a word ready for transfer to the processor.
- $\overline{\text{PARCSRO}}$ .... Loads into transmitter and receiver devices the word controlling serial format parameters.
- $\overline{\text{TXCSRI}}$ .... Interrogates the Data Available status of the transmitter device.
- $\overline{\text{TXCSRO}}$ .... Loads the Transmitter Status Register.
- $\overline{\text{TXDBUF}}$ .... Loads transmitter buffer register with bits 0 through 7 of the character to be transmitted.

Interrupt Request Logic

Interrupt Request Logic, along with Interrupt Vector Logic, permits the DUV11 to gain control of the bus, and to cause the processor to branch to an interrupt vector for routines for servicing the DUV11.

The following paragraphs describe logic that generates a receive interrupt, and a transmit interrupt.

Receive Interrupt Request Logic – A receive interrupt request (RXINTR) is generated when either:

a. The RVRINB (receiver interrupt enable) bit is set in the RXCSR register, and RXDONE is asserted indicating that a received word is ready for transfer to the processor.

b. The DSETINB (dataset interrupt enable) bit is set in the RXCSR register, and DATSETCH is asserted indicating a status change at the modem interface.

Either event sets flip-flop 3C/9, asserting RXINTR and IRQST. IRQST is applied through a bus driver to the BIRQL line. The flip-flop is cleared after the processor has acknowledged the request and RCVRVECT puts the vector on the bus.

Transmit Interrupt Request Logic – A transmit interrupt request (TXINTR) is asserted when either:

a. The TXINB (transmit interrupt enable) bit is set in the TXCSR register, and TXDONE is asserted indicating that a word has been transmitted and the next word may be transferred into the transmitter register.

b. The DNAINB bit is set in the TXCSR register, and a "fill" character is to be transmitted (DNA is asserted).

Either event sets flip-flop 3C/5, asserting TXINTR. TXINTR is ORed to the IRQST line along with the RXINTR signal. The flip-flop is cleared when the interrupt has been acknowledged and the vector is placed in the bus (TXRVECT is asserted).

Interrupt Vector Control Logic

This logic permits a DATIN instruction to cause the assigned vector address to be put on the bus, in response to a stored interrupted request.

When a receive-interrupt request is stored (RXINTR), the DATIN signal sets flip-flop 2A/5. If the DUV11 has priority (BIAKIL is asserted at the interface), term IAKI enables the flip-flop output to assert RCVRVECT. RCVRVECT appears as a bit in the vector address word, and gates the vector address bits onto the bus. At the same time, VECT causes DURPLY to be asserted on the BRPLY interface line.

Similarly, the stored transmit-interrupt request (TXINTR) permits DATIN to set flip-flop 2A/9, asserting TXRVECT. TXRVECT gates the vector address to the bus.

IAK0 (BIAK0L) is asserted whenever BIAKL is asserted and no interrupt request is stored in the DUV11. BIAKOL grants priority to the device having the next-lower priority. Whenever either RCVRVECT or TXRVECT is asserted, however, IAK0 (BIAKOL) is negated, reserving priority to the DUV11.

The vector address is programmed by setting six switches in location 8D. A switch in the "off" position leaves the related multiplexer input high (bit true) while the interrupt vector is asserted, and the switch in the ON position grounds the Multiplexer input. Bit DATI02 is used to distinguish a receive interrupt vector from a transmit interrupt vector; that is, that bit is *false* for a receive interrupt vector, and *true* for a transmit interrupt vector.

The interrupt vector address is put on the bus as a result of the following cycle:

1. Interrupt request IRQSI is stored, and sent (as BIRQL) to the processor, and either RXINTR or TXINTR is asserted.

2. The processor responds to the request by asserting BIAKIL and then BDINL.

3. The vector address is placed on the bus, and VECT causes the term DURPLY to assert BRPYL on the bus.

Multiplexer Logic

The multiplexer selects any one of four words onto the DATIn lines. The DATIn bits are applied through drivers to the bus.

Each type of word is selected by bits BDAL01L and BDAL02L (A01 and A02) on the bus, or by a DATIN command sent in response to an interrupt request (refer to *Interrupt Vector Control Logic*).

Conditions for selecting each type of word are as follows:

- Vector Address . . . . . RCVRVECT + TXRVECT + (A01•A02)
- Receiver Data Buffer Register . . . . . A01•$\overline{A02}$
- Transmitter Status Register . . . . . $\overline{A01}$•A02
- Receiver Status Register . . . . . $\overline{A01}$•$\overline{A02}$.

The selected bits are enabled onto the DATI lines by $\overline{GDATI}$ which is RCVRVECT or TXRVECT for the vector address; or by A01 or A02 true, along with ADR and DATIN true, for other input words.

Note that when RCVRVECT is not asserted, levels on "3" inputs to the multiplexer devices are held true. This is to provide the "all ones" vector compatible with existing DEC operating systems.

Modem Electrical Interface

The serial interface connector links the DUV11 with either a Bell 303/313-type, or 201/202-type dataset. At circuits that receive signals from the dataset, 110-ohm pull-down resistors are required only for 303/313-type units, and are omitted for other types.

There are two possible sets of output drivers, with the appropriate device type selected to match dataset requirements. For a 303/313-type dataset, 8T13 TTL drivers are used. For 201/202-type units, 1488 devices (open-collector outputs) are installed in the DUV11.

Modem Interface Input Logic

Input circuits condition and remove transients from input signals. Control signals are sensed for level transitions to detect any change of status. Sensing is performed for each input control signal by a circuit comprising an inverter, a delay network, and an exclusive-OR gate. Gate outputs are wire-ORed so that any status change causes flip-flop 5C/5 to assert the DATSETCH flag at the multiplexer as bit 15 of the RXCSR word. This bit raises a receive interrupt request.

The DATSETCH flag is cleared by the RXCSR read strobe RXCSRI which is generated when the interrupt request is serviced.

In normal operation the received serial data (REC DATA) is gated to the SDI line (to the receiver device). In test and maintenance modes REC DATA is inhibited and test signals are sent to the receiver device (refer to *Receiver Data Source Select Logic*).

The received clocks (TRS CLOCK and TEC CLOCK) are gated to transmitter and receiver devices in normal operation (refer to *Clock Source Select Logic*).

Receiver Data Source Select Logic

Receiver data (SDI) may originate at one of three possible sources, as follows:

- at the modem (REC DATA);
- in the program (MITDAT bit in the TXCSR word);
- at the DUV11 transmitter (SDO).

The source is selected by bits 12 and 13 in the TXCSR word as follows:

Mode	MS00	MS01	Source
Normal	0	0	Modem
Internal Loop	1	0	MITDAT or SDO
External Loop	0	1	MITDAT, or modem (looping from transmitter via test cable)
System Test	1	1	SDO

Note that in either maintenance mode both MITDAT and SD0 can be simultaneously applied to the receiver. The programmer must be sure that only one source is active. The BREAK bit is to be set in the TXCSR register to inhibit the transmitter output when MITDAT is set.

Clock Source Select Logic

The receiver and transmitter clocks are selected from different sources for different modes of operation. Selection is controlled by MS00 and MS01 bits (from the TXCSR register) which select sources at multiplexer device 13D. The selected receiver clock is gated to the RXCLK line, and the selected transmitter clock is gated to the TXCLK line.

Sources are as follows:

- the modem (REC CLOCK and TRS CLOCK)
- the program (SSCLK bit in the TXCSR register)
- an internally generated clock TSTCLK (approximately 2 kHz).

Mode	MS00	MS01	Transmitter Source	Receiver Source
Normal	0	0	REC CLOCK	TRS CLOCK
Internal Loop	1	0	SSCLK	SSCLK
External Loop	0	1	REC CLOCK Clocks are SSCLK, gated through EXTERNAL CLOCK output and cable looped back to REC CLOCK and TRS CLOCK inputs.	TRS CLOCK
System Test	1	1	TSTCLK	TSTCLK

Note that HLFDUP (the half-duplex control bit from the TXCSR register) disables the REC CLOCK input while the DUV11 is transmitting in the half-duplex mode (SEND is asserted).

Transmitter Logic

Transmitter logic comprises a PT148B Synchronous Transmitter device, flip-flop 7D, and associated gates. (See figure 3.)

The transmitter control register (PARCSR) is loaded by the decoded $\overline{\text{PARCSRO}}$ command received with the device address. The bits loaded from the bus control the following parameters of transmitter operation:

a. Mode of operation (Internal or External Synchronous, or Isochronous).

b. Data character length (5, 6, 7, or 8 bits).

c. Parity enabled or disabled.

d. Parity sense (odd or even).

e. Sync character code (used as a "fill" character in Synchronous mode transmission.

Synchronous Mode Transmission – In either Internal or External Synchronous mode of operation, the character on the bus is loaded into the TXDBUF register by $\overline{\text{TXDBUFO}}$ (decoded from the device address and DATOUT). If the SEND bit is set, TXCLK shifts the stored bits out to the TRO line and transmitter output gating.

As the last bit of the character is transmitted, the TXDONE flag is raised to generate a transmitter interrupt request for the next character.

If the program still holds SEND set, but does not load a new character before the current character is transmitted, the internally stored sync character is multiplexed to the transmitter register and transmitted in order to maintain synchronization at the receiving station.

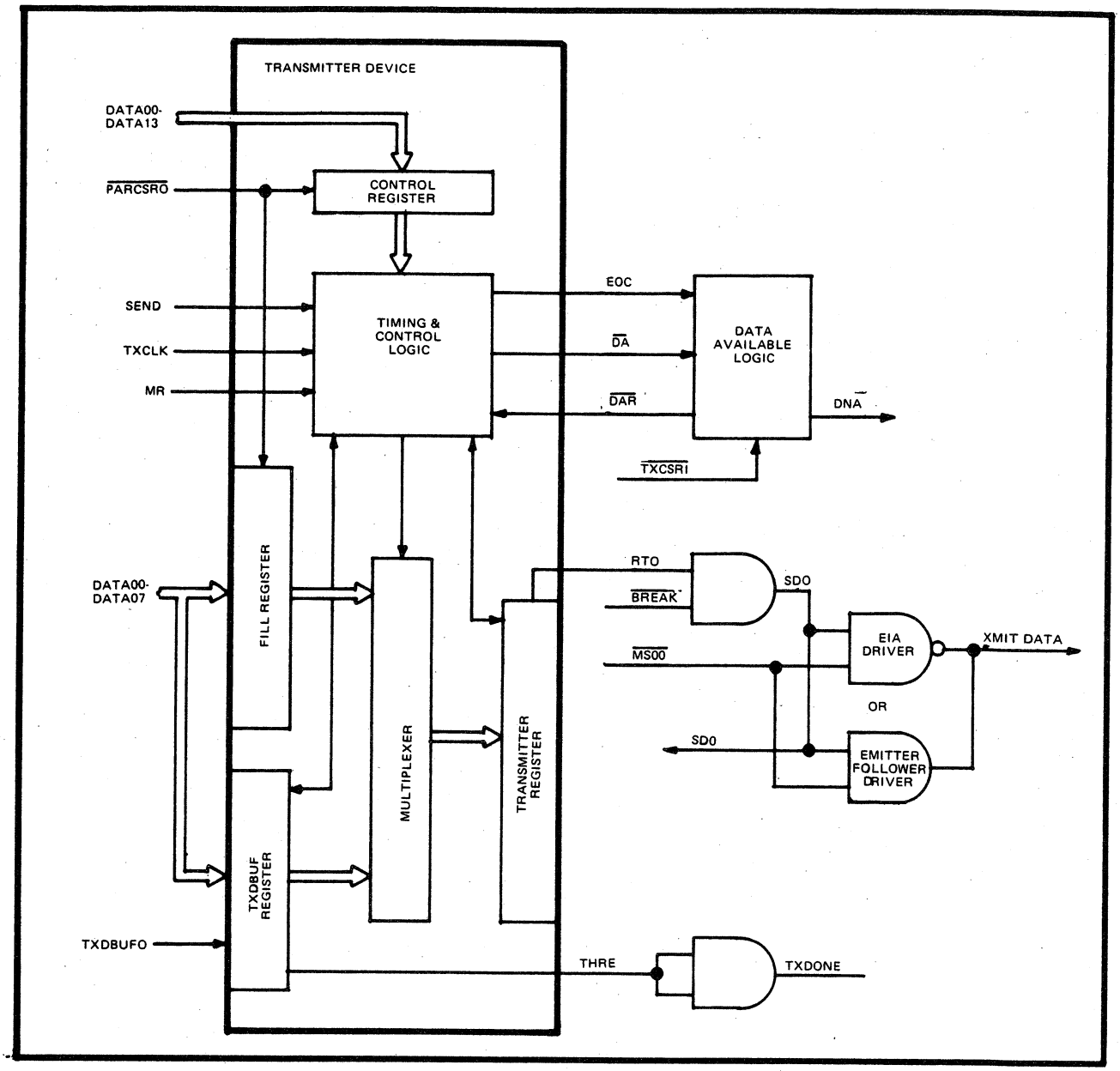


Figure 3 Transmitter Logic, Block Diagram

A Data Not Available (DNA) flag is raised and stored in the Transmitter Status Register to notify the program that the character transmitted is a fill character. The DNA signal is stored in flip-flop 7D, which is reset by $\overline{\text{TXCSRI}}$ as the program reads the contents of the Transmitter Status Register. The Data Not Available logic is reset (at $\overline{\text{DAR}}$) as the last bit of the fill character is transmitted (EOC is asserted).

The serial data SRO is gated to the XMIT DATA line (and the modem) unless either:

a. $\overline{\text{BREAK}}$ is low (at the EIA modem interface only), or

b. A maintenance mode is programmed ($\overline{\text{MS00}}$ is low at either type of modem interface).

A BREAK bit is set in the TXCSR register for Internal and External Loop Maintenance modes, to permit sending serial data directly to the DUV11 receiver without sending it to the modem.

Isochronous Mode Transmission - In Isochronous operation, a character is loaded into the TXDUB register and output serially, following a Start bit, and followed by a single Stop bit. TXDONE is set as the Stop bit is transmitted.

In this mode of operation, if the program does not load the next character before the current character is transmitted, the transmitter simply pauses and waits for the next character. The DNA flag is not used in this mode of operation.

As in either type of transmitter operation, transmission is enabled only for as long as the SEND bit is set in the TXCSR register.

Receiver Logic

Receiver logic (figure 4) is initialized by BINITL or MRST (DATA08•UTXCS0), which causes MR to clear receiver logic to the idle state, and to set the RXDBUF register to all 1's.

The receiver control register (PARCSR) is loaded by the decoded PARCSRO command received with the device address. The bits loaded from the bus control the following parameters of receiver operation.

a. The sync character code (bits 00 - 07).

b. The operating mode: That is, Internal or External Synchronous, or Isochronous, operation (bits 12 and 13).

c. Word Length: That is, 5, 6, 7, or 8 bits (bits 10 and 11).

d. Parity enabled (bit 09).

e. Parity sensing, odd or even (bit 08).

The receiver serial data input is selected at modem interface logic and may be either: (1) the modem's serial data output, in normal operation; or (2) either the programmed bit MITDAT, or the transmitter output signal SDO, in maintenance mode operation.

Note that the SCHSYNC bit must be set in the RXCSR register, and the RXINB bit must be set to enable recognition of an RXDONE flag.

If the STRPSYNC bit is set, a received sync character will *not* cause a receiver interrupt request unless an error is detected.

Operation in Internal Synchronous Mode

In this mode of operation characters must arrive in an unbroken stream in order for synchronization to be maintained. Synchronization is achieved when the receiver has recognized two consecutive sync characters in the incoming serial data stream.

When the PARCSR register is loaded by $\overline{\text{PARCSRO}}$, the SYNCINTR flip-flop is set, and the flip-flop 9F is set to hold the SS input to the receiver device at a high level.

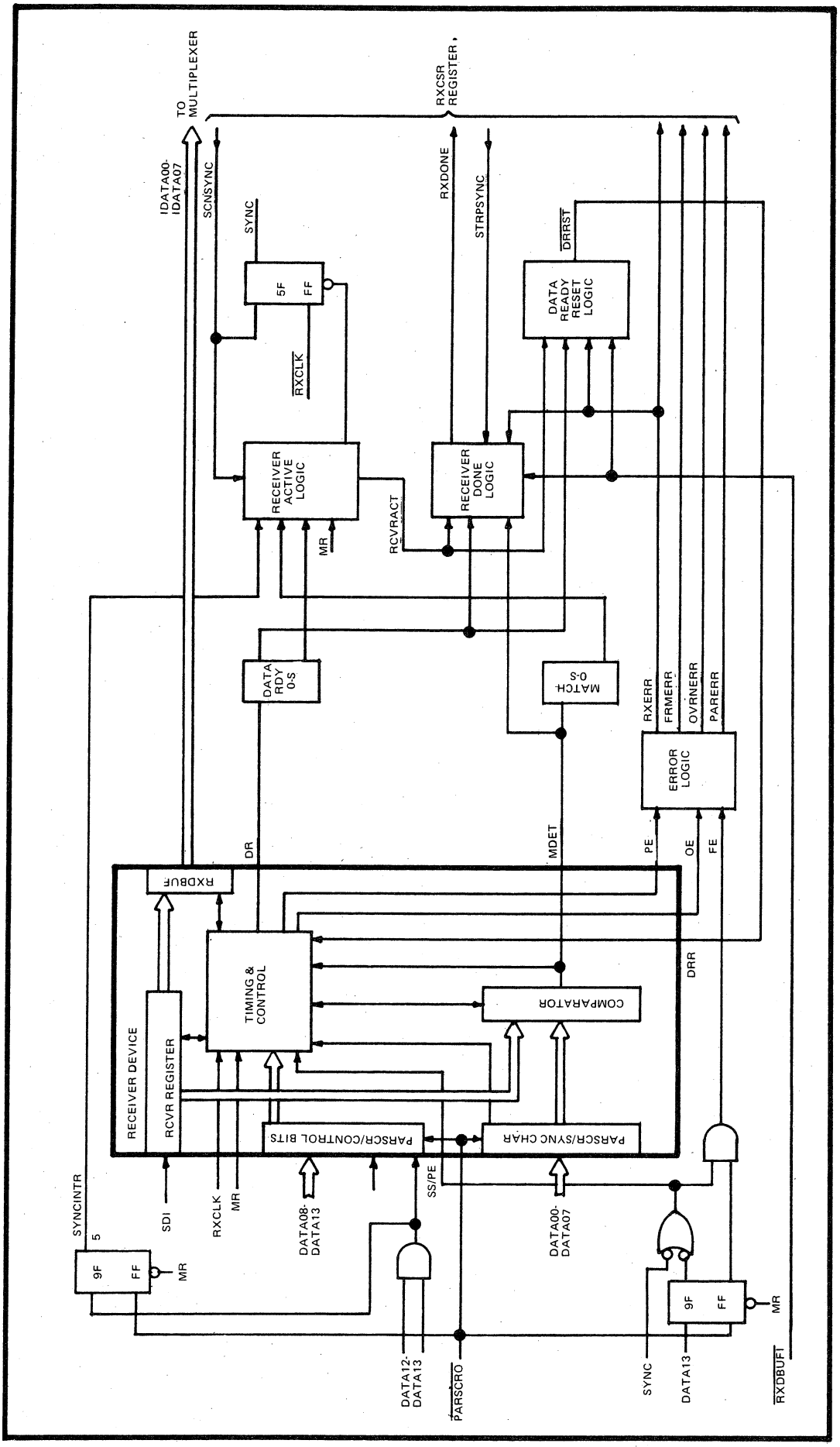


Figure 4 Receiver Logic, Block Diagram

Receiver operation begins when the program asserts SCHSYNC. SCHSYNC causes the next negative-going transition of RXCLK to set SYNC. SYNC appears at the SS input of the receiver device, enabling the receiver.

As each incoming bit is shifted into the transmitter register (and into the RXDBUF register), the contents of the receiver register are compared with the sync character stored in the PARCSR register. When the characters match, the receiver asserts $\overline{\text{MDET}}$, which triggers one-shot 6E. After approximately 400 nsec, the one-shot output sets flip-flop 6D.

The receiver then begins to frame the next character, but no longer shifts the incoming bits into the RXDBUF register. When an entire character has been received, the receiver device does the following:

a. Checks parity of the received character;

b. Compares the received character to the stored character;

c. Transfers the character in the receiver register to the RXDBUF register.

If the second character is a sync character, MDET remains high, and the Data Ready (DR) flag is asserted. RCVRACT is asserted, triggering the $\overline{\text{DRRST}}$ one-shot. After a 400 nsec delay, DRRST clears the DR flag.

If STRPSYNC is set in the RXCSR register, the term RCVRACT•DR•MDET will *not* raise the RXDONE flag and the character in the RXDBUF register will not be read by the program. If STRPSYNC is not set, the second sync character will be read by the program in response to RXDONE.

The next (data) character is framed in the same manner, but with MDET cleared. For each character RXDONE is set as the last bit is received. When the program responds to the resulting interrupt request, $\overline{\text{RXDBUFI}}$ clears RXDONE, and triggers the $\overline{\text{DRRST}}$ one-shot to clear the DR flag.

SYNC remains asserted for as long as synchronization is maintained.

A detected parity error is flagged (by both PARERR and RXERR) for transfer to the bus.

An overrun error (OVRNERR) flag is set in the RXDBUF register (and RXERR is set) if a new character is framed and transferred into the RXDBUF register with RCVRACT asserted, and before DR is asserted. OVRNERR and RXERR are applied to the multiplexer for transfer to the bus.

An error of any kind defeats STRPSYNC logic so that RXDONE is raised for a sync character received with an error.

Note that a framing error (FRMERR) is not defined in Synchronous mode operation.

Operation in External Synchronous Mode

As in Internal Synchronous operation, the receiver is initialized by loading the PARCSR register. As the register is loaded by $\overline{\text{PARCSRO}}$, the SYNCINTR flip-flop is cleared and the SYNC flip-flop is enabled.

When the program has acquired synchronization with the incoming data stream, it asserts SCHSYNC. RCVRACT is immediately asserted to enable the RXDONE flip-flop, and the next negative-going transition of RXCLK sets the SYNC flip-flop. SYNC, enabled by flip-flop 9F, appears at the SS input to the receiver device and causes the receiver to begin framing on the next bit received.

Subsequent operation is essentially the same as in the Internal Synchronous mode of operation.

Operation in Isochronous Mode

In Isochronous operation, each character received is preceded by a Start bit and followed by a Stop bit. A Start bit is a line level (usually low) lasting for one bit time. A Stop bit is a line level (usually high) lasting for one, one and a half, or two bit times. Because the receiver can synchronize on each character, individually, data need not be received in an unbroken stream.

In this mode, PARCSR bits 12 and 13 are cleared. Therefore, $\overline{\text{PARCSRO}}$ clears both the SYNCINTR flip-flop and the SYNC flip-flop. Receiver operation is enabled when the program asserts SCHSYNC. RCVRACT rises immediately and SYNC is set by the next negative-going transition of RXCLK.

SYNC has no effect on the receiver device because flip-flop 9F (storing DATA13) holds the SS terminal low. This permits the Framing Error flag (FRMERR) to control the SS terminal.

Receiver operation is begun when the receiver detects a Start bit. Any level transition from Mark to Space causes the receiver to sample the input line. A low level causes the receiver to begin framing succeeding data bits. When the programmed number of bits has been shifted into the receiver register the receiver tests the next bit for a valid Stop bit, checks parity, and transfers the contents of the receiver register into the RXDBUF register (minus Start and Stop bits).

If the character is a valid character, the receiver asserts DR, and RXDONE raises a receiver interrupt request. When the program reads the character on the bus, DRRST is triggered to reset DR in the receiver device, and RXDONE is reset.

The receiver device then idles until the next Start bit is received.

As in Synchronous mode operation, a parity error flag is raised if invalid parity is detected in any character. An Overrun Error (OVRNERR) is defined just as in Synchronous operation.

A Framing Error flag (enabled by the stored bit DATA13) is raised when the receiver does not find a Stop bit at the predicted time. Two-hundred nanoseconds later, DR is asserted and RXDONE requests a receiver interrupt. The program responds by reading the FRMERR and RXERR bits on the bus, $\overline{\text{DRRST}}$ is raised to clear DR, and RXDONE is cleared.

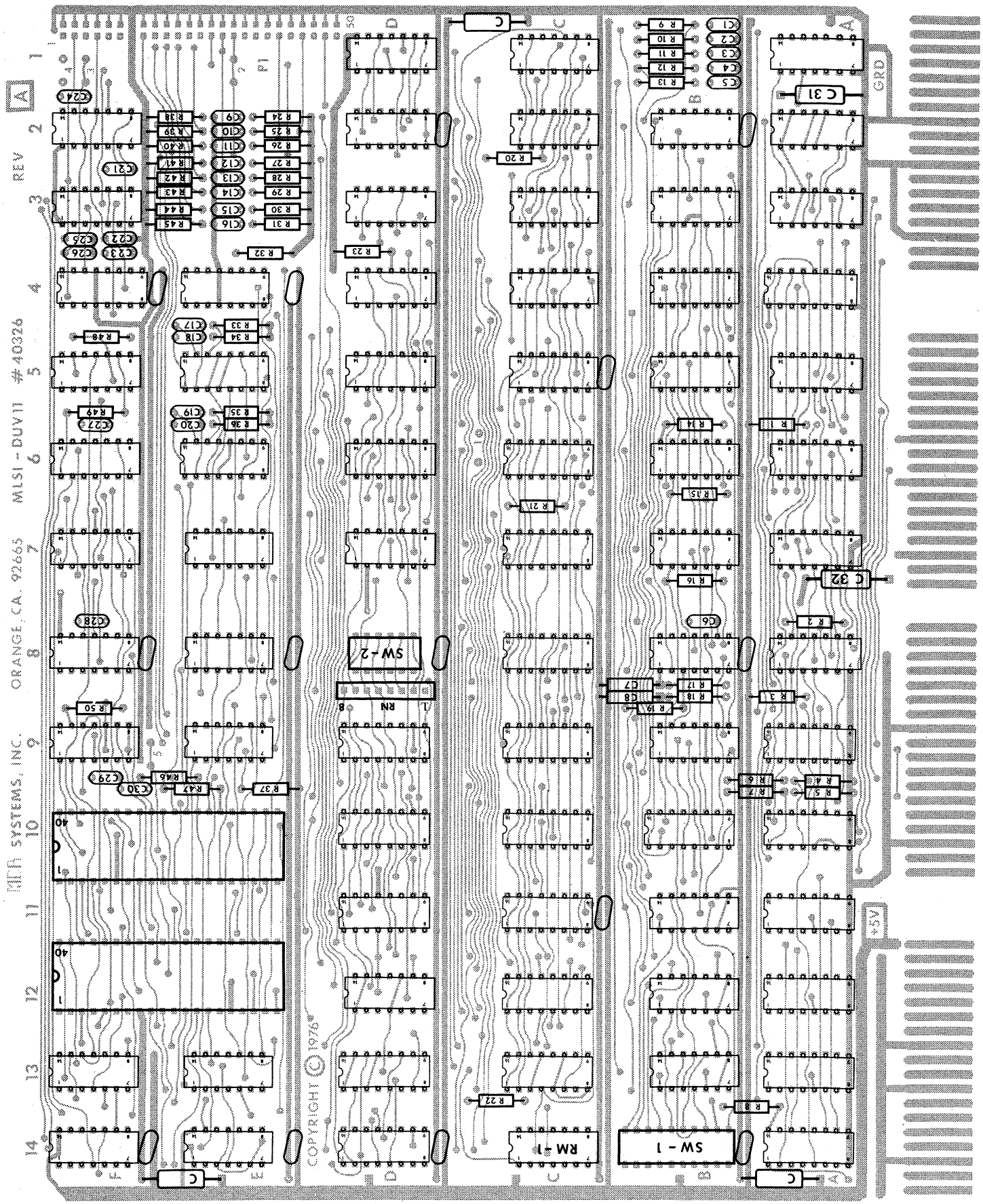
MAINTENANCE

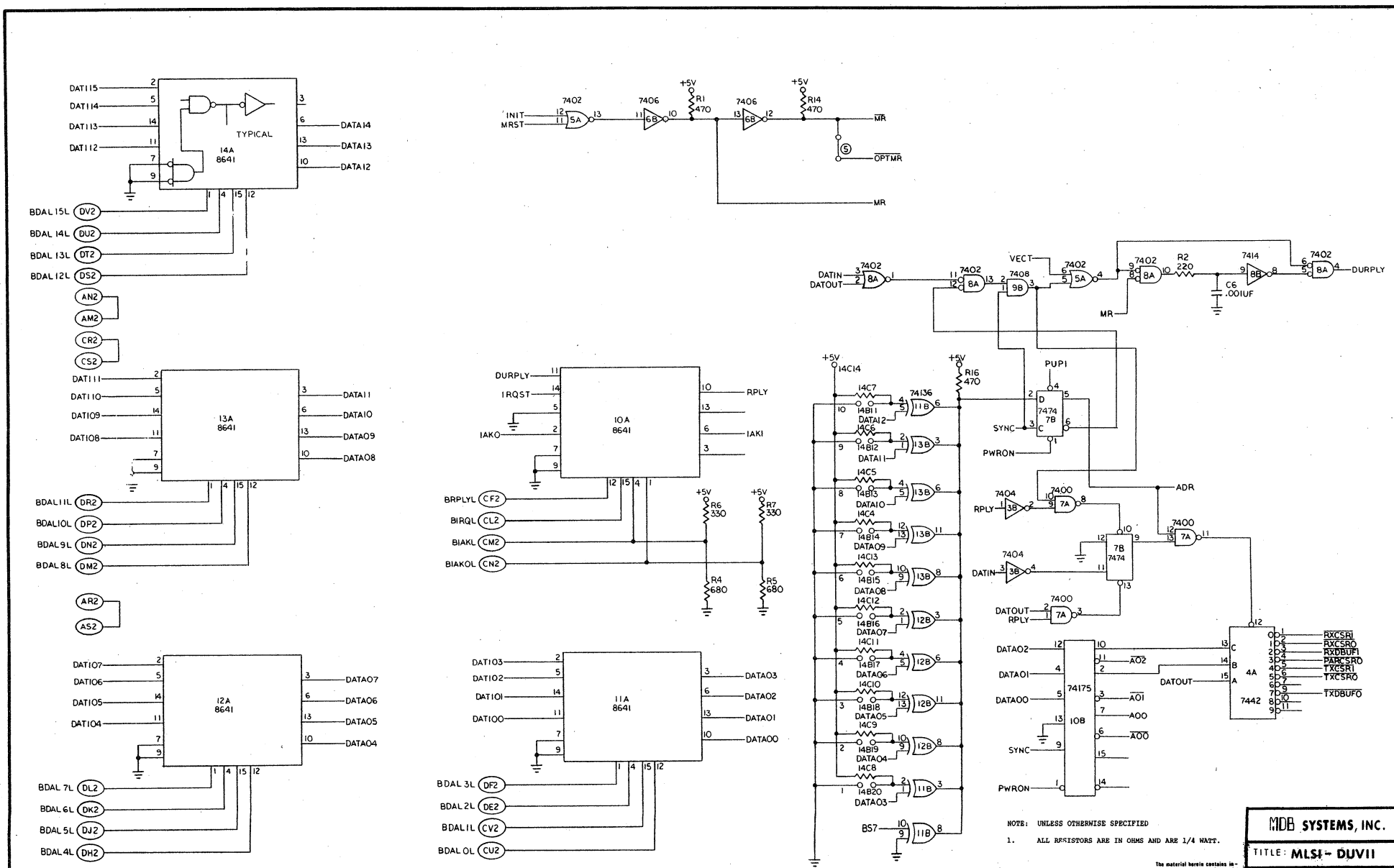
The MDB MLSI-DUV11 Synchronous Serial Line Interface is completely compatible with existing DEC LSI-11 and PDP11/03 diagnostic software. Use that software to verify correct operation and for trouble analysis.

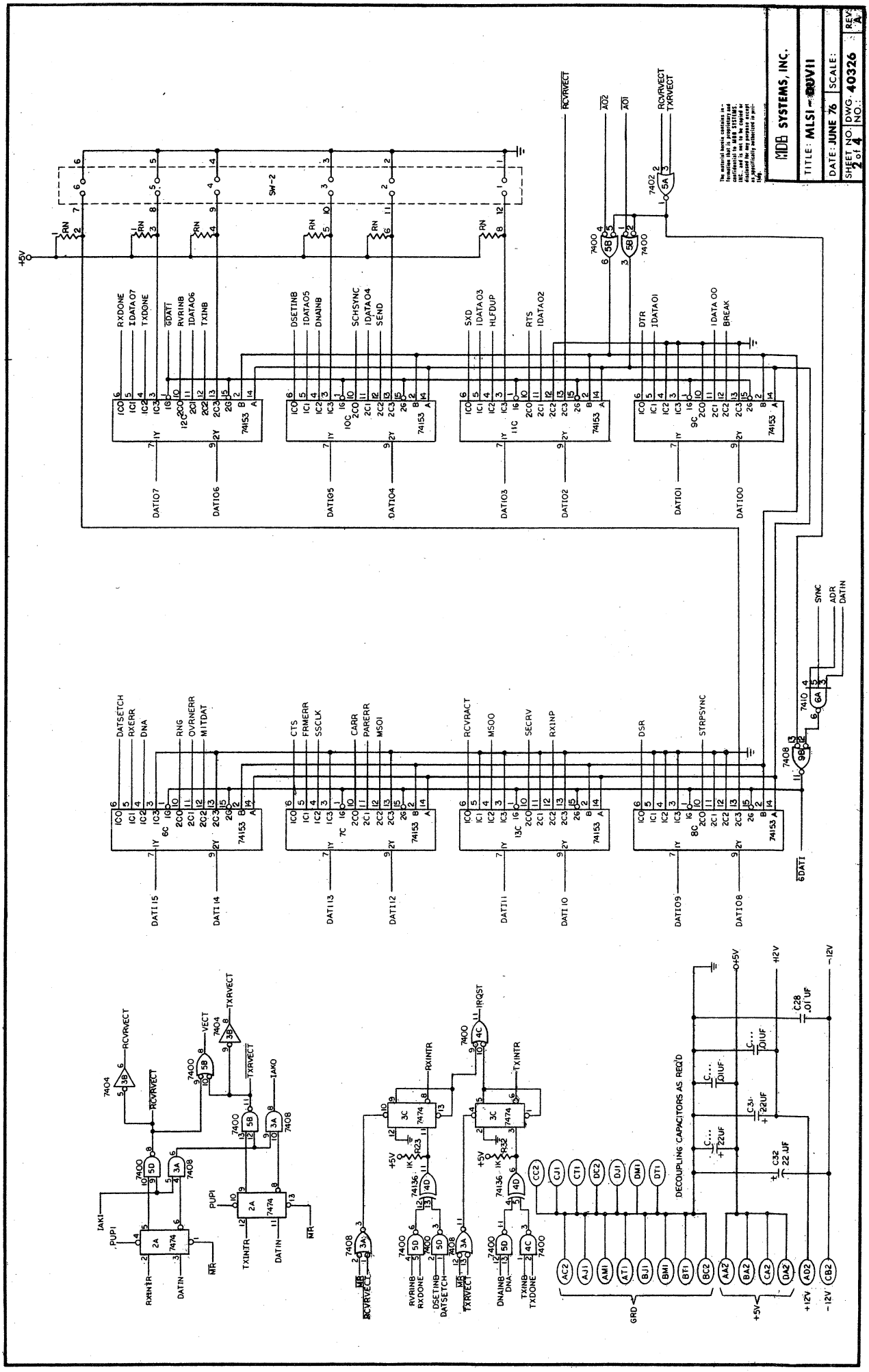
Repair the module using appropriate skills, techniques, and materials. If you wish MDB Systems to repair the module, first notify MDB System's Customer Service, then pack the module carefully, along with your best evaluation of trouble symptoms, and ship it, prepaid, to MDB Systems.

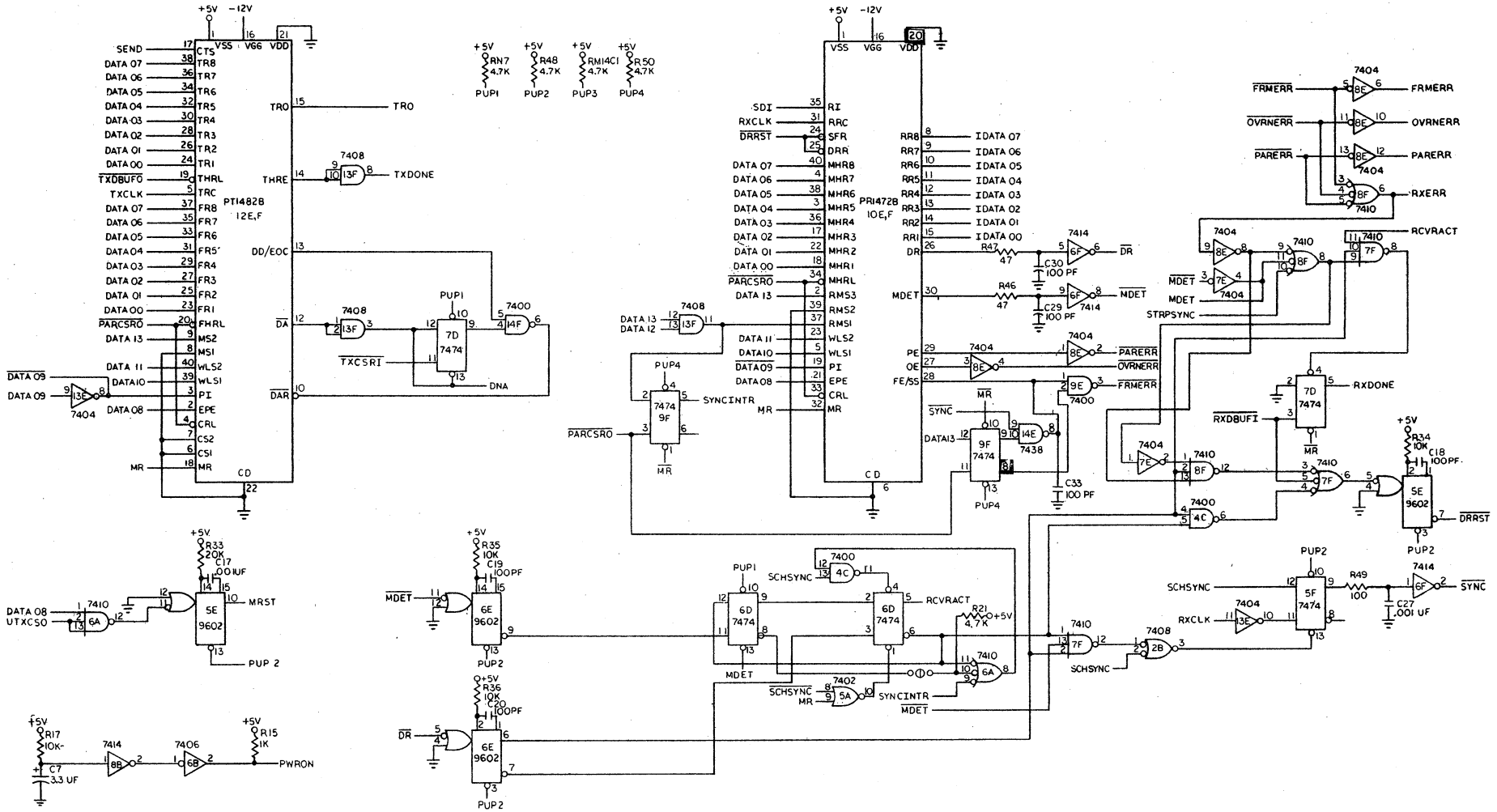
DRAWINGS

The following pages contain logic diagrams and assembly drawings useful in maintaining and repairing the module.

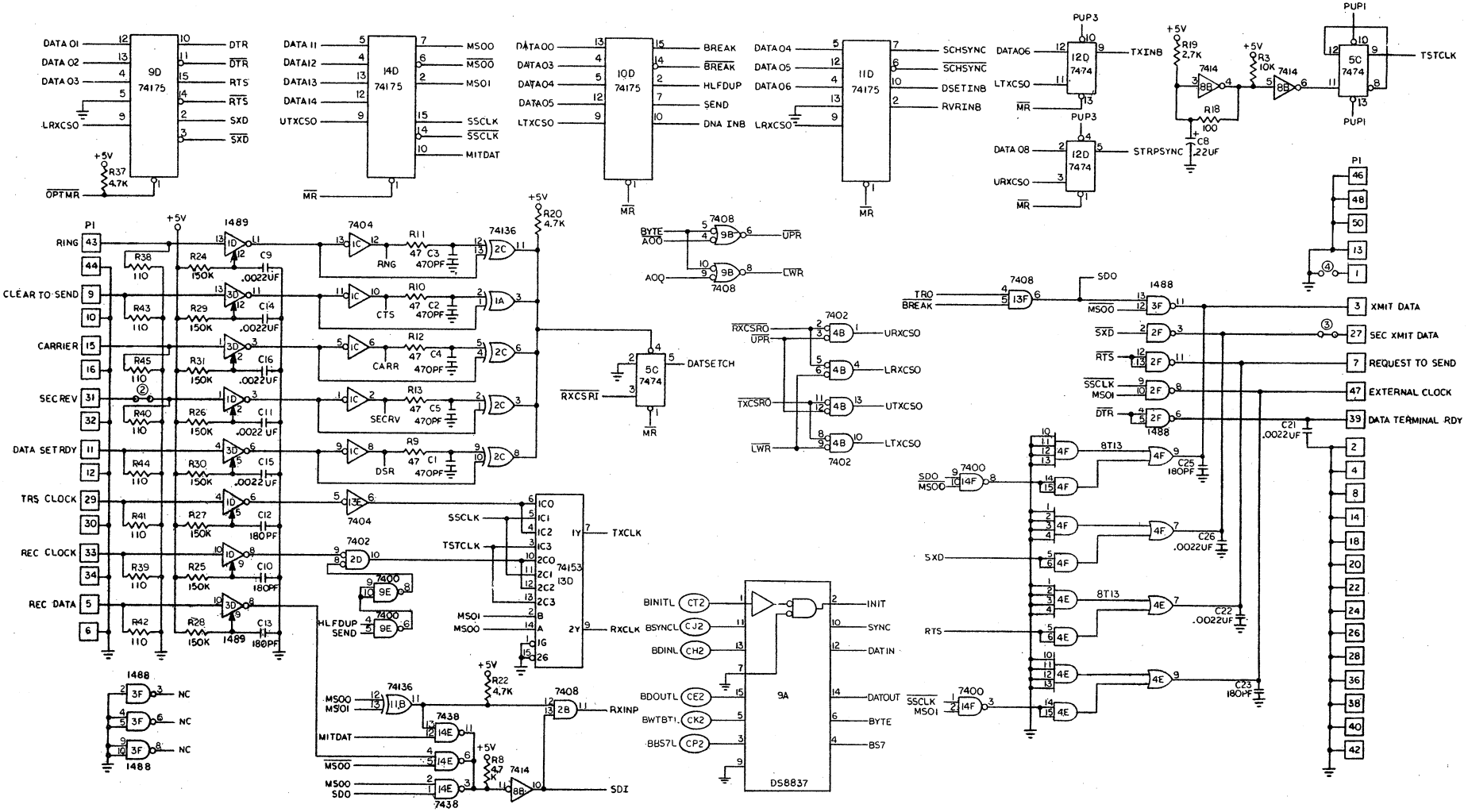

MLSI - DUV11 #40326
REV
ORANGE, CA. 92665
MTI SYSTEMS, INC.
COPYRIGHT © 1976
SW - 2
SW - 1
RM - 1
RN
GRD
+5V


DAT115
DAT114
DAT113
DAT112
TYPICAL
14A
8641
DATA14
DATA13
DATA12
BDAL15L DV2
BDAL 14L DU2
BDAL 13L DT2
BDAL 12L DS2
AN2
AM2
CR2
CS2
DAT111
DAT110
DAT109
DAT108
13A
8641
DATA11
DATA10
DATA09
DATA08
BDAL11L DR2
BDAL10L DP2
BDAL 9L DN2
BDAL 8L DM2
AR2
AS2
DAT107
DAT106
DAT105
DAT104
12A
8641
DATA07
DATA06
DATA05
DATA04
BDAL 7L DL2
BDAL 6L DK2
BDAL 5L DJ2
BDAL 4L DH2
7402
7406
+5V
R1
470
INIT
MRST
5A
6B
R14
MR
OPTMR
DURPLY
IRQST
IAKO
10A
8641
RPLY
IAKI
BRPLYL CF2
BIRQL CL2
BIAKL CM2
BIAKOL CN2
R6
330
R7
330
R4
680
R5
680
DAT103
DAT102
DAT101
DAT100
11A
8641
DATA03
DATA02
DATA01
DATA00
BDAL 3L DF2
BDAL 2L DE2
BDAL1L CV2
BDAL 0L CU2
DATIN
DATOUT
8A
7408
9B
VECT
5A
7414
8B
R2
220
C6
.001UF
14C14
R16
470
14C7
74136
14B11
DATA12
14C6
14B12
DATA11
14C5
14B13
DATA10
14C4
14B14
DATA09
14C13
14B15
DATA08
14C12
14B16
DATA07
14C11
14B17
DATA06
14C10
14B18
DATA05
14C9
14B19
DATA04
14C8
14B20
DATA03
BS7
11B
13B
12B
PUP1
D
7474
7B
SYNC
PWRON
ADR
7404
7400
7A
RPLY
DATIN
DATOUT
74175
10B
AO2
AO1
AO0
7442
4A
RXCSR1
RXCSR0
RXDBUF1
PARCSR0
TXCSR1
TXCSR0
TXDBUF0
NOTE: UNLESS OTHERWISE SPECIFIED
1. ALL RESISTORS ARE IN OHMS AND ARE 1/4 WATT.
The material herein contains information that is proprietary and confidential to MDB SYSTEMS, INC., and is not to be copied or disclosed for any purpose except as specifically authorized in writing.
MDB SYSTEMS, INC.
TITLE: MLSI - DUVII
DATE: JUNE 76
SCALE:
SHEET NO. 1 of 4
DWG. NO.: 40326
REV. A


MDB SYSTEMS, INC.
TITLE: MLSI-DUVII
DATE: JUNE 76
SCALE:
SHEET NO. 2 of 4
DWG. NO. 40326
REV A
SW-2
+5V
RXDONE
IDATA07
TXDONE
6DATI
RVRINB
IDATA06
TXINB
DSETINB
IDATA05
DNAINB
SCHSYNC
IDATA04
SEND
SXD
IDATA03
HLFDUP
RTS
IDATA02
DTR
IDATA01
IDATA00
BREAK
RCVRVECT
TXRVECT
A02
A01
DATI07
DATI06
DATI05
DATI04
DATI03
DATI02
DATI01
DATI00
DATSETCH
RXERR
DNA
RNG
OVRNERR
MITDAT
CTS
FRMERR
SSCLK
CARR
PARERR
MSOI
RCVRACT
MSOO
SECRV
RXINP
DSR
STRPSYNC
DATI15
DATI14
DATI13
DATI12
DATI11
DATI10
DATI09
DATI08
SYNC
ADR
DATIN
RCVRVECT
VECT
TXRVECT
IAKO
IAKI
PUPI
RXINTR
TXINTR
DATIN
NPR
IRQST
RVRINB
RXDONE
DSETINB
DATSETCH
DNAINB
DNA
TXINB
TXDONE
DECOUPLING CAPACITORS AS REQ'D
GRD
+5V
+12V
-12V
AC2
AJ1
AM1
AT1
BJ1
BM1
BT1
BC2
AA2
BA2
CA2
DA2
A02
CB2



MDB SYSTEMS, INC.	
TITLE: MLSI - DUVII	
DATE: JUNE 76	SCALE:
SHEET NO. 3 of 4	DWG. NO.: 40326 REV. A



MDB SYSTEMS, INC.

TITLE: MLSI – DUV11

DATE: JUNE 76 | SCALE:

SHEET NO. 4 of 4 | DWG. NO.: 40326 | REV. A

MDB SYSTEMS INC.
1995 N. Batavia Street
Orange, California 92665
714-998-6900
TWX:910-593-1339