As you are now the owner of this document which should have come to you for free, please consider making a donation of £1 or more for the upkeep of the (Radar) website which holds this document. I give my time for free, but it costs me money to bring this document to you. You can donate here https://blunham.com/Misc/Texas

Many thanks.

Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website https://blunham.com/, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (https://blunham.com/Misc/Texas). **Please do not point them at the file itself as it may move or the site may be updated.**

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you.

Colin Hinson
In the village of Blunham, Bedfordshire.

TEXAS INSTRUMENTS

TM 990

TM 990/308
Industrial Communication Module

MICROPROCESSOR SERIES™

June 1980

TABLE OF CONTENTS

SECTION TITLE PAGE

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

SECTION 1

INTRODUCTION

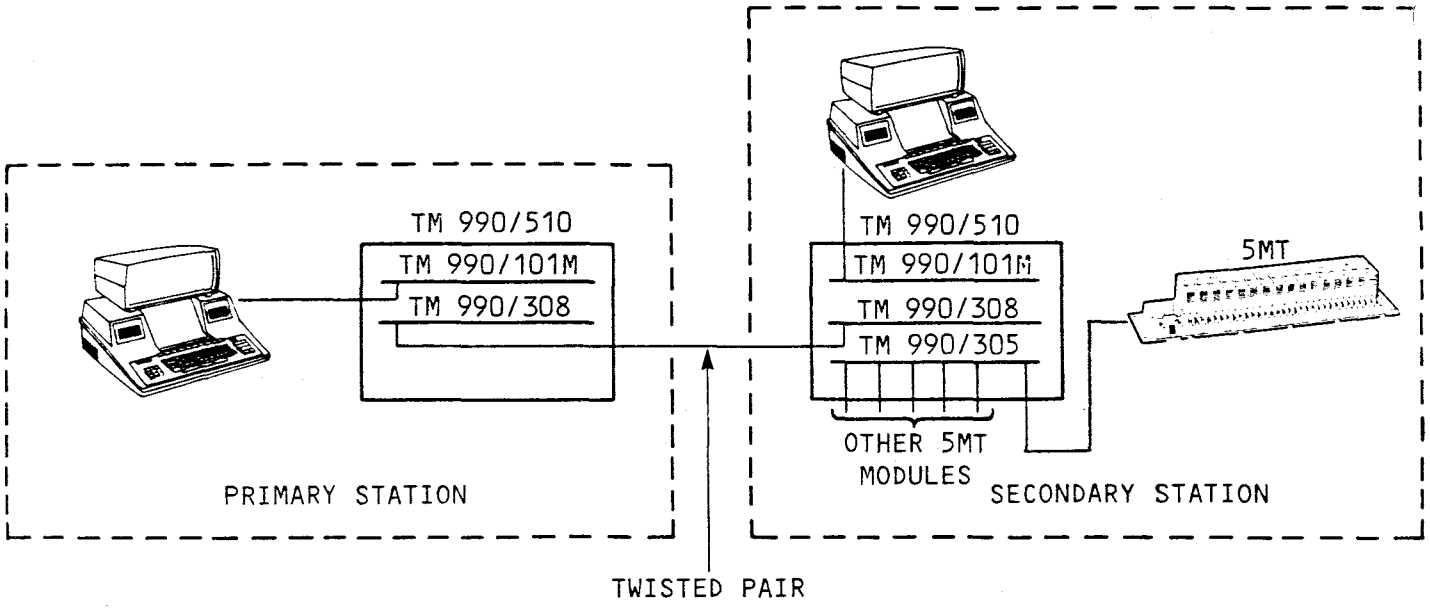
1.1 GENERAL

The TM 990/308 Industrial Communications Module (ICM) provides the necessary interface logic for implementing a serial, synchronous communications link between two or more TM 990 systems. The ICM extends the communication range of the CPU module by providing multidrop and RS-232-C EIA interfaces that can be used with twisted pair line and telephone lines respectively to provide a long-range communication link. Figure 1-1 shows a system that uses the multidrop interface. This interface allows several ICMs (up to 32) to communicate using a twisted pair line. The line may extend to 10,000 feet while using as small as 22 gauge wire. In this system, the secondary station is being used to affect numerous industrial control processes where push buttons, limit switches, sensors, and TTL logic are being used to control contactors, solenoids, pilot lights, and motor starters. The primary station can supervise up to 31 secondary stations.

Figure 1-2 illustrates another use of the ICM. In this system which uses the RS-232-C EIA interface, the ICM provides the necessary interface so that two CPU modules can communicate with each other using telephone lines. Figure 1-2a shows the familiar telephone system where voice is converted into an analog voice signal by a transducer in one phone , then sent over telephone lines to another phone whose transducer re-converts the voice signal to sound. A modem essentially performs the same type of service for this computer system as the transducers do for the telephone system. Figure 1-2b shows a digital message from a CPU module being converted to an analog signal by a modem (acronym for MOdulator, DEModulator) prior to being sent over telephone lines to the remote location. At the remote location, the modem recovers the original digital message from the analog signal prior to relaying it to the CPU module. Additional systems that use the ICM are presented in Section 2 of this manual.

Some of the additional features of the ICM are as follows:

- Half-duplex serial synchronous operation
- Point-to-point or multi-point operation
- Switch selectable baud rates: 1200, 2400, 4800, or 9600 bits/second
- Character length of 8 bits
- Self-test capability
- Two-wire interface utilizing isolated differential line driver/line receiver stage - 1500 volts of isolation provided
- Immune to a 100 V common mode noise impulse with a 100 ns rise time
- Separate EIA RS-232-C interface for modem operation - Bell 208 Modem compatible using TM 990/502 cable



TM 990/101M - CPU Module using TMS 9900 microprocessor with 4 K bytes of RAM

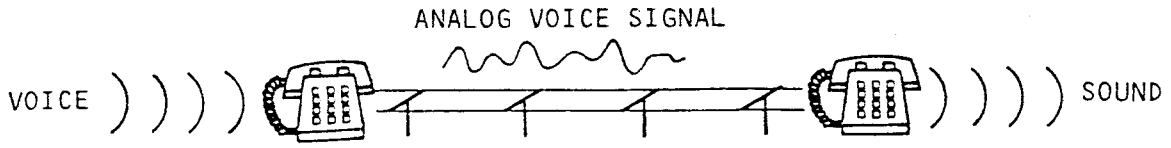
TM 990/308 - Industrial Communication Module (ICM) featuring TMS 9980 Microprocessor and TMS 9903 synchronous serial I/O controller

TM 990/305 - Optically isolated I/O expansion module with expansion memory

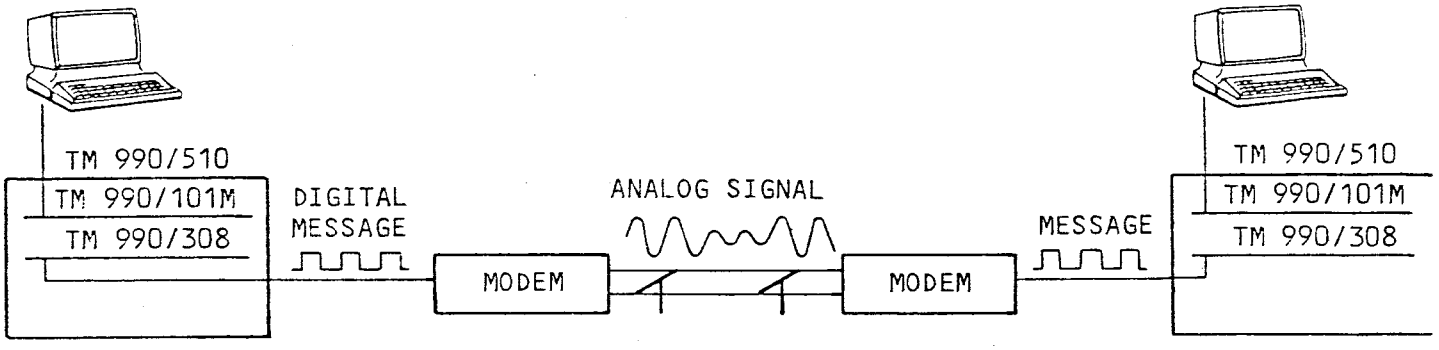
TM 990/510 - 4-slot card cage that provides interface between CPU and other boards in system via card cage backplane

TM 990/5MT - Industrial I/O that uses a variety of input sensors to control contactors, motor starters, and the like. Several of these could be connected via the TM 990/305

FIGURE 1-1. SAMPLE SYSTEM USING ICM'S MULTIDROP INTERFACE



a) SIMPLE TELEPHONE SYSTEM



b) MODEM APPLICATION

FIGURE 1-2. SAMPLE SYSTEM USING ICM'S RS-232-C EIA INTERFACE

- Hardware cyclic redundancy checking (CRC) provided by the TMS 9903 (CRC-CCITT)

- Address decoding provided by TMS 9980A software - 64 switch selectable addresses (00-3F) plus broadcast address (FF) - up to 32 ICMs per multidrop line.

- Download command decoding provided by TMS 9980A software

- 928 bytes of RAM used for data buffering, 1952 bytes with expansion. Board software determines size by a RAM search.

- 2K bytes of EPROM - expandable to 4K bytes

- Jumper selectable line terminators

- Communication up to 10,000 feet at 9600 bits/second.

1.2 MANUAL ORGANIZATION

This manual is organized as follows:

- Section 1 presents a description of the module along with its specifications and characteristics.

- Section 2 presents several systems that use the ICM, it also explains how to install, power-up, self-test, and operate the TM 990/308.

- Section 3 explains programming techniques for systems that use both the multidrop (local line) or an EIA interfaces.

- Section 4 covers the theory of operation for the TM 990/308. An overview of the internal circuitry is followed by detailed explanations covering the various circuits which comprise the ICM.

1.3 MODULE CHARACTERISTICS

Figure 1-3 is a photograph of the TM 990/308 module while Figure 1-4 shows the principal components and interfaces of the module. The system bus connector is P1, a 100-pin (50 each side) PC board edge connector spaced on 0.125 inch centers. Connector P2 provides an interface for systems that use either the EIA or multidrop interface.

Figure 1-5 shows the module's dimensions.

1.4 GENERAL SPECIFICATIONS

- Power Requirements

	V_{CC} +5 V ±3%		V_{DD} +12 V ±3%		V_{BB} -12 V ±3%	
	MAX	TYP	MAX	TYP	MAX	TYP
	2.0 A	1.1 A	0.5 A	0.2 A	0.1 A	0.05 A

- Operating Temperature: 0°C to 70°C.

- Module Dimensions: See Figure 1-5.

- Memory*: RAM - 1K bytes expandable to 2K bytes using TMS 4045s or TMS 4014s.

  EPROM - 2K bytes expandable to 4K bytes using TMS 2716s.

  * The RAM and EPROM on board the TM 990/308 is not connected to the system bus. The EPROM is used exclusively for implementing ICM control routines. It is not available for user applications involving other modules in the system. The RAM can be expanded by the user to increase the length of message buffered on the TM 990/308 from 928 bytes to 1952 bytes. Upon exection of an IORST at powerup or at reset, the TM 990/308 software does a RAM search to determine the bounds.

- External Interfaces: Multidrop (local line) interface - capable of driving an additional 31 ICMs and communicating up to 10,000 feet at 9600 bits/second.

  RS-232-C EIA interface - Used for Bell 208 modem operation.

1.5 APPLICABLE DOCUMENTS

- TM 990 System Specification

- Model 990 Computer/TMS 9900 Microprocessor Assembly Language Programmer's Guide, 943441-9701

- TMS 9980A/TMS 9981 Microprocessor Data Manual

- TMS 9903 Synchronous Communication Controller (Appendix D of this manual)

- EIA Standard RS-232-C

- Bell 208 Modem Handbook.

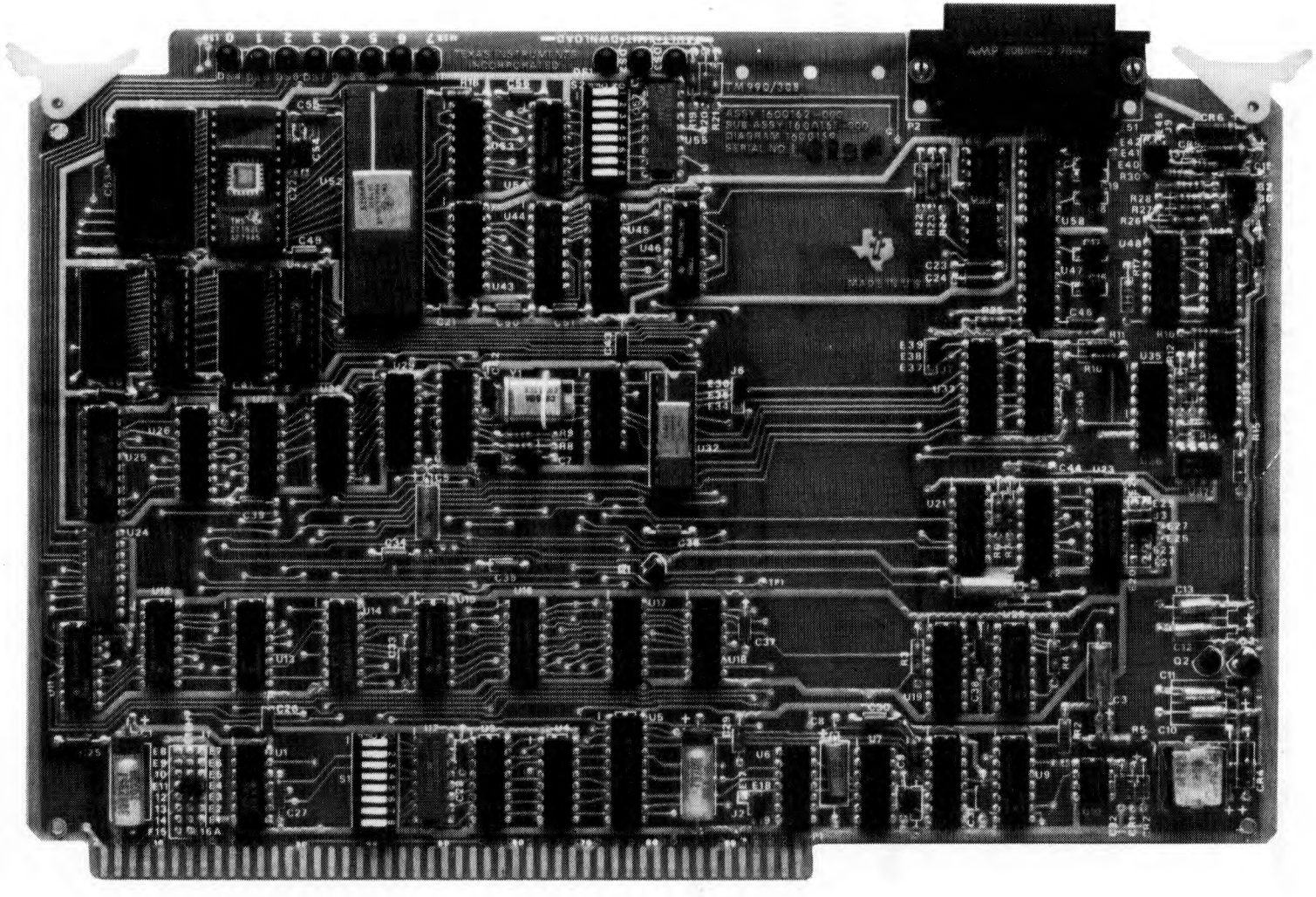


FIGURE 1-3. TM 990/308 MODULE

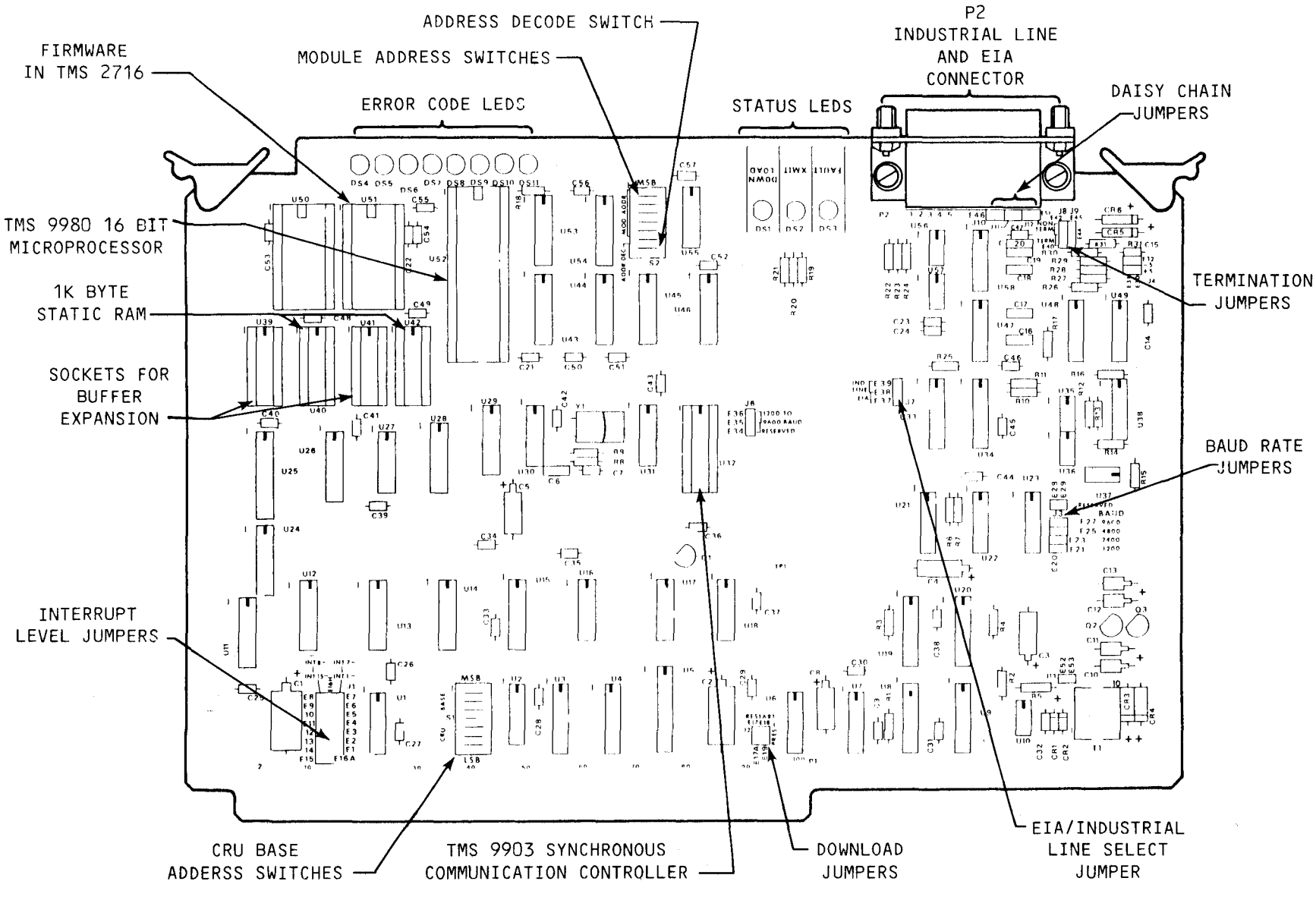


FIGURE 1-4. TM 990/308 PRINCIPAL COMPONENTS

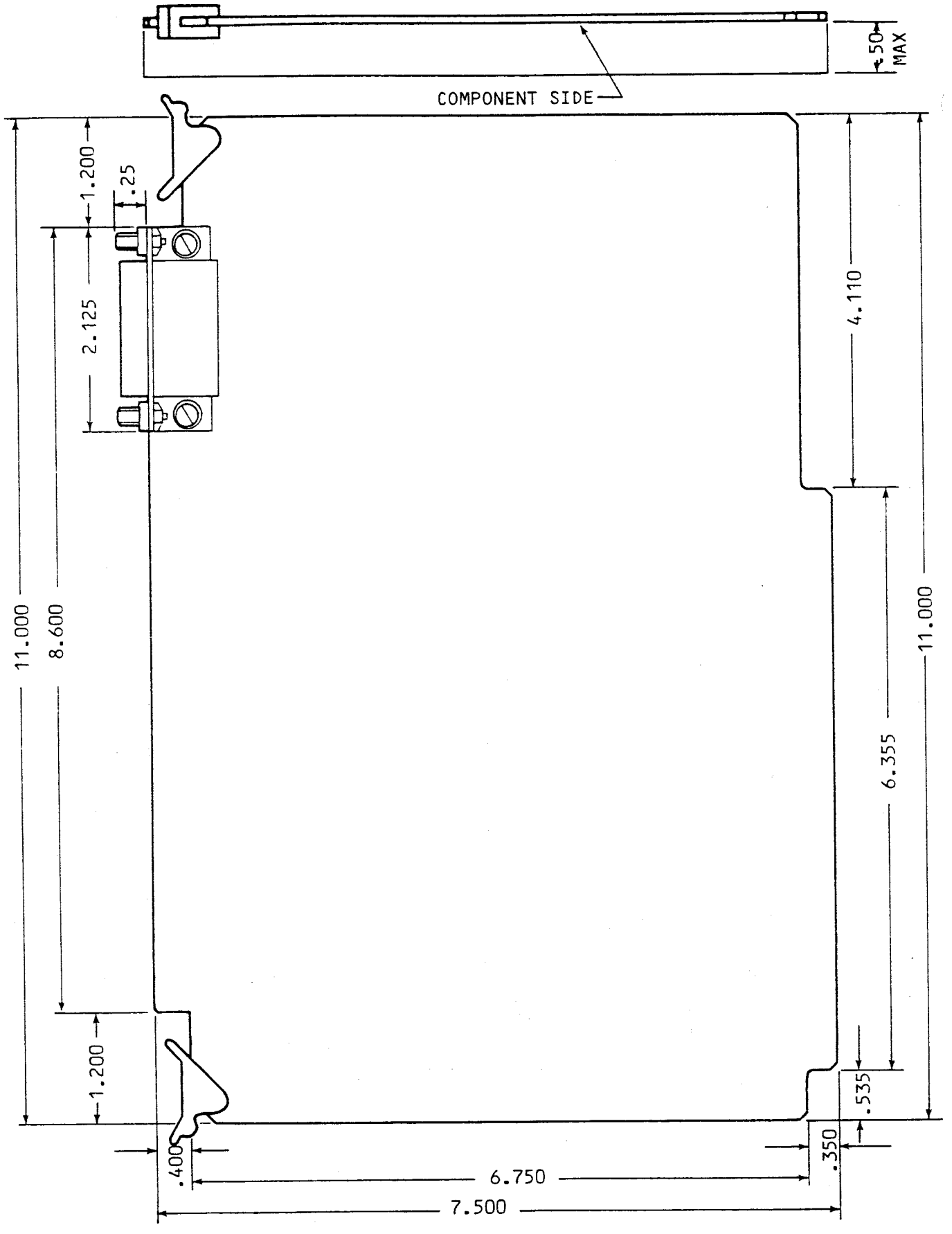


FIGURE 1-5. TM 990/308 MODULE DIMENSIONS (IN INCHES)

SECTION 2

INSTALLATION AND CHECKOUT OF TM 990/308 MODULE

2.1 GENERAL

The following topics are covered in this section:

- Unpacking and inspection
- Module options
- System for testing the ICM
- Typical systems using the ICM
- Required equipment
- Additional systems using the ICM.

It is assumed that the user is familiar with the hardware and programmming aspects of the host computer (TM 990/101M or TM 990/100M), the I/O module (TM 990/305), and the Bell 208 Modem. If not, consult their respective user's guides.

2.2 UNPACKING AND INSPECTION

Remove the TM 990/308 from its carton and remove the protective wrapping. Check the module for shipping damage. If any damage is found, notify your TI distributor.

2.3 SWITCH/JUMPER DESCRIPTIONS

The switches and jumpers on the ICM provide the following functions:

1) Data rate selection
2) Module address decode selection
3) Module address selection
4) CRU software base address selection
5) Interrupt level selection
6) Line terminations
7) EIA/multidrop interface selection
8) Multidrop daisy chain jumpers
9) Clock jumper
10) Download jumper
11) Isolated power jumpers.

Each of these switch/jumper functions are described in the paragraphs that follow. A switch/jumper position summary is given in Table 2-1: however, the user should refer to the descriptions for specific details, limitations, and precautions prior to using this table.

TABLE 2-1. JUMPER/SWITCH POSITION SUMMARY

SELECT	GENERAL INFORMATION		SECTION
DATA RATE (J3)	DATA RATE (BPS)	JUMPER POSITIONS	2.3.1
	1200	E20 to E21	
	2400	E22 to E23	
	4800	E24 to E25	
	9600	E26 to E27	
	19200*	E28 to E29	
ADDR DECODE (S2)	FUNCTION	S2-7 POSITION	2.3.2
	ON	No address decoding	
	OFF	Address deconding	
MODULE ADDR (S2)	MODULE ADDR (HEX) (DEC)	SWITCHES S2-1 through S2-6 BINARY PROGRESSION	2.3.3
	00 0	All positions ON	
	. .	. .	
	. .	. .	
	. .	. .	
	3F 63	All positions OFF	
CRU BASE ADDR (S1)	CRU SOFTWARE BASE ADDRESS (HEX)	SWITCHES S1-1 through S1-8 BINARY PROGRESSION	2.3.4
	0000	All positions ON	
	.	.	
	.	.	
	.	.	
	1FE0	All positions OFF	
INTERRUPT LEVEL (J1)	INT LEVEL	JUMPER POSITIONS	2.3.5
	INT1-	E1 to E16B	
	.	.	
	.	.	
	.	.	
	INT7-	E7 to E16H	
	INT8-	E8 to E16H	
	.	.	
	.	.	
	.	.	
	INT15-	E15 to E16A	

* See note, Section 2.3.1 Data Rate Selection, page 2-4.

TABLE 2-1. JUMPER/SWITCH POSITION SUMMARY (CONCLUDED)

SELECT	GENERAL INFORMATION		SECTION
LINE TERMINATIONS (J8,J9)			
	LINE	JUMPER POSITIONS	2.3.6
	Terminated	E41 to E40, E44 to E43	
	Non-terminated	E41 to E42, E44 to E45	
	Default to non-terminated	No jumpers E40 to E45	
EIA/MULTIDROP (J7)			
	INTERFACE SELECTED	JUMPER POSITION	2.3.7
	Industrial Line	E38 to E39	
	EIA	E38 to E37	
	Default to Industrial Line	No Jumpers E37-E39	
MULTIDROP DAISY CHAIN (J10,J11,J12)			
	FUNCTION	JUMPER POSITIONS	2.3.8
	Daisy chain tie points	E47 to E46, E48 to E49, E51 to E50	
	No daisy chain tie points	No jumpers E46-E51	
CLOCK (J6)			
	FUNCTION	JUMPER POSITION	2.3.9
	Selects on-board clock	E35 to E36	
	Reserved, DO NOT USE!	E35 to E34	
DOWNLOAD (J2)			
	FUNCTION	JUMPER POSITIONS	2.3.10
	RESET	E18 to E19	
	LOAD	E18 to E17	
	No download	No jumpers E17-E17A	
ISOLATED POWER (J4,J5,J13)			
	FUNCTION	JUMPER POSITIONS	2.3.11
	Connects isolated power supply to circuits that require this supply	E30 to E31, E32 to E33	
	Applies power to isolated power transformer. This jumper must be connected from E52 to E53 if isolated power is to work.	E52 to E53	

2.3.1 Data Rate Selection (J3)

The data rate is selected by setting jumper J3 as called for in Table 2-2.

TABLE 2-2. DATA RATE SELECTION JUMPER POSITIONS AT J3

DATA RATE (BITS PER SECOND)	JUMPER POSITIONS
1200	E20 to E21
2400	E22 to E23
4800	E24 to E25
9600	E26 to E27
19200*	E28 to E29

* This rate is produced by the baud rate generator but the TMS 9903 is not characterized to operate in the NRZ mode at this data rate.

2.3.2 Address Decode Selection (S2-7)

A special address decode selection feature allows each ICM to have a unique address assigned to it. This feature is required when more than two ICMs are used on the same line. If only two ICMs are used, address decoding is not needed.

Switch S2-7 controls the selection of the address decode feature. If the switch is set to the OFF position, the module will perform address decoding. In this mode, only data blocks with the correct unique address (the module address or the broadcast address, FF_{16}) will be buffered and presented to the CPU module. If the switch is set to the ON position, the ICM will not perform address decoding and all incoming data blocks will be buffered and presented to the CPU module.

2.3.3 Module Address Selection (S2-1 to S2-6)

The module address is selected by setting switches S2-1 through S2-6 as required by Table 2-3. It should be apparent that Table 2-2 follows a binary progression where ON is defined to be 0 and OFF is defined to be 1. Hence the settings of S2-1 through S2-6 for a required address of 15_{10} would be - 001111 which corresponds to a switch setting of ON ON OFF OFF OFF OFF.

Even though up to 64 different addresses are available, only 32 ICMs may be used on the multidrop line.

TABLE 2-3. MODULE ADDRESS SELECTION SWITCH POSITIONS AT S2

S2-1	S2-2	S2-3	S2-4	S2-5	S2-6	HEX ADDR	DEC ADDR
0	0	0	0	0	0	00	0
0	0	0	0	0	1	01	1
0	0	0	0	1	0	02	2
0	0	0	0	1	1	03	3
0	0	0	1	0	0	04	4
0	0	0	1	0	1	05	5
		•					
		•					
		•					
1	0	0	0	0	0	20	32
		•					
		•					
		•					
1	1	1	1	1	1	3F	63

ON = 0 OFF = 1

2.3.4 CRU Software Base Address Selection (S1)

The TM 990 CRU software base address may be selected by the setting of switches S1-1 through S1-8 as given in Table 2-4. The first address that is available is 0000_{16}. Addresses that are available follow in steps of 20_{16} up to the highest address value of $1FE0_{16}$. Careful study of the partial table shows that the switch positions follow a binary progression.

TABLE 2-4. CRU SOFTWARE BASE ADDRESS SELECTION AT S1

S1-1	S1-2	S1-3	S1-4	S1-5	S1-6	S1-7	S1-8	HEX ADDR
0	0	0	0	0	0	0	0	0000
0	0	0	0	0	0	0	1	0020
0	0	0	0	0	0	1	0	0040
0	0	0	0	0	0	1	1	0060
0	0	0	0	0	1	0	0	0080
0	0	0	0	0	1	0	1	00A0
0	0	0	0	0	1	1	0	00C0
0	0	0	0	0	1	1	1	00E0
				•				
				•				
				•				
0	1	1	1	1	1	1	1	0FE0
1	0	0	0	0	0	0	0	1000
				•				
				•				
				•				
1	1	1	1	1	1	1	1	1FE0

The following procedure can be used to determine the switch positions for selecting a CRU software base address that is not given in Table 2-4.

1) Convert the three most significant digits of the desired CRU software base address into their binary equivalents (CRU base address nomenclature is shown in Figure 3-4).

2) Shift the binary equivalents one bit to the right and truncate.

3) Convert the eight least significant bits of the shifted binary equivalent (obtained in Step 2) into ON/OFF equivalents to determine the switch settings for S1-1 through S1-8. The following switch definitions apply: 0 = ON, 1 = OFF.

EXAMPLE: Determine the switch (S1) settings to select CRU software base address $0FE0_{16}$ (numbers correspond to paragraphs above):

1)		0				F				E			0
	0	0	0	0	1	1	1	1	1	1	1	0	
2)	0	0	0	0	0	1	1	1	1	1	1	1	
3)	0	0	0	0	0	1	1	1	1	1	1	1	
					↓	↓	↓	↓	↓	↓	↓	↓	
					ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	

S1-1 ← through → S1-8

2.3.5 Interrupt Level Selection (J1)

The interrupt level (1 through 15) is selected by placing a jumper at J1 between the pin corresponding to the desired interrupt level and its mating pin of the interrupt common pins. Figure 2-1 shows the interrupt level selection pins with a jumper selecting interrupt 4.

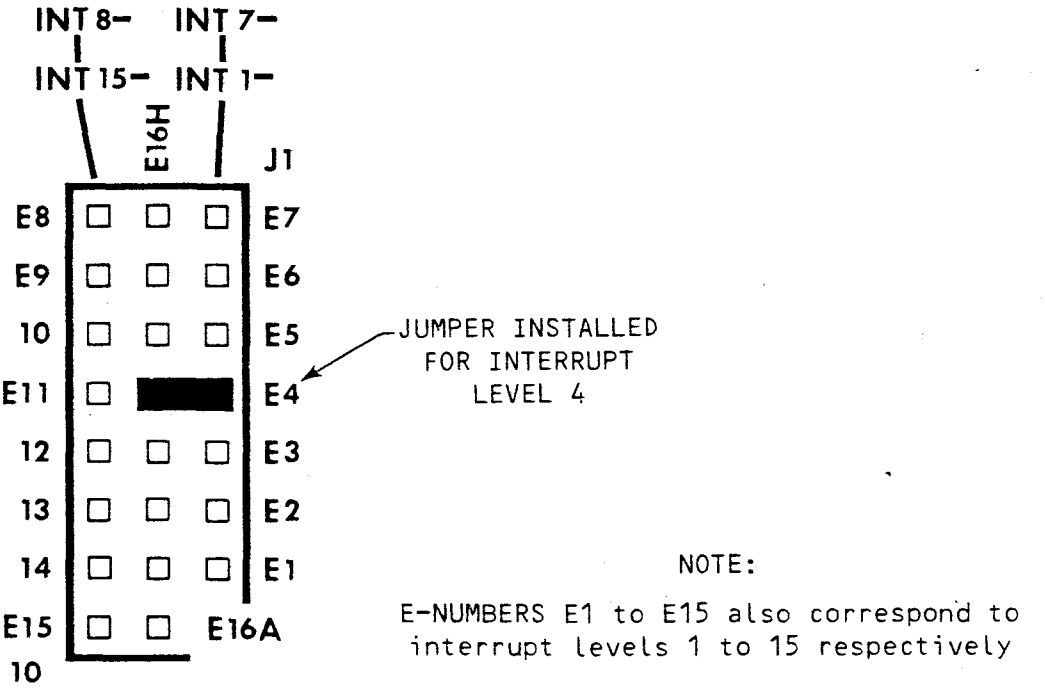


FIGURE 2-1. INTERRUPT SELECTION PINS

2.3.6 Line Terminations (J8 & J9)

The multidrop line should be properly terminated so that maximum transfer of signal occurs with mininal interference. Line termination should be included at both ends of the transmission line (see Figure 2-2). The two boards at the ends of the lines are considered terminating boards; whereas, the boards in the middle are non-terminating. Terminating a board that should be non-terminated will result in needless signal absorption while failing to terminate a board that should be terminated leaves the system susceptable to data errors due to transmission line reflections and loss of bias for the line receivers. Table 2-5 shows the proper jumper positions at J8 and J9 for either terminating or non-terminating boards.

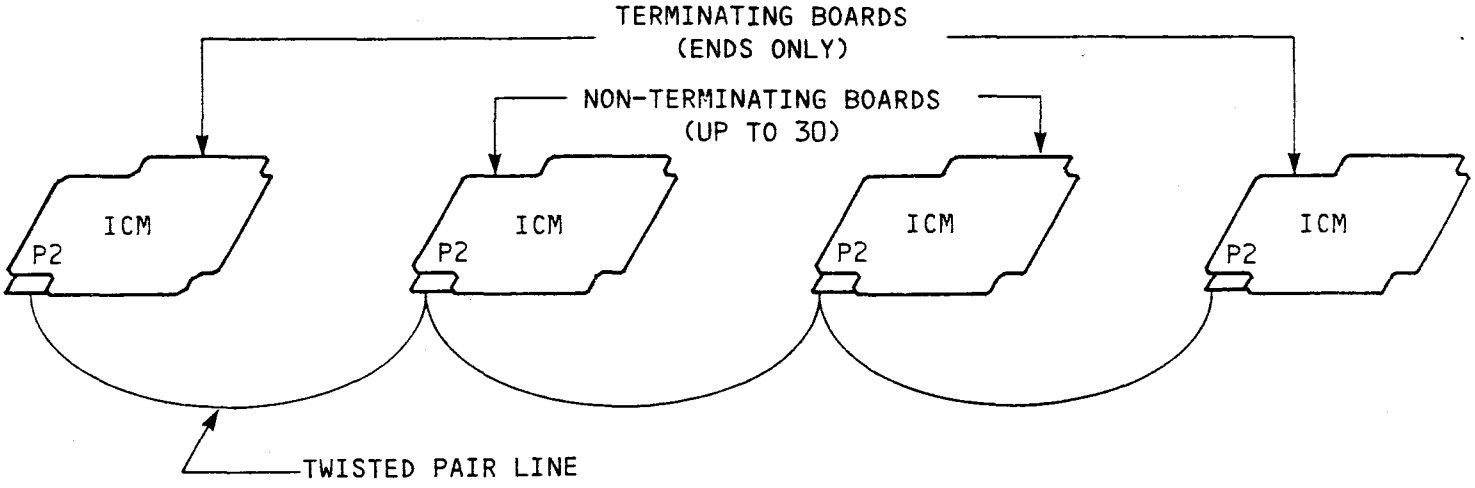


FIGURE 2-2. MULTIDROP SYSTEM TERMINATIONS

TABLE 2-5. LINE TERMINATION SELECT JUMPERS AT J8 & J9

LINE	JUMPER CONNECTIONS
Terminated	E41 to E40, E44 to E43
Non-terminated	E41 to E42, E44 to E45
Default to non-terminated	No jumpers installed at E40 to E45

2.3.7 Connector P2 as EIA/Multidrop Interface Selection (J7)

Connector P2 can be designated as either an EIA or multidrop interface by positioning jumper J7 as indicated in Table 2-6. Since the multidrop and EIA interfaces share connector P2, only one interface can be selected at a time.

TABLE 2-6. EIA/MULTIDROP INTERFACE SELECTION JUMPER POSITIONS AT J7

INTERFACE SELECTED	JUMPER POSITION
Multidrop	E38 to E39
EIA	E38 to E37
Default to multidrop	No jumpers, E37-E39

2.3.8 Multidrop Daisy Chain Jumpers (J10, J11, & J12)

Three special jumper plugs (J10, J11, and J12) are provided to facilitate connections in systems where more than two ICMs are used. These jumper plugs connect pins at the EIA/industrial line port, P2. Connected are P2-12 to P2-13, P2-22 to P2-23, and P2-24 to P2-25. Pins P2-13, -23, and -25 connect directly to board circuitry while P2-12, -22, and -24 connect to circuits only if jumpered. Figure 2-3 shows the use of these jumpers in a system using three ICMs. On ICM 2, pins 23, 25, and 13 connect to one multidrop line while pins 22, 24, and 12 connect to the other. Pin 14 on one end only can be used for an earth ground for the shield.

CAUTION

If a TM 990/308 is used in the EIA mode, the user should ascertain whether pins 12, 22, and 24 are used by the EIA device prior to making any connections to connector P2. If any of these pins are used by the EIA device, then the appropriate tie point jumper must be removed. Failure to heed this warning could result in damage to the equipment or a non-operative system.

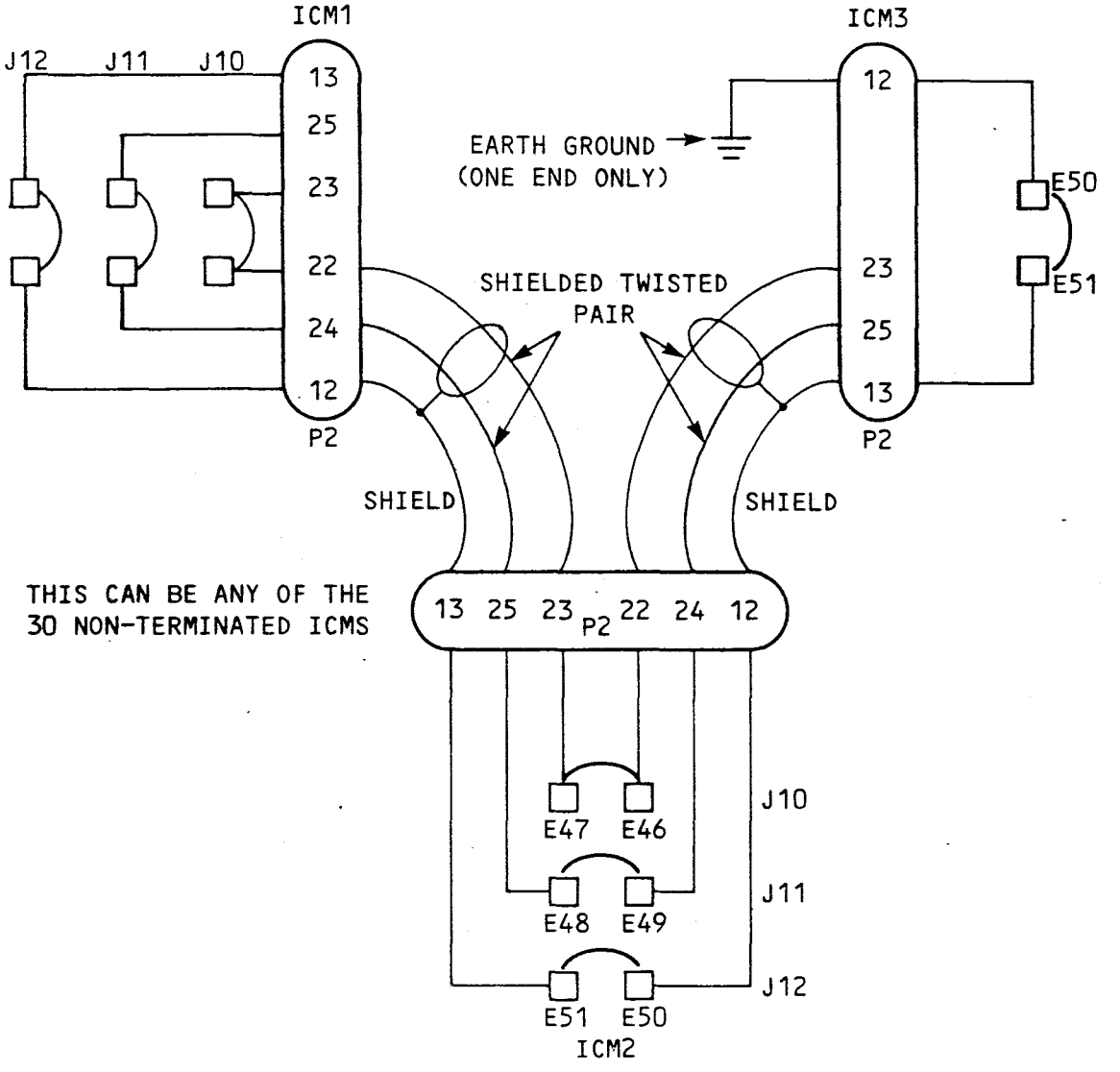


FIGURE 2-3. MULTIDROP DAISY CHAIN JUMPER EXAMPLE

2.3.9 Clock Jumper (J6)

The clock jumper (J6) must be connected from E35-E36. Jumper position E35-E34 is reserved and should not be used.

2.3.10 Download Jumper (J2)

The download jumper (J2) is used to select either the host's PRES- or RESTART- lines when a download function is commanded to a system over the industrial line. The download command essentially serves two purposes: 1) It can be used to initialize the ICM and, 2) It can be used to force the ICM to a known state in case the ICM gets locked up due to power surges or other line transients. When active (low), PRES- resets the TM 990's processor which issues an I/O reset (IORST-) to the bus. However, IORST- is inhibited from affecting the ICM whenever the ICM initiates the PRES- signal. PRES- is primarily intended for system initialization or full system reset if activated after initialization. When active (low), RESTART- causes the host's processor to perform a load function; reset is not activated. An active (low) at either the host's PRES- or RESTART- line will cause a hardware vector to locations 0000_{16} or at $FFFC_{16}$, respectively. Table 2-7 lists settings for jumper J2.

TABLE 2-7. DOWNLOAD JUMPERS AT J2

SIGNATURE	DESCRIPTION	JUMPER POSITION
PRES-	Reset	E18 to E19
RESTART-	Load	E18 to E17
--	No Download	E17 to E17A

2.3.11 Isolated Power Jumpers

Jumpers E30 to E31, E32 to E33, and E52 to E53 must always be in place.

2.4 LEDS AND SELF-TEST

Eleven LEDs on the board's outer edge (shown in Figure 2-4) indicate the following:

- FAULT (DS3): A self-test (not transmit or receive) error has occurred (the data LEDs will contain the error number, further defined in Table 3-3).
- XMIT (DS2): The ICM is currently transmitting.
- DWNLOAD (DS1): The ICM has received a download command; if jumpered at J2, the host will be requested to do a load or a reset. The LED will be illuminated for approximately 1 second.
- 0 to 7 (DS4 to DS11): Value at the data bits on the CRU interface (shown in Figure 3-5).

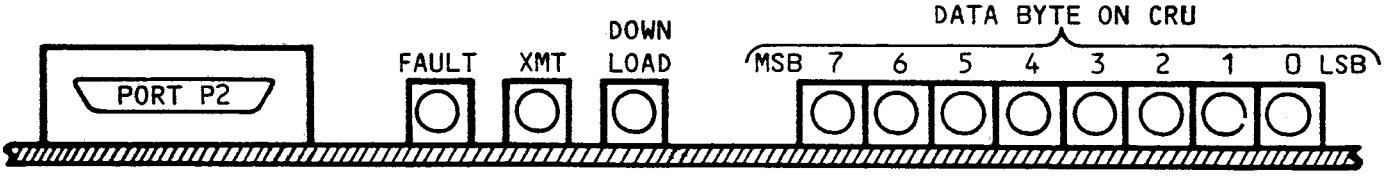


FIGURE 2-4. BOARD LEDS

When the board is reset by an IORST- from the processor (e.g., when the microcomputer RESET switch is enabled), the board self-test will execute. During this test, the FAULT light will illuminate for approximately one second while the eight data LEDs blink. Following this test, LEDs DS1 to DS3 will extinguish unless an error occurs. Should an error occur during the self-test for approximately 1 second, the data LEDs will show the error code with the FAULT light on until reset by software.

2.5 CONSTRUCTING A MULTIDROP CABLE

Figure 2-5 schematically shows an example cable that connects six ICMs. Note that all connectors are soldered as follows:

- Pin 12 to pin 13 Shield
- Pin 22 to pin 23 Black (Lo)
- Pin 24 to pin 25 White (Hi)
- The earth ground is soldered to the final-connector's unused pin 12

This design can be standard for any system; however, it requires that jumpers J10, J11, and J12 be installed on all boards (as shown in Figure 2-3).

Materials that can be used:

- Connectors (one for each ICM): ITT DB25P
  TRW CINCH DB25P
- Cable: Belden 8761 twisted pair, shielded, 22 AWG minimum

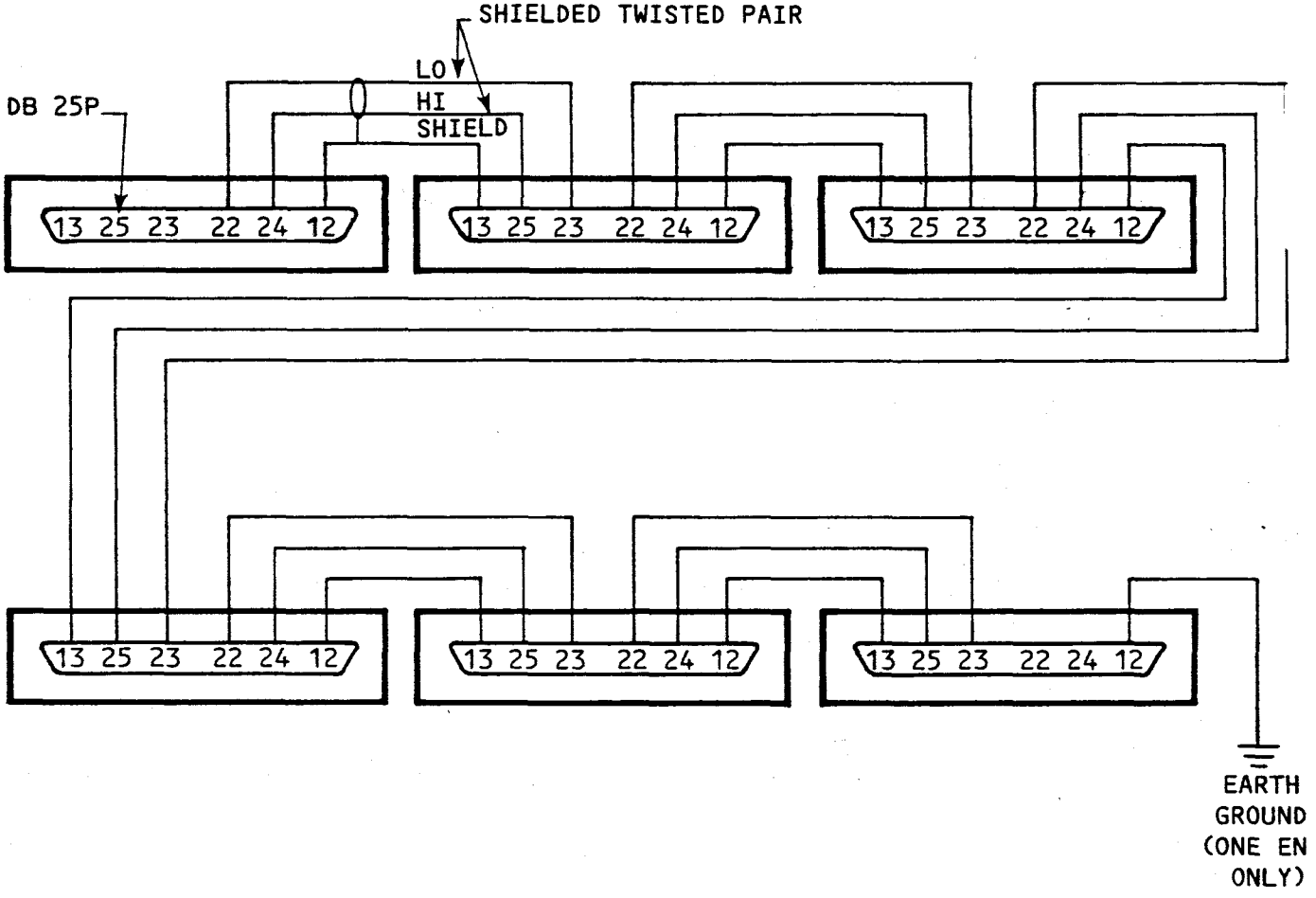
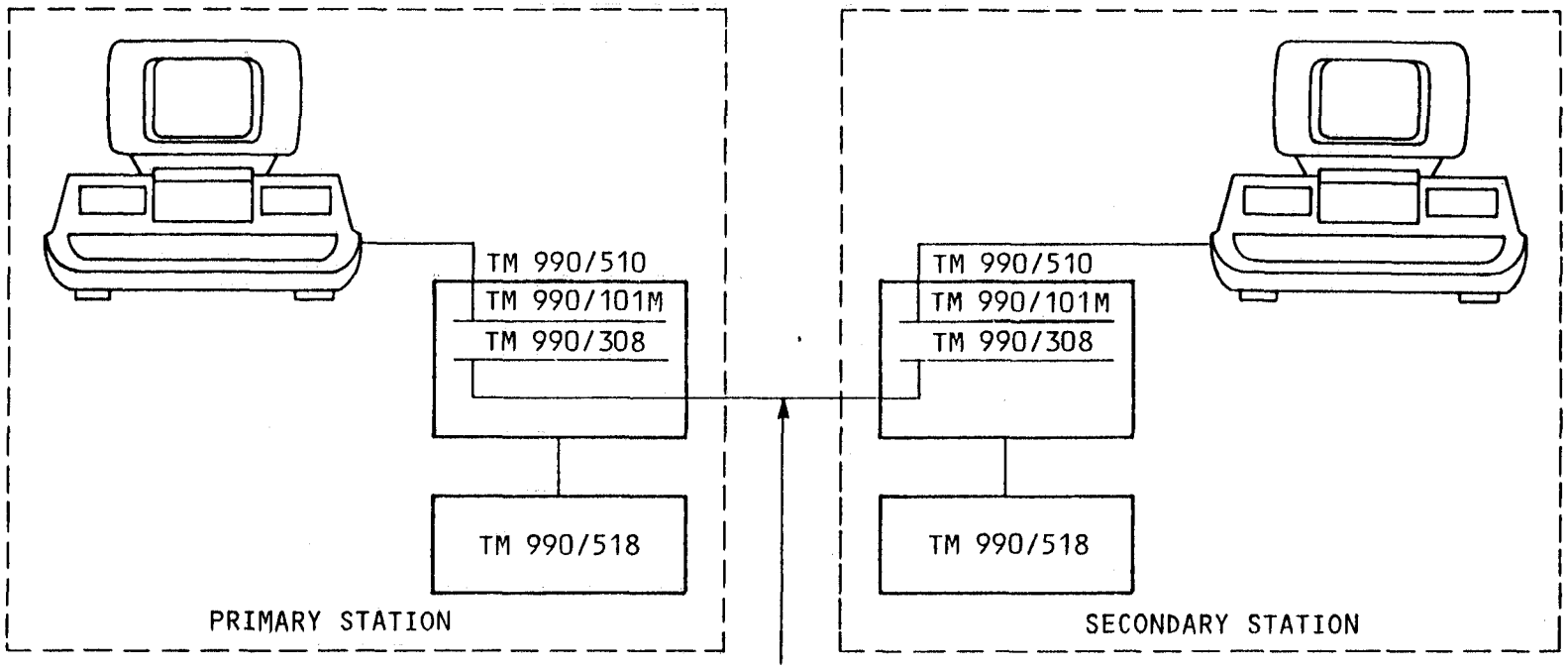


FIGURE 2-5. EXAMPLE CABLE FOR CONNECTING SIX ICMS



TERMINAL - EIA OR TTY
TM 990/101M - CPU MODULE
TM 990/308 - ICM
TM 990/510 - 4-SLOT CARD CAGE
TM 990/518 - DC POWER SUPPLY

FIGURE 2-6. SAMPLE TEST SYSTEM

2.6 SYSTEM FOR TESTING THE ICM

Figure 2-6 shows a test system that can be used to initially check out the ICM. In this system, data can be sent from one terminal to the other in order to evaluate the ICM and the industrial line (twisted pair line). The primary and secondary stations require the same components: terminal, CPU module, ICM, card cage, and power supply. System component descriptions, set-up procedures, and a test routine are presented in the paragraphs that follow.

2.6.1 CPU Modules

Either a TM 990/100M or TM 990/101M CPU module can be used. A TM 990/101M is used here for illustrative purposes.

2.6.2 Power Supply

The power supply requirements for each of the two stations is given in Table 2-8. Typical values for the currents are used. Figure 2-7 shows a TM 990/518 power supply attached to the back of a TM 990/510 card cage.

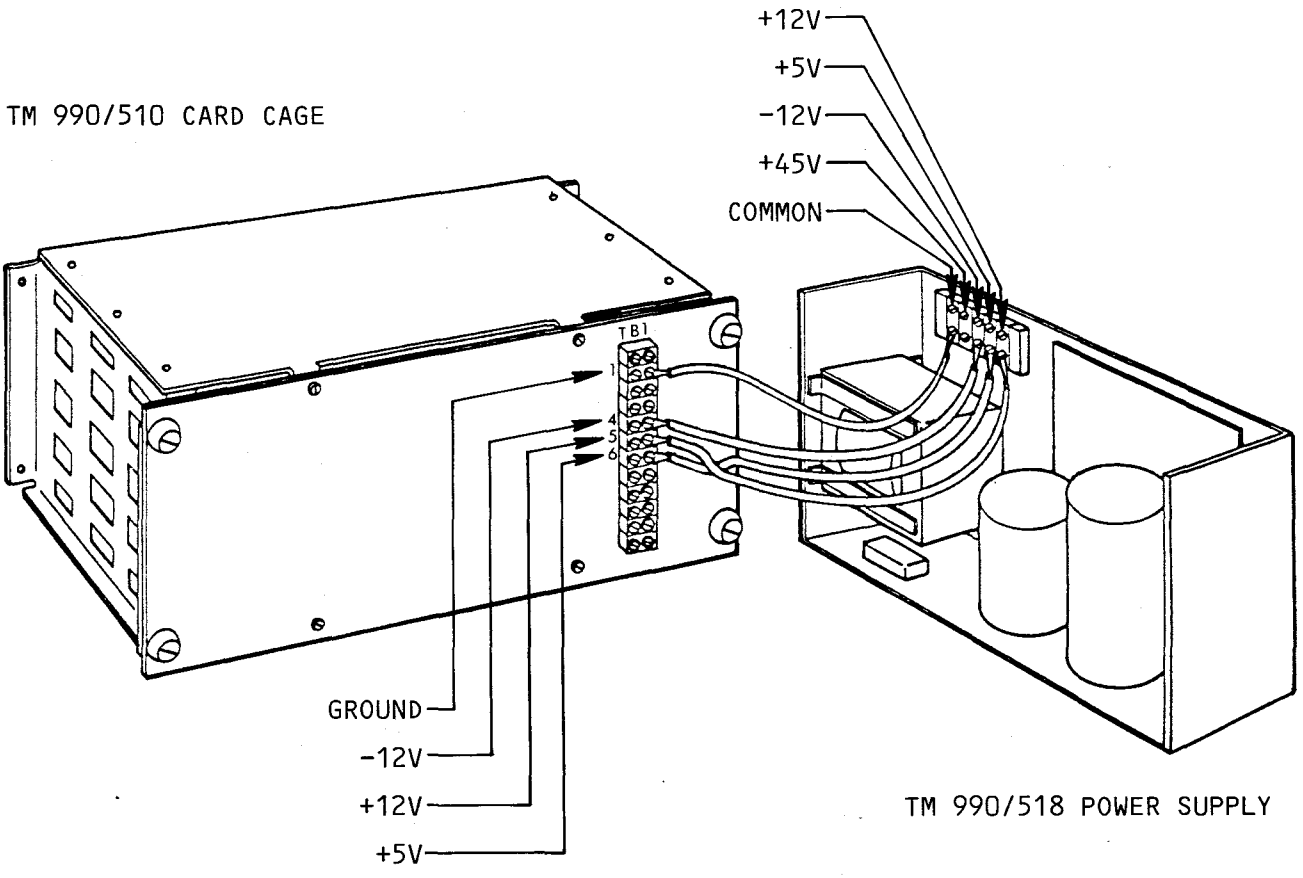


FIGURE 2-7. POWER SUPPLY CONNECTIONS (TEST STATION)

TABLE 2-8. POWER SUPPLY REQUIREMENTS (EACH TEST STATION)

	PRIMARY OR SECONDARY STATION		
	+5 V	+12 V	-12 V
TM 990/101M	1.8 A	0.4 A	0.3 A
TM 990/308	1.1 A	0.2 A	0.1 A
	2.9 A	0.6 A	0.4 A

A TM 990/518 OEM power supply provides the following dc voltages: +5 V @ 6 A, +12 V @ 0.9 A, -12 V @ 0.9 A, and 45 V @ 0.1 A. This power supply meets the requirements for the test system.

2.6.3 Card Cage, Cabling, and Power Supply Connections

The card cage (chassis) provides a protective housing for TM 990 modules, a terminal strip (TB1) for power supply connections and a 100-pin bus interface socket for communications between the CPU module and the ICM. Two card cages are available: 1) TM 990/510 (4-slot card cage) and 2) TM 990/520 (8-slot card cage). The TM 990/510 provides adequate storage for the test system. Figure 2-7 shows the connections between the TM 990/518 power supply and the TM 990/510 card cage.

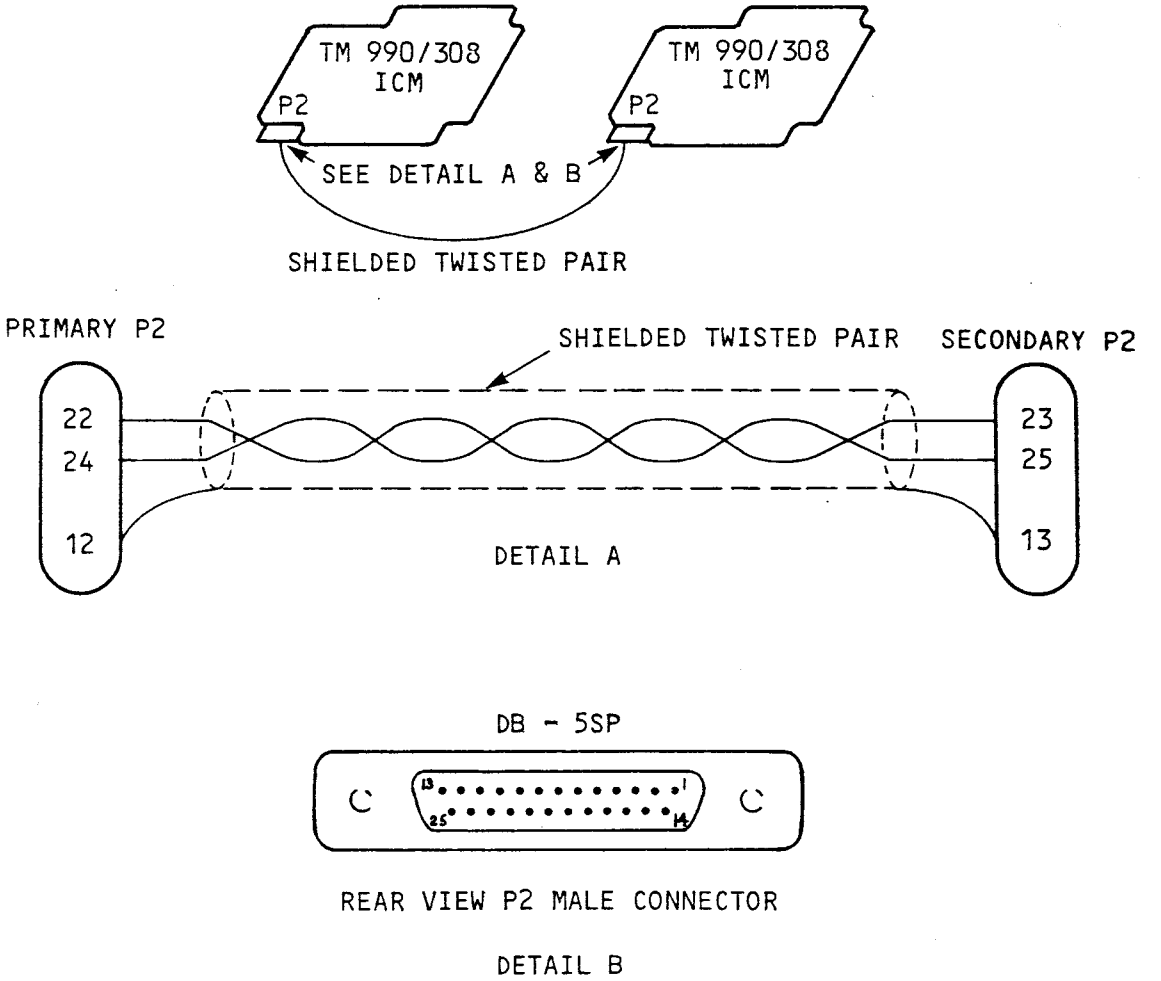


FIGURE 2-8. INDUSTRIAL LINE CONNECTIONS FOR TESTING TWO ICMS

The communication link between ICMs is a shielded twisted pair line (construction of an ICM cable is presented in 2.5). A Belden 8761 shielded two conductor twisted pair or equivalent line could be used. The multidrop and RS-232-C interfaces both use connector P2. However, only one type of interface can be selected at a time (Jumper selection of EIA/multidrop interface is explained in Section 2.3.7). A 25-pin RS-232 male plug, type DB25P, is required to make the proper connection to P2. The shield for the twisted pair connects to P2-13, and the twisted pair lines connect to P2-25 and P2-23. Jumpers J10, J11, and J12 must be installed, especially on the board where the cable is soldered to at the connector to P2-12, -22, and 24. Figure 2-8 shows the required connections between the ICMs.

2.6.4 Switch/Jumper Settings

Set the following switches and jumpers at the two ICM boards:

		Jumper/ Switch	Setting at ICM 1 & 2
1)	Data rate selection of 1200 baud	J3	E20-E21
2)	Address decode deselected	S2-7	ON
3)	Module address	S2-1/-6	N/A*
4)	CRU base address selection	S1	$1FE0_{16}$ (all OFF)
5)	Interrupt level selection	J1	N/A*
6)	Line termination selected	J8,J9	E41-E40,E44-E43
7)	Multidrop interface selection	J7	E38-E39
8)	Clock jumper	J6	E35-E36
9)	Download jumper	J2	E17-E17A
10)	Isolated power jumpers	J4,J5,J13	Jumpered
11)	P2 board-to-board connections	J10,J11,J12	Jumpered

*The broadcast mode is used for simplicity; this mode addresses all ICMs on the industrial line. No interrupts are used; this uses polling only.

If necessary, review subsection 2.3 for information relevant to these switch/jumper settings.

2.6.5 Test Routine

Figure 2-9 shows example code to test each board in an ICM system. This program can be assembled and object loaded by cassette, or the object can be loaded using the memory inspect/change TIBUG command. Load the code in RAM at all ICM systems beginning at address $FE00_{16}$, the address shown in the first column of the listing (the microcomputer board memory can be used). After loading, set the program counter to $FE00_{16}$ and issue the TIBUG execute (E) command at all ICM setups. The program alternately polls the ICM and the terminal for a message. The message, terminated by a carriage return, is entered at one of the terminals. After the carriage return is entered, the message will be sent to and written on the destination terminal.

Object Code

Address

```
                *
                * CHECK FOR MESSAGE AT TERMINAL AND HOST
                *
FE00 02E0   START  LWPI WP              SET WORKSPACE POINTER
FE02 FEA8
FE04 020C          LI   R12,>80         CRU ADDRESS OF TMS 9902
FE06 0080
FE08 0201          LI   R1,BUFFER+2     MESSAGE DATA START IN R1
FE0A FEC4
FE0C 04C3          CLR  R3              CLEAR BYTE COUNTER
FE0E 1F15          TB   21              IS CHARACTER AT TERMINAL?
FE10 1306          JEQ  TRMNAL          YES, GO RECEIVE CHARACTER
FE12 020C          LI   R12,>1FE0       NO, LOAD CRU ADDR OF ICM
FE14 1FE0
FE16 1E0E          SBZ  14              MASK OUT ICM INTERRUPTS
FE18 1F0C          TB   12              READ REQUEST COME IN?
FE1A 1311          JEQ  ICM             YES, RECEIVE MESSAGE FROM ICM
FE1C 10F1          JMP  START           NO, RESTART POLLING
                *
                * ROUTINE FOR HOST TO RECEIVE MESSAGE FROM SENDING TERMINAL
                *
FE1E 04C2   TRMNAL CLR  R2              INITIALIZE CHARACTER RECEIVER
FE20 2EC2   LOOP1  XOP  R2,11           RECEIVE/ECHO CHARACTER
FE22 DC42          MOVB R2,*R1+         MOVE CHARACTER TO BUFFER
FE24 0583          INC  R3              INCREMENT CHAR COUNTER
FE26 0282          CI   R2,>0D00        CAR RET ENTERED?
FE28 0D00
FE2A 16FA          JNE  LOOP1           NO, GET NEXT CHARACTER
FE2C DC60          MOVB @LFZERO,*R1+ YES, ENTER LINE FEED
FE2E FEA6
FE30 D460          MOVB @LFZERO+1,*R1  FOLLOW WITH ZERO BYTE
FE32 FEA7
FE34 0223          AI   R3,4            ADD FIRST BYTE, LF, & 00
FE36 0004
FE38 2FA0          XOP  @LFZERO,14      LINE FEED TO SENDING TERMNL
FE3A FEA6
FE3C 1016          JMP  SEND            GO WRITE MESSAGE TO ICM
                *
                * ROUTINE TO RECEIVE A MESSAGE FROM THE ICM
                *
FE3E 1F09   ICM    TB   9               IS THIS THE FINAL BYTE?
FE40 1308          JEQ  LAST1           YES, GO TO ERROR ROUTINE
FE42 3631          STCR *R1+,8          STORE BYTE IN BUFFER
FE44 1D0B          SBO  11              ACKNOWLEDGE RRQ
FE46 1F0C   LOOP2  TB   12              HAS RRQ BEEN RESET TO ZERO?
FE48 13FE          JEQ  LOOP2           NO, WAIT FOR RRQ TO BE RESET
FE4A 1E0B          SBZ  11              ACKNOWLEDGE RRQ RESET
FE4C 1F0C   LOOP3  TB   12              NEW BYTE RECEIVED?
FE4E 16FE          JNE  LOOP3           NO, WAIT FOR NEW BYTE
FE50 10F6          JMP  ICM             YES, GET NEXT BYTE
FE52 3603   LAST1  STCR R3,8            STORE BYTE
FE54 C0C3          MOV  R3,R3           BYTE = 0 (LAST BYTE)
FE56 1301          JEQ  EXIT            YES, EXIT
FE58 1020          JMP  ERROR           NO, GO TO ERROR ROUTINE
FE5A 1D0B   EXIT   SBO  11              ACKNOWLEDGE RRQ
FE5C 1F0C   LOOP4  TB   12              HAS RRQ BEEN RESET TO ZERO?
FE5E 13FE          JEQ  LOOP4           NO, WAIT FOR RRQ TO BE RESET
FE60 1E0B          SBZ  11              ACKNOWLEDGE RRQ BEING RESET
```

FIGURE 2-9. EXAMPLE CODE TO READ/WRITE OVER INDUSTRIAL LINE (Sheet 1 of 2)

Object Code

Address

```
                *
                * ROUTINE TO WRITE MESSAGE TO RECEIVING TERMINAL
                *
FE62 0201   WRITE   LI    R1,BUFFER+4   ADDRESS OF MESSAGE START
FE64 FEC6
FE66 2F91           XOP   *R1,14        WRITE MESSAGE TO TERMINAL
FE68 10CB           JMP   START         GO BACK TO POLLING
                *
                * ROUTINE TO WRITE MESSAGE FROM HOST TO ICM
                *
FE6A 020C   SEND    LI    R12,>1FE0     ICM/HOST CRU ADDRESS IN R12
FE6C 1FE0
FE6E 0201           LI    R1,BUFFER     ADDRESS OF MESSAGE IN R1
FE70 FEC2
FE72 0603   SEND1   DEC   R3            COUNTER=0? (LAST BYTE?)
FE74 1308           JEQ   LAST2         YES, LAST BYTE AT CRU
FE76 3231           LDCR  *R1+,8        LOAD BYTE ON CRU
FE78 1D0B           SBO   11            SET DAV TO ONE
FE7A 1F0B   LOOP5   TB    11            HAS WRQ GONE TO ONE?
FE7C 16FE           JNE   LOOP5         NO, LOOP UNTIL WRQ A ONE
FE7E 1E0B           SBZ   11            YES, RESET DAV TO ZERO
FE80 1F0B   LOOP6   TB    11            IS WRQ RESET TO ZERO?
FE82 13FE           JEQ   LOOP6         NO, LOOP UNTIL A ZERO
FE84 10F6           JMP   SEND1         YES, GET NEXT BYTE
FE86 3211   LAST2   LDCR  *R1,8         LOAD LAST BYTE ON CRU
FE88 1D0A           SBO   10            TRANSMIT A ONE (LAST BYTE)
FE8A 1D0B           SBO   11            DAV A ONE, ICM GET LAST CHAR
FE8C 1F0B   LOOP7   TB    11            HAS WRQ GONE TO ONE?
FE8E 16FE           JNE   LOOP7         NO, LOOP UNTIL WRQ A ONE
FE90 1E0A           SBZ   10            YES, RESET TRANSMIT BIT
FE92 1E0B           SBZ   11            AND RESET DAV
FE94 1F0B   LOOP8   TB    11            IS WRQ RESET TO ZERO?
FE96 13FE           JEQ   LOOP8         NO, LOOP UNTIL WRQ IS RESET
FE98 10B3           JMP   START         RETURN TO PROGRAM START
FE9A 2FA0   ERROR   XOP   @ERRMSG,14    OUTPUT ERROR MESSAGE
FE9C FEA0
FE9E 10B0           JMP   START         RETURN TO POLLING
FEA0   45   ERRMSG  TEXT  'ERROR '
FEA1   52
FEA2   52
FEA3   4F
FEA4   52
FEA5   20
FEA6   0A   LFZERO  BYTE  >0A           LINE FEED
FEA7   00           BYTE  >00           MESSAGE DELIMIITER (ZERO BYTE)
FEA8        WP      BSS   26            RESERVE 26 BYTES FOR WORKSPACE
FEC2 FF00   BUFFER  DATA  >FF00         BROADCAST ADDRESS, NO DOWNLOAD
```

FIGURE 2-9. EXAMPLE CODE TO READ/WRITE OVER INDUSTRIAL LINE (Sheet 2 of 2)

2.7 TYPICAL SYSTEMS USING THE ICM

Two systems that use the TM 990/308 ICM are described in order to familiarize the user with typical setup procedures. These systems are:

- System using multidrop interface
- System using modem and telephone lines

The first system allows several (2 up to 32) computers (CPU modules) to talk with each other over a twisted pair line. This system uses the multidrop interface and allows communications up to 10,000 feet in an electrically noisy industrial environment. The second system allows two CPU modules to communicate with each other using telephone lines. This system uses a modem (MOdulator, DEModulator) to encode the CPU's output to audio tones that are then sent over telephone lines to a remote location which has another modem to decode the audio tones, then relay the initial message to the second CPU module. This system uses the RS-232-C EIA interface and allows two CPU modules to communicate using telephone lines.

2.7.1 System Using Multidrop Interface

Figure 2-10 shows a system which uses the multidrop interface. The system consists of two main sections: the primary station and the secondary station. The primary station serves as "master" and supervises the control processes that are implemented by the secondary station. In addition, the primary station performs data logging and error reporting. The secondary station is a programmable control system that provides a variety of industrial control routines.

2.7.1.1 CPU Modules. System 1 requires either a TM 990/101M or TM 990/100M CPU module for both the primary and secondary stations.

2.7.1.2 Power Supply. The power supply requirements (typical values) for both the primary and secondary stations are given in Table 2-9.

TABLE 2-9. POWER SUPPLY REQUIREMENTS (SYSTEM 1)

PRIMARY STATION				SECONDARY STATION			
	+5 V	+12 V	-12 V		+5 V	+12 V	-12 V
TM 990/101M	1.8 A	0.3 A	0.25 A	TM 990/101M	1.8 A	0.30 A	0.25 A
TM 990/308	1.1 A	0.2 A	0.05 A	TM 990/308	1.1 A	0.20 A	0.05 A
				TM 990/305	1.5 A	0.75 A*	
	2.9 A	0.5 A	0.30 A		4.4 A	1.25 A	0.30 A

* To interface with a TM 990/5MT industrial control system, a +12 V ±3% @ 0.75 A power supply is required.

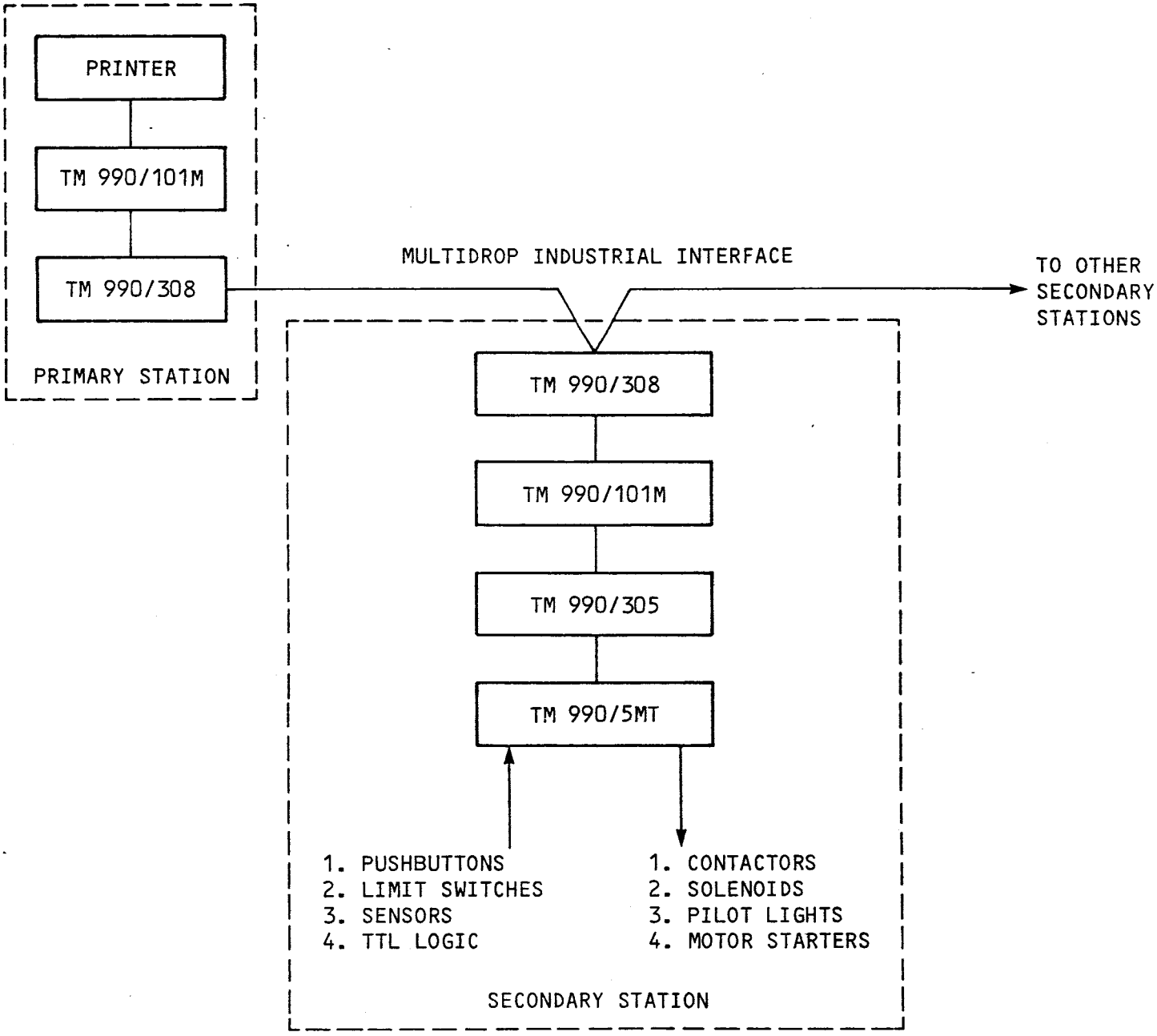


FIGURE 2-10. SAMPLE SYSTEM USING MULTIDROP INTERFACE

2.7.1.3 Card Cage, Cabling, and Power Supply Connections. Use of either the TM 990/510 or TM 990/520 card cage greatly simplifies operation and setup. The CPU module communicates with the ICM and I/O module via the 100-pin bus interface socket of the card cage.

Data communications between ICMs or between an ICM and other modules will occur over a twisted pair line (See Figure 2-5). Up to 10,000 feet of shielded twisted pair wire (22 gage minimum) can be used.

If either a TM 990/510 or TM 990/520 card cage is used, power supply connections can be made to terminal block (TB1) on the back side of the card cage (See Figure 2-7 for power supply connections).

2.7.1.4 Switch/Jumper Settings. The switches/jumpers discussed in Section 2.3 must be properly set/positioned.

2.7.2 System Using RS-232-C EIA Interface

Figure 2-11 shows a system that uses the RS-232-C EIA interface. This system allows two CPU modules to communicate using telephone lines. The TM 990/308 provides the necessary interface for the Bell 208 modem. The modems at both ends of the telephone line provide the necessary modulation/demodulation for communication over telephone lines.

2.7.2.1 CPU Modules. Either TM 990/101M or TM 990/100M CPU modules can be used in System 2 also.

2.7.2.2 Power Supply. The power supply requirements (typical values) for the CPU module and ICM are given in Table 2-10.

TABLE 2-10. POWER SUPPLY REQUIREMENTS (SYSTEM 2)

	+5 V	+12 V	-12 V
TM 990/101M	1.8 A	0.3 A	0.25 A
TM 990/308	1.1 A	0.2 A	0.05 A
	2.9 A	0.5 A	0.30 A

2.7.2.3 Card Cage, Cabling, and Power Supply Connections. Either the TM 990/510 or TM 990/520 card cage can be used in system 2. The CPU module communicates with the ICM via the 100-pin bus interface socket of the card cage. The ICM communicates with the Bell 208 modem via a TM 990/502 cable connected to the ICM's P2 connector (EIA interface).

If either a TM 990/510 or TM 990/520 card cage is used, power supply connections can be made to terminal block (TB1) on the back side of the card cage (see Figure 2-7).

2.7.2.4 Switch/Jumper Settings. The switches/jumpers discussed in Section 2.3 must be properly set/positioned.

2.7.2.5 Phone Connections. Consult the modem manufacturers instructions for establishing the connection and carrier on the phone line.

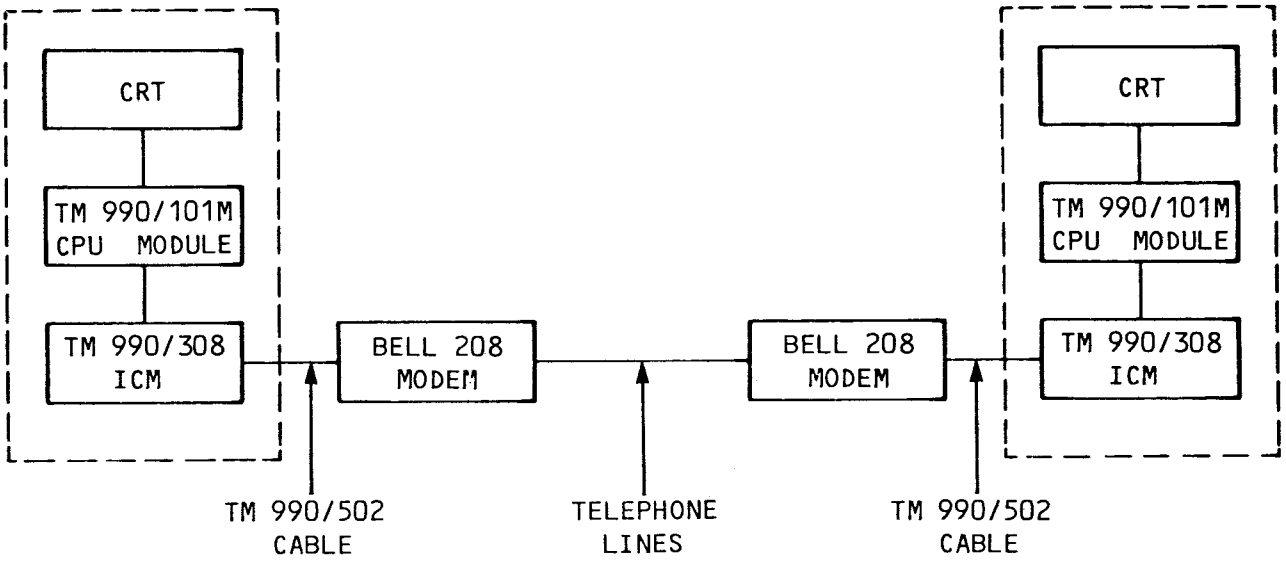


FIGURE 2-11. SAMPLE SYSTEM USING RS-232-C EIA INTERFACE

2.7.3 Sample System Using ICM and Mass Storage

ICMs can be used to provide the interface between several industrial control systems and an on-line bulk storage medium such as a floppy disk drive. A system using bulk storage and ICMs is shown in Figure 2-12. The TM 990/303 floppy disk controller provides the necessary control capability to interface a maximum of four double-sided, dual-density floppy disk drives to the TM 990 system bus. In the system shown, the floppy disk controller is providing the interface between a floppy disk drive and a TM 990/101M CPU module. The TM 990/201/203 memory expansion module provides the necessary storage required by the floppy disk controller. The ICM in the primary station provides the interface between the TM 990 system bus and the multidrop line. The multidrop line can accommodate up to 31 additional ICMs that communicate with terminals or industrial I/O control systems via the TM 990/101M CPU module. In this particular system, a control routine stored on a floppy disk is relayed to the primary station via the TM 990/303 floppy disk controller. The ICM in the primary station provides the interface to the multidrop line which links the primary and secondary stations. The secondary station receives the necessary control routine and sends it to the 5MT I/O controller which provides the proper control for the load.

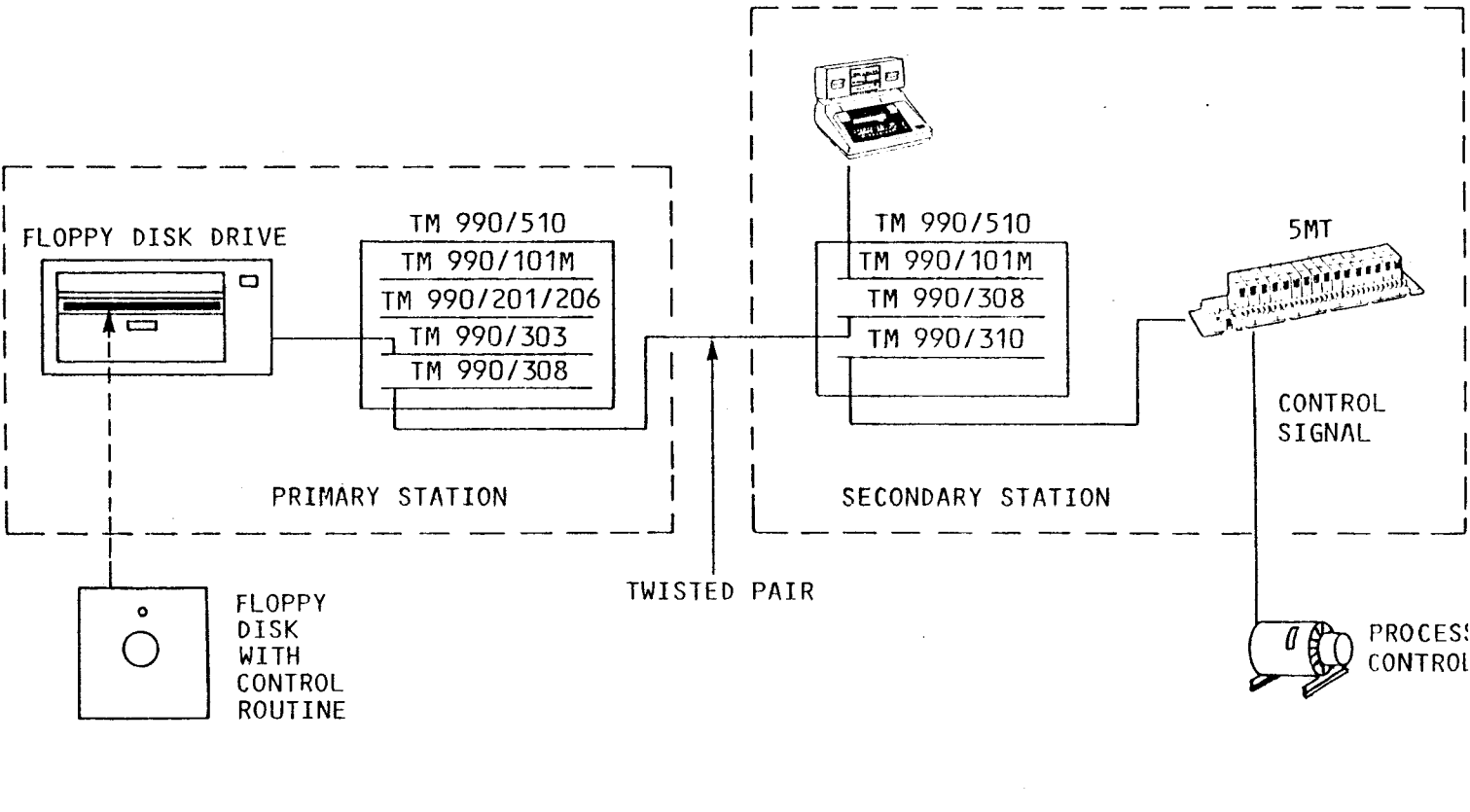


FIGURE 2-12. SAMPLE SYSTEM USING ICMS AND BULK STORAGE

3.2.2 Message Length and ICM Memory Size

If a message is too large for the ICM's buffer, message receipt will stop and an error message will be presented to the host (buffer full error, error code 08_{16} or 09_{16}). The TM 990/308 is shipped with 1024 bytes of RAM; of which, 928 bytes are available for message buffering. The user can add another 1024 bytes in exapnsion sockets U39 and U41 (TMS 4040-45s) for a total of 1952 bytes. Paragraph 4.2.5 further explains RAM expansion.

CAUTION

Both considerations listed below (3.2.3 and 3.2.4) can cause either loss of data or injury to equipment. It is necessary, therefore, that the user system include system protocol that will avoid this.

3.2.3 Possible Data Loss During Transmission

While receiving data from the industrial line, the TM 990/308 transfers the data from the line to its buffer area. It next transfers the buffered contents to its host computer. During this latter transfer, the industrial line is not being monitored by the ICM; thus, while the ICM is communicating to its host, it cannot recognize any other messages being sent to it on the industrial line, resulting in possible loss of data transmitted. Therefore, the system protocal must include a method to ascertain that the ICMs are not busy before sending a message over the industrial line. This could require the ICM that last received data to send back an acknowledgement verifying that it has completed its "housekeeping" chores and is ready to receive again (is monitoring the communication link again). In effect, this ICM will retain possession of the industrial line until it states that it is through with all the facets of receiving data.

3.2.4 Multiple Transmissions on Industrial Line

CAUTION

Without a protecting system protocal, two ICMs can contend for industrial line use, resulting in both ICMs transmitting at the same time. This can damage the ICM boards as well as result in no communication.

3.2.5 Possible System Configurations

In a communication system of two ICMs, system protocal is relatively simple. In larger systems, more consideration is needed, especially to ensure that only one ICM has access to the industrial line at any one time. Two possible system configurations that could meet the system requirements listed in this subsection are shown in Figures 3-1 and 3-2.

Figure 3-1 shows a system using one central primary microcomputer and its ICM (primary ICM) to control the industrial line. Secondary ICMs on the line communicate between one another only through the primary ICM. The primary ICM continously polls all secondary ICMs to see if one of them desires the communication link. Each inquiry receives a response. When a secondary ICM

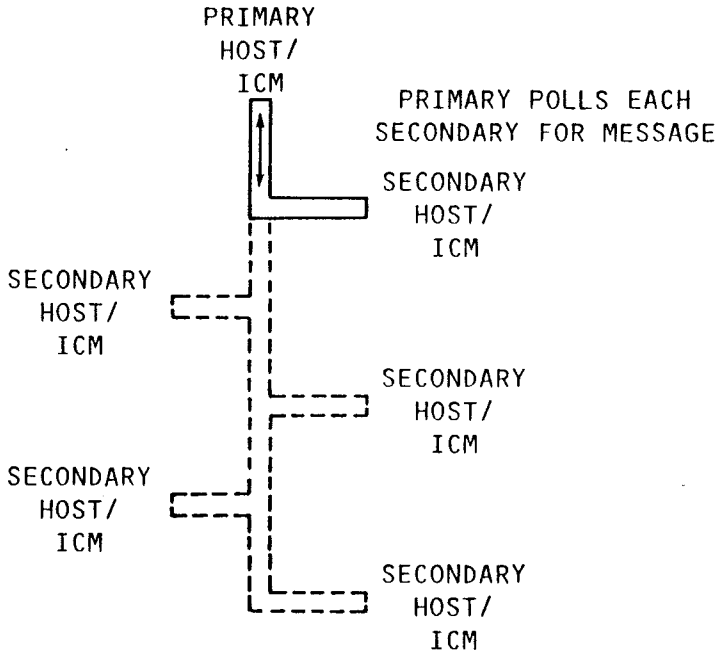


FIGURE 3-1. POLLED SYSTEM USING PRIMARY AND SECONDARY ICMs

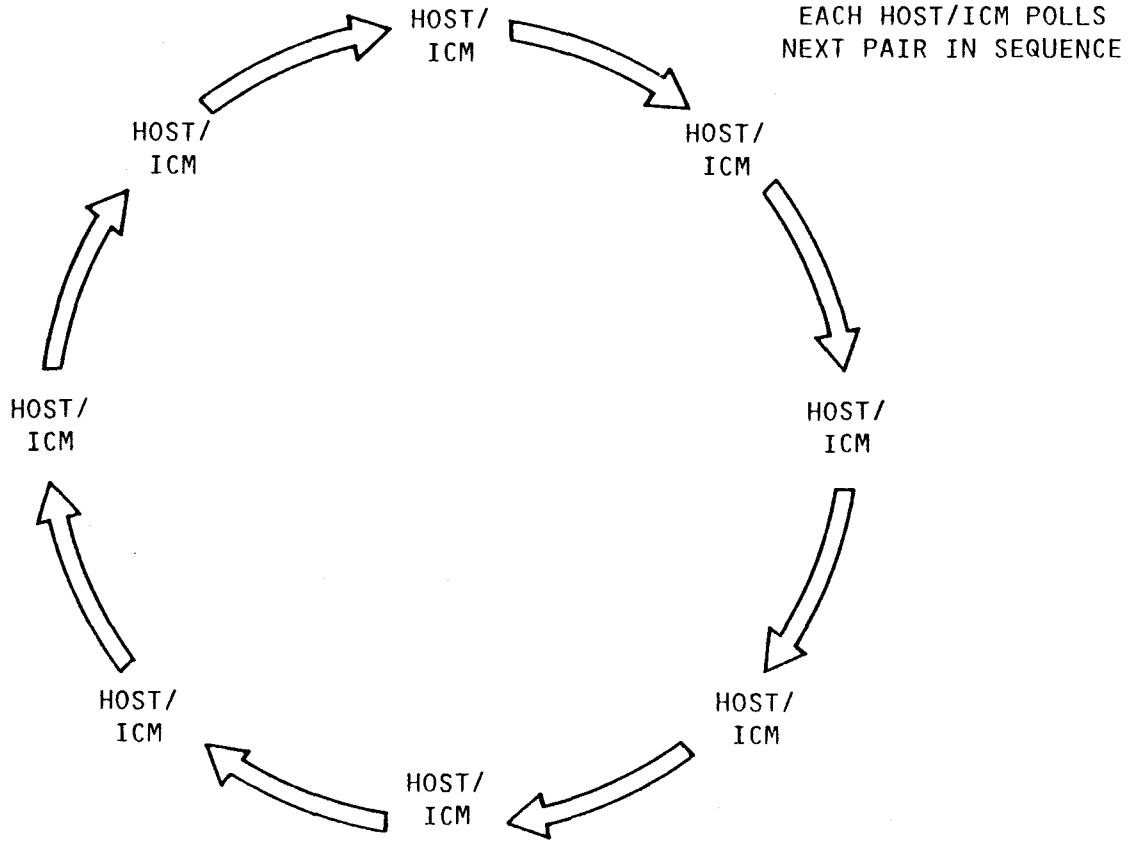


FIGURE 3-2. SYSTEMS USING INTERCONNECTED ICMs

states that it wants to communicate with another ICM, it sends the message (and destination ICM number) to the primary ICM which in turn retransmits it to the designated ICM and then resumes polling. In this manner,industrial-line traffic is fully managed by a central authority. A timeout routine could be used by the primary ICM and host in case an answer is not received from the polled secondary ICM.

Figure 3-2 shows a similar system but which does not employ a central authority. In this system, an ICM communicates with only the next ICM in a specified series. Each ICM is allowed industrial line transmission only after receiving a message from the preceding ICM in the series; thus, only one ICM will be on the line at one time. This message link is in continuous transmission from one ICM to the next; thus allowing the receiving subsystem to either use the line for a data message or (via message) to relinquish line usage to the next ICM in the series. It could be possible to make two message classifications: one for relinquishing line access which goes only to the next ICM in the series, or a second for data messages which goes directly to the destination ICM; this receiving ICM then restarts passing the line access message to the next designated ICM in the series, and so forth.

3.3 SUMMARY OF COMMUNICATION MODES

Three modes of communication are used in a TM 990/308 system:

- Serial transmission over the industrial line or EIA interface from one ICM to another,
- Host-to-ICM and ICM-to-host interrupts,
- Serial transmission using the CRU for byte and status bit transmission between (to and from) the host microcomputer and ICM.

These modes are described in general in this paragraph, and in more detail in other referenced parts in this section.

3.3.1 Serial Transmission Over Industrial Line or EIA Interface

The TMS 9903 is the interface between the ICM and either the industrial line or EIA interface. Following a reset via software (from host) or hardware, the ICM begins monitoring for an incoming message to its TMS 9903 (receive mode). The ICM will monitor either the industrial line or an EIA device at P2 depending upon the setting of jumper J7.

Before transmitting over the industrial line or EIA interface, the sending ICM first receives the message over the CRU from its host. Then the ICM is commanded by its host to send the message. The ICM transmits the data in the serial format shown in Figure 3-3 using its TMS 9903 as the synchronous interface with the industrial line or asynchronous interface with an EIA device. (Paragraph 3.6 discusses message format in detail.) In data receipt, the ICM receives an interrupt from the industrial line or EIA device (via the TMS 9903). If the message is for the ICM (address byte matches ICM switch-selected address and address decoding is jumpered), the ICM reads the bytes, stores these in its buffer memory, and then signals its host that a message is available for it.

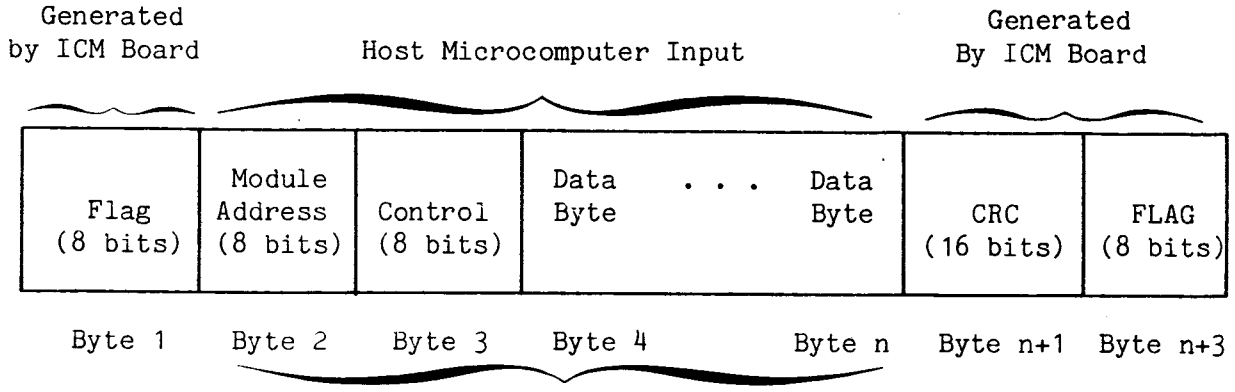


FIGURE 3-3. MESSAGE FORMAT

3.3.2 Host/ICM Communication Through Interrupts

The host microcomputer can be interrupted by its ICM, or the host can send an interrupt to its ICM. This is the usual means for either the ICM or the host to initiate handshaking before transmitting a message over the CRU (summarized in paragraph 3.3.3). All such interrupts are triggered through a CRU bit.

The level of the interrupt from the ICM to the host is jumper selectable at the ICM board (J2). Before the host can recognize an interrupt, the host must have first initialized the correct interrupt vector scheme, and it also must have enabled that interrupt at both its TMS 9901 and microprocessor status register interrupt mask as well as at an interrupt-enabling CRU bit (output bit 14).

The host can interrupt the ICM by setting a bit at the CRU interface (DAV, output bit 11). However, the host cannot interrupt the ICM if the ICM is servicing a higher priority interrupt than the host-to-ICM interrupt (level 4 at ICM). Higher-level interrupts to the ICM include:

- hardware reset (level 0),
- software reset which can be issued by the host (level 1),
- ICM internal timer (level 2),
- interrupt from the ICM's TMS 9903 when an incoming message is sensed over the industrial line or EIA interface (interrupt 3). (The TMS 9903 is the ICM's interface with the industrial line or EIA device.)

It is good programming practice for the host to have an interrupt-driven timeout routine to prevent the host from being locked up while waiting for a response from the ICM (the ICM could fail to set or reset the bit value of RRQ or WRQ which are polled for a change by the host during handshaking in reads and writes). With a timeout routine, the locked up host will be interrupted after a designated time. The host can use its TMS 9901 or TMS 9902 timers for this. Of course, the TMS 9901 interrupt (level 3) or the TMS 9902 interrupt (level 4) must be enabled at the host. Note that the host interrupt mask must be at a lower or equal priority than the timer; thus, if the host was answering an interrupt from the ICM during handshaking, the timer's interrupt must be a higher level than the ICM interrupt or the host must change its interrupt mask to accommodate the timer. Since data transfer is most efficiently handled by polling CRU bits once the ICM or host is interrupted, the host's interrupt mask can be modified to accommodate this.

Interrupt level from ICM to host is selected by the jumper setting at J1, explained in paragraph 2.3.5.

3.3.3 Host/ICM Serial Transmission Using the CRU

The CRU is the serial input/output mechanism for TM 990 products and is described (along with the CRU instructions) in detail in your microcomputer user's guide. Subsections 3.4 to 3.7 in this manual describe the host/ICM handshaking using the CRU. Transmission includes data bytes and error codes as well as status bits.

Figure 3-4 shows the derivative of the CRU software and hardware base addresses and the CRU bit address. These terms are used throughout this section. Figure 3-5 and subsection 3.4 further describe the CRU.

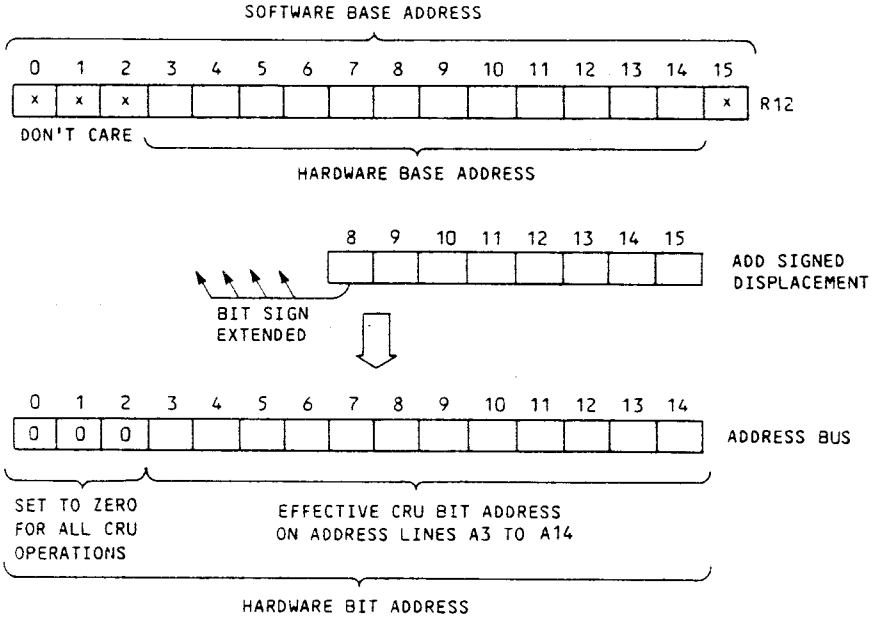


FIGURE 3-4. CRU ADDRESS NOMENCLATURE

3.4 CRU INTERFACE

Figure 3-5 depicts the CRU interface which is the communication link between the host microcomputer and its ICM. These CRU bits are further defined in detail in Tables 3-1 and 3-2. This 16-bit CRU space is located at the base address as set on the CRU Base Address Select switch (switch S-3). The setting of this switch is described in paragraph 2.6.4. For example purposes in this section, a CRU software base address of 0400_{16} (CRU hardware base address of 0200_{16}) is used. Error codes, which are read via the CRU, are listed in Table 3-3.

Output Bits From Host	Bit No.	Input Bits To Host
Output Data Bit (LSB)	0	Input Data Bit (LSB)
Output Data Bit (MSB)	7	Input Data Bit (MSB)
Enable Reading ICM Address	8	(zero)
Faulty Data Request	9	End of Block (EOB)
Transmit Data (Output Enable)	10	(zero)
Data Available (DAV)	11	Write Request (WRQ)
Self Test of ICM	12	Read Request (RRQ)
Echo Enable	13	(zero)
Enable CRU Interrupt to Host	14	CRU Interrupt Enabled at ICM
ICM Board Reset	15	CRU Interrupt Active at ICM

FIGURE 3-5. CRU INTERFACE

TABLE 3-1. CRU BITS, OUTPUT FROM HOST (Sheet 1 of 3)

Bit	Description
0 to 7	Byte to be Transmitted to ICM This is the byte to be buffered by the ICM and then transmitted over the industrial link. Bit 0 is the LSB; bit 7 is the MSB.
8	ICM Address Number If this bit is a one, the 6-bit value of the ICM's Module Address Select switch can be read on input bits 0 to 7 with the two most-significant bits (6 and 7) set to zeroes.
9	Faulty Data Request The ICM will transmit an error message to the host should an error occur during receipt of a message (e.g., buffer full, CRC error, timeout). The ICM will not transfer the data received in its buffer to the host; however, the host has the option of asking for the buffer data by setting bit 9 to a one. In response, the ICM will present the faulty data in the normal way; receipt of data is explained in detail in sub-section 3.7.2.
10	Transmit Message (Output Enable) If this bit is a one when the DAV bit is set to a one, the ICM will read the byte in bits 0-7 and buffer it as the last byte in the message. Then the ICM will send the entire buffer contents over the industrial line along with the flag and CRC bits as shown in Figure 3-3. After data transmission, the ICM returns to its receive mode and monitors for an incoming message on the industrial line or EIA interface.
11	Data Available from Host (DAV) If this bit is set to a one, an interrupt (level 4) is sent to the ICM. The ICM's reaction is dependent upon the value at output bits 9, 10, 12, and 13 (with DAV a one) as shown below: Output Control Bits Set to One — ICM Reaction 11 & 9 — Sends the ICM's buffer contents to the host in the normal manner. (An error had occurred during data reception by the ICM; now normal transfer of the data bytes will resume). 11 & 10 — Buffers the byte being transferred from host to ICM; and then transmit the entire ICM buffer contents over the industrial line.

TABLE 3-1. CRU BITS, OUTPUT FROM HOST (Sheet 2 of 3)

Bit	Description
	11 Bits 0-7 are a data byte to be received by the ICM. 11 & 12 Executes an ICM self-test. When complete, place an error/OK code on bits 0-7 to show test results, then set to one CRU input bits 9 (EOB) and 12 (RRQ) to show the host that an error/OK code is available. 11 & 13 Reads the the byte at bits 0-7, then transmits back to host the same byte and sets read request (RRQ, bit 12). In essence, this is an echo routine.
12	Self Test If this bit is a one when the DAV bit is set to a one, the ICM will execute a self test. At the end of the test, the ICM will communicate test results by placing the resulting error code on bits 0 to 7 (code 00 indicates no errors), setting the end-of-block bit, and then setting RRQ high. Error code interpretations are listed in Table 3-3. If an error occurs (a code other than 00), the FAULT light is lit. After receiving the error code, the host exits this mode by resetting to zero both this bit and the DAV bit; the ICM will respond by resetting RRQ and extinguish the FAULT light if lit. Then the ICM enters the receive mode and begins monitoring for an incoming message on the industrial line or EIA interface.
13	Echo Enable If this bit is a one when the DAV bit is set to a one, the ICM will read the byte at bits 0 to 7, buffer the byte, then write the same byte back to the host by applying it to bits 0 to 7 and setting RRQ (read request, bit 12) to a one. This mode of operation will continue as long as this bit is set.
14	Enable Host to Receive CRU Interrupt from ICM When this bit is set to a zero, interrupts from the ICM are masked (RRQ or WRQ going to a one causes an interrupt to the host). When bit 14 is a one, the CRU interrupt generated by the ICM is enabled. The level of the interrupt is jumper selectable at the ICM. Naturally, this level must also be enabled at the host microcomputer via its TMS 9901 and its status register interrupt mask. The setting of bit 14 can be checked by reading input bit 14 (Table 3-2). In other words, if this bit is set to a one, input bit 14 must also be a one; if this bit is a zero, input bit 14 must also be a zero.

TABLE 3-1. CRU BITS, OUTPUT FROM HOST (Sheet 3 of 3)

Bit	Description
15	ICM Reset When set to a one, a level one interrupt is issued to the ICM. It responds by entering the receive mode (checks the EIA or industrial line for incoming character). Setting this bit causes termination of any present mode of the ICM. Reset this bit to zero for normal operation.

TABLE 3-2. CRU BITS, INPUT TO HOST (Sheet 1 of 3)

Bit	Description
0 to 7	Input Data These bits can be a data byte, a code indicating an error or the last byte has been received, or the address number (0 to 63) in binary of the host's ICM (bits 0 to 5 only with bits 6 and 7 zeroes). The ICM address number will be that value set at the Module Address Select switch (S2-1 to S2-6) on the ICM. Bit 0 is the LSB. Table 3-3 lists error codes. Interpretation is as follows:

(Output Bit)	(Input Bits)			
Module Address Select Bit 8	End of Block Bit 9	RRQ Bit 12	Bits 0-7	Meaning of Bits
1	X	X		Bits 0-5 = ICM address no.
0	0	1		Data byte to host
0	1	1	YY	Error code
0	1	1	00	Previous byte was last character transferred to host

X = Don't care
YY = Error Code

8	Not Used (Zero)

TABLE 3-2. CRU BITS, INPUT TO HOST (Sheet 2 of 3)

Bit	Description
9	End of Block or Error Bit As shown below, RRQ has to be active to interpret this bit. When this bit is a one and bits 0-7 contain a byte of all zeroes then the previous character received was the last character. If this bit is a one and bits 0-7 contain a value other than 00_{16}, then bits 0-7 contain an error code. Error codes are interpreted in Table 3-3. If this bit is a zero when RRQ is a one, then bits 0 to 7 contain data.

(Output Bit)	(Input Bits)			
Module Address Select Bit 8	End of Block Bit 9	RRQ Bit 12	Bits 0-7	Meaning of Bits
0	0	1		Data byte to host
0	1	1	YY	Error code
0	1	1	00	Previous byte was last character of transmission to host

YY = Error Code

10	Not used (Zero).
11	Write Request (WRQ) When set to one by the ICM: (a) The ICM indicates that it has read the previous byte from the host microcomputer to write a byte to it over the CRU. (b) An interrupt is issued to the host. Naturally, the host must provide for reading and answering the interrupt request from the ICM. Interrupts to the host can be masked by CRU output bit 14. When reset from a one to a zero by the ICM: The ICM has seen the host's acknowledgement of the write request. After WRQ is issued by the ICM, the host answers the write request (places a data byte on CRU) then states this by setting its DAV bit low; then the ICM resets WRQ low to indicate it has seen DAV go low. The handshaking during a host-to-ICM data transmission is explained in paragraph 3.7.1.

TABLE 3-2. CRU BITS, INPUT TO HOST (Sheet 3 of 3)

Bit	Description
12	Read Request (RRQ) When set to a one by the ICM: (a) An interrupt is issued to the host. Naturally, the host must provide for reading and answering the interrupt request from the ICM. Interrupts can be masked by CRU output bit 14. (b) The ICM indicates that it has a byte ready for the host microcomputer to read at the CRU. When reset from a one to a zero by the ICM: The ICM has seen the host's acknowledgement of the read request. After RRQ is issued (set), the host reads the byte and sets its DAV bit high which says that it has answered the read request; then the ICM sets RRQ low to indicate it has seen DAV go high. The handshaking during a host-to-ICM data transmission is explained in paragraph 3.7.1.
13	Not used (Zero).
14	ICM Interrupt Mask Status The logic levels of this bit are interpreted as follows: 1 Interrupt by ICM to host can be generated. Specifically, this bit states that CRU output bit 14 is set; this can be read as a test of the ICM and bit 14. 0 Interrupt by ICM to host will not be generated.
15	CRU Interrupt Active from ICM This bit indicates that an interrupt is presently being generated by the ICM to the host; this condition is activated by the ICM issuing a RRQ or WRQ (read or write request). This bit is interpreted as follows: 0 Interrupt from ICM is active 1 Interrupt from ICM is not active

TABLE 3-3. ICM ERROR CODES (Sheet 1 of 2)

Code (Hex)	Description
	NOTE If an error occured during data reception, the ICM sets the End of Block bit and the RRQ bit and places the error code on CRU input bits 0 to 7. A 00_{16} "error" code indicates successful data transfer.
00	No errors.
01	Receiver Overrun Error. A character was lost during the receipt of a data block.
02	CRC Error. A cyclic redundancy check (CRC) resulted in an error.
03	Line to ICM Transfer Timeout Error. In a data transfer from the industrial line to the ICM, 20 seconds have elapsed since transfer of the last byte.
04	ICM to Line Transfer Timeout Error. In a data transfer from the ICM to the industrial line, 20 seconds have elapsed since transfer of the last byte.
05	Not used.
06	Host to ICM Transfer Timeout Error. In a data transfer from the host to the ICM, 20 seconds have elapsed since transfer of the last byte.
07	Buffer Full (Input from Line). A message block greater than the ICM buffer has been received from the industrial line.
08	Not Used.
09	Buffer Full (Input from Host). A message block greater than the ICM buffer has been transferred from the host microcomputer.
0A to 0F	Not used.

TABLE 3-3. ICM ERROR CODES (Sheet 2 of 2)

Code (Hex)	Description
	NOTE The following errors (10 to 18) occur during the ICM self-test requested by software or at powerup.
10	ROM Checksum Error. The ROM test failed
11	RAM1 Error. Error in RAM in socket U42.
12	RAM2 Error. Error in RAM in socket U40.
13	RAM3 Error. Error in RAM in socket U41.
14	RAM4 Error. Error in RAM in socket U39.
	NOTE Loopback is where the ICMs TMS 9903 asynchronous controller is placed in a test mode with outputs tied to inputs; then a read-after-write operation is executed.
15	Complete Frame Not Received During Loopback. The data read indicated that bits were lost during loopback.
16	Data Error During Loopback. The data read indicated that a data error occurred during loopback.
17	CRC Error During Loopback. A cyclic redundancy check error occurred during loopback.
18	Instruction Error. Indicates failure of a 9980 instruction.
19	Memory Error. Bad RAM or no RAM in U40 or U42 for the self-test workspace. No further operation is possible.

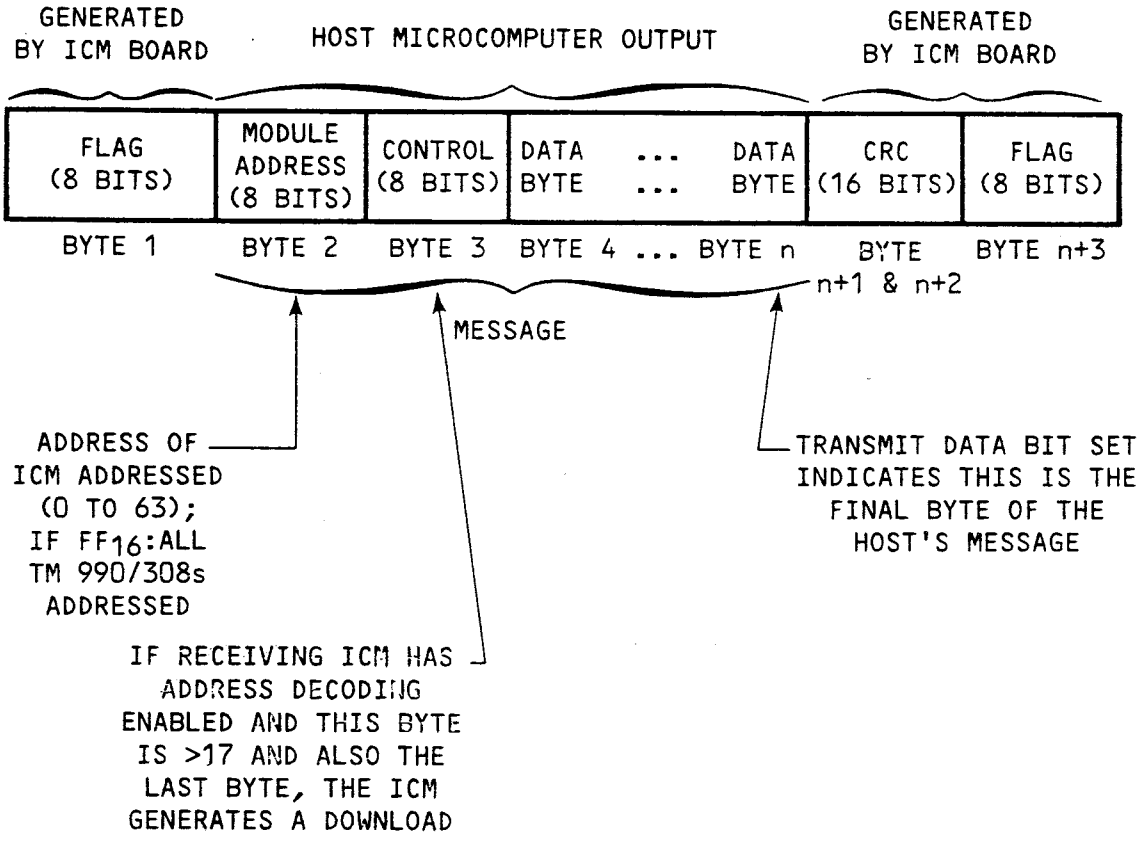


FIGURE 3-6. DETAIL OF MESSAGE FORMAT

3.5 MESSAGE FORMAT

Message format from the host to its ICM (and to the subsequent ICM addressed) is as shown in Figure 3-6. The transmitting ICM will generate the flag byte (byte 1) at the beginning of the message and the CRC word and flag byte at the end of the message. The host will receive from and pass to the ICM messages in the following sequence:

- First Message Byte (ICM addressed):
  - 00_{16} to $3F_{16}$: send message only to ICM with the same address as set in the ICM's MODULE ADDR switch (S2) explained in 2.3.3.
  - FF_{16}: send message to all ICMs on industrial line (broadcast mode).
- Second Message Byte (a control code):
  - 17_{16}: the destination ICM will generate a download (paragraph 3.7.3). If this is the last byte, and address decoding is selected at the ICM whose address is in the first byte.
  - Not 17_{16}: Any other code means this is simply a data message.
- Third and Following Message Bytes:
  These are the data bytes. The final byte is indicated to the ICM by the host setting output bit 10 (Transmit Message/Output Enable bit) high at transmission (DAV then set high).

3.6 BOARD SETUP FOR SOFTWARE EXAMPLES

3.6.1 Switches and Jumpers

Several switch and jumper settings at the ICM must be considered before host software can be written. These include:

- Setting of the CRU's software base address at switch S1. This is covered in detail in paragraph 2.3.4. For examples in this section, a software base address of 0400_{16} will be set as shown in the top of Figure 3-7 for the primary ICM used in this section.

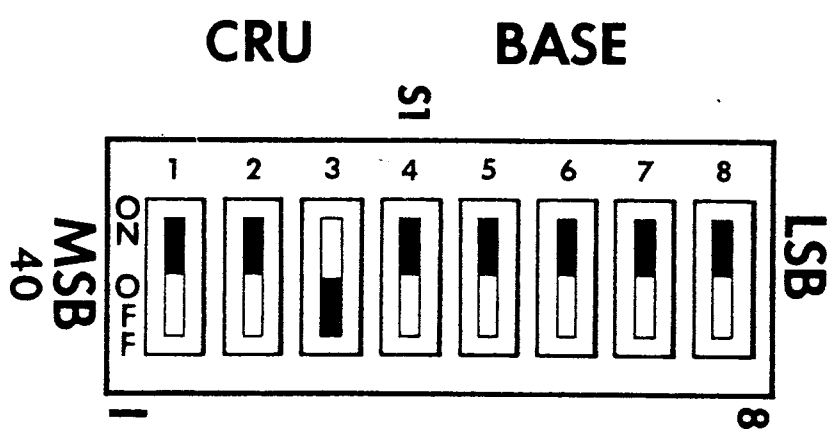
- Setting of the level of the interrupt from the ICM to the host. This is covered in detail in paragraph 2.3.5. For examples in this section, interrupt level 4 will be set at jumper J1 as shown in Figure 2-1. TM 990 host software must also enable interrupts at the host TMS 9901, at the microprocessor status register interrupt mask, and at the CRU interface (set using output bit 14, explained in Table 3-3). The TM 990/100M comes with TM 990/401-1 TIBUG software; this software has vectors for only interrupts 3 and 4. The TM 990/101M TIBUG (401-3) contains an interrupt vector scheme that allows using all interrupts.

- If the system contains more than two ICMs that are addressed individually, a table of each ICM's module address possibly may be part of the program. Also, all ICMs to be addressed must have their module address switch set with the proper address value and with address decoding selected at switch S2. The bottom of Figure 3-7 shows switch S2 set for address decoding and a module address of one.

- A download command can be sent to an ICM over the industrial line which will cause the ICM's host CPU to either do a load (branch to vectors at upper memory) or a reset (branch to vectors at memory address 0000_{16}) depending upon the ICM's jumper J2. If J2 is jumpered, host software must inlcude a corresponding service routine. If J2 is jumpered E17 to E17A, a download command will not be sent. The download command is further described in subsection 3.7.3.

3.6.2 Available Buffer Memory

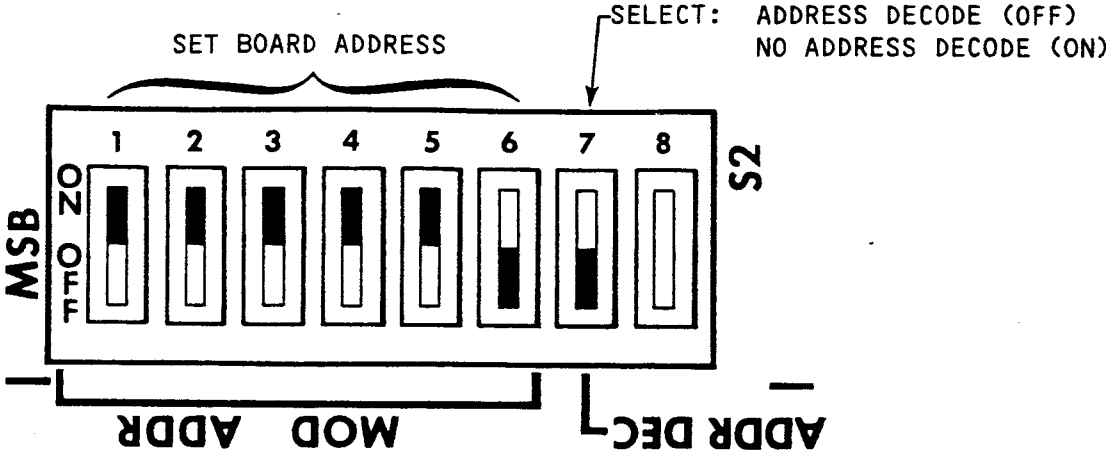
The TM 990/308 ICM's RAM is used for firmware workspace and message buffering. The board is shipped with 1 K bytes of RAM, of which 928 bytes are available for buffer space. The RAM can be expanded to 1952 bytes by inserting two additional TMS 4045s in U39 and U41. The user must be aware of his available buffer space to avoid buffer-full errors (error codes 07 and 09). No firmware change is necessary when adding RAM; existing firmware does a RAM search after an IORST at reset or powerup.



NOTES:

1. ON = zero (ground); OFF = one
2. Software base address 0400_{16} selected (Hardware base address 0200_{16})
3. Switch setting is dark position

(a) CRU Software Base Address 0400_{16} Selected at Switch S1



NOTES:

1. Switch 8 (S2-8) is reserved
2. ON = zero (ground); OFF = one
3. Board address one selected
4. Address decode selected
5. Switch setting is dark position

(b) Board Address One Selected at Switch S2

FIGURE 3-7. EXAMPLE SWITCH SETTINGS USED IN THIS SECTION

3.7 PROGRAMMING EXAMPLES

3.7.1 Write Bytes to ICM From Host

Figure 3-8 is on two sheets. The first sheet shows the handshaking for the host to transfer a message of data to its ICM. This example uses polling between the host and ICM -- the most efficient method. With polling, transmission will be quicker and less "overhead" will be required in the interrupt service routine.

Format of the message to the ICM (and subsequent ICM addressed) will be as shown in Figure 3-6. The first byte is the address of the destination (receiving) ICM as set at its Module Decode switch (shown in Figure 3-7 (c) . The sending ICM will generate the flag byte at the beginning of the message and the CRC word and flag byte at the end of the message.

In the handshaking routine shown in sheet 1 of Figure 3-8, it is presumed that register 12 already has been loaded with the ICM's software base address. The second sheet shows example source code to effect a transfer using polling by the host. The host first uses DAV to interrupt the ICM, then the handshaking is maintained by polling over the CRU interface. When the last byte to be sent is loaded on the CRU, the host can tell its ICM to transmit the entire message by setting its Transmit Message (Output Enable) bit high before setting DAV high. Since the host polls the CRU interface rather than receives interrrupts from the ICM in this routine, the host first masks out interrupts from the ICM by resetting CRU output bit 14 to a zero.

NOTE

During a write or read operation between the ICM and the host, it is possible for the ICM to receive a higher priority interrupt (such as character received on the industrial line). This would force the ICM to make a context switch to answer the interrupt. In such a case, the host's program should contain a timeout routine that will interrupt the host should the host wait too long for an indicator change from the ICM (the ICM would fail to set or reset the bit value of RRQ or WRQ which are polled for a change by the host during handshaking in reads and writes). A summary of using a timeout routine is covered in paragraph 3.3.2.

3.7.2 Read Bytes from ICM to Host

Figure 3-9 is on two sheets. The first sheet shows the handshaking for the ICM to interrupt the host and give it a read request (read bytes from ICM). After being interrupted, the host polls RRQ in the handshaking sequence. When the EOB (End of Block) bit is set, the host checks the byte and links to an error checking routine that determines whether the byte indicates an error occurred (byte not all zeroes) or if that was the final byte in the transmission (byte all zeroes).

Note that the first two bytes in the buffer, sent to the host, are the ICM address (00_{16} to $3F_{16}$) followed by the control character (00_{16} for data message.

3.7.3 Issuing a Download

Should the status of a particular ICM and its host be unknown, the download request can bring them into a known state. The download request causes the the selected ICM to command its host microcomputer to do either:

- a reset (do a BLWP to vectors at memory addresses 0000_{16} and 0002_{16})
- a load (do a BLWP to vectors in the top two memory addresses (for a TM 990/100M and TM 990/101M this would be memory addresses $FFFC_{16}$ and $FFFE_{16}$, -- unless the TM 990/101M is jumpered otherwise).

The selection of which of the above (or neither) occurs depends upon the setting of jumper J2:

- E18 to E19: a microcomputer board reset is executed
- E18 to E17: a microcomputer board load is executed
- E17 to E17A: download will not occur.

The code to write a message as shown in Figure 3-8, sheet 2, can be used to do a download by substituting a download message in the message block. A download message contains only two bytes:

- First byte: address of ICM to be downloaded
- Second byte: a control field of 17_{16} (any other value in the control byte indicates a data message)

The second byte (above) is the last byte to be sent; thus, this byte is loaded on the CRU, the Transmit Message (Output Enable) bit is set to one, and then the DAV (Data Available) bit is set to start the byte transfer to the ICM. Completion of the handshaking is the same as shown for a data message in Figure 3-8. To have the code in Figure 3-8, sheet 2, represent a download request, change only the message block as shown in the following example:

```
            .
            .
            .

* MESSAGE AREA
MESSGE   BYTE   02              MESSAGE ADDRESS IS ICM 02
         BYTE   >17              DOWNLOAD ICM-02'S HOST
ENDMSG   EQU    $               END OF MESSAGE LOCATION VALUE
            .
            .
            .
```

TM 990 HOST CPU	TM 990/308 ICM
1. Host resets CRU bit 14 (ICM interrupt enabled) to mask out interrupt from ICM. (If interrupts from ICM are desired, set this bit to a one -- each RRQ or WRQ at the ICM will interrupt the host.)	
2. Loads byte (LDCR) on bits 0-7 of CRU.	
3. Sets DAV (Data Available) bit on CRU (bit 11, causes interrupt at ICM)	
4. Begins testing for WRQ going high from ICM (TB 11; loop if a zero).	
	5. ICM checks bits 9 to 13; it finds only bit 11 is set indicating bits 0-7 are a data byte.
	6. Stores byte in ICM buffer.
	7. Sets WRQ bit high indicating byte received.
	8. Begins testing for DAV going low.
9. Sees WRQ go high; resets DAV low (SBZ 11).	
10. Tests for WRQ reset low (TB 11; loop if a one).	
	11. Sees DAV go low; resets WRQ low.
12. Sees WRQ reset to a zero. Loads next byte on bits 0-7 of CRU. a. If this is <u>not</u> the last byte, the cycle repeats the same as starting with step 3 above. b. If this is the last byte, the host sets the Transmit Message (Output Enable) bit, then the last cycle repeats the same as starting with step 3 above.	

NOTE: Example source code to effect a data transfer is shown on the next page.

FIGURE 3-8. HANDSHAKING FOR HOST TO WRITE BYTES TO ICM (SHEET 1 OF 2)

```
* EQUATES FOR CRU VALUES
CRUADR   EQU   >1400            CRU SOFTWARE BASE ADDRESS
DAV      EQU   11               CRU OUTPUT BIT FOR DATA AVAILABLE
WRQ      EQU   11               CRU INPUT BIT FOR WRITE REQUEST
TRNSMT   EQU   10               CRU TRANSMIT MESSAGE BIT
INTMSK   EQU   14               CRU INTERRUPT MASK BIT (TO HOST)
         .
         .
         .

* MESSAGE AREA
MESSGE   BYTE  02               MESSAGE ADDRESS IS ICM 02
         BYTE  00               CONTROL BYTE (NO DOWNLOAD IF NOT >17)
         TEXT  'MESSAGE CONTENTS'
ENDMSG   EQU   $                END OF MESSAGE LOCATION VALUE
         .
         .
         .

* ROUTINE TO WRITE MESSAGE TO THE ICM
** SET REGISTERS (CRU, MESSAGE, BYTE COUNT)
         LI    R12,CRUADR       LOAD SOFTWARE BASE ADDRESS IN R12
         LI    R1,MESSGE        LOAD ADDRESS OF MESSAGE IN R1
         LI    R2,ENDMSG-MESSGE LOAD MESSAGE BYTE COUNT IN R2
** MASK OUT INTERRUPTS FROM ICM, USE POLLING OF WRQ
         SBZ   INTMSK           WRQ/RRQ WILL NOT CAUSE INTERRUPT
** CHECK FOR LAST BYTE, SEND BYTES UNTIL LAST BYTE TO BE SENT
SEND     DEC   R2               IS COUNTER = 0? (LAST BYTE AT CRU)
         JEQ   LAST             YES, IF COUNTER = 0, LAST BYTE AT CRU
         LDCR  *R1+,8           LOAD BYTE ON CRU BITS 0-7
         SBO   DAV              SET DAV TO ONE, INTERRUPT ICM
LOOP2    TB    WRQ              HAS WRQ GONE TO ONE?
         JNE   LOOP2            NO, LOOP UNTIL WRQ IS A ONE
         SBZ   DAV              YES, RESET DAV TO A ZERO
LOOP3    TB    WRQ              IS WRQ RESET TO ZERO?
         JEQ   LOOP3            NO, LOOP UNTIL A ZERO
         JMP   SEND             YES, GET NEXT BYTE
**** LAST BYTE REMAINS TO BE SENT, SET TRANSMIT BIT AND SEND LAST BYTE
LAST     LDCR  *R1,8            LOAD LAST BYTE ON TO BITS 0-7
         SBO   TRNSMT           TRNSMT A ONE = LAST BYTE, SEND MESSAGE
         SBO   DAV              SET DAV TO ONE TO TELL ICM TO GET CHAR
**** FINAL HANDSHAKING WITH ICM AFTER LAST BYTE SENT
LOOP4    TB    WRQ              HAS WRQ GONE TO ONE?
         JNE   LOOP4            NO, LOOP UNTIL WRQ IS A ONE
         SBZ   TRNSMT           YES, RESET TRANSMIT MESSAGE BIT
         SBZ   DAV              AND RESET DAV
LOOP5    TB    WRQ              IS WRQ RESET TO ZERO?
         JEQ   LOOP5            NO, LOOP UNTIL WRQ IS RESET
         RTWP                   YES, EXIT
         .
         .
         .
```

FIGURE 3-8. HANDSHAKING FOR HOST TO WRITE BYTES TO ICM (SHEET 2 OF 2)

TM 990 HOST CPU	TM 990/308 ICM
	1. Loads byte on bits 0-7 of CRU from ICM's buffer. 2. Sets to one the RRQ (read request) bit (bit 12, sends interrupt to host). 3. Waits for DAV bit from host to be set high.
4. Tests for RRQ being high on CRU. 5. Checks end-of-block (EOB) bit (bit 9) to determine meaning of data bits 0-7. 6. Stores byte in memory (if EOB bit was high, this byte is an error/end-of-block code). 7. Sets DAV bit high. 8. Begins testing for RRQ bit going low.	
	9. Sees DAV go high. 10. Resets RRQ low. 11. Begins testing for DAV going low.
12. Sees RRQ go low. 13. Resets DAV low. a. If the EOB bit was a one, host exits from data receipt. b. If the EOB bit was a zero, host returns to step 4 above.	
	14. Sees DAV go low. 15. a. If the EOB bit was high on this transfer, the ICM goes into the receive mode. b. If EOB bit was low, it loads next byte on bits 0-7 Then the cycle repeats the same as starting with step 2 above.

NOTE: Example code to effect a data transfer is shown on the next page.

FIGURE 3-9. HANDSHAKING FOR HOST TO RECEIVE BYTES FROM THE ICM (SHEET 1 OF 2)

```
* EQUATES FOR CRU VALUES
CRUADR   EQU   >400            CRU SOFTWARE BASE ADDRESS OF ICM
DAV      EQU   11              CRU OUTPUT BIT FOR DATA AVAILABLE
RRQ      EQU   12              CRU READ REQUEST BIT
EOB      EQU   9               CRU END-OF-BLOCK BIT
INTMSK   EQU   14              CRU INTERRUPT MASK BIT (TO HOST)
STORE    EQU   >FC00           BEGINNING OF MESSAGE STORAGE AREA
         .
         .
         .

* ENABLE INTERRUPTS AT TMS 9901, TMS 9900, AND CRU
         LI    R12,TM9901      LOAD TMS 9901 CRU ADDRESS IN R12
         SBZ   0               ENTER THE INTERRUPT MODE
         SBO   4               ENABLE INTERRUPT 4
         LI    R12,CRUADR      CRU SOFTWARE BASE ADDRESS OF ICM
         SBO   INTMSK          ENABLE CRU INTERRUPTS TO HOST
         .
         .
         .

* A READ (RRQ) INTERRUPT HAS BEEN RECEIVED AT HOST FROM ICM
** INTERRUPT SERVICE ROUTINE TO RECEIVE A MESSAGE FROM THE ICM
**** SET REGISTERS
         LI    R12,CRUADR      LOAD SOFTWARE BASE ADDRESS IN R12
         LI    R1,STORE        LOAD STORAGE ADDRESS OF MESSAGE
         CLR   R2              LOAD COUNTER WITH ALL ZEROES
**** DISABLE INTERRUPTS TO HOST OVER CRU
         SBZ   INTMSK          WRQ/RRQ WILL NOT CAUSE INTERRUPT
**** CHECK TO SEE IF BYTE IS LAST BYTE (EOB BIT SET)
READ     TB    EOB             IS THIS THE FINAL (OR ERROR) BYTE?
         JEQ   LAST            YES, GO TO END/ERROR ROUTINE
**** EOB BIT NOT SET            NO, GO STORE BYTE
         STCR  *R1+,8          STORE BYTE IN STORAGE BUFFER
         SBO   DAV             ACKNOWLEDGE RRQ
LOOP1    TB    RRQ             HAS RRQ BEEN RESET TO ZERO?
         JEQ   LOOP1           NO, WAIT FOR RRQ TO BE RESET
         SBZ   DAV             ACKNOWLEDGE RRQ BEING RESET
LOOP2    TB    RRQ             HAS RRQ BEEN SET TO ONE? (NEW BYTE)
         JNE   LOOP2           NO, WAIT FOR RRQ TO BE SET
         JMP   READ            YES, GET NEXT BYTE
**** LAST BYTE TO BE RECEIVED
LAST     CLR   R3              CLEAR REGISTER TO RECEIVE BYTE
         STCR  R3,8            STORE BYTE IN R3,BYTE=0=LASTBYTE
         JEQ   EXIT            YES, EXIT, THAT WAS LAST BYTE
         BL    @ERROR          NO, GO TO ERROR ROUTINE
EXIT     SBO   DAV             ACKNOWLEDGE RRQ
LOOP3    TB    RRQ             HAS RRQ BEEN RESET TO ZERO?
         JEQ   LOOP3           NO, WAIT FOR RRQ TO BE RESET
         SBZ   DAV             ACKNOWLEDGE RRQ BEING RESET
         RTWP                  RETURN FROM INTERRUPT SERVICE
         .                     ROUTINE
         .
         .
```

FIGURE 3-9. HANDSHAKING FOR HOST TO RECEIVE BYTES FROM ICM
(SHEET 2 OF 2)

3.7.4 Request ICM Module Address

Each ICM has a module address specified by the setting of switch S2 (explained in 2.3.3). The example code in Figure 3-10 shows how the host can find out the settings of this DIP switch via the CRU. This allows the host to test for the ICM address or a change in the ICM address.

```
* EQUATES FOR CRU VALUES
CRUADR   EQU    >400              CRU SOFTWARE BASE ADDRESS
ICMADR   EQU    8                 CRU ICM ADDRESS BIT
         .
         .
         .

* ROUTINE TO REQUEST ICM ADDRESS VALUE FROM ICM
         LI     R12,CRUADR        LOAD SOFTWARE BASE ADDRESS IN R12
         CLR    R1                CLEAR REGISTER TO STORE ADDRESS
         SBO    ICMADR            SET ICM ADDRESS BIT TO ONE
         STCR   R1,8              YES, ICM ADDRESS TO R1 (LEFT BYTE)
         SBZ    ICMADR            RESET BIT
         .
         .
         .
```

FIGURE 3-10. EXAMPLE CODE TO OBTAIN ICM ADDRESS NUMBER

SECTION 4

THEORY OF OPERATION

4.1 GENERAL

This section covers the theory of operation of the TM 990/308. Information in the following manuals can be used to supplement material in this section:

- TMS 9980A/TMS 9981 Microprocessor Data Manual
- TMS 9903 Synchronous Communication Controller
- The TTL Data Book For Design Engineers
- The Line Driver and Line Receiver Data Book For Design Engineers.

4.2 TM 990/308 ARCHITECTURE

The TM 990/308 ICM uses a TMS 9903 to perform the serial synchronous transmit and receive functions and a TMS 9980A to control the TMS 9903 and to provide the interface between the ICM and a TM 990 computer. The TMS 9980A performs the address decoding, the download command decoding, and the data buffering functions provided by the ICM.

Figure 4-1 shows the data flow within the TM 990/308, highlighting the major sections and the two sets of main buses: internal buses (address, control, data, and communications register unit (CRU)) and the interbuses (CRU interbus, address interbus, and interrrupt interbus).

4.2.1 TMS 9980A Microprocessor

The TMS 9980A is a single-chip 16-bit central processing unit (CPU) which has an 8-bit data bus, uses a 16-bit instruction word, and is packaged in a 40-pin package. The instruction set of the TMS 9980A is exactly the same as the 9900's. The memory-to-memory architecture features multiple register files, resident in memory, which allow faster response to interrupts and increased programming flexibility.

The CPU has the capability to make decisions and take different courses of action based on those decisions. In addition to decision making, the CPU is responsible for:

- Instruction acquisition and interpretation
- Timing of most control signals and data transfers
- Data, address, and CRU bus control.

The TMS 9980A internal structure is shown in Figure 4-2. Figure 4-3 shows the TMS 9980A pin assignments and Table 4-1 describes the function of each pin.

In the discussions that follow, whenever the "processor" is referred to it is understood to be the TMS 9980A located on board the ICM. The processor located on the TM 990/100 or TM 990/101 CPU modules will be referred to as the host processor.

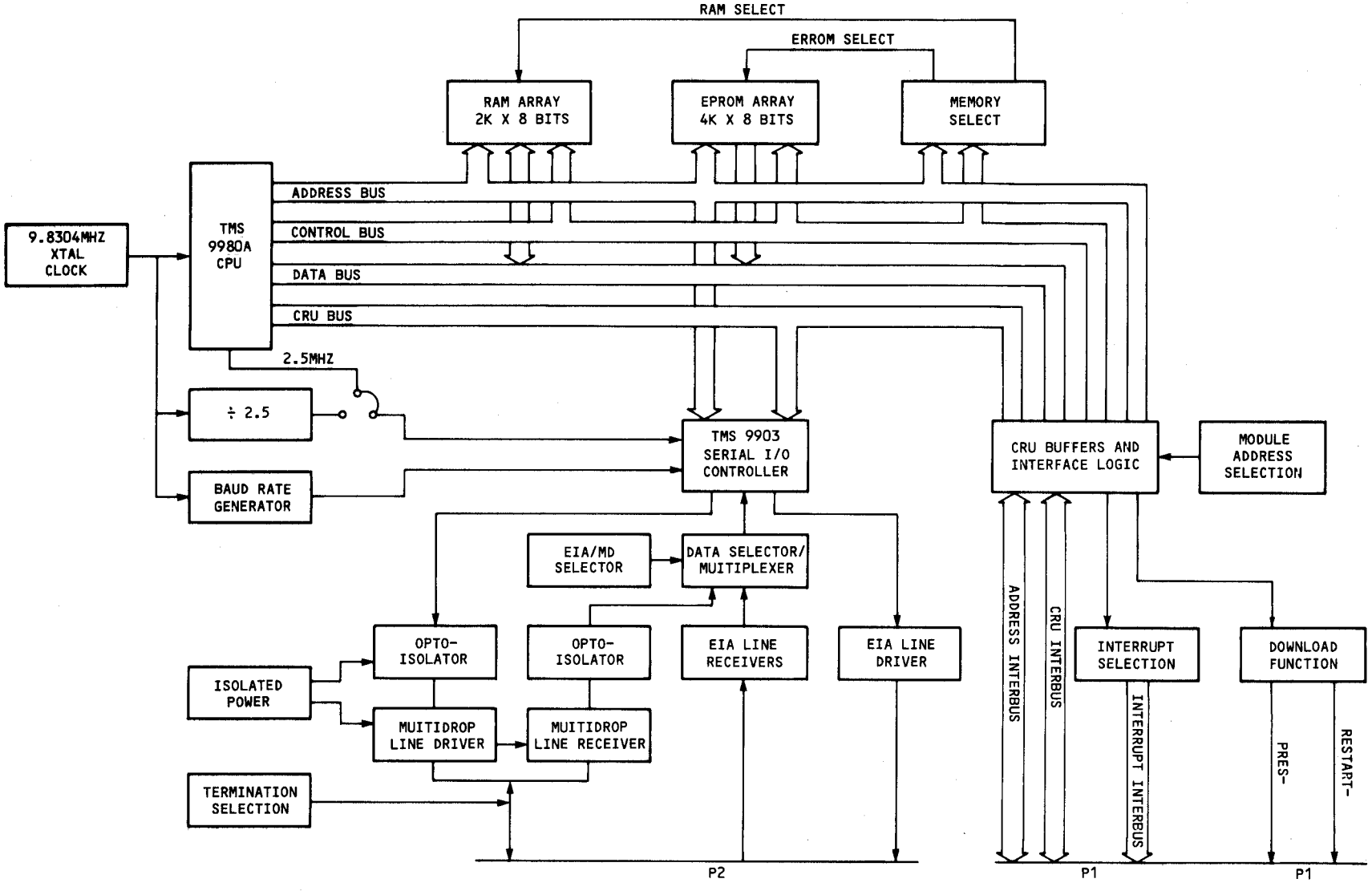


FIGURE 4-1. TM 990/308 BLOCK DIAGRAM

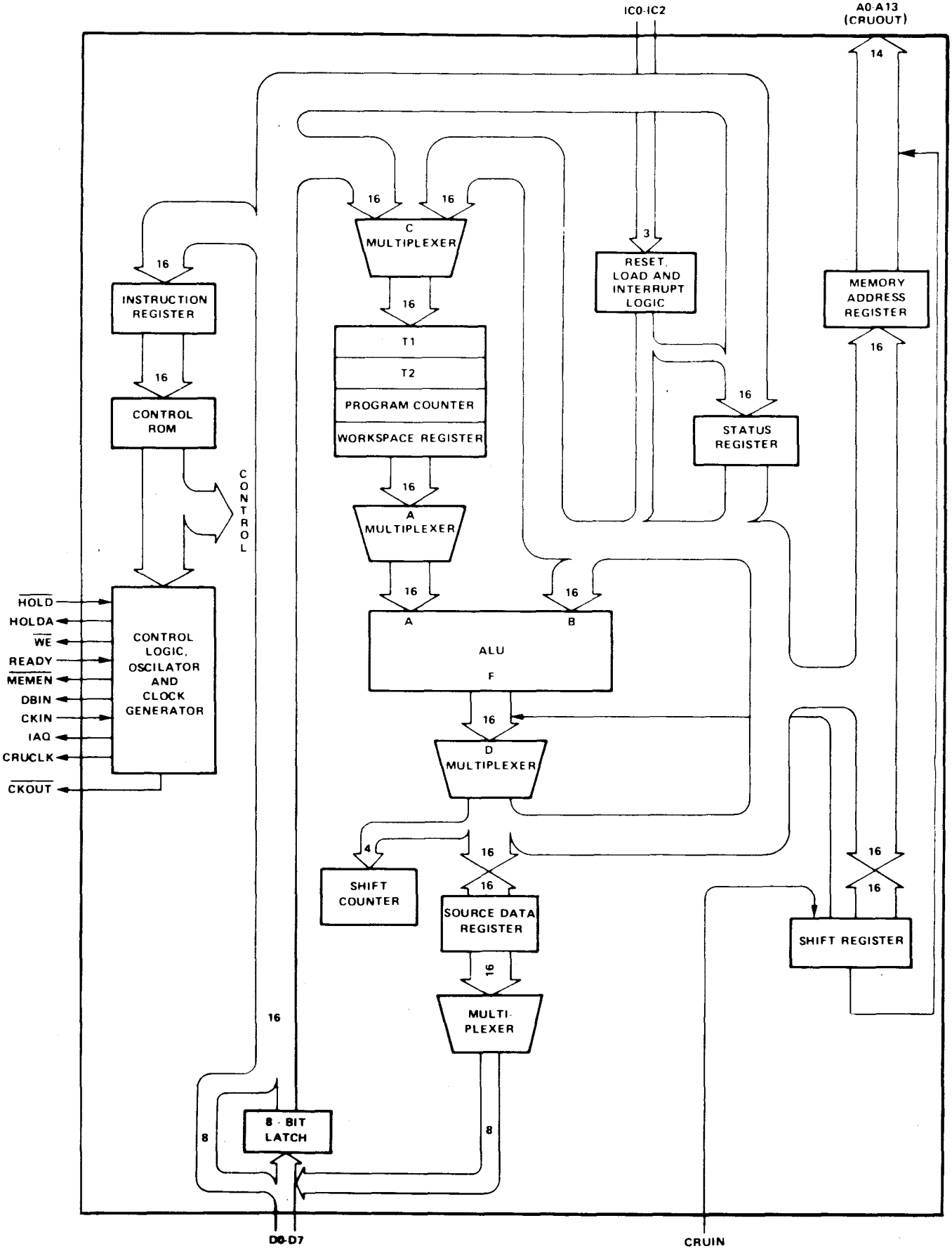


FIGURE 4-2. TMS 9980A ARCHITECTURE

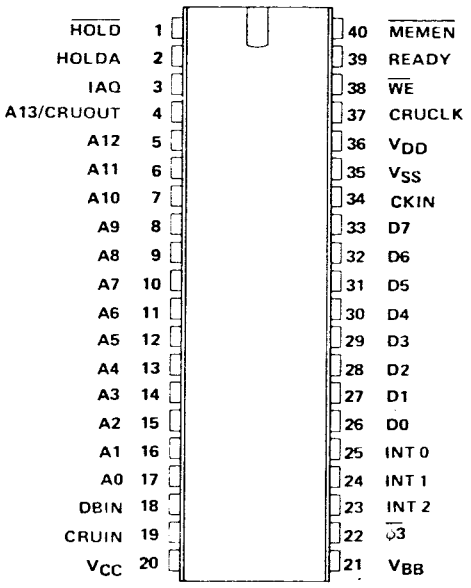


FIGURE 4-3. TMS 9980A PIN ASSIGNMENTS

TABLE 4-1. TMS 9980A PIN FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
			ADDRESS BUS
A0(MSB)	17	OUT	A0 through A13 comprise the address bus. This three-
A1	16	OUT	state bus provides the memory-address vector to the
A2	15	OUT	external-memory system when MEMEN is active and I/O
A3	14	OUT	bit addresses and external-instruction addresses to
A4	13	OUT	the I/O system when MEMEN- is inactive. The address
A5	12	OUT	bus assumes the high-impedance state when HOLDA is
A6	11	OUT	active.
A7	10	OUT	
A8	9	OUT	
A9	8	OUT	
A10	7	OUT	
A11	6	OUT	
A12	5	OUT	
A13/CRUOUT	4	OUT	CRUOUT
			Serial I/O appears on A13 when an LDCR, SBZ, and SBO instruction is executed. This data should be sampled by the I/O interface logic when CRUCLK goes active (high). One bit of the external instruction code appears on A13 during external instruction execution.

TABLE 4-1. TMS 9980A PIN FUNCTIONS (CONTINUED)

SIGNATURE	PIN	I/O	DESCRIPTION
			DATA BUS
D0(MSB)	26	I/O	D0 through D7 comprise the bidirectional 3-state data bus. This bus transfers memory data to (when writing) and from (when reading) the external memory system when MEMEN- is active. The data bus assumes the high-impedance state when HOLDA is active.
D1	27	I/O	
D2	28	I/O	
D3	29	I/O	
D4	30	I/O	
D5	31	I/O	
D6	32	I/O	
D7(LSB)	33	I/O	
			POWER SUPPLIES
V_{BB}	21		Supply voltage (-5 V NOM)
V_{CC}	20		Supply voltage (+5 V NOM)
V_{DD}	36		Supply voltage (12 V NOM)
V_{SS}			
			CLOCKS
CKIN	34	IN	Clock in. A TTL compatible input used to generate the internal 4-phase clock. CKIN frequency is 4 times the desired system frequency.
o3	22	OUT	Clock phase 3 (o3) inverted; used as a timing reference
			BUS CONTROL
DBIN	18	OUT	Data bus in. When active (high), DBIN indicates that the TMS 9980A has disabled its output buffers to allow the memory to place memory-read data on the data bus during MEMEN. DBIN remains low in all other cases except when HOLDA is active at which time it is in the high-impedance state.
MEMEN-	40	OUT	Memory enable. When active (low), MEMEN- indicates that the address bus contains a memory address. When HOLDA is is active, MEMEN- is in the high-impedance state.
WE-	38	OUT	Write enable. When active (low), WE- indicates that memory-write data is available from the TMS 9980A to be written into memory. When HOLDA is is active, WE- is in the high-impedance state.
CRUCLK	37	OUT	CRU clock. When active (high), CRUCLK indicates that external interface logic should sample the output data on CRUOUT or should decode external instructions on A0, A1, A13.
CRUIN	19	IN	CRU data in. CRUIN, normally driven by 3-state open collector devices, receives input data from external interface logic. When the processor executes a STCR or TB instruction, it samples CRUIN for the level of the CRU input bit specified by the address bus (A2 - A12).
INT2	23	IN	Interrupt code.
INT1	24	IN	
INT0	25	IN	

TABLE 4-1. TMS 9980A PIN FUNCTIONS (CONCLUDED)

SIGNATURE	PIN	I/O	DESCRIPTION
			MEMORY CONTROL
HOLD-	1	IN	Hold. When active (low), HOLD- indicates to the processor that an external controller (e.g., DMA device) desires to utilize the address and data buses to transfer data to or from memory. The TMS 9980A enters the hold state following a hold signal when it has completed its present memory cycle.* The processor then places the address and data buses in the high-impedance state (along with WE-, MEMEN-, and DBIN) and responds with a hold-acknowledge signal (HOLDA). When HOLD- is removed the processor returns to normal operation.
HOLDA	2	OUT	Hold acknowledge. When active (high), HOLDA indicates that the processor is in the hold state and the address and data buses and memory control outputs (WE-, MEMEN-, and DBIN) are in the high-impedance state.
READY	39	IN	Ready. When active (high), READY indicates that memory will be ready to read or write during the next clock cycle. When not-ready is indicated during a memory operation, the TMS 9980A enters a wait state and suspends internal operation until the internal memory systems indicated ready.
IAQ	3	OUT	Instruction acquisition. IAQ is active (high) during any memory cycle when the TMS 9980A is acquiring an instruction. IAQ can be used to detect illegal op codes. It may also be used to synchronize LOAD stimulus.

* If the cycle following the present memory cycle is also a memory cycle it, too, is completed before the TMS 9980A enters hold state.

4.2.2 Crystal Clock and Baud Rate Generator

System timing is regulated by a crystal-controlled logic network (See Figure 4-4) which produces a signal of four times the frequency of the desired system frequency. This signal is input to the TMS 9980A (through CKIN), which contains internal logic to shape and divide the clock rate by four to the desired system speed of operation. P22 (o3 or CKOUT) on the TMS 9980A provides an output for this clock to the rest of the system.

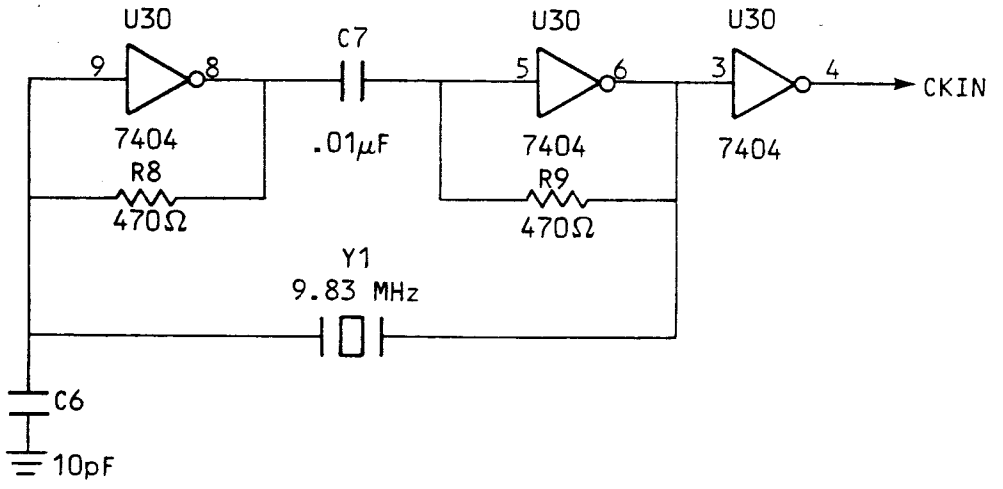


FIGURE 4-4. CRYSTAL CLOCK

The CKIN signal goes to an 8 bit counter (U23) which generates the various baud rates (See sheet 5 of the schematics). A jumper selects which baud rate clock will be connected to the TMS 9903 via selector U34 when the industrial line mode is selected. The crystal frequency of 9.8304 MHZ was chosen to meet the needs of both the baud rate generator and the 9980 microprocessor.

CKIN also feeds a divide by 2.5 circuit consisting of synchronous counters U8 and U9 (Sheet 3 of the schematics). This circuitry produces 3.98 MHZ for a reserved function.

4.2.3 TM 990/308 Buses

The TM 990/308 has two sets of major buses: the four internal buses and the three interbuses. The four internal buses connect the 9980 microprocessor to the memory, memory select, serial I/O controller, and buffers and interface sections. The three interbuses provide the interface to the host TM 990 processor at connector P1. The four internal buses are: 1) address, 2) control, 3) data, and 4) communication register unit (CRU). The three interbuses are: 1) CRU interbus, 2) address interbus, and 3) interrupts interbus. Each of these buses will be described in the paragraphs that follow.

4.2.3.1 Data Bus. The data bus consists of eight bidirectional lines, D0 through D7, used to transfer information between the 9980 processor and memory or memory mapped I/O. D0 is the most significant bit and D7 is the least significant bit.

The direction of data transfer is controlled by the processor and indicated by the state of control bus signal DBIN which is set to a logic one when the processor has disabled its data bus drivers and enters an input mode.

4.2.3.2 Address Bus. The address bus is comprised of fourteen lines, A0 through A13/CRUOUT, which are driven by the 9980 processor and are used to reference individual memory, CRU, and memory mapped I/O locations. Memory references are distinguished from CRU operations by observing the state of control bus signal MEMEN- which the processor sets to a logic zero during memory cycles. During memory references A0 through A13/CRUOUT present the address of the byte being accessed, where A0 is the most significant bit and A13/CRUOUT is the least significant bit of the address. During CRU cycles, A0 and A1 are set to zero and A2 through A12 contain the effective CRU bit address being referenced, where A2 is the most significant bit and A12 is the least significant bit of the address. A13/CRUOUT is shared with the CRU bus during CRU operations.

4.2.3.3 CRU Bus. The CRU bus consists of three signals: A13/CRUOUT, CRUCLK, and CRUIN. During 9980 CRU output instructions (SBZ, SB0, and LDCR), A13/CRUOUT contains the value of the bit being output. After a delay to allow the CRU bit address (A2 through A12) and the output bit (A13/CRUOUT) to stabilize, the 9980 strobes CRUCLK to latch the output bit in the output device. During CRU input instructions (TB and STCR) the 9980 processor again sets up the CRU bit address on A2 through A12 and, after a delay for settling, reads the input bit from CRUIN.

4.2.3.4 Control Bus. The control bus consists of a number of bus, memory, and timing and control signals used by the processor and support circuitry. These signals are listed in Table 4-2. A description of each control signal can be found in Table 4-1.

TABLE 4-2. CONTROL BUS SIGNALS

	BUS CONTROL		MEMORY CONTROL
DBIN	Data bus in	HOLD	Hold
MEMEN-	Memory enable	HOLDA	Hold acknowledge
WE-	Write enable	READY	Ready
INT2			TIMING AND CONTROL
INT1	Interrupt code	IAQ	
INT0		ø3	

4.2.3.5 Address Interbus. The address lines A3.B through A14.B provide for the host TM 990 processor board selection and CRU bit addressing. These address lines must be distinguished from those of section 4.2.3.2.

4.2.3.6 CRU Interbus. The CRUIN.B, CRUOUT.B, and CRUCLK.B- lines provide CRU comunication for the host processor in the same way the CRU bus provides communication for the 9980. These CRU lines must be distinguished from the CRU bus lines described in section 4.2.3.3.

4.2.3.7 Interrupt Interbus. These lines are simply interrupt lines to the host processor. Either a read request or a write request can generate an

interrupt if not masked. This interrupt can be connected via jumper to any external level INT1- through INT15-. The interrupt selection jumpers are shown at the top of Figure 4-12.

4.2.4 System Bus Edge Connector (P1)

Connector P1 is the system bus edge connector. It contains the system power, interrupt, address, and control signals. Table 4-3 lists pins and their functions.

TABLE 4-3. CHASSIS INTERFACE CONNECTOR (P1) SIGNAL ASSIGNMENTS

P1 PIN	SIGNAL	P1 PIN	SIGNAL	P1 PIN	SIGNAL
60	A3.B	16	INT1.B-	1	GND
61	A4.B	13	INT2.B-	2	GND
62	A5.B	15	INT3.B-	21	GND
63	A6.B	18	INT4.B-	23	GND
64	A7.B	17	INT5.B-	25	GND
65	A8.B	20	INT6.B-	27	GND
66	A9.B	6	INT7.B-	31	GND
67	A10.B	5	INT8.B-	77	GND
68	A11.B	8	INT9.B-	79	GND
69	A12.B	7	INT10.B-	81	GND
70	A13.B	10	INT11.B-	83	GND
71	A14.B	9	INT12.B-	85	GND
		12	INT13.B-	89	GND
29	CRUIN.B	11	INT14.B-	91	GND
30	CRUOUT.B	14	INT15.B-	99	GND
80	MEMEN.B-			100	GND
87	CRUCLK.B-	3	+5V		
88	IORST.B-	4	+5V	75	+12V
93	RESTART.B-	97	+5V	76	+12V
94	PRES.B-	98	+5V	73	-12V
				74	-12V

4.2.5 RAM and EPROM Memory

The TM 990/308 memory consists of RAM and EPROM sections as shown in Figure 4-5. Four sockets (U39-U42) are provided for RAM memory. The RAM memory section can accommodate up to four TMS 4045 chips, each consisting of 1024 X 4 bits. These are grouped into two blocks of two TMS 4045's each, providing a total of 2048 eight-bit bytes of RAM if fully populated. The TM 990/308 is supplied with one block (U40, U42) populated that provides nearly 1024 eight-bit bytes of RAM to buffer incoming and outgoing messages. Populating U39 and U41 will give an additional 1024 eight-bit bytes of data buffering. Control lines RAM 1 and RAM 2 determine which block of RAM is selected. A low logic signal on RAM 1 selects U40 and U42 while a low logic level on RAM 2 selects U39 and U41. When the chip select is active (low), all eight data outputs are enabled and the eight-bit addressed word can be read. When the chip select is inactive (high), all eight data outputs are in a high-impedance state.

The EPROM memory section can accommodate up to two TMS 2716 EPROM chips (U50, U51) each consisting of 1024 X 8 bits. These two chips provide 2048 eight-bit bytes of EPROM. The TMS 990/308 is supplied with U51 populated with firmware to implement the features described in this manual. A low logic signal on EPROM 1 selects U51 while a low logic signal on EPROM 2 selects U50. When the chip select is active (low), all eight data outputs are enabled and the eight-bit addressed word can be read. When the chip select is inactive (high), all eight data outputs are in the high-impedance state.

4.2.6 Memory Select

The memory select section (See Figure 4-6) is responsible for selecting and enabling the proper banks of RAM and EPROM.

4.2.6.1 RAM Select. When DBIN- and WE- are both high, the upper section of the LS139 will be inhibited from decoding inputs A0 and A3. The output signals RAM 1 and RAM 2 will remain high and no RAM will be selected. If however, DBIN- is high while WE- is low, the upper section of the LS139 will be enabled. Inputs A0 and A3 will be decoded by the LS139 which provides RAM select signals as required.

4.2.6.2 EPROM Select. When DBIN is high, the lower section of the LS139 will be inhibited from decoding inputs A2 and ENEPROM- . The output signals EPROM 1 and EPROM 2 will remain high and no EPROM memory will be selected. If however, DBIN- is low, the lower section of the LS139 will be enabled. Inputs A2 and ENEPROM- will be decoded by the LS139 which provides select EPROM signals as needed.

4.2.7 TMS 9980 Interrupts

Latch U28 and priority encoder U29 encode the on-board interrupts for the 9980 processor. The possible interrupts listed in the order of priority from the highest to the lowest are RESET- (from the IORST- bus signal), RSTBD- (a software board reset), DTINT- (from the timeout timer), INTPT- (from the 9903), and DAV- (the data available interrupt from the host shown on sheet 3 of the schematics).

4.2.8 Memory Mapped I/O Select

Decoders U12 and U13 decode memory addresses to generate enables for the 9980's memory mapped I/O. ENINBYTE1- selects a memory mapped input; ENOUTBYTE1 selects memory mapped output (Sheet 3 of the schematics).

4.2.9 CRU Base Address Select

Decoder U12 also selects the 9980's CRU. ENINBYTE2- selects the CRU input and ENOUTBYTE2- selects the CRU output (Sheet 3 of the schematics).

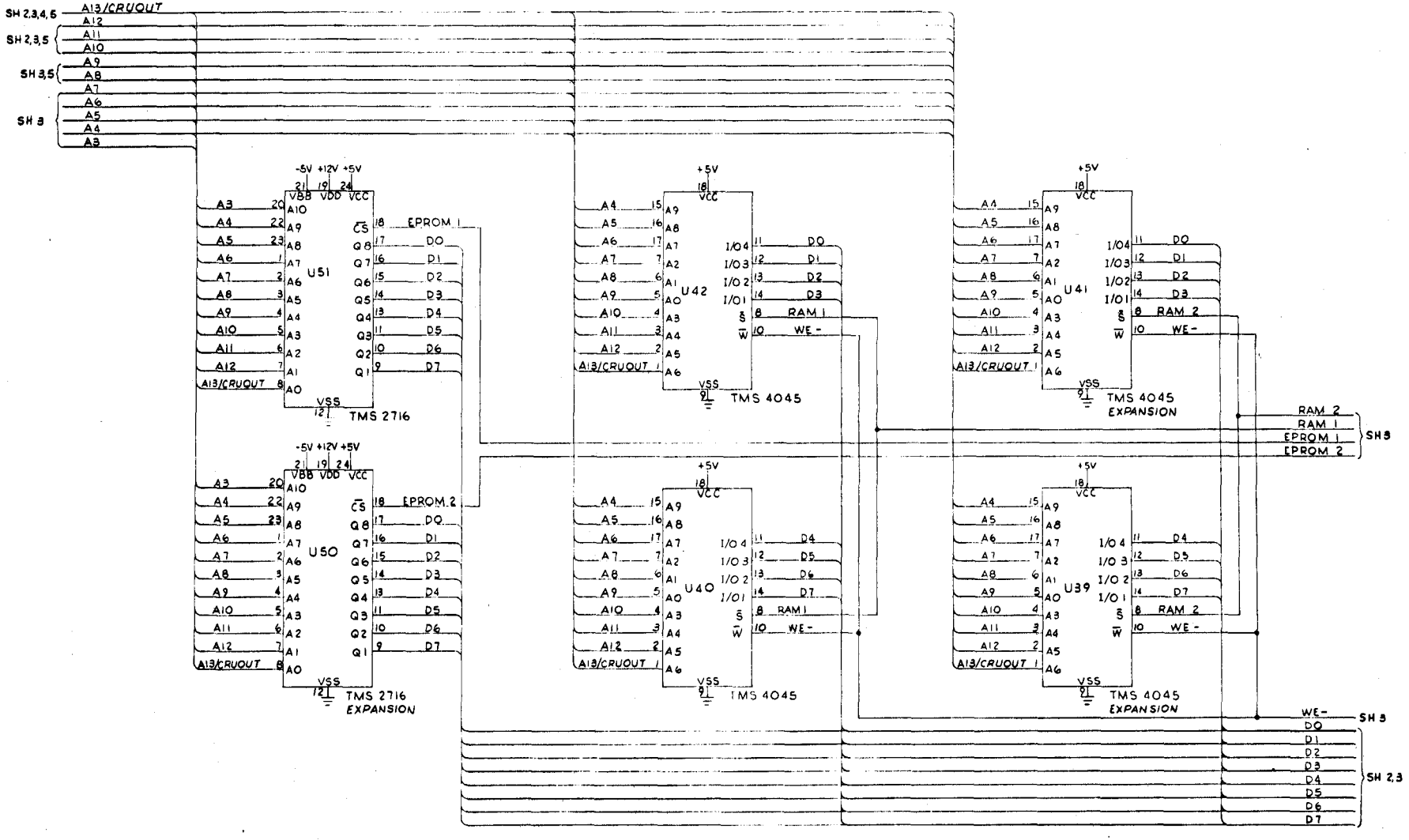


FIGURE 4-5. RAM AND EPROM MEMORY

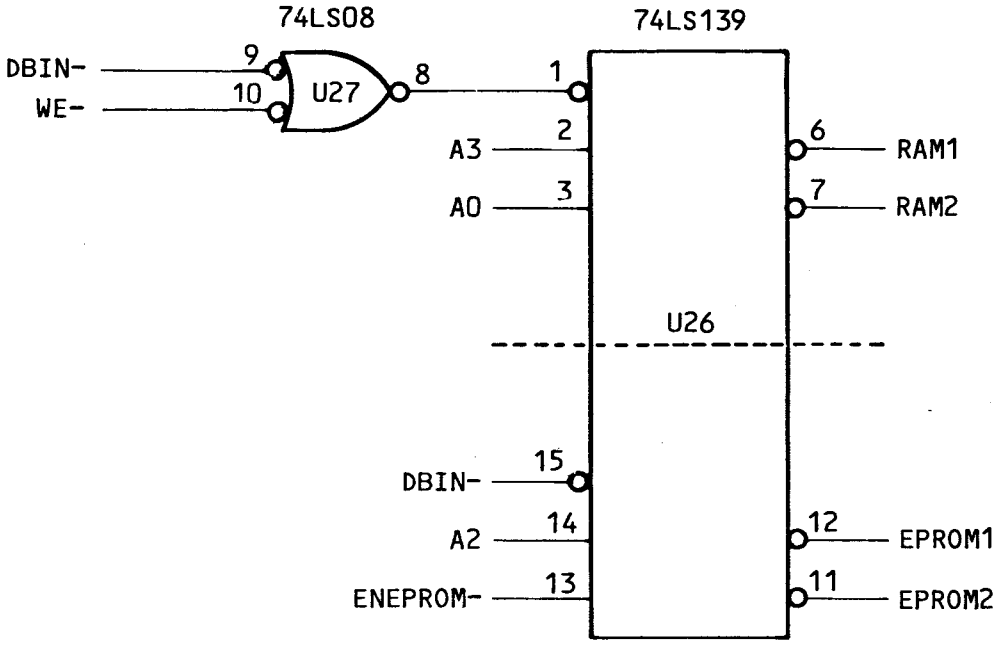


FIGURE 4-6. MEMORY SELECT CIRCUITRY

4.2.10 TMS 9903 Serial I/O Controller

The TMS 9903 synchronous communication controller (SCC) provides an interface between the microprocessor and the serial synchronous channel. It performs data serialization and deserialization while facilitating microprocessor control of the communications channel.

Figure 4-7 shows the interface between the TMS 9980A and the TMS 9903. The TMS 9903 is selected by a low signal from address line A2. Data transfers occur through the CRU bus. The CPU sends serial data via CRUOUT and receives data via CRUIN. CRUCLK is used as a strobe for CRUOUT operations. CRU locations 0 through 31 of the TMS 9980A are used to interface to the TMS 9903. The TMS 9903 operates in mode 1 (SDLC mode). The configuration will be SDLC normal with a character length of eight bits. Data is transferred from the TMS 9903 via XOUT (transmitter serial data output line) to either the EIA or multidrop interface or between the EIA or multidrop interface to RIN (receiver serial data input line). More information on the TMS 9903 can be found in the TMS 9903 Synchronous Communication Controller Data Manual.

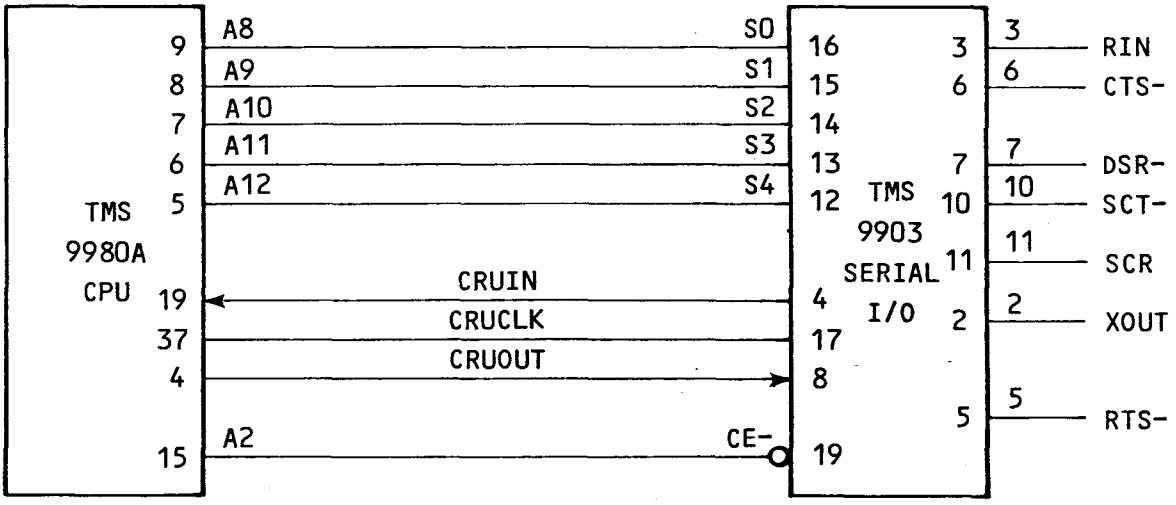


FIGURE 4-7. TMS 9903/TMS 9980A INTERFACE CIRCUITRY

4.2.11 EIA Interface

The EIA interface circuitry is shown in Figure 4-8. Table 4-4 provides a listing of several TMS 9903 signal descriptions pertinent to the following discussion.

TABLE 4-4. TMS 9903 SIGNAL DESCRIPTIONS

SIGNATURE	DESCRIPTION
XOUT	Transmitter serial data output line.
RIN	Receiver serial data input line.
RTS-	Request-to-sent output from TMS 9903 to modem. This output is enabled by the CPU and remains active (low) during data transmission from the TMS 9903.
CTS-	Clear-to-send input from modem to TMS 9903. When active (low), it enables the transmitter section of the TMS 9903.
DSR-	Data set ready input from modem to TMS 9903. This input generates an interrupt when going ON or OFF.
SCT-	Transmit clock - transmitter data is shifted out on one-to-zero transition of SCT-.
SCR	Receiver clock - receiver serial data (RIN) is sampled at zero-to-one transition of SCR.

If the EIA interface is selected (jumper connected from E37 to E38), then data from the modem will be transferred via connector P2 to line receivers U58 and U47. EIA data is then transferred to U33 and U34. U33 and U34 are 2-line to 1-line data selectors/multiplexers which transfer the selected EIA data to the TMS 9903.

For transmit operation, RTS- is enabled by the CPU and remains active (low) during data transmission from the TMS 9903. This signal activates transmit LED DS2 after passing through inverters U31 and U22. At the same time, U14 decodes address lines A10-A12 and provides the TIMETRIG signal. Either the TIMETRIG or RTS- signal can activate timer U7 which provides a 20 second enable signal to NAND gate U17. U17 provides the RTS1- signal input to U56 (uA 9636) while the XOUT signal is buffered and then fed to the other input of U56. The outputs from the dual single-ended line driver (U56) provide EIA signals EIAXOUT and EIARTS to connector P2. The timer (U7) stops relaying the RTS1- signal that provides the EIARTS signal after a 20 second time out. The primary function of the timer is to provide a time out escape mechanism in case something goes wrong during data transmission. The timer will prevent the ICM's transmitter from tying up the line in case of difficulty. It also generates an interrupt to the 9980 processor that will terminate any infinite loop that the processor might be caught in.

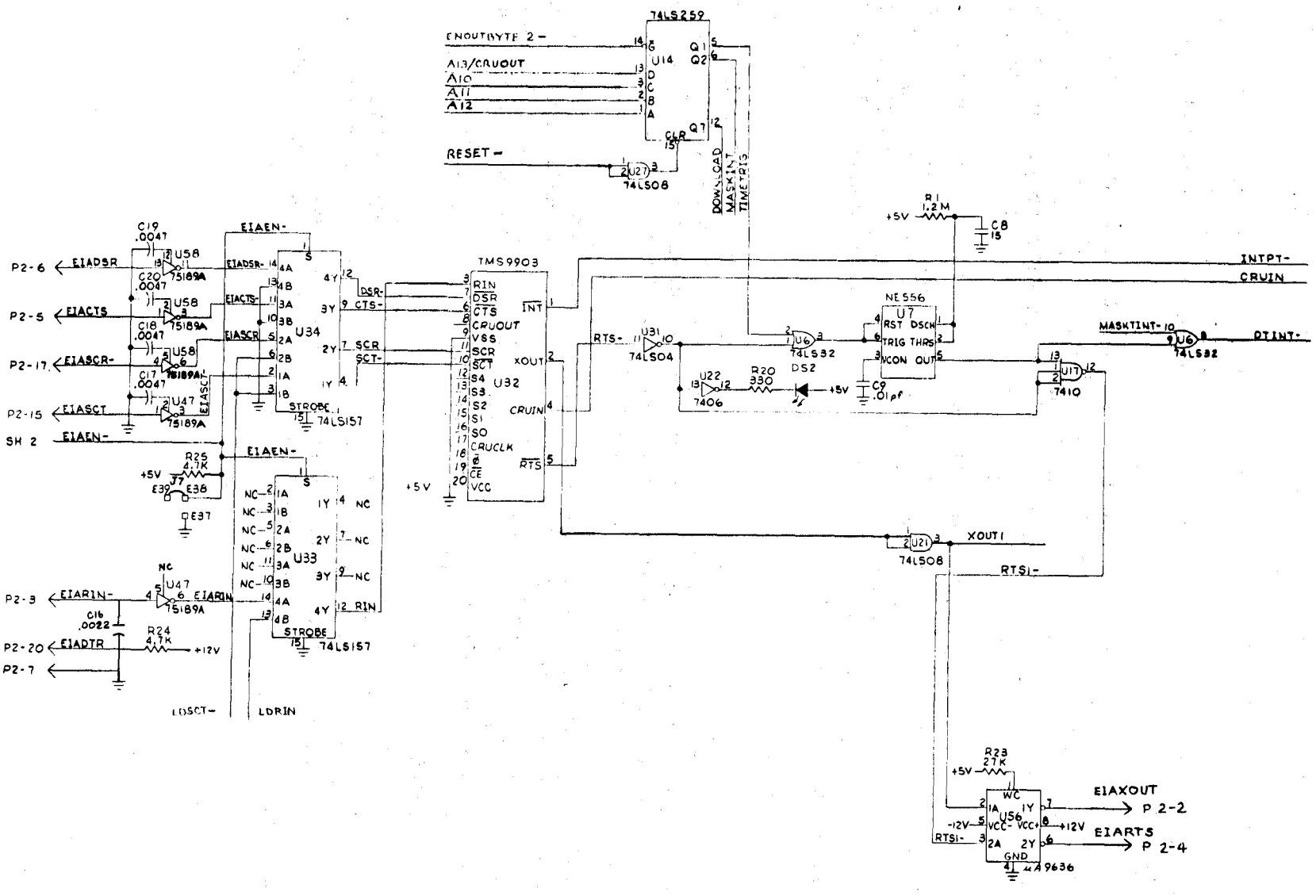


FIGURE 4-8. EIA INTERFACE CIRCUITRY

4.2.12 Multidrop Interface Circuitry

The multidrop interface circuitry is shown in Figure 4-9. Both the multidrop line receiver U48 (75107) and the multidrop line driver U49 (75112) are optically isolated. U35 and U36 provide optical isolation for the driver while U37 provides optical isolation for the receiver. Both the driver and receiver operate on dc voltages from the isolated power supply contained on the ICM. The multidrop receiver continually reads the multidrop line whereas the multidrop line driver is only enabled when a transmit operation is in progress.

Transmit operations are timed-out by timer U7. An active RTS- signal turns on transmit LED DS2. This signal is also used to energize the timer. The XOUT signal is optically coupled to pin 1A of the line driver which is enabled by the signal from the timer that is optically coupled to pin 1C of the line driver. After a 20 second time out the line driver is turned off by U7.

The line receiver constantly reads the multidrop line feeding AND gate U21 via opto-isolator U37. When a transmit operation is in progress, U21 is inhibited by an active RST1- signal but when a receive operation is in progress U21 is enabled by an inactive RTS1- signal. U21 provides signal LDRIN to 2-line to 1-line data selector U33 which couples the multidrop signal RIN to the TMS 9903 serial I/O controller.

4.2.13 Isolated Power

Figure 4-10 shows the circuitry that develops +5 V and -5 V floating for the multidrop line drivers and receivers. The 76.8 KHz square wave is fed in true and inverted form as inputs for U10 (75462) dual peripheral positive-AND drivers. U10 provides push-pull drive for transformer T1 while CR1 and CR2 provide transient voltage protection for U10.

Diodes (CR3 and CR4) are half-wave rectifiers that develop the positive and negative dc voltages needed. On one half cycle, CR4 conducts and charges C11 to positive V_{MAX} and on the other half cycle, CR3 conducts and charges C10 to negative V_{MAX}. The positive voltage from C11 is then regulated by the 78L05 positive voltage regulator thus providing +5 V floating. The negative voltage from C10 is regulated by the 79L05 negative 5 volt regulator thus providing -5 V floating.

This isolated power supply provides the 1500 volt RMS isolation between the ICM-host processor and the industrial line required for immunity to noise and differential ground potentials that are common in an industrial environment. At the same time, it allows the significant drive currents needed for a 10,000 foot line.

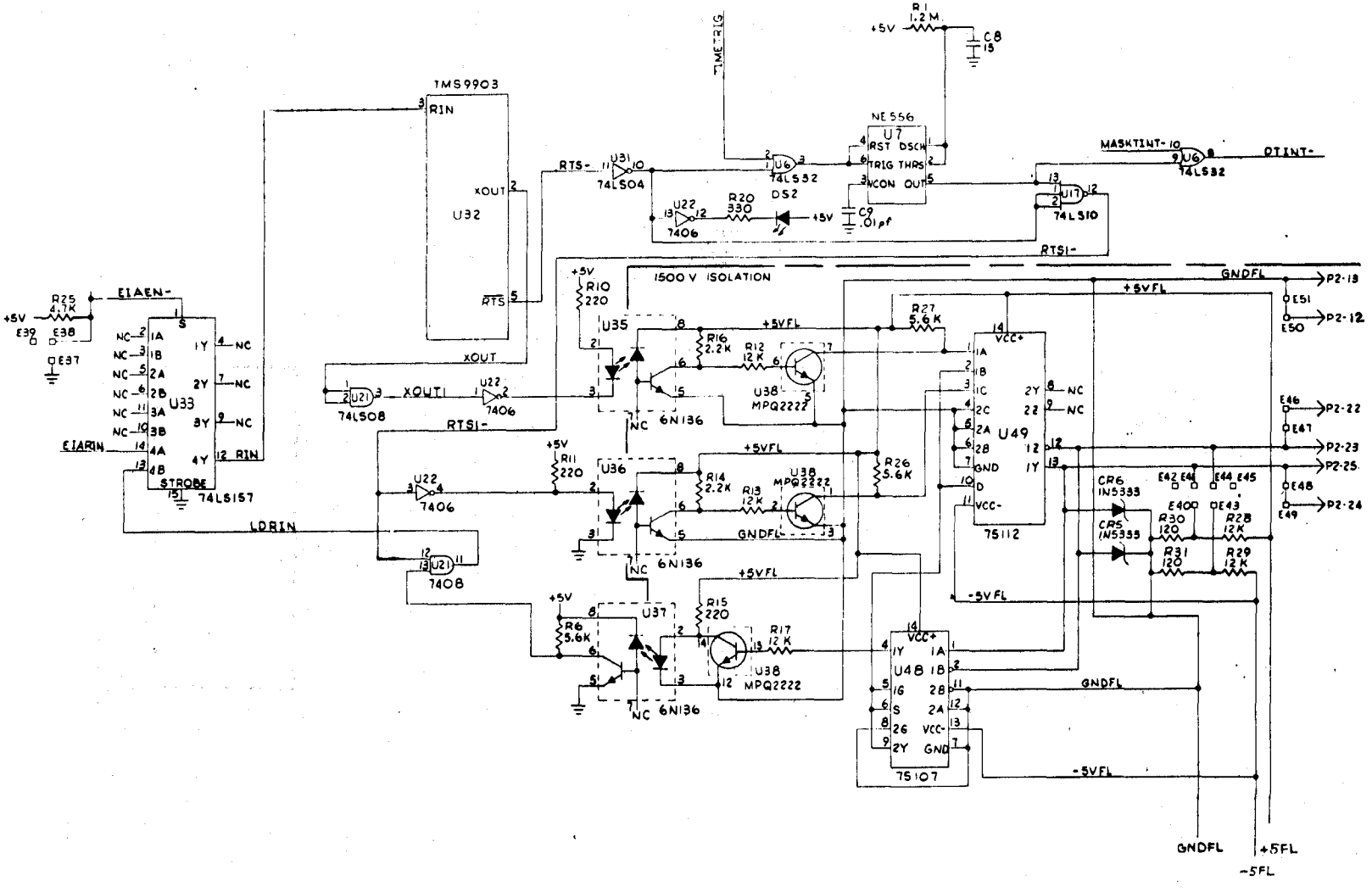


FIGURE 4-9. MULTIDROP INTERFACE CIRCUITRY

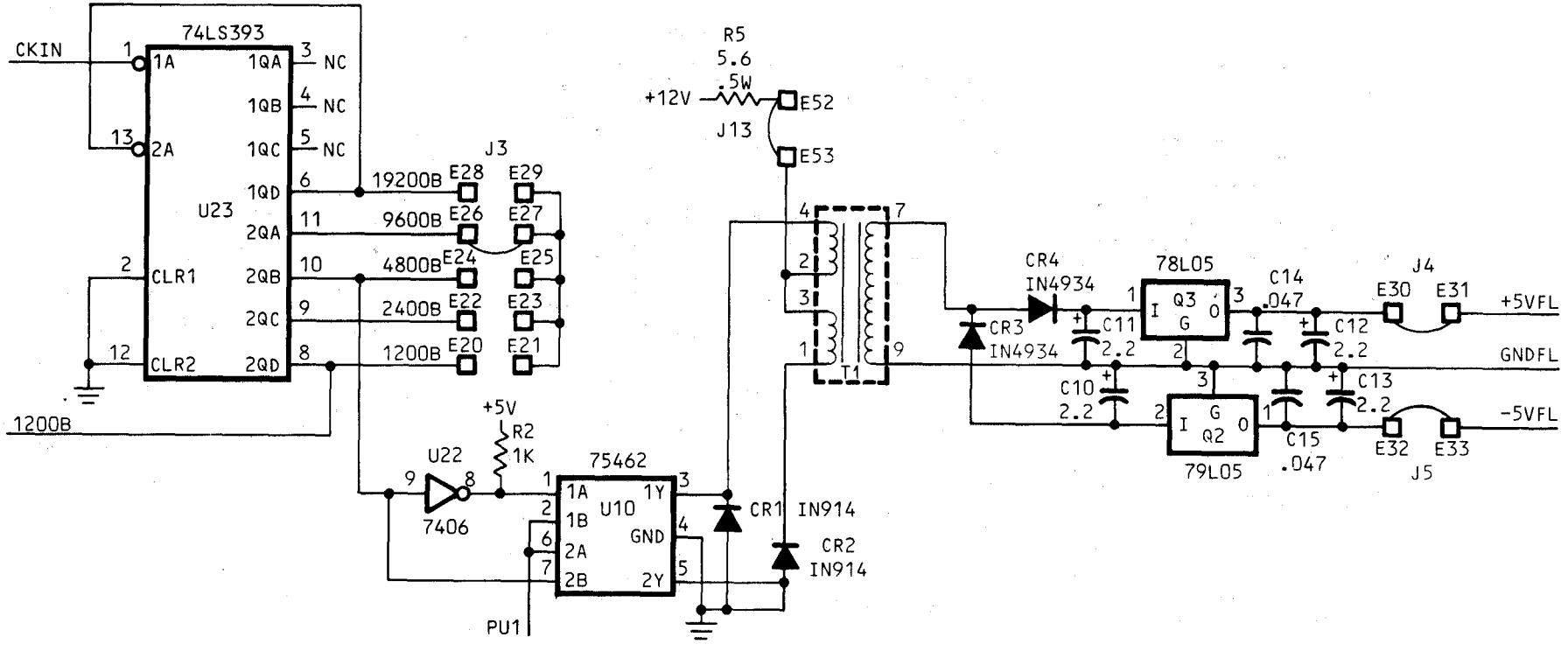


FIGURE 4-10. ISOLATED POWER CIRCUITRY

4.2.14 CRU Base Address Decoding

CRU base address decoding is accomplished by comparing the value of the eight address bits from the host TM 990 to a DIP-switch (S1) setting (Refer to Figure 4-11). When the value of address bits A3.B through A10.B equals the CRU base address selected on S1-1 through S1-8, U4 provides the CRUSEL signal to U5. This provides a board select function for the host processor.

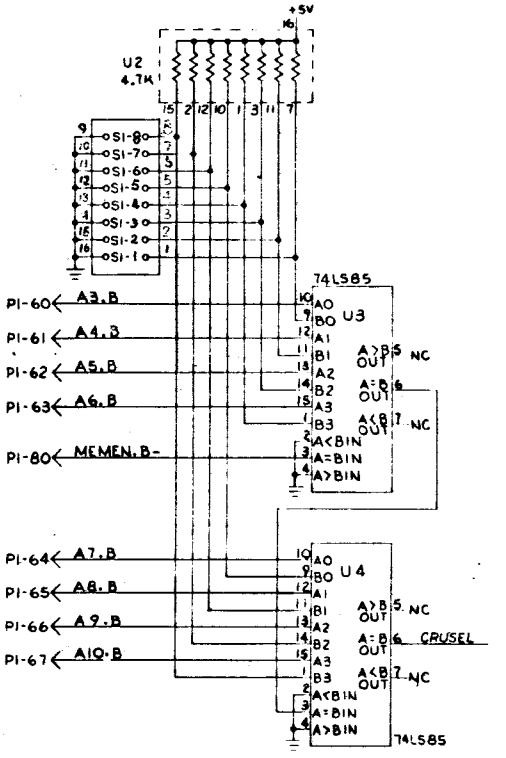


FIGURE 4-11. CRU ADDRESS DECODING

4.2.15 Host processor and 9980 Interactive I/O

Data (bits 0-7) is transferred in and out of the 9980 via a memory mapped I/O location in the 9980's address space. This same data is transferred in and out of the host TM 990 processor via CRU. Refer to Figure 4-12 for the explanations contained in sub-sections 4.2.15.1 through 4.2.15.7.

4.2.15.1 9980 Memory Mapped Input. Data is read onto the 9980's data bus (D0-D7) through data selectors U43 and U53 when they are enabled by ENINBYTE1-. This data can come first from the host via CRU output addressable latch U44 and second from the six address switches on the ICM. The two sources are selected by ADDSEL which is a 9980 CRU output bit from addressable latch U14.

4.2.15.2 9980 Memory Mapped Output. Data is output from the 9980 data bus (D0-D7) to latch U25 when it is enabled by ENOUTBYTE1. This data goes to the host CRU input multiplexer U11 and to LEDs DS4 to DS10 to display an error code (See Table 3-3).

4.2.15.3 Host CRU Data Input. The host can either read data from the 9980 on CRU input multiplexer U11 or it can read the address switches on input multiplexer U54 depending on the state of ADDRESS which is a host CRU output from addressable latch U45.

4.2.15.4 Host CRU Control Output. The addressable latch U45 is the host CRU output device which outputs several control and status signals to the ICM. ADDRESS was discussed in sub-section 4.2.15.3 previously. FAULTDR, OUTEN, TESTEN, and ECHOEN are status bits which are read by the 9980 CRU input multiplexer U46. DAV is also read by U46. In addition, DAV along with RSTBD are inverted by U31 to generate interrupts to the 9980. INTMSK will mask interrupts from the ICM at U1 pin 2.

4.2.15.5 9980 CRU Status Input. The 9980 CRU input multiplexer U46 reads the signals mentioned in sub-section 4.2.15.4. It also reads the state of the EIA enable jumper (J7) and the address decode enable switch S2-7.

4.2.15.6 9980 CRU Control Output. The 9980 CRU output multiplexer (U14) outputs control bits for the ICM and status bits to the host processor. A description of these control/status bits follows. DOWNLOAD starts the download sequence. TIMETRIG starts the 20 second timeout timer. MASKINT masks the interrupt to the 9980 generated by a timeout condition. ENDOB is a status bit which is read by the host CRU input multiplexer (U15). ADDSEL was described in sub-section 4.2.15.1. FAULT lights the FAULT LED. Both RRQ and WRQ are status bits which can be read by the host CRU input multiplexer (U15). Either of these signals will also generate the INT- interrupt to the host if host interrupts are enabled.

4.2.15.7 Host CRU Status Input. The host CRU input multiplexer (U15) reads the ENDOB, WRQ, and RRQ status bits as described in sub-section 4.2.15.6. It also reads the INTMASK signal described in sub-section 4.2.15.4 and the INT- status described in 4.2.15.6.

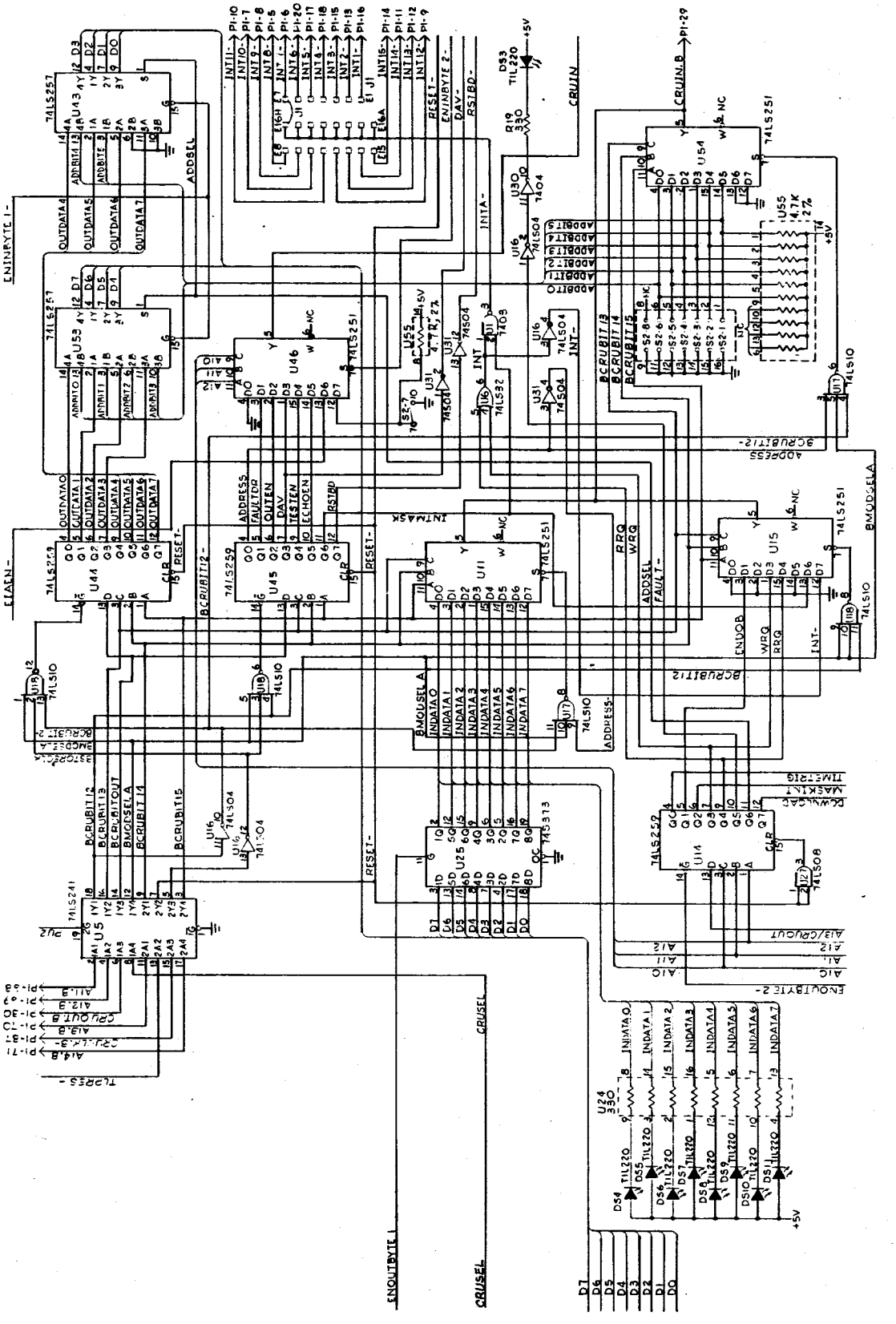


FIGURE 4-12. BUFFERS, INTERFACE LOGIC, MODULE ADDRESS SELECT

FIGURE 4-12. BUFFERS, INTERFACE LOGIC, MODULE ADDRESS SELECT

4.2.16 Download Circuitry

The download circuitry is shown in Figure 4-13. The download circuitry is used to generate PRES- or RESTART- signals for the TM 990/100/101 CPU module when the download command (described in sub-section 3.7.3) is received by the ICM. Upon correct reception, the 9980 sets its' "download" CRU bit that lights the LED and toggles flip-flop U9 pin 1. This generates a pulse of three clock periods at U9 pin 9 which blocks the IORST- signal from coming on-board while a pulse of one clock period at U20 pin 5 creates either the PRES- or RESTART- signals. PRES- on the host processor generates an IORST- which is undesireable immediately after the download command. The clock is derived from the baud rate generator. This circuitry allows either a "reset" or a "load" of the host processor. When active (low), PRES- resets the host processor and issues an I/O reset (IORST-) to all peripherals. PRES- is primarily intended for system initialization or full system reset if activated after initialization. When active (low), RESTART- causes the host processor to perform a load function; reset is not activated. An active signal (PRES- or RESTART-) will cause a context switch through the vectors located at 0000_{16} or $FFFC_{16}$, respectively.

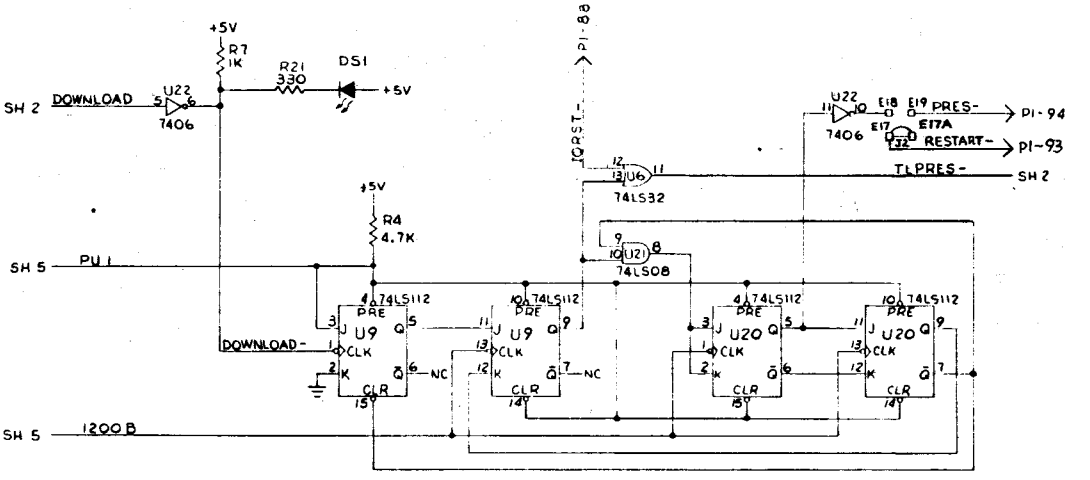


FIGURE 4-13. DOWNLOAD CIRCUITRY

APPENDIX A

SCHEMATICS

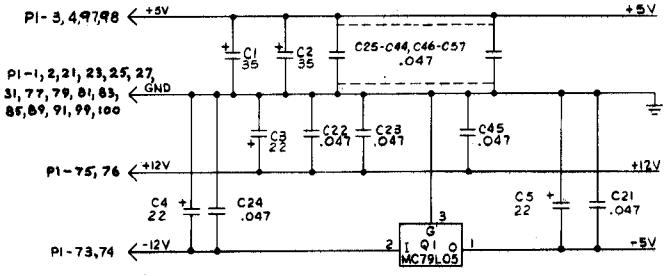
NOTES: UNLESS OTHERWISE SPECIFIED:

1. CAPACITOR VALUES ARE IN MICROFARADS
2. ALL RESISTOR VALUES ARE IN OHMS
3. ALL RESISTORS TO BE .25W, 5%
4. SOCKETS AT U39, U41 AND U50 ARE PROVIDED FOR FUTURE EXPANSION

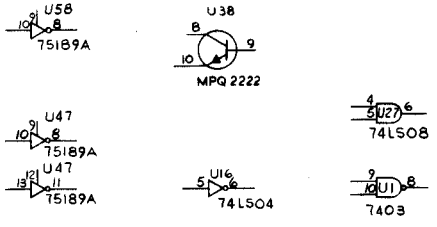
JUMPER CONFIGURATION	
JUMPER	POSITION
J1	E7, E16 H
J2	E17, E17 A
J3	E26, E27
J4	E30, E31
J5	E32, E33
J6	E35, E36
J7	E38, E39
J8	E41, E42
J9	E44, E45
J10	E46, E47
J11	E48, E49
J12	E50, E51
J13	E52, E53

REFERENCE DESIGNATORS	
USED	NOT USED
C1 - C57	
CR1 - CR6	
DS1 - DS11	
E1 - E53	
P1 - P2	
Q1 - Q3	
R1 - R31	
S1 - S2	
T1	
TP1	
U1 - U58	
Y1	
J1 - J13	

REVISIONS			
REV	DESCRIPTION	DATE	APPROVED
A	DESIGN CHANGE, INFORMAL UPDATE TO AGREE WITH PWB REV B	3/14/80	
B	ECN 471779		



SPARES



SEQ NO	IDENT	F.SPEC	NO	ADDITIONAL	NOTES
	PROCESS			CLASSIFICATION	

PROCESSES — FOR CORRELATION TO GOVT/IND SPECIFICATIONS SEE TI DRAWING 729467

QTY	ITEM NO	PART OR IDENTIFYING NUMBER	NOMENCLATURE OR DESCRIPTION	PROCUREMENT SPECIFICATION	NOTES
			PARTS LIST		
1		1600162	8117		
1		1600161	8117		
NEXT ASSY		USED ON			
APPLICATION					

UNLESS OTHERWISE SPECIFIED
- DIMENSIONS ARE IN INCHES
- TOLERANCES ANGLES ± 1°
  2 PLACE DECIMALS ± .010
  3 PLACE DECIMALS ± .127
- INTERPRET DRAWING PER MIL-D-1 DNC
- REMOVE ALL BURRS AND SHARP EDGES
- CONCENTRICITY: MACHINED DIAMETERS .010 FIM
- DIMENSIONAL LIMITS APPLY BEFORE PROCESSES
- PARENTHETICAL INFO FOR REF ONLY

HOLE TOLERANCE

.013 THRU .129	+.004 -.003	.131 THRU .250	+.005 -.001	.251 THRU .500	+.006 -.001
.501 THRU .750	+.008 -.002	.751 THRU 1.000	+.010 -.001	1.001 THRU 1.500	+.012 -.001

	NAME	DATE
DWN		3/5/79
ENGR		3/14/80
ENGR		4/1/80
CHK		4/1/80
QA		4/1/80
MFG		4/4/80
MGR		4/1/80
		4-2-80

TEXAS INSTRUMENTS INCORPORATED Dallas, Texas

DIAGRAM, LOGIC
TM 990/308

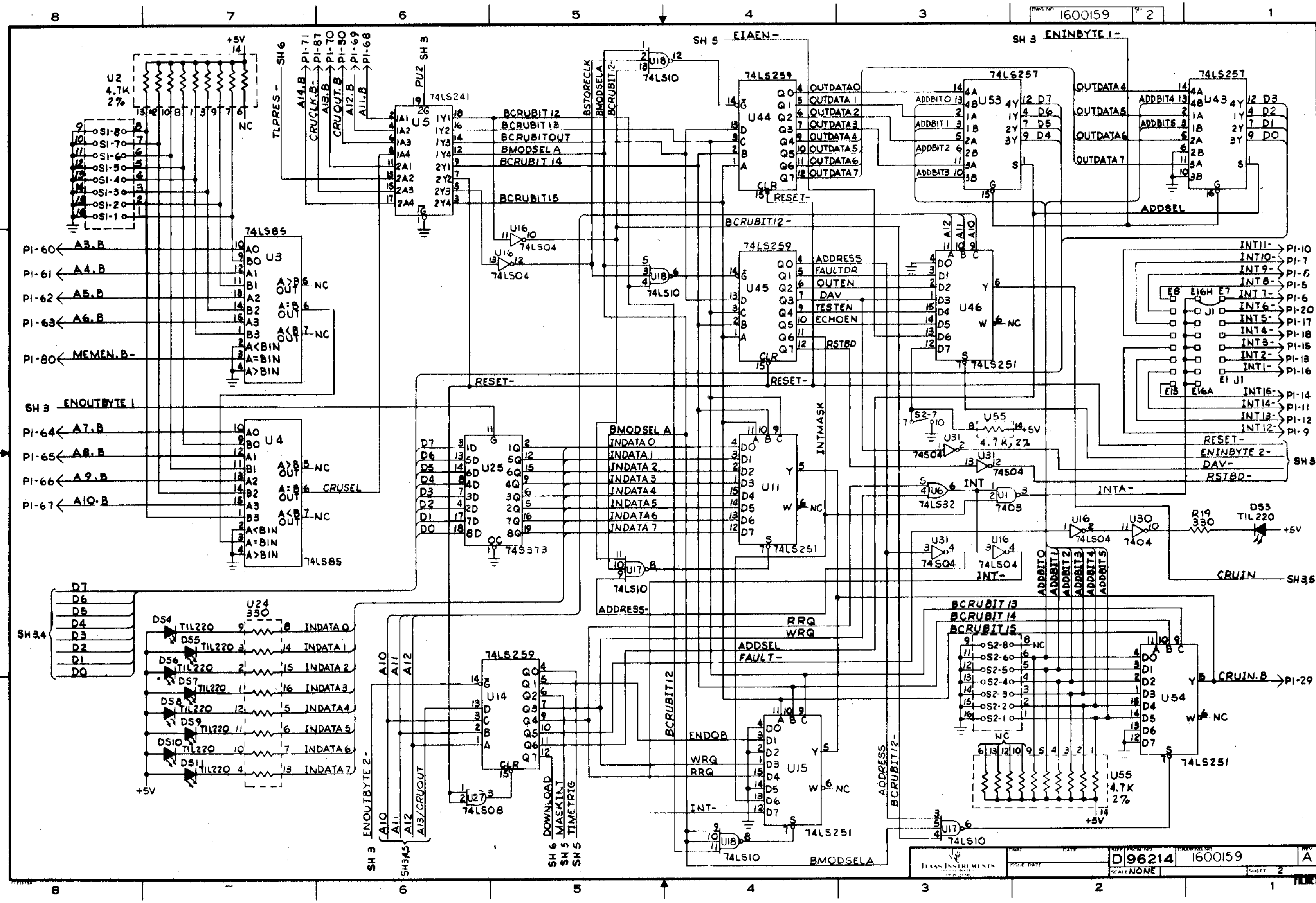
SIZE	FSCM NO	DRAWING NO
D	96214	1600159

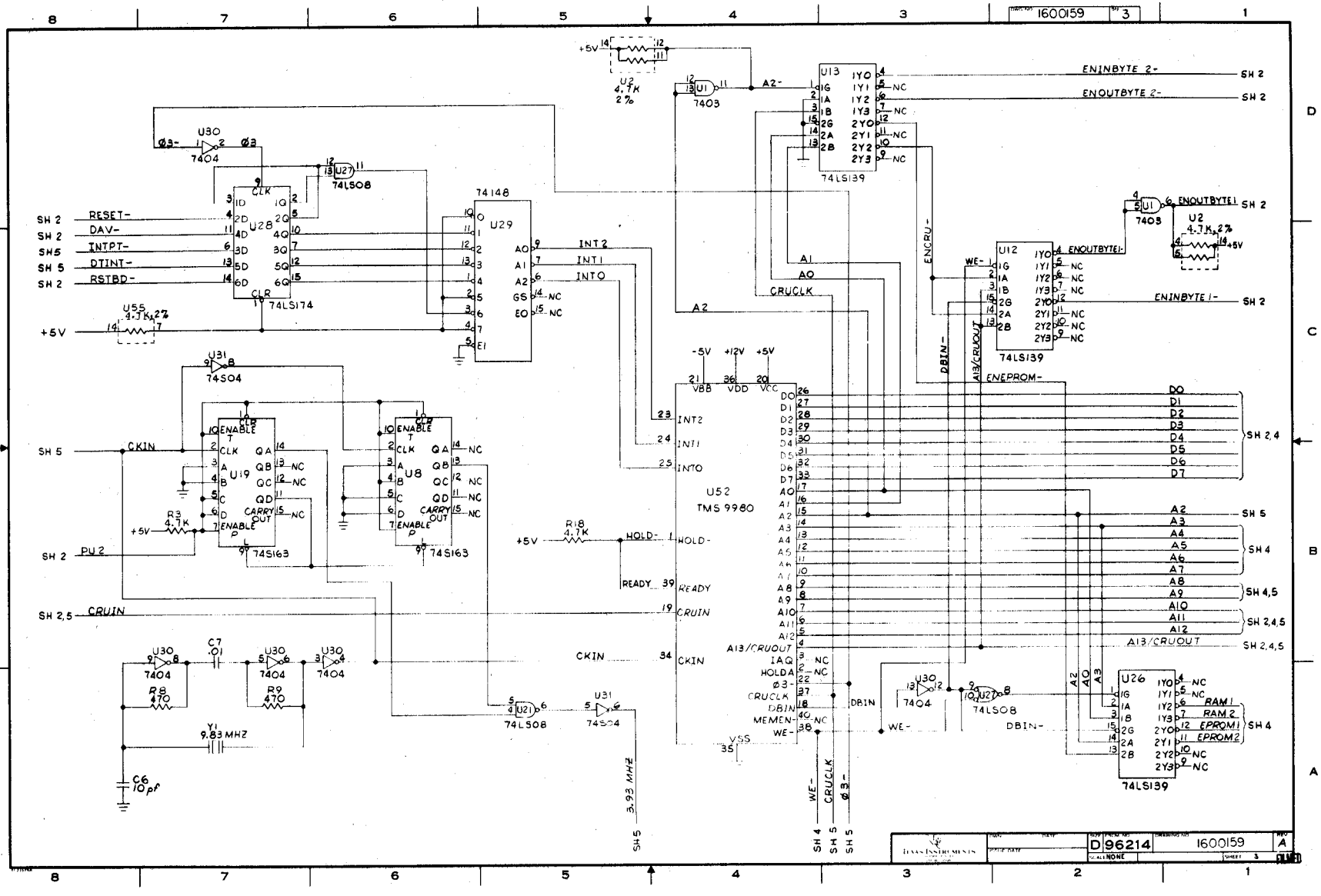
SCALE NONE SHEET 1 OF 6

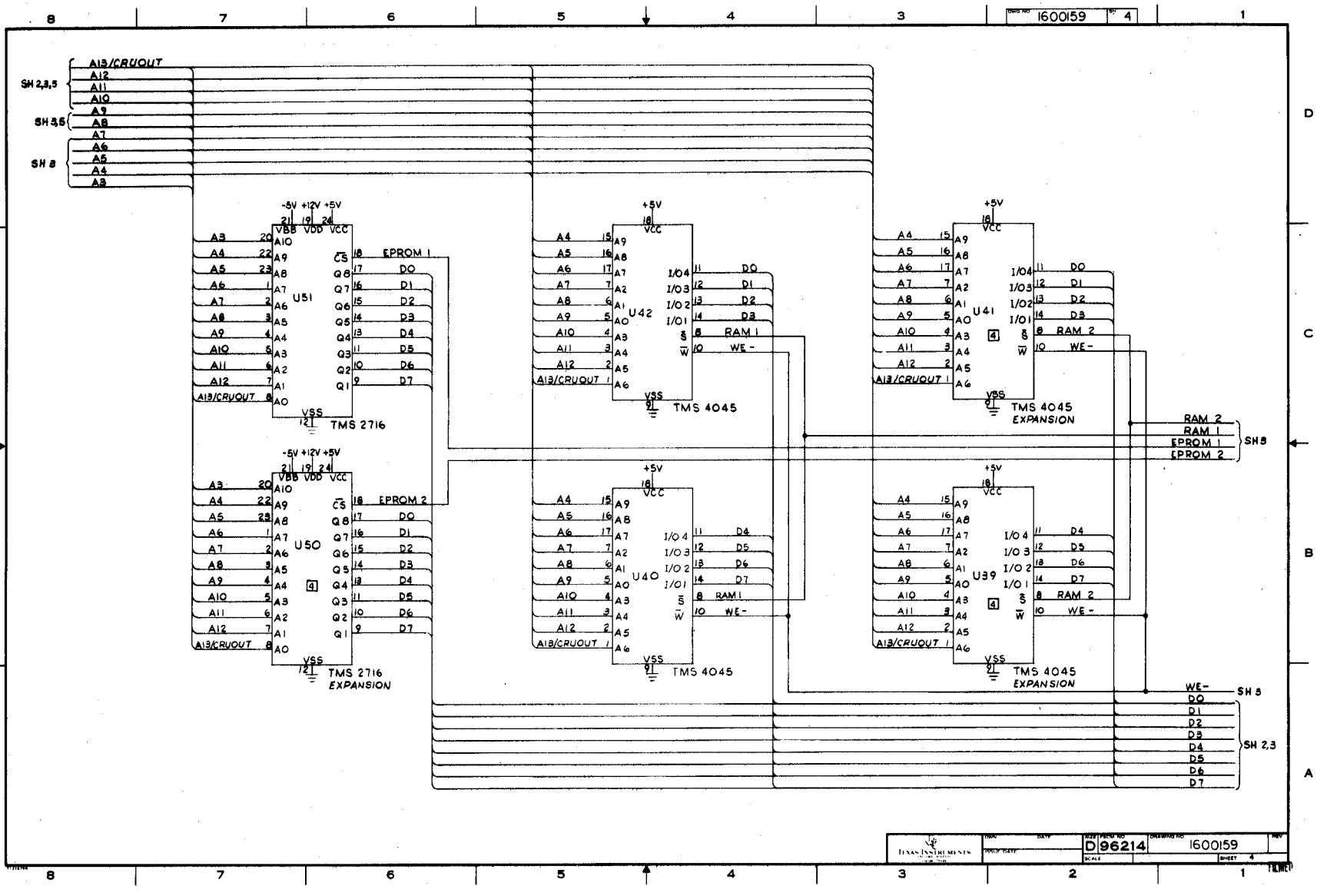
FILMED

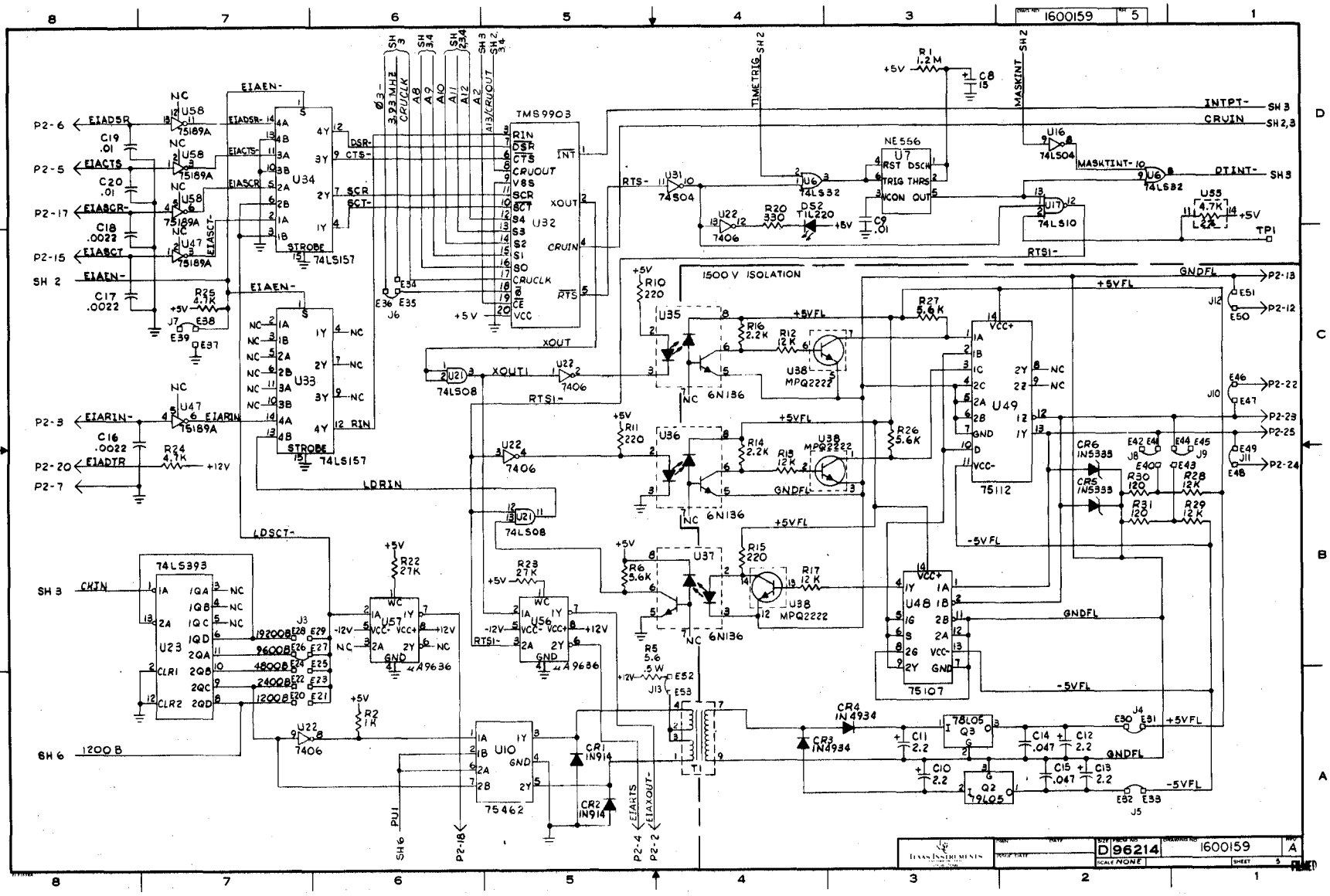
A-2

43

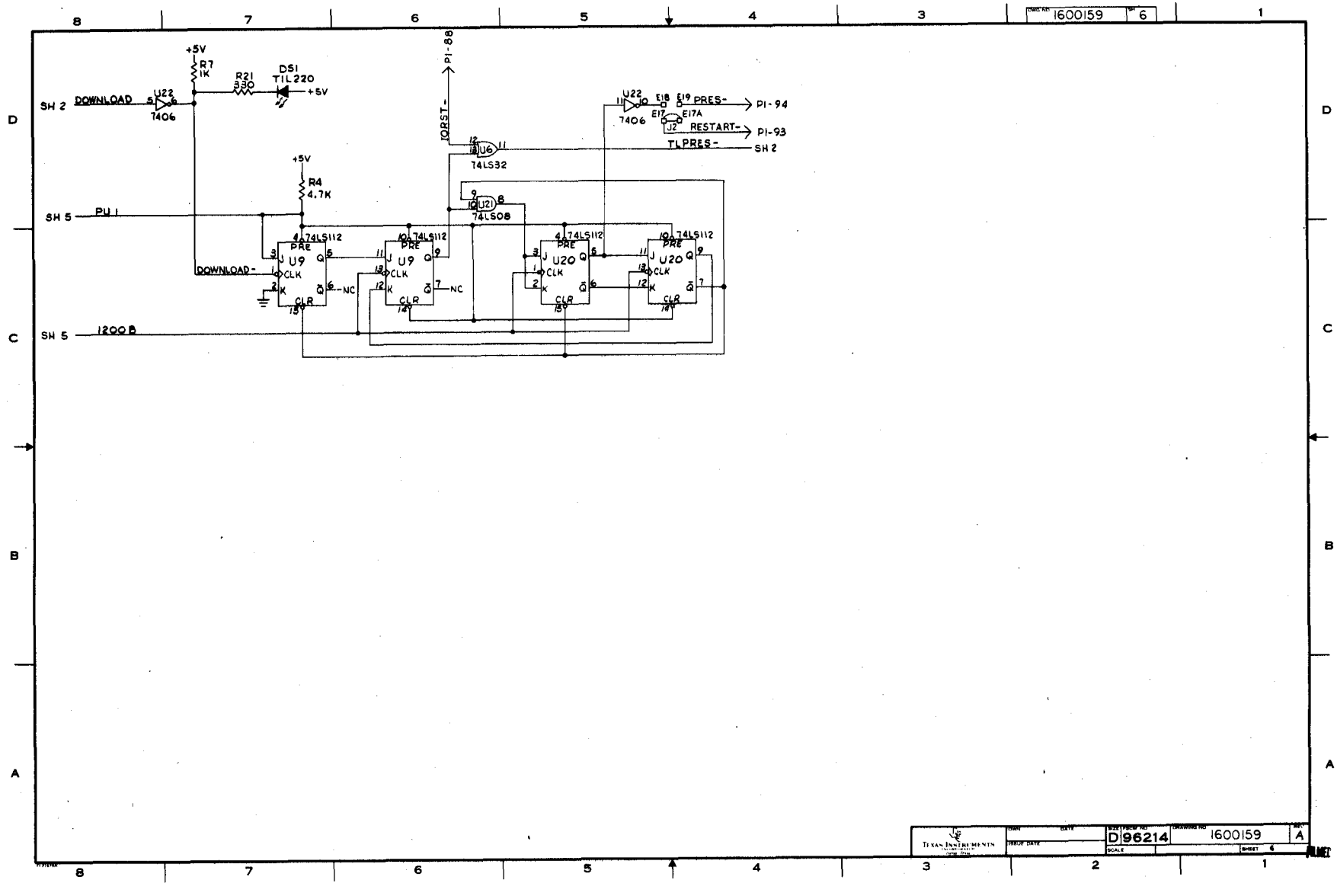

1600159
SH 5 EIAEN-
SH 3 ENINBYTE 1-
74LS259
U44
74LS257
U53
U43
74LS241
U5
U2
4.7K
2%
74LS85
U3
U4
74LS259
U45
U46
74LS251
U25
74S373
U11
U55
4.7K
2%
U14
U15
U54
U24
330
74LS10
74LS04
74LS32
7408
7404
74LS08
BCRUBIT12
BCRUBIT13
BCRUBITOUT
BMODSEL A
BCRUBIT14
BCRUBIT15
OUTDATA0
OUTDATA1
OUTDATA2
OUTDATA3
OUTDATA4
OUTDATA5
OUTDATA6
OUTDATA7
ADDSEL
ADDRESS
FAULTDR
OUTEN
DAV
TESTEN
ECHOEN
RSTBD
RESET-
INDATA0
INDATA1
INDATA2
INDATA3
INDATA4
INDATA5
INDATA6
INDATA7
ENOUTBYTE 1
MEMEN.B-
CRUSEL
CRUIN
CRUIN.B
INTMASK
INT
INTA-
INT-
RRQ
WRQ
FAULT-
ENDOB
DOWNLOAD
MASKINT
TIMETRIG
TIL220
+5V
D 96214
1600159
A
SCALE NONE
SHEET 2
TEXAS INSTRUMENTS


1600159
3
U30
7404
U27
74LS08
U28
74LS174
74148
U29
U55
4.7K 2%
U31
74S04
U19
74S163
U8
74S163
R3
4.7K
U52
TMS 9980
U13
74LS139
U12
74LS139
U26
74LS139
U1
7403
U2
4.7K 2%
R18
4.7K
R8
470
R9
470
C7
.01
C6
10pf
Y1
9.83 MHZ
RESET-
DAV-
INTPT-
DTINT-
RSTBD-
CKIN
CRUIN
PU2
READY
HOLD-
INT2
INT1
INT0
A2
A1
A0
CRUCLK
DBIN
WE-
MEMEN-
ENINBYTE 2-
ENOUTBYTE 2-
ENOUTBYTE1
ENINBYTE 1-
ENCRU-
A13/CRUOUT
ENEPROM-
RAM 1
RAM 2
EPROM1
EPROM2
D0
D1
D2
D3
D4
D5
D6
D7
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
D96214
1600159
A
NONE


DWG NO 1600159 REV 4
A13/CRUOUT
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
SH 2,3,5
SH 3,5
SH 3
-5V +12V +5V
VBB VDD VCC
U51
U50
TMS 2716
TMS 2716 EXPANSION
EPROM 1
EPROM 2
U42
U40
U41
U39
TMS 4045
TMS 4045 EXPANSION
RAM 1
RAM 2
WE-
D0
D1
D2
D3
D4
D5
D6
D7
SH 3
SH 2,3
TEXAS INSTRUMENTS
SIZE D
PROD NO 96214
DRAWING NO 1600159
SHEET 4


1600159
TMS9903
TMS9902
74LS157
74LS393
74LS08
74LS04
74LS32
74LS10
NE556
6N136
MPQ2222
TIL220
IN4934
IN914
IN5333
75189A
75112
75107
75462
A9636
A9686
7406
1500 V ISOLATION
INTPT-
CRUIN
MASKINT
MASKTINT
DTINT-
RTS-
RTSI-
XOUT
XOUTI
LDRIN
LDSCT-
CKIN
1200 B
6H 6
+5VFL
-5VFL
GNDFL
P2-6
P2-5
P2-17
P2-15
SH 2
P2-3
P2-20
P2-7
P2-18
P2-4
P2-2
P2-13
P2-12
P2-22
P2-23
P2-25
P2-24
TP1
D96214
A

A-6


DWG NO 1600159
SH 2
DOWNLOAD
U22
7406
+5V
R7
1K
R21
330
DS1
TIL220
+5V
P1-88
IORST-
U22
E18
E19
PRES-
P1-94
7406
E17
E17A
J2
RESTART-
P1-93
U6
74LS32
TLPRES-
SH 2
+5V
R4
4.7K
SH 5
PU 1
U21
74LS08
74LS112
U9
U20
PRE
CLK
CLR
J
K
Q
NC
DOWNLOAD-
SH 5
1200 B
TEXAS INSTRUMENTS
D 96214
1600159
A
SHEET 6

APPENDIX B

PARTS LIST

Symbol	Description
C1, C2	Capacitor, 35 uF @ 25V
C3, C4, C5	Capacitor, 22 uF @ 15V
C6	Capacitor, 10 pF @ 200V
C7, C9, C19, C20	Capacitor, .01 uF @ 100V
C8	Capacitor, 15 uF @ 20V
C10, C11, C12, C13	Capacitor, 2.2 uF @ 20V
C14, C15, C21-C55, C56, C57	Capacitor, .047 uF
C16, C17, C18	Capacitor, .0022 uF @ 100V
CR1, CR2	Diode, 1N914B
CR3, CR4	Diode, 1N4934-1
CR5, CR6	Diode, 1N5333B
DS1-DS11	Right angle LED, red
E1-E15, E16A-E16H, E17, E17A, E18-E53, TP1	Pin .025 square
J1-J13	Jumper plug, connector black
P2	Connector, RS-232, PWB MTG.
Q1, Q2	IC, MC79L05ACP
Q3	IC, UA78L05ACLP
R1	Resistor, 1.2 megohm, .25W
R2, R7	Resistor, 1 kilohm, .25W
R3, R4, R18, R24, R25	Resistor, 4.7 kilohm, .25W
R5	Resistor, 5.6 ohm, .5W
R6, R26, R27	Resistor, 5.6 kilohm, .25W
R8, R9	Resistor, 470 ohm, .25W
R10, R11, R15	Resistor, 220 ohm, .25W
R12, R13, R17, R28, R29	Resistor, 12 kilohm, .25W
R14, R16	Resistor, 2.2 kilohm, .25W
R19, R20, R21	Resistor, 330 ohm, .25W
R22, R23	Resistor, 27 kilohm, .25W
R30, R31	Resistor, 120 ohm, .25W
S1, S2	Switch, dual in line, 8 position
T1	Transformer, ferrite isolation Suggested sources: 1. GFS Manufacturing Co. Sixth Street Industrial Park Dover, N.H. 03820 P/N 2261953-1 2. National Transformer Corp. P.O. Drawer 130 500 Follis St. Johnston, Ill. 62951 P/N NT3487

PARTS LIST (CONTINUED)

Symbol	Description
U1	Network, SN7403N
U2, U55	Network, resistor, 4.7 kilohm
U3, U4	Network, SN74LS85N
U5	IC, SN74LS241N
U6	Network, SN74LS32N
U7	IC, NE556
U8, U19	Network, SN74S163N
U9, U20	Network, SN74LS112N
U10	IC, 75462
U11, U15, U46, U54	Network, SN74LS251N
U12, U13, U26	Network, SN74LS139N
U14, U44, U45	IC, SN74LS259N
U16	Network, SN74LS04N
U17, U18	Network, SN74LS10N
U21, U27	Network, SN74LS08N
U22	Network, SN7406N
U23	IC, 74LS393N
U24	Network, resistor, 330 ohm (Bournes 4114R-802-472)
U25	IC, SN74LS373N
U28	Network, SN74LS174N
U29	Network, SN74148N
U30	Network, SN7404N
U31	Network, SN74S04N
U32	IC, TMS 9903
U33, U34	Network, SN74LS157N
U35, U36, U37	IC, 6N136
U38	Transistor, MPQ2222/SQ1062
U40, U42	IC, 2114, static RAM
U43, U53	IC, SN74LS257N
U47, U58	Network, SN75189AN
U48	Network, SN75107N
U49	IC, SN75112N
U51	EPROM, (TMS 2716)
U52	IC, TMS 9980
U56, U57	IC, EIA line driver (Fairchild 9636TC)
XU32	Socket, 20 pin IC, low profile solder tail
XU39, XU40, XU41, XU42	Socket, 18 pin IC, low profile solder tail
XU50, XU51	Socket, 24 pin IC, low profile solder tail
XU52	Socket, 40 pin IC, low profile solder tail
Y1	Crystal, quartz, 9.8304 MHz (MIL CR60A/U 9.8304)

APPENDIX C

TM 990/428 DEMONSTRATION SOFTWARE FOR THE TM 990/308 INDUSTRIAL COMMUNICATIONS MODULE

C.1 INTRODUCTION

The TM 990/428 demonstration software for the TM 990/308 industrial communications module (ICM) has two purposes. The first is to give our customer a quick and easy way to determine if his board is working properly. The second is to provide via the source listing an example of how to program his board in a realistic application.

C.2 NECESSARY HARDWARE

C.2.1 Industrial Line Mode

	Qty	Hardware
1.	Two	TM990/308's
2.	Two	Microcomputers: TM 990/100M or TM 990/101
3.	Two	Memory boards: TM 990/201 or TM 990/203 (with at least 3K bytes of RAM)
4.	Two	Copies of TM 990/428 demonstration software
5.	Two	Chassis: TM 990/510, TM 990/520, or TM 990/530
6.	Two	Power supply: TM 990/518 or equivalent
7.	Two	Terminals: 733 ASR, Lear Seigler ADM, or equivalent
8.	One	Cable as described in subsections 2.5, 2.3.6 and 2.3.8.

C.2.2 EIA Mode (Bell 208 MODEM)

	Qty	Hardware
1.	Two	TM990/308's
2.	Two	Microcomputers: TM 990/100M or TM 990/101
3.	Two	Memory boards: TM 990/201 or TM 990/203 (with at least 3K bytes of RAM)
4.	Two	Copies of TM 990/428 demonstration software
5.	Two	Chassis: TM 990/510, TM 990/520, or TM 990/530
6.	Two	Power supply: TM 990/518 or equivalent
7.	Two	Terminals: 733 ASR, Lear Seigler ADM, or equivalent
8.	Two	TM990/502 cables
9.	Two	Bell 2088-L1A data sets or equivalent

C.3 OPERATION OF TM900/428 DEMO SOFTWARE LOCAL LINE MODE

Set jumpers and switches on the TM 990/308 as follows:

Jumper	Setting	
J1	E4-E16E	Interrupt selection to Interrupt 4
J2	E17-E17A	Disable download
J3	E26-E27	Baud Rate to 9600
J4	E30-E31	Isolated power jumper on
J5	E32-E33	Isolated power jumper on
J13	E52-E53	Connects +12 V power to isolated power circuitry
J6	E35-E36	9903 Clock standard (Never use 19.2 mode).
J7	E38-E39	In the industrial line mode
	or	
	E37-E38	In EIA mode
J8, J9		Jumper the terminators as described in subsection 2.3.6. Although the demo software works with two stations at a time, up to 32 stations, each with its unique address, may be on the same multidrop line.
J10	E46-E47	Install these three jumpers as shown.
J11	E48-E49	
J12	E50-E51	
SW1		CRU Base to $1FE0_{16}$ (all OFF)
SW2-1 to SW2-6		Module address switches as desired. Each /308 board on an industrial line should have a unique address set on switches SW2-1 through SW2-6.
SW2-7		Set SW2-7 to OFF for enabling address decoding.

The Demo Software will work over either the industrial line or over modems like the Bell 208. For industrial line operation, connect jumper E38-E39 and install the industrial line as described in subsection 2.6.3. For use with a modem, connect jumper E37-E38 and connect a TM 990/502 cable between the ICM board and modem. Follow the manuafacturer's instructions to establish communmication from modem to modem and to get the carrier. After establishing communication, the Demo Software works the same via modem as it does over an industrial line.

Set jumpers on the microcomputer boards as shown in Tables C-1 and C-2.

TABLE C-1. JUMPER SETTINGS AT TM 990/101M BOARD

Jumper	Position
BANK1	2716
BANK2	2716
E1-E2	Interrupt 4 to P1-18
E9-E10	2716
E13-E14	EPROM ON
E16-E17	RAM HI
E39-E40	EIA

TABLE C-2. JUMPER SETTINGS AT TM 990/100M BOARD

Jumper	Position
J1	P1-18
J2	2716 Position
J3	2708 Position
J4	2716 Position
J7	EIA mode

C.4 TM990/428 DEMO SOFTWARE EPROM PLACEMENT

The Demo Software is designed to work with TIBUG, and the demo software EPROM's can be placed in any 4K (decimal) bytes of memory address space not already occupied by TIBUG or RAM. The program when executed finds itself and a free 6K byte block of RAM below $F000_{16}$ (search starts at upper address and seeks downward for first 6 K block). A suggested Memory Map is implemented as follows:

1. Install TM990/428 EPROMS on a TM 990/100M or /101 microcomputer board in sockets U-43 and U-45 as marked on the EPROMs.

2. Install a contiguous 6 K byte block of RAM on a 2 K byte ($X800_{16}$ or $X000_{16}$) memory bound between 0000_{16} and $F000_{16}$(this area must not interfere with either the demo software and TIBUG in EPROM or with user RAM). For a suggested installation at 2800_{16}, set switches as follows:

   On a TM 990/201/206 board, populate RBLK0 and RBLK1 and set the RAM configuration switches 5 to 8 with OFF, ON, OFF, ON respectively.

   On a TM 990/203 board, populate bank A and set switches as follows:
   - S1 and S3 to all OFF
   - S2 to ON, ON, OFF, ON
   - S4 to ON, ON, OFF, OFF

   NOTE

   Set other jumpers and switches on the memory boards according to instructions in the user manuals.

C.5 EXECUTION

The following sequence describes how to execute the Demo Software.

1. Install the boards (microcomputer, memory, and ICM) in a chassis with a properly connected power supply. Make the necessary cabling as specified in Section 2 of this manual.

2. Turn on the power supply and press the RESET pushbutton on the microcomputer board. The ICM board will execute its self test diagnostic. Observe the LEDs on the ICM board during the self test. The FAULT LED should come on for approximately one second then go off. During this same time, LEDs D0 - D7 should flicker briefly then come on brightly. The XMIT and DWNLOAD LED's should both be off when the FAULT LED goes off. If the diagnostic fails, the FAULT LED will stay on and LED's D0 - D7 will display an error code, LED Off = 1, LED On = 0. Error codes are defined in Table 3-3.

3. After the ICM passes the self test, do the following at the keyboard:
   a. Type an "A" key on the terminal; the TIBUG banner will come up.
   b. Type "R" to observe the hardware register contents.
   c. Type a space to view the Program Counter contents.
   d. Type the hex memory address of the starting address of the TM990/428 Demo Software; in the example used herein, this is 1000_{16}.
   e. Type a carriage return to exit the register inspect/change command.
   f. Type "E" to execute the Demo Software program.

The top part of Figure C-1 shows this sequence.

```
TIBUG REV.A

?R
W=FFB0
P=0170  1000
?E
TM990/428 DEMO SOFTWARE  REL. 1.1
COPYRIGHT 1980 BY TEXAS INSTRUMENTS, INC.
EPROM AT >1000   DEMO RAM AT >2800

PRIMARY OR SECONDARY STATION (P,S) (S)        P
RANDOM OR USER DATA          (R,U) (R)
FIXED OR VARIABLE LENGTH     (F,V) (V)
LENGTH OF MESSAGE                  (039F)
308 CRU BASE ADDRESS               (1FE0)
ADDRESS OF OTHER STATION           (003F)
INTERRUPT 4 MUST BE JUMPERED
KEYBOARD TEST OR VERIFY      (K,V)
??
```

DEMO RAM LOCATION

FIGURE C-1. EXAMPLE OF INITIALIZATION PROCEDURE AT TERMINAL

The Demo Software banner should now come up. The first time the Demo Software is executed you will be asked to initialize the parameters listed below and on Figure C-1. The last parenthetical expression of each parameter prompt is the default value; this can be chosen by entering a carriage return.

PRIMARY OR SECONDARY STATION (P,S) (S)
- P = This is the primary station issuing the messages.
- S = This is the secondary station receiving the messages (default).

RANDOM OR USER DATA (R,U) (R)
- R = Message bytes will be constructed using the Demo Software's random number generator (default).
- U = The user will enter the message contents from the keyboard.

FIXED OR VARIABLE LENGTH (F,V) (V)

NOTE

These parameters are effective only with the V (Verify) command.

- F = Messages will be of a fixed length of bytes; this length to be specified in the next prompt.
- V = Demo software will specify length of message; this length will intentionally vary (default).

LENGTH OF MESSAGE (039F)

Enter the length in bytes of a fixed length message; this cannot be a hexadecimal value larger than 079F, the message buffer size with expansion RAM installed. Default is buffer size without expansion RAM (039F).

308 CRU BASE ADDRESS (1FE0)

Enter the software CRU base address of the ICM board; this is the setting of switch S1 as specified in paragraph 2.3.4; default is 1FE0 or all switches set to OFF.

ADDRESS OF OTHER STATION (003F)

Enter the address on switches S2-1 to S2-6 of the ICM to receive the message; default is 003F or all switches set to OFF.

INTERRUPT 4 MUST BE JUMPERED

This is a reminder to the user that interrupt 4 must be set by setting E4 to E-16E at jumper J1 on the ICM board.

KEYBOARD TEST OR VERIFY (K,V)
??

This is the same as for the K or V commands further explained in paragraphs C.6.4 and C.6.5. These commands or any of the other commands can be entered at this time.

These prompts can be repeated by using the "I" initialize command (paragraph C.6.3). After intialization, the default values will be the last value entered from the keyboard, and this will be the value shown in the last parentheses of the prompt. This allows the user to review (and change) the values presently being used.

C.6 COMMANDS

Each paragraph heading lists the command name and the character to enter to execute the command.

C.6.1. Help (H)

This command displays the Demo Software commands shown in Figure C-2.

C.6.2 Quit (Q)

This command returns progam control to the TIBUG monitor.

C.6.3 Initialize Parameters (I)

The I command allows the user to change the values set up at initialization (shown in Figure C-1 and described in C.5). The values in the last parentheses of the prompt are the current values (these will change to the new input) and are also the default values to be chosen by a carriage return.

C.6.4 Send/Receive Keyboard Routine (K)

The K command allows messages to be sent and received alternately at each terminal. This command allows the user to enter at the primary terminal a message, terminated by a carriage return. The message is then sent to the secondary terminal. The secondary terminal, in turn, then asks for a message to be entered at its keyboard while the primary terminal becomes the message recipient. In this fashion, a message is sent from the primary to the secondary, then the modes reversed so that the previous sender becomes the receiver and vice versa, and then the modes are reversed again, etc. To exit this command, enter a carriage return without a message. The K command must be entered at both systems with the secondary system enabled before the first message is sent from the primary system. Figure C-3 shows message transmission started at the primary terminal. Note that if a message is not sent or received within 25 seconds from the previous operation, a timeout will occur and control will return to the Demo Software command scanner.

```
H

COMMANDS:

H = HELP, PRINT THIS HELP LIST
Q = QUIT, RETURN TO TIBUG
I = INITIALIZE MODULE ADDRESS, PRIMARY/SECONDARY FLAG,
    LENGTH, AND RECEIVING STATION ADDRESS
K = KEYBOARD MESSAGE ENTRY MODE
V = VERIFY - AUTOMATIC TEST MODE
P = INITIALIZE RAM MEMORY WITH PATTERN
C = MEMORY TO MEMORY DATA COMPARE
M = MEMORY TO MEMORY BLOCK MOVE
N = FILL A BUFFER WITH RANDOM DATA
S = SELF TEST: RAM, EPROM, 9903, AND 9980
R = RESET, REINITIALIZES THE 308
A = SEND RECORD FROM BUFFER
B = RECEIVE RECORD INTO BUFFER

??
```

FIGURE C-2. HELP COMMAND PRINTOUT

```
??K

KEYBOARD SEND-RECEIVE UTILITY
KEY IN MESSAGE FOR STATION NUMBER 003F
HOW IS THE WEATHER THERE, GEORGE?

WAIT FOR MESSAGE FROM STATION NUMBER 003F
NOT BAD MARTHA!!!

KEY IN MESSAGE FOR STATION NUMBER 003F


??
```

FIGURE C-3. INITIATING MESSAGE FOR KEYBAORD (K) COMMAND AT PRIMARY TERMINAL

C.6.5 Verify System (Auto Test) (V)

The V command starts the Automatic Test Mode. This mode transmits either pseudorandom data or data from the 1952-byte transmit buffer RAM (buffer starts at the RAM base address shown in the Demo Software opening banner). The user can enter data in this buffer with the P command (initialize RAM memory with pattern, explained in C.6.6).

Data is inverted and sent from the primary terminal to the secondary terminal where it is returned to the primary terminal and inverted again; the received data is compared to the transmitted data and a message printed at each terminal showing the results of the comparison.

Data sent may be data patterns which the user suspects of causing problems in his communications system. Figure C-4 shows a sample run with random data. A carriage return stops execution of the test. A note on coordination: the secondary station must also be in the automatic test mode (via the V command) before the primary is placed in this mode, but once the V command is issued at the secondary terminal, a message from the primary must be sent to it within 25 seconds or the secondary will timeout and return to the monitor command scan prompt.

C.6.6 Initialize RAM Memory With Pattern (P)

The P command allows the user to initialize memory with a data pattern. This command is useful for initializing memory before read and write operations. The user will be prompted for a memory address, byte count, and data pattern. All entries are in hexadecimal and are terminated with a space character. The start address and byte count must be even numbers. Shown below are the prompts for this command.

```
??P
ADDR1 = XXXX LENGTH = LLLL PATTERN = PPPP
??
```

```
??V

AUTOMATIC SEND-RECEIVE TEST

PRIMARY STATION
PASSES  00000001    BYTES  00000001    ERRORS  0000    ERROR CODE  0000

PASSES  00000002    BYTES  00000100    ERRORS  0000    ERROR CODE  0000

PASSES  00000003    BYTES  00000200    ERRORS  0000    ERROR CODE  0000

PASSES  00000004    BYTES  000003FF    ERRORS  0000    ERROR CODE  0000

PASSES  00000005    BYTES  000005FF    ERRORS  0000    ERROR CODE  0000

PASSES  00000006    BYTES  000008FE    ERRORS  0000    ERROR CODE  0000

PASSES  00000007    BYTES  00000BFE    ERRORS  0000    ERROR CODE  0000

PASSES  00000008    BYTES  00000F9D    ERRORS  0000    ERROR CODE  0000

PASSES  00000009    BYTES  00000FA5    ERRORS  0000    ERROR CODE  0000

PASSES  0000000A    BYTES  00000FB4    ERRORS  0000    ERROR CODE  0000

PASSES  0000000B    BYTES  00000FD4    ERRORS  0000    ERROR CODE  0000

PASSES  0000000C    BYTES  00001013    ERRORS  0000    ERROR CODE  0000

PASSES  0000000D    BYTES  00001053    ERRORS  0000    ERROR CODE  0000

PASSES  0000000E    BYTES  000010D2    ERRORS  0000    ERROR CODE  0000

PASSES  0000000F    BYTES  00001152    ERRORS  0000    ERROR CODE  0000

25 SECOND TIMEOUT. TEST ABORTED
??
```

FIGURE C-4. VERIFY (V) COMMAND SEQUENCE AT PRIMARY TERMINAL

C.6.7 Memory To Memory Compare (C)

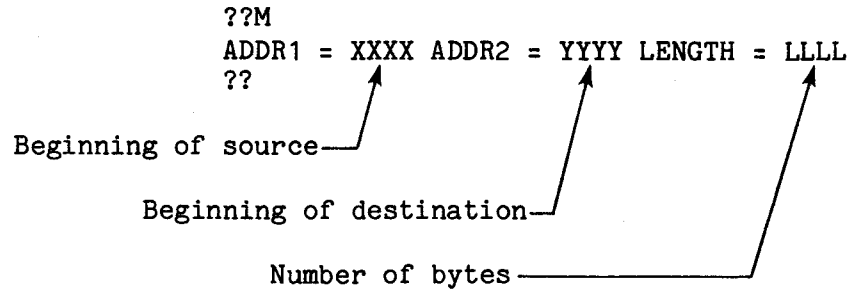
The C command is entered from the command scanner by entering the character "C". This command allows the user to verify the contents of two blocks of memory by comparing the contents of the two blocks on a word by word basis. If two corresponding memory locations do not have the same contents an error message is printed. This command is useful in comparing data after send and receive operations. The user will be prompted for the starting addresses of the two blocks of data and the number of bytes to be compared. All entries are in hexadecimal are terminated by the space character. Shown below are the prompts for this command and the error print out.

```
??C
ADDR1 = XXXX ADDR2 = YYYY LENGTH = LLLL

**ERROR
ADDR1 = XX42 DATA = ABCD ADDR2 = YY42 DATA = ABCC
??
```

C.6.8 Memory To Memory Block Move (M)

The M command, memory to memory block move, is entered from the command scanner by entering the character 'M'. This command allows the user to move a block of data from one memory location to another. The user will be prompted for the source address, destination address, and the number of bytes to be transferred. The user should terminate all entries with a space character. All entries are hexadecimal and must be an even number. Shown below are the prompts for this command.



C.6.9 Fill Memory With Random Data (N)

The N command is entered from the command scanner by entering the character "N". This command allows the user to fill a section of memory with pseudorandom data. The user will be prompted to enter the starting address and a length to be buffered. The user should terminate all entries with a space character. All entries are hexadecimal and must be an even number. Shown below are the prompts for this command.

```
??N
ADDR1 = XXXX LENGTH = LLLL
??
```

C.6.10 Self-Test (S)

The S, self-test command is entered from the command scanner by entering the character "S". This command forces the ICM to execute its internal self-test. This self-test is a routine in the ICM firmware which checks the ICM EPROM, RAM, the 9903, and the 9980 for proper operation. When completed, the user is prompted with a "PASSED" message or an "ERROR" message followed by an error code. The LED's can also be observed at this time for proper operation. See section 5.1 and 5.2. The self-test at a minimum must pass for the /308 to work properly. Shown below is a sample output.

```
??S
SELF-TEST PASSED
??
??S
SELF-TEST ERROR XXXX
??
```

C.6.11 Reset (R)

The R reset command is entered from the command scanner by entering the character "R". This command allows the user to escape from unknown error conditions on the ICM and put it back in the receive mode. It toggles the software RESET bit 15 of the ICM CRU.

C.6.12 Send Record From Buffer (A)

The A command sends a single record when the character "A" is typed in the command mode. The length of the record is determined by the length entry in the I command. The contents of the record is determined by user initialization of the transmit buffer in memory. The memory address of the transmit buffer is the same as the address shown in the DEMO RAM AT XXXX banner that is written when the demo software is entered from TIBUG. See Figure C-1. This transmit buffer can be initialized by either the "P" or "N" commands in the demo software or by the TIBUG "M" command. This command is useful for sending a spicific binary sequence.

C.6.12 Receive Record, Place in Buffer (B)

The B receive command will receive one record and return to the command scanner. The record is returned in the receive buffer which has a memory address equal to the transmit buffer memory address plus $079F_{16}$. The "B" character must be typed before the message is sent from another station to this station, but the receive routine will time out in 25 seconds. The received data can be inspected only by Quitting "Q" and using the TIBUG "M" command. This command is useful in inspecting specific binary data.

C.7 ERROR MESSAGES

There are several possible error messages generated by the demo software.

C.7.1 NO RAM EXISTS, CANNOT USE DEMO

This error can occur immediately after the demonstration software is entered from TIBUG. This error identifies an error in the memory map configuration in that the demonstration software could not find a 4 K block of memory to use. If this error occurs the user should turn the power off, extract the CPU

and memory boards, and recheck the jumpers and switch configurations. This error does not indicate any problem with the ICM board.

C.7.2 BAD EPROM, CANNOT USE DEMO

This error can also occur immediately after the demonstration software is entered from TIBUG. This error indicates the contents of the demonstration software EPROM's is incorrect. The contents of the EPROM's are verified by performing a checksum on the contents of the EPROM. When this occurs the user should not use the demonstration software because the results will be unpredictable.

C.7.3 0 THROUGH 3F AND FF ARE VALID

During the initialize routine an out-of-range address for the other station causes this message to be printed and the address querry is repeated.

C.7.4 ADDRESS SHOULD NOT BE THE SAME AS THIS BOARD

During initialization an address of another station that is keyed in that matches the address of this station will cause this message to be printed and the user will again be prompted for the address. A default address will not cause this message.

C.7.5 MAKE 0 < LENGTH < >39F OR >79F IF EXPANSION RAM IS INSTALLED

During initialization if a message length less than 1 byte or greater than $079F_{16}$ bytes is entered this message is printed and the user is again prompted for the message length. The ICM can buffer only $3A0_{16}$ bytes of data, one byte address and $39F_{16}$ bytes of message. If expansion RAM is installed on the ICM, up to $79F_{16}$ bytes of message can be buffered.

C.7.6 TRANSMIT AND RECEIVE DATA DOES NOT COMPARE

During the automatic send-receive test, the record sent by the primary is compared to the record it later receives back from the secondary station. If the records do not match, this message is printed and the error counter is incremented before the pass count banner is displayed.

C.7.7 25 SECOND TIMEOUT, TEST ABORTED

If any message transfer from processor-to-ICM or from ICM-to-processor takes longer than 25 seconds, it is assumed that something has failed and the test will be aborted. Also, if any receive mode is maintained longer than 25 seconds, it is assumed that no message is going to be sent and the test is again aborted.

C.7.8 /308 INTERRUPT MASK BIT DOES NOT RESPOND
CHECK THE CRU BASE ADDRESS

Whenever the command scanner is entered, the interrupt mask bit is toggled and the interrupt mask status bit is checked. The message could indicate that the CRU base address specified durring initialization does not match the CRU base address switches, or it could mean a hardware failure on the ICM.

The Engineering Staff of
TEXAS INSTRUMENTS INCORPORATED
Semiconductor Group

TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER DATA MANUAL

DECEMBER 1978

TEXAS INSTRUMENTS
INCORPORATED

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

TMS 9903

Synchronous Communications Controller

1. INTRODUCTION

1.1 DESCRIPTION

The TMS 9903 Synchronous Communications Controller (SCC) is a 20 pin peripheral device for the Texas Instruments TMS 9900 family of microprocessors. The TMS 9903 is TTL compatible on all inputs and outputs, including the power supply (+5V) and single phase clock. The SCC provides an interface between the microprocessor and a serial synchronous or asynchronous channel, performing data serialization and deserialization, facilitating microprocessor control of the communications channel. The TMS 9903 is fabricated using N-channel, silicon gate, MOS technology.

1.2 KEY FEATURES

- DC to 250 kilobits per second (kb/s) data rate, half or full duplex
- Dynamic character length selection
- Multiple line protocol capabilities: SDLC, Bi-Sync, HDLC, ADCCP, SNAP, or Asynchronous
- Programmable cyclic-redundancy-check (CRC) generation and detection
- Interface to unclocked or NRZI data
- Programmable sync registers
- Interval timer with resolution from 64-16,320 microseconds (μs)
- Automatic zero insert and delete for SDLC, HDLC
- Fully TTL-compatible, including single +5V power supply and clock
- Standard 20-pin plastic or ceramic package

1.3 TYPICAL APPLICATION

Figure 1 shows a general block diagram of a TMS 9900 based system incorporating a TMS 9903 SCC; Figure 2 is a similar diagram depicting a TMS 9980A or TMS 9981 based system. Following is an introductory discussion of the 9900 based application. Subsequent sections of this Data Manual detail all aspects of TMS 9903 usage.

The TMS 9903 interfaces with the CPU through the *communications register unit* (CRU). The CRU interface consists of five address select lines (S0-S4) chip enable ($\overline{CE}$), and three CRU lines (CRUIN, CRUOUT, CRUCLK). An additional input to the CPU is the SCC interrupt line ($\overline{INT}$). The TMS 9903 occupies 32 bits of CRU space; each of the 32 bits are selected individually by processor address lines A10-A14 which are connected to SCC select lines S0-S4, respectively. Chip enable ($\overline{CE}$) is generated by decoding address lines A0-A9 on CRU cycles. Under certain conditions the TMS 9903 causes interrupts, the SCC $\overline{INT}$ line is sent to the TMS 9901 for prioritization and encoding.

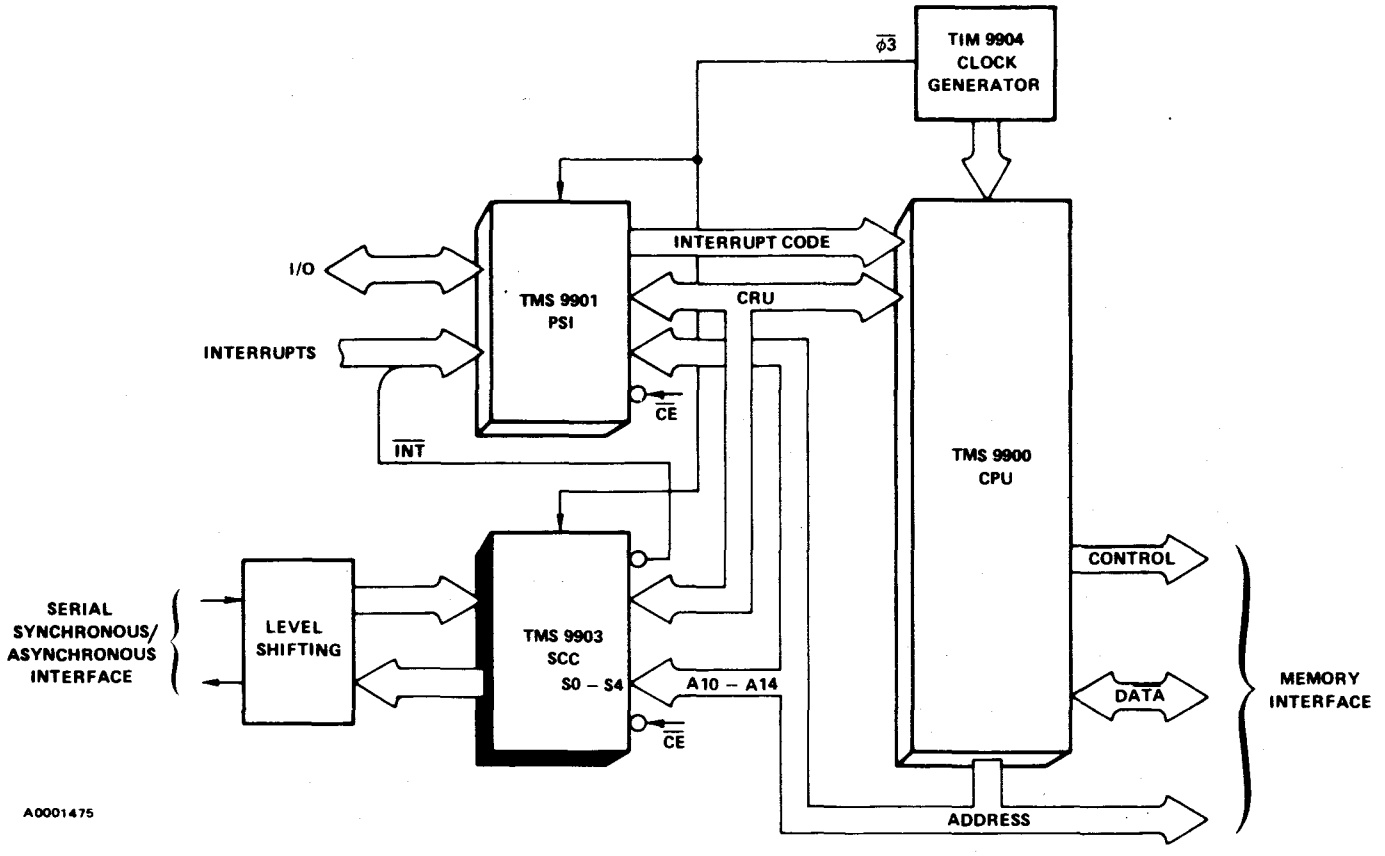


FIGURE 1. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER IN A TMS 9900 SYSTEM

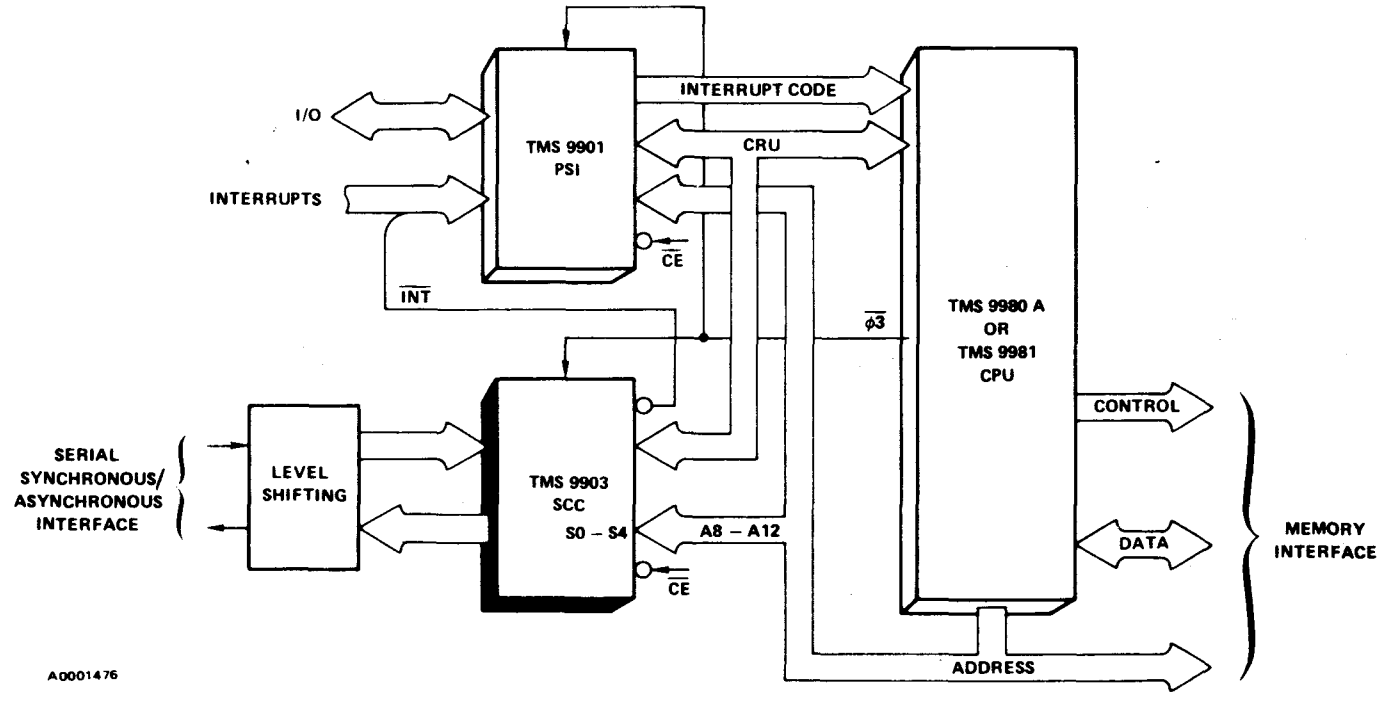


FIGURE 2. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER IN A TMS 9980 A, 9981 SYSTEM

The SCC interfaces to the synchronous communications channel on seven lines: request to send ($\overline{RTS}$), data set ready ($\overline{DSR}$), clear to send ($\overline{CTS}$), serial transmit data (XOUT), serial receive data (RIN), receiver clock (SCR), and transmitter clock ($\overline{SCT}$). The request to send ($\overline{RTS}$) goes active (LOW) whenever the transmitter is activated. However, before data transmission begins, the clear to send ($\overline{CTS}$) input must be active. The data set ready ($\overline{DSR}$) input does not affect the receiver or transmitter. When $\overline{DSR}$, $\overline{CTS}$, or automatic request-to-send (RTSAUT) changes level, an interrupt is generated, if enabled.

The TMS 9903 is capable of six different modes of operation, including two asynchronous modes. Standard synchronous protocols such as SDLC, HDLC, Bi-Sync, and ADCCP can be directly implemented on the SCC.

2. ARCHITECTURE

The TMS 9903 synchronous communications controller (SCC) is designed to provide a low cost, serial, synchronous or asynchronous interface to the 9900 family of microprocessors. A block diagram for the TMS 9903 is shown in Figure 3. The SCC has five main subsections: CRU interface, transmitter section, receiver section, interval timer, and interrupt section.

2.1 CRU INTERFACE

The communications register unit (CRU) is the means by which the CPU communicates with the TMS 9903 SCC. The SCC occupies 32 bits of CRU read and write space. Figure 4 illustrates the CRU interface between a TMS 9903 and a TMS 9900 CPU; Figure 5 illustrates the CRU interface for a TMS 9980A or TMS 9981 CPU. The CRU lines are tied directly to each other as shown in Figures 4 and 5. The least significant bits of the address bus are connected to the select lines. In a TMS 9900 CPU system A14-A10 are connected to S4-S0 respectively. The most significant address bits are decoded to select the TMS 9903 via the chip enable ($\overline{CE}$) signal. When $\overline{CE}$ is inactive (HIGH), the SCC CRU interface is disabled.

NOTE

When $\overline{CE}$ is inactive (high) the 9903 places the CRUIN line in its high impedence state and disables CRUCLK from coming on chip. Thus CRUIN can be used as an OR tied bus. $\overline{CE}$ being inactive will not disable the select lines from coming on chip, although no device action is taken.

For those unfamiliar with the CRU concept, the following is a discussion of how to build a CRU interface. The CRU is a bit addressable (4096 bits), synchronous, serial interface over which a single instruction can transfer between one and 16 bits serially. Each one of the 4096 bits of the CRU space has a unique address and can be read and written to. During multibit CRU transfers, the CRU address is incremented at the beginning of each CRU cycle to point to the next consecutive CRU bit.

When a 9900 CPU executes a CRU instruction, the processor uses the contents of workspace register 12 as a base address. (Refer to the 9900 Microprocessor Data Manual for a complete discussion on how CRU addresses are derived.) The CRU address is brought out on the 15-bit address bus; this means that the least significant bit of R12 is not brought out of the CPU. During CRU cycles, the memory control lines ($\overline{MEMEN}$, $\overline{WE}$, and DBIN) are all inactive; $\overline{MEMEN}$ being inactive (HIGH) indicates the address is not a memory address and therefore is a CRU address or external instruction code. Also, when $\overline{MEMEN}$ is inactive (HIGH) and a valid address is present, address bits A0-A2 must all be zero to constitute a valid CRU address; if address bits A0-A2 are other than all zeros, they are indicating an external instruction code. In summary, address bits A3-A14 contain the CRU address to be decoded, address bits A0-A2 must be zero and $\overline{MEMEN}$ must be inactive (HIGH) to indicate a CRU cycle.

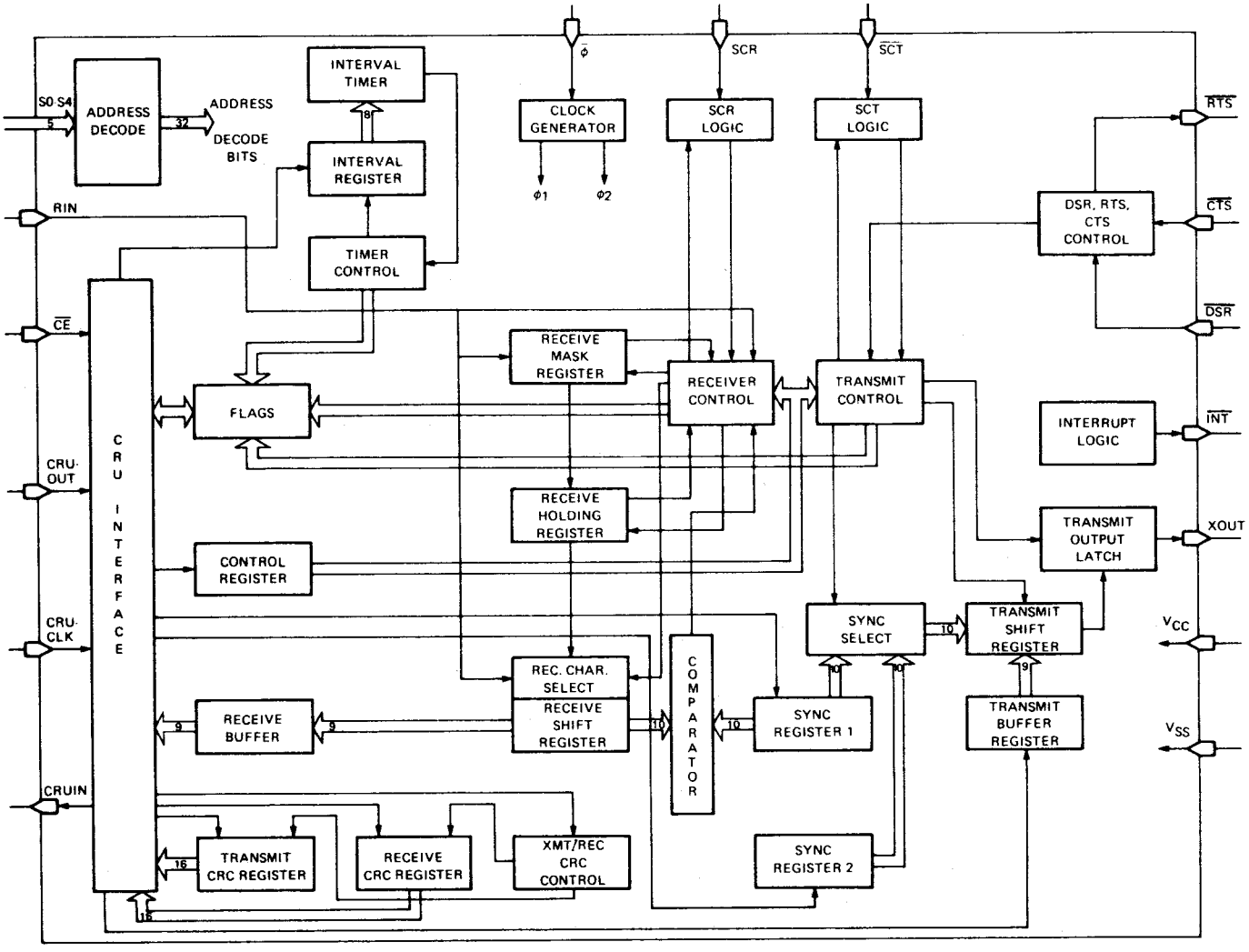


FIGURE 3. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER BLOCK DIAGRAM

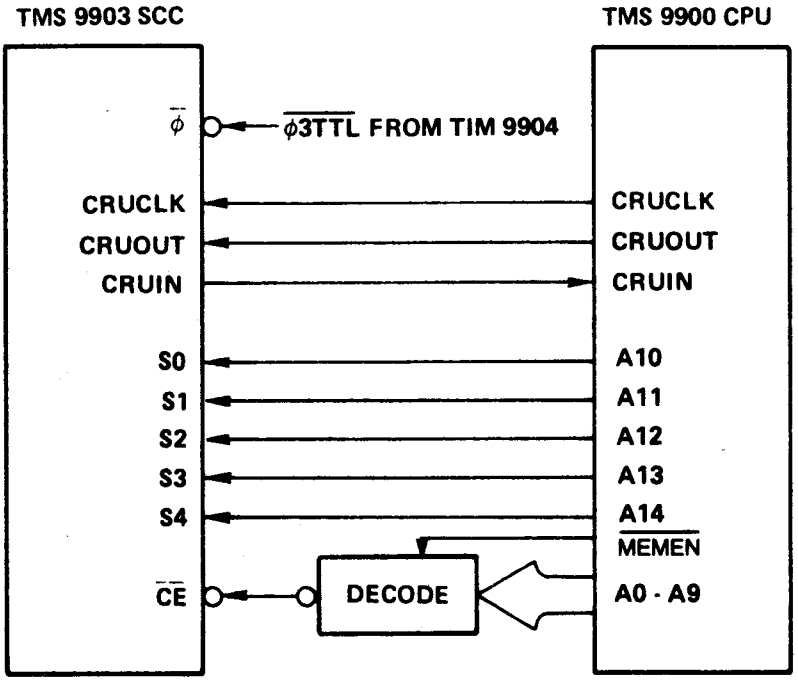


FIGURE 4. TMS 9903 CONTROL SIGNALS (TMS 9900 SYSTEM)

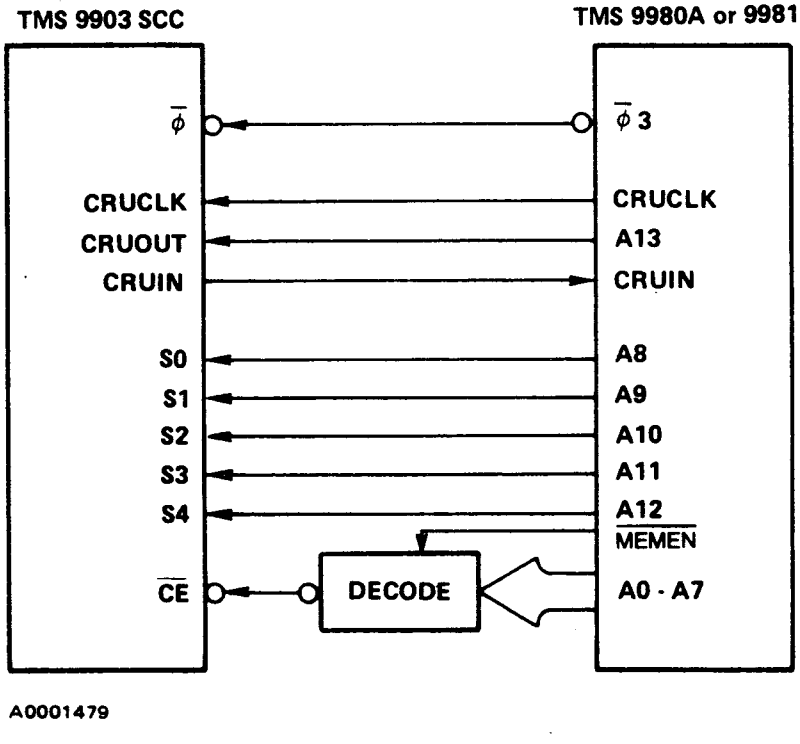


FIGURE 5. TMS 9903 CONTROL SIGNALS (TMS 9980A or 9981 SYSTEM)

2.1.1 CPU Output for CRU

The TMS 9903 SCC occupies 32 bits of output CRU space, of which all are used. These bits are employed by the CPU to communicate command and control information to the TMS 9903. Table 1 shows the mapping between CRU select (S lines) and SCC functions by operational mode; modes 4 and 7 are not implemented. Each CRU selectable output bit on the TMS 9903 is described in detail following Table 1.

TABLE 1. TMS 9903 OUTPUT SELECT BIT ASSIGNMENTS

SELECT	NAME	MODE						DESCRIPTION
		0	1	2	3	5	6	
31	RESET	X	X	X	X	X	X	Reset Device
30	CLRXMT (1)	X	X	X	X	X	X	Clear Transmitter
	CLRRCV (0)	X	X	X	X	X	X	Clear Receiver
29	CLXCRC (1)	X	X	X	X	X	X	Clear Transmitter CRC Register
	CLRCRC (0)	X	X	X	X	X	X	Clear Receiver CRC Register
28	—	X		X		X	X	Not Used
	XZINH		X					Transmitter Zero Insertion Inhibit
	RSYNDL				X			Received Sync Character Delete
27	LDSYN2	X	X	X	X			Load Sync Character Register 2
	—					X	X	Not Used
26	—	X				X	X	Not Used
	RHRRD		X					Receiver Holding Register Read
	LDSYN1			X	X			Load Sync Character Register 1
25	LXBC	X	X	X	X	X	X	Load Transmitter Buffer and Transmitter CRC Register
24	LXCRC	X	X	X	X	X	X	Load Transmitter CRC Register
23	XPRNT	X	X		X			Transparent
	—			X				Not Used
	BRKON					X	X	Break On
22	XAIENB	X	X					Transmitter Abort Interrupt Enable
	—			X	X	X	X	Not Used
21	DSCENB	X	X	X	X	X	X	Data Set Status Change Interrupt Enable
20	TIMENB	X	X	X	X	X	X	Timer Interrupt Enable
19	XBIENB	X	X	X	X	X	X	Transmitter Buffer Register Empty Interrupt Enable
18	RIENB	X	X	X	X	X	X	Receiver Interrupt Enable
17	RTS	X	X	X	X	X	X	Request To Send
16	XMTON	X	X	X	X	X	X	Transmitter On
15	TSTMD	X	X	X	X	X	X	Test Mode
14	LDCTRL	X	X	X	X	X	X	Load Control Register
13	LDIR	X	X	X	X	X	X	Load Interval Register
12	LRCRC	X	X	X	X	X	X	Load Receiver CRC Register
11-0	DATA							Data To Selected Register

Bit 31

All modes (RESET) — **Reset.** Writing a one or zero to bit 31 causes the device to reset, disabling all interrupts, initializing all controllers, and resetting all flags except LDCTRL and XBRE which are set.

Bit 30

All modes (CLRXMT) — **Clear Transmitter.** Writing a one to bit 30 initializes the transmitter and clears transmit interrupts.

(CLRRCV) — **Clear Receiver.** Writing a zero to bit 30 initializes the receiver and clears all receive interrupts.

Bit 29

All modes (CLXCRC) — **Clear Transmitter CRC Register (XCRC).** Writing a one to bit 29 in all modes clears the XCRC register to all zeros.

(CLRCRC) — **Clear Receiver CRC Register (RCRC).** Writing a zero to bit 29 in all modes clears the RCRC register to all zeros.

Bit 28

Modes 0, 2, 5, 6 — Not Used.

Modes 1 (XZINH) — **Transmitter Zero Insertion Inhibit.** Writing a one to bit 28 in mode 1 causes the contents of the transmitter buffer register (XBR) to be transmitted without the insertion of a zero after five consecutive ones. Writing a zero to bit 28 in mode 1 causes the transmitter to insert a zero after five consecutive ones are transmitted.

Mode 3 (RSYNDL) — **Received Sync Character Delete.** Writing a one to bit 28 in mode 3 causes received characters which are identical to the contents of sync character register 1 (SYNC1) to be ignored. This function is disabled when XPRNT (bit 23) is set. Writing a zero to bit 28 in mode 3 causes RSYNDL (Receiver sync character delete) to be reset.

Bit 27

Modes 0, 1, 2, 3 (LDSYN2) — **Load Sync Character Register 2.** Writing a one to bit 27 in mode 0, 1, 2, or 3 enables loading of sync character register 2 (SYNC2) from output select bit addresses 0-9. Writing a zero to bit 27 in mode 0, 1, 2, 3 resets LDSYN2.

Modes 5, 6 — Not Used.

Bit 26

Mode 1 (RHRRD) — **Receiver Holding Register Read.** Writing a one to bit 26 in mode 1 enables reading of the receiver-holding register (RHR) contents at input bit addresses 0-15. Writing a zero to bit 26 in mode 1 resets RHRRD, RHRL (receive holding register loaded), RHROV (receive holding register overrun), and RZER (receive zero error).

Modes 2, 3 (LDSYN1) — **Load Sync Character Register 1.** Writing a one to bit 26 in mode 2 or 3 enables loading of sync character register 1 (SYNC1) from output select bit addresses 0-9. Writing a zero to bit 26 in mode 2 or 3 resets LDSYN1.

Bit 25

All modes (LXBC) — **Load Transmit Buffer and CRC Register.** Writing a one to bit 25 in all modes enables loading of XBR (transmit buffer register) and XCRC (transmit CRC register) from output select bit addresses 0-8, and enables reading of XCRC at input select bit addresses 0-15. Writing a zero to bit 25 in all modes resets LXBC and XBRE (transmit buffer register empty).

Bit 24

All modes (LXCRC) — **Load Transmit CRC Register.** Writing a one to bit 24 in all modes enables loading the XCRC register from output select bit addresses 0-9, and enables reading XCRC at input select bit addresses 0-15. Writing a zero to bit 24 in all modes resets LXCRC.

Bit 23

Mode 0 (XPRNT) — **Transparent.** Writing a one to bit 23 in mode 0 causes the contents of SYNC2 to be transmitted whenever no data is available and the transmitter is active. Writing a zero to bit 23 in mode 0 causes the transmitter abort signal (XABRT) to set and transmitter operation to be suspended when no data is available and the transmitter is active.

Mode 1 (XPRNT) — **Transparent.** Writing a one to bit 23 in mode 1 causes the contents of SYNC2 to be transmitted without zero insertion when no data is available and the transmitter is active. Writing a zero to bit 23 in mode 1 causes XABRT to be set and transmit operations to be suspended when no data is available.

Mode 2 — Not Used.

Mode 3 (XPRNT) — **Transparent.** Writing a one to bit 23 in mode 3 causes the fill sequence of (contents of SYNC2) followed by (contents of SYNC1) to be transmitted when no data is available. Writing a zero to bit 23 in mode 3 causes the fill sequence of (contents of SYNC1) followed by (contents of SYNC1) to be transmitted when no data is available.

Modes 5 and 6 (BRKON) — **Break ON.** Writing a one to bit 23 in mode 5 or 6 causes the output to go to a constant zero level when no data is available and the transmitter is active. Writing a zero to bit 23 in mode 5 and 6 causes BRKON to be reset. The transmit buffer register should not be loaded during transmission of a break.

INTERRUPT ENABLE FLAGS

INTERRUPT ENABLE	SELECT BIT	INTERRUPT FLAG	INTERRUPT NAME	DESCRIPTION
XAIENB	22	XABRT	XAINT	Transmitter Abort
DSCENB	21	DSCH	DSCINT	Data Set Status Change ($\overline{CTS}$, $\overline{DSR}$, RTSAUT)
TIMENB	20	TIMELP	TIMINT	Timer Elapsed
XBIENB	19	XBRE	XBINT	Transmitter Buffer Register Empty
RIENB	18	RBRL	RINT	Receiver Buffer Register Loaded
RIENB	18	RHRL	RINT	Receiver Holding Register Loaded
RIENB	18	RABRT	RINT	Receiver Abort

Refer to Section 2.6

Bit 22

Modes 0 and 1 (XAIENB) — **Transmitter Abort Interrupt Enable.** Writing a one to bit 22 in mode 0 or 1 resets XABRT (transmitter abort) and enables XABRT interrupts. Writing a zero to bit 22 in mode 0 or 1 resets XABRT and disables XABRT interrupts.

Modes 2, 3, 5 and 6 — Not Used.

Bit 21

All modes (DSCENB) — **Data Set Status Change Interrupt Enable.** Writing a one to bit 21 in all modes resets DSCH (data set status change) and enables DSCH interrupts. Writing a zero to bit 21 in all modes resets DSCH and disables DSCH interrupts.

Bit 20

All modes (TIMENB) — **Timer Interrupt Enable.** Writing a one to bit 20 in all modes resets TIMELP (timer elapsed) and TIMERR (timer error) and enables TIMELP interrupts. Writing a zero to bit 20 in all modes resets TIMELP and TIMERR and disables TIMELP interrupts.

Bit 19

All modes (XBIENB) — **Transmitter Buffer Register Empty Interrupt Enable.** Writing a one to bit 19 in all modes enables XBRE interrupts. Writing a zero to bit 19 in all modes disables XBRE interrupts.

Bit 18

Modes 0, 2, 3, 5, 6 (RIENB) — **Receiver Interrupt Enable.** Writing a one to bit 18 in mode 0, 2, 3, 5, or 6 resets RBRL (receiver buffer register loaded) and ROVER, (receiver overrun), and enables RBRL interrupts. Writing a zero to bit 18 in mode 0, 2, 3, 5, 6 resets RBRL and ROVER, and disables RBRL interrupts.

Mode 1 (RIENB) — **Receiver Interrupt Enable.** Writing a one to bit 18 in mode 1 resets RBRL, RFLDT, ROVER, and RABRT (receiver abort), and enables RBRL, RABRT, and RHRL (receiver holding register loaded) interrupts. Writing a zero to bit 18 in mode 1 resets RBRL, RFLDT, ROVER, and RABRT, and disables RBRL, RABRT, and RHRL interrupts.

Bit 17

All modes (RTS) — **Request to Send.** Writing a one to bit 17 in all modes resets the $\overline{\text{RTS}}$ output (LOW) and disables automatic control of $\overline{\text{RTS}}$ by the internal RTSAUT (automatic RTS control) signal. Writing a zero to bit 17 in all modes sets the $\overline{\text{RTS}}$ output HIGH and disables automatic control by RTSAUT.

Bit 16

All modes (XMTON) — **Transmitter On.** Writing a one to bit 16 in all modes enables data transmission. Writing a zero to bit 16 in all modes disables data transmission when no data is available.

Bit 15

All modes (TSTMD) — **Test Mode.** Writing a one to bit 15 in all modes causes the timer to decrement at 32 times the normal rate, and internally connects XOUT to RIN, RTSAUT to CTS, and SCR to $\overline{\text{SCT}}$. $\overline{\text{SCT}}$ is internally generated at the frequency to which TIMELP is set. Writing a zero to bit 15 in all modes resets TSTMD and enables normal device operation. The test mode should not be used in a loop configuration of mode 1; test mode is useful for testing and inspection purposes.

TABLE 2. REGISTER LOAD CONTROL FLAGS

FLAG*	CRUOUT BIT ADDRESS	REGISTER LOADED	BITS/REGISTER
LDSYN2	27	Sync Register 2 (SYNC2)	10
LDSYN1	26	Sync Register 1 (SYNC1)	10
LXBC	25	Xmt CRC Register (XCRC) and Xmt Buffer Reg. (XBR)	9
LXCRC	24	XCRC	10
LDCTRL	14	Control Register (CTRL)	12
LDIR	13	Interval Register	8
LRCRC	12	Receive CRC Register (RCRC)	10
None	—	XBR	9

*It is recommended that no more than one register load control flag be set at any one time.

Bit 14
All modes (LDCTRL) — **Load Control Register.** Writing a one to bit 14 in all modes enables the loading of the control register from output select bit addresses 0-11. Writing a zero to bit 14 in all modes resets LDCTRL. When a bit is written to select bit 11 (when loading the control register), the LDCTRL flag is automatically reset.

Bit 13
All modes (LDIR) — **Load Interval Register.** Writing a one to bit 13 in all modes enables the loading of the interval register from output select bits 0-7. Writing a zero to bit 13 in all modes resets LDIR and causes the contents of the interval register to be loaded into the interval timer.

Bit 12
All modes (LRCRC) — **Load Receiver CRC Register.** Writing a one to bit 12 in all modes enables the loading of the receiver CRC register from output select bit addresses 0-9, and enables reading the RCRC (receiver CRC register) on input select bits 0-15. Writing a zero to select bit 12 in all modes resets LRCRC.

2.1.2 Control and Data Registers

Loading the internal control and data registers is controlled by one of the single bit control function flags described in Section 2.1.1 and summarized in Table 2. The registers must be carefully loaded to ensure that no more than one flag is set at a time. Unlike the TMS 9902, when the MSB of a register is loaded, the load flag is not automatically reset except for the control register which is the only register which will automatically reset the load flag when the MSB of the register is written to.

The TMS 9903 SCC is capable of performing dynamic character length operations. The receiver character length is set by bits 2-0 of the control register. Transmitted character and sync character registers are maintained internally to determine the character length. The length of the character to be transmitted is determined by the number of bits loaded into the transmitter buffer register before the transmitter buffer register empty flag is reset. Similarly, the character length of the two sync registers is determined by the number of bits loaded into the most recently loaded SYNC character. Thus, for transmission purposes the length of the two SYNC characters is the same. NOTE: When the receiver is comparing received data to SYNC1, only the number of bits selected as the received character length are compared [i.e., RSCL (2-0) plus parity, if enabled].

2.1.2.1 Control Register

The control register is loaded to select the mode, configuration, CRC polynomial, received character length, data rate clock, and internal device clock rates of the TMS 9903. Table 3 shows the bit address assignments for the control register.

TABLE 3. CONTROL REGISTER BIT ADDRESS ASSIGNMENTS

ADDRESS (S0-S4)	NAME	DESCRIPTION
11	DRCK32	32X Data Rate Clock
10	CRC1	CRC Polynomial Select
9	CRC0	
8	MDSL2	Mode Select
7	MDSL1	
6	MDSL0	
5	CSL1	Configuration Select
4	CSL0	
3	CLK4M	4X System Clock Select
2	RSCL2	Receive Character Length Select
1	RSCL1	
0	RSCL0	

11	10	9	8	7	6	5	4	3	2	1	0
DRCK32	CRC1	CRC0	MDSL2	MDSL1	MDSL0	CSL1	CSL0	CLK4M	RSCL2	RSCL1	RSCL0

MSB LSB

CONTROL REGISTER BIT ADDRESS

Bit 11

Modes 0, 1, 2, 3 (DRCK32)— **32X Data Rate Clock.** Setting control bit 11 to one in mode 0, 1, 2, or 3 sets the $\overline{SCT}$ frequency at 32 times the transmit-data rate and the SCR frequency at 32 times the receive-data rate. SCR is set to resync on every transition of RIN. Also, if bit 11 is a one, zero-complementing NRZI data encoding is used (to send a one, the signal remains in the same state; to send a zero, the signal changes state). Setting bit 11 to zero in mode 0, 1, 2, or 3 causes the receive data to be sampled on every zero-to-one transition of SCR, and the transmit data to be shifted out on the one-to-zero transition of $\overline{SCT}$. DRCK32 should always be reset when in a loop configuration of mode 1.

Modes 5, 6 (DRCK32)— **32X Data Rate Clock.** Setting control bit 11 to one in mode 5 or 6 sets the $\overline{SCT}$ frequency to 32 times the transmit data rate, and the SCR frequency to 32 times the receive data rate. SCR is resynced on every start bit received. Setting control bit 11 to zero in mode 5 or 6 causes receive data to be sampled on the zero-to-one transition of SCR, and transmit data to be shifted out on the one-to-zero transition of $\overline{SCT}$.

All modes — Setting control bit 11 to a one or zero resets LDCTRL (load control register). The control register is the only register that resets its load flag in this fashion.

Bits 10 and 9

All modes (CRC1 and CRC0)— **CRC Polynomial Select.** The polynomial used in the generation of the transmit and receive CRC's is selected by bits 10 and 9 of the control register, as shown below.

CRC POLYNOMIAL BIT SELECT

CRC	CRC1	CRC0	NAME	POLYNOMIAL
0	0	0	CRC-16	$X^{16}+X^{15}+X^2+1$
1	0	1	CRCC-12*	$X^{12}+X^{11}+X^3+X^2+X+1$
2	1	0	REV. CRCC-16	$X^{16}+X^{14}+X+1$
3	1	1	CRC-CCITT	$X^{16}+X^{12}+X^5+1$

*NOTE: When using CRCC-12, the four most-significant bits of the CRC register will contain data that must be masked to assure validity of CRC comparisons.

Bits 8, 7 and 6
All modes (MDSL2, MDSL1, MDSL0)

Mode Select. The mode of operation for the transmitter and receiver is selected by bits 8, 7, and 6 of the control register as shown below.

TRANSMIT/RECEIVE MODE SELECT

MODE	MDSL2	MDSL1	MDSL0	EXAMPLE PROTOCOL	SYNC CHARACTER	FILL-CHARACTER
0	0	0	0	GENERAL	NONE	(SYNC2) or NONE
1	0	0	1	SDLC	$7E_{16}$	(SYNC2) or NONE
2	0	1	0	GENERAL	(SYNC1)	(SYNC2)
3	0	1	1	BI-SYNC	(SYNC1-SYNC1)	(SYNC1-SYNC1) or (SYNC2-SYNC1)
4	1	0	0	NOT USED		
5	1	0	1	ASYNCHRONOUS OPERATION WITH TWO STOP BITS		
6	1	1	0	ASYNCHRONOUS OPERATION WITH ONE STOP BIT		
7	1	1	1	NOT USED		

Bits 5 and 4
All modes (CSL1, CSL0)—

Configuration Select. The configuration of the transmitter and receiver within each mode is set by bits 5 and 4 of the control register, as shown below. CSL1 is forced to zero on RESET.

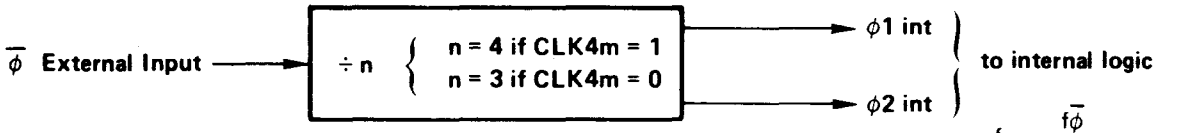
TRANSMITTER/RECEIVER CONFIGURATION BIT SELECT

CONFIGURATION	CSL1	CSL0	MODE 0 1 2 3 5 6	DESCRIPTION
0	0	0	X X X X X X	No Parity Generation or Detection SDLC Normal (Non-Loop)
1	0	1	X X X X X X	No Parity Generation or Detection SDLC Loop Master
2	1	0	X X X X X X	Even Parity Generated on Transmission and Detected on Reception SDLC Loop Slave — Pending Synchronization
3	1	1	X X X X X X	Odd Parity Generated on Transmission and Detected on Reception SDLC Loop Slave - Active

Bit 3

All modes (CLK4M) —

Input Divide Select. The $\overline{\phi}$ input to the TMS 9903 SCC is used to generate internal dynamic logic clocking and to establish the time base for the interval timer. The $\overline{\phi}$ input is internally divided by either 3 or 4 to generate the two phase internal clocks required for MOS logic, and to establish the basic internal operating frequency (f_{int}) and internal clock period (t_{int}). When bit 3 of the control register is set to a logic one (CLK4M = 1), $\overline{\phi}$ is internally divided by 4, and when CLK4M = 0, $\overline{\phi}$ is divided by 3. For example, when $f\overline{\phi}$ = 3 MHz, (as in a standard 3 MHz TMS 9900 system) and CLK4M = 0, $\overline{\phi}$ is internally divided by 3 to generate an internal clock period t_{int} of 1 μs. The figure below shows the operation of the internal clock divider circuitry. The internal clock frequency should be no greater than 1.1 MHz; thus, when $f\overline{\phi} > 3.3$ MHz, CLK4M should be set to a logic one.



$$f_{int} = \frac{f\overline{\phi}}{n}$$

$$t_{int} = \frac{1}{f_{int}} = \frac{n}{f\overline{\phi}}$$

A0001471

INTERNAL CLOCK DIVIDER CIRCUITRY

Bits 2, 1, and 0

All modes (RSCL2, RSCL1, and RSCL0)

Received Character Length Select. The number of data bits in each received character is determined by bits 2, 1, and 0 of the control register, as shown below.

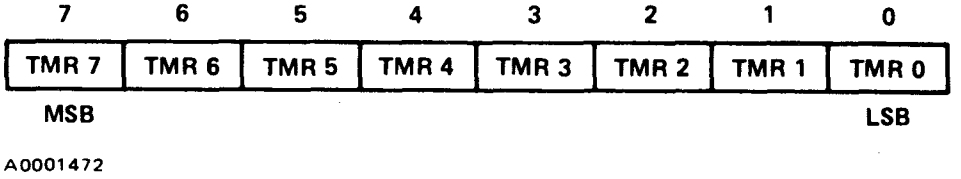
RECEIVE CHARACTER LENGTH SELECTION

RSCL2	RSCL1	RSCL0	BITS/CHAR.
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	0	0	9

Note: $f\phi$ denotes frequency of ϕ.

2.1.2.2 *Interval Register*

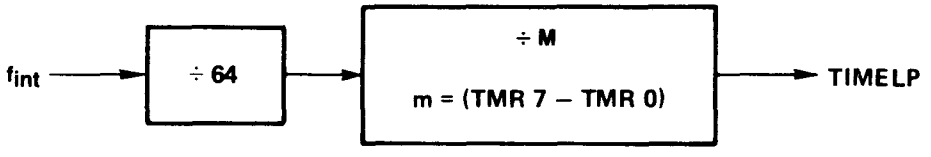
The interval register is enabled for loading whenever LDIR = 1. The interval register is used to select the rate at which timer interrupts are generated by the SCC interval timer. The figure below shows the bit assignments for the interval register when enabled for loading.



INTERVAL REGISTER BIT ADDRESS

The figure below illustrates the establishment of the interval for the timer. For example, if the interval register is loaded with a value of $40_{16}(64_{10})$ with t_{int} = 1 μs, the interval at which the timer decrements to zero and interrupts the CPU is

$$\begin{aligned} t_{interval} &= t_{int} \times 64 \times M \\ &= (1\ \mu s) \times 64 \times 64 \\ &= 4.096\ ms \end{aligned}$$



TIME INTERVAL SELECTION

2.1.2.3 *Receiver CRC Register*

The receiver CRC register is enabled for loading when LRCRC = 1. The receiver CRC register is used to verify data integrity in the synchronous communication channel. When LRCRC is set, output to select bits 0-9 updates the contents of the receiver CRC register according to the CRC polynomial selected by the control register. Also, when LRCRC is set, the receiver CRC register can be read on CRU input select bits 0-15. When read, the MSB of the register is read first, and the LSB is read last. The receiver CRC register block diagram is shown in Figure 6.

2.1.2.4 *Transmitter CRC Register*

The transmitter CRC register is enabled for loading when either LXCRC = 1 (load transmitter CRC register) or LXBC = 1 (load transmitter buffer register and transmitter CRC register). When either LXBC or LXCRC is set, output to bit addresses 0-9 updates the contents of the transmitter CRC register according to the CRC polynomial selected by the control register. When set, the LXBC or LXCRC flag selects the transmitter CRC register contents to be read by the CPU at input select bits 0-15. LXBC and LXCRC flags are reset by a command from the CPU.

Operation of the transmitter CRC register is analogous to that of the receiver CRC register shown in Figure 6.

2.1.2.5 *Sync Character Register 1*

Sync character register 1 is enabled for loading when LDSYN1 = 1. The sync character register 1 is used for synchronization and as a fill sequence for the transmitter. When LDSYN1 is set, output to select bits 0-9 is loaded into sync character register 1. The LDSYN1 flag is reset by a command from the CPU.

RCRCS is the CRC shift clock and is a function of CRUCLK; it is used to shift the CRC polynomial in as data is shifted in the CRUOUT line by CRUCLK.

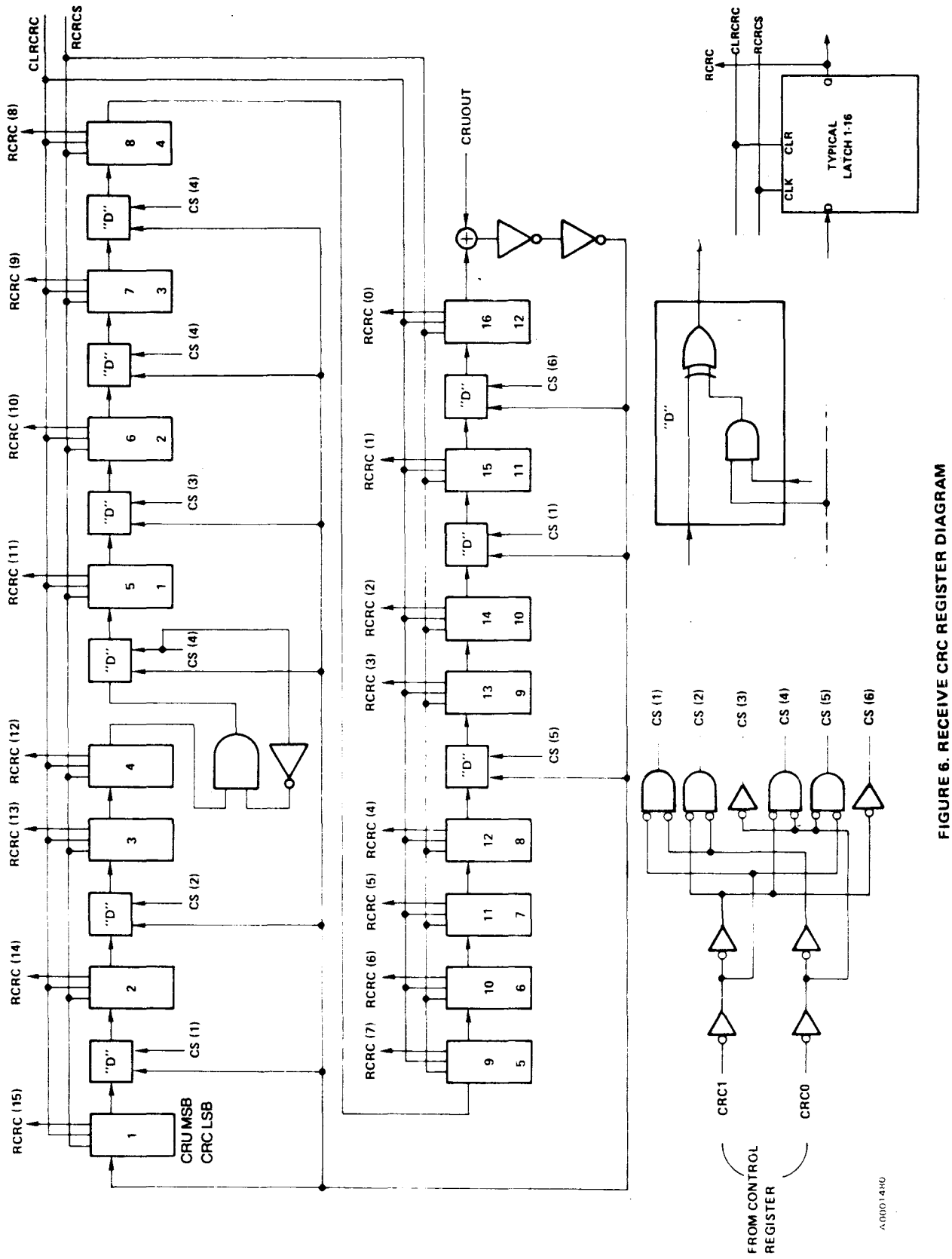


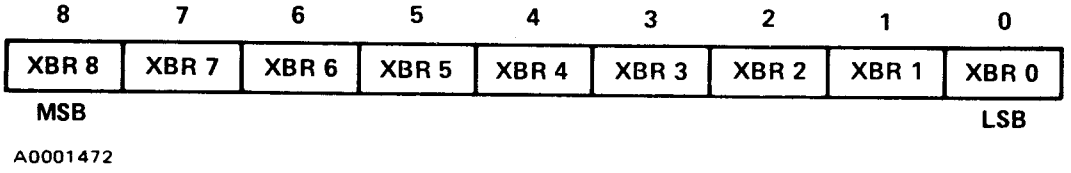
FIGURE 6. RECEIVE CRC REGISTER DIAGRAM

2.1.2.6 *Sync Character Register 2*

Sync character register 2 is enabled for loading whenever LDSYN2 = 1. The contents of sync character register 2 are used for a fill sequence for the transmitter. When LDSYN2 is set, output to select bits 0-9 is loaded into sync character register 2. The LDSYN2 flag is reset by a command from the CPU.

2.1.2.7 *Transmitter Buffer Register*

Two conditions enable the transmitter buffer register for loading. If all flags are zero or if LXBC = 1, the transmitter buffer register is enabled for loading. The transmitter buffer is used for the storage of the next character to be transmitted. When the transmitter is active, the contents of the transmitter buffer register are transferred to the transmitter shift register when the previous character has been completely transmitted. When LXBC is set, the output to select bits 0-8 is loaded into the transmitter buffer register and simultaneously updates the contents of the transmitter CRC register, according to the CRC polynomial selected by the control register. Also, when LXBC is set, the transmitter CRC register contents are enabled for reading on input select bits 0-15. The LXBC flag is reset by a command from the CPU.



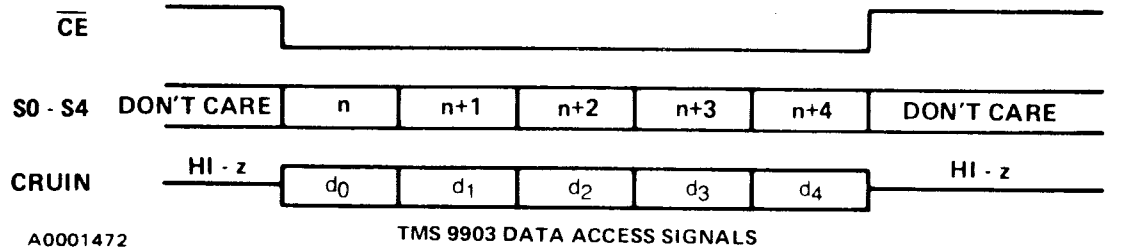
TRANSMIT BUFFER REGISTER BIT ADDRESSES

TABLE 4. CRU OUTPUT ADDRESS ASSIGNMENTS

ADDR	LDCTRL = 1	LDSYN1 = 1	LDSYN2 = 1	LDIR = 1	LRCRC = 1	LXCRC = 1	LXBC = 1		ADDR FLAGS=0
11	DRCK32								
10	CRC(1)								
9	CRC(0)	S1(9)	S2(9)		RCRC(9)	XCRC(9)			
8	MDSL(2)						XCRC(8)	XBR(8)	XBR(8)
7	MDSL(1)			IR(7)					
6	MDSL(0)								
5	CSL(1)								
4	CSL(0)								
3	CLK4M								
2	RSCL(2)								
1	RSCL(1)								
0	RSCL(0)	S1(0)	S2(0)	IR(0)	RCRC(0)	XCRC(0)	XCRC(0)	XBR(0)	XBR(0)

2.1.3 Status and Data Input to CPU

Status and data information is read from the SCC by the CPU on the CRUIN line whenever $\overline{CE}$ is active (LOW). The input bit is selected from one of 32 input select bits using the five select lines S0-S4. When $\overline{CE}$ is high, CRUIN is in its high-impedance state, permitting the CRUIN control line to be wire-ORed with other input devices. The following figure illustrates the relationships of the signals used to access data from the SCC. Table 5 describes the input select bit assignments of the SCC. Following Table 5 is a detailed discussion of each bit. Select bits 0-15 can be read as shown only when all load flags are reset.



TMS 9903 DATA ACCESS SIGNALS

TABLE 5. TMS 9903 INPUT BIT ADDRESS ASSIGNMENTS

ADDRESS	NAME	MODE 0	1	2	3	5	6	DESCRIPTION
31	INT	X	X	X	X	X	X	Interrupt
30	FLAG	X	X	X	X	X	X	Any Register Load Control Flag Set
29	DSCH	X	X	X	X	X	X	Data Set Status Change
28	CTS	X	X	X	X	X	X	Clear to Send
27	DSR	X	X	X	X	X	X	Data Set Ready
26	RTSAUT	X	X	X	X	X	X	Automatic Request to Send
25	TIMELP	X	X	X	X	X	X	Timer Elapsed
24	TIMERR	X	X	X	X	X	X	Timer Error
23	XABRT	X	X					Transmitter Abort
	—			X	X	X	X	Not Used
22	XBRE	X	X	X	X	X	X	Transmitter Buffer Register Empty
21	RBRL	X	X	X	X	X	X	Receiver Buffer Register Loaded
20	DSCINT	X	X	X	X	X	X	Data Set Status Change Interrupt
19	TIMINT	X	X	X	X	X	X	Timer Interrupt
18	XAINT	X	X					Transmitter Abort Interrupt
	—			X	X	X	X	Not Used (Always = 0)
17	XBINT	X	X	X	X	X	X	Transmitter Buffer Interrupt
16	RINT	X	X	X	X	X	X	Receiver Interrupt
15	RIN	X	X	X	X	X	X	Receiver Input
14	RABRT		X					Receiver Abort
	—	X		X	X			Not Used (Always = 0)
	RSBD					X	X	Receiver Start Bit Detect
13	RHRL		X					Receiver Holding Register Loaded
	—	X		X	X			Not Used (Always = 0)
	RFBD					X	X	Receiver Full Bit Detect
12	RHROV		X					Receiver Holding Register Overrun
	—	X		X	X			Not Used (Always = 0)
	RFER					X	X	Receiver Framing Error
11	ROVER	X	X	X	X	X	X	Receiver Overrun
10	RPER	X		X	X	X	X	Receiver Parity Error
	RZER		X					Receiver Zero Error
9	RCVERR	X		X	X	X	X	Receiver Error
	RFLDT		X					Receiver Flag Detect
8-0	RBR	X	X	X	X	X	X	Receiver Buffer Register (Received Data)

Bit 31
All modes (INT) — **Interrupt.** All modes INT = DSCINT + TIMINT + RBINT + XAINT + XBINT. The interrupt output control line ($\overline{INT}$) is active (LOW) when this status signal is a logic one.

Bit 30
All modes (FLAG) — **Register Load Control Flag Set.** In all modes FLAG = LDCTRL + LDSYN1 + LDSYN2 + LDIR + LRCRC + LXBC + LXCRC. Flag = 1 when any of the register load control flags is set.

Bit 29
All modes (DSCH) — **Data Set Status Change.** In all modes DSCH is set when the $\overline{DSR}$ or $\overline{CTS}$ inputs, or RTSAUT changes state. To ensure recognition of the state change, $\overline{DSR}$ or $\overline{CTS}$ must remain stable in its new state for a minimum of two internal clock cycles. DSCH is reset by an output to select bit 21 (DSCENB).

Bit 28
All modes (CTS) — **Clear To Send.** The CTS signal indicates the inverted status of the $\overline{CTS}$ device input in all modes.

Bit 27
All modes (DSR) — **Data Set Ready.** The DSR signal indicates the inverted status of the $\overline{DSR}$ device input in all modes.

Bit 26
All modes (RTSAUT) — **Automatic Request to Send.** The RTSAUT signal indicates the output status of RTSAUT, the automatic RTS controller, in all modes.

Bit 25
All modes (TIMELP) — **Timer Elapsed.** The TIMELP signal is set in all modes each time the interval timer decrements to 0. TIMELP is reset by an output to select bit 20 (TIMENB).

Bit 24
All modes (TIMERR) — **Timer Error.** The TIMERR signal is set in all modes when the selected time interval elapses and TIMELP is already set. TIMELP is reset by an output to output select bit 20 (TIMENB).

Bit 23
Modes 0, 1 (XABRT) — **Transmitter Abort.** The XABRT signal is set by the transmitter in modes 0 and 1 when no data is available for transmission and no provisions have been made to identify a fill sequence (i.e., XPRNT is not set). XABRT is reset by an output to output select bit 22 (XAIENB).

Modes 2, 3, 5, 6 — Not used.

Bit 22
All modes (XBRE) — **Transmit Buffer Register Empty.** The XBRE signal is set in all modes when the transmit buffer register (XBR) contents are transmitted to the transmit shift register (XSR) and when the transmitter is initialized. XBRE is reset by a zero output to output select bit 25 (LXBC).

Bit 21
All modes (RBRL) — **Receiver Buffer Register Loaded.** The RBRL signal is set in all modes when a complete character has been transferred from the receiver shift register (RSR) to the RBR. RBRL is reset by an output to output select bit 18 (RIENB).

Bit 20

All modes (DSCINT) — **Data Set Status Change Interrupt.** In all modes DSCINT = DSCH (input bit 29) and DSCENB (output bit 21). DSCINT indicates the presence of an enabled interrupt caused by the change in status of $\overline{DSR}$, RTSAUT, or $\overline{CTS}$.

Bit 19

All modes (TIMINT) — **Timer Interrupt.** In all modes TIMINT = TIMELP (input bit 25) and TIMENB (output bit 20). TIMINT indicates the presence of an enabled interrupt caused by the interval timer.

Bit 18

Modes 0, 1 (XAINT) — **Transmitter Abort Interrupt.** In modes 0 and 1 XAINT = XABRT (input bit 23) and XAIENB (output bit 22). XAINT indicates the presence of an enabled interrupt caused by a transmitter abort.

Modes 2, 3, 5, 6 — Not Used.

Bit 17

All modes (XBINT) — **Transmitter Buffer Interrupt.** In all modes XBINT = XBRE (input bit 22) and XMTON (output bit 16) and XBIENB (Output bit 19). XBINT indicates the presence of an enabled interrupt caused by an empty transmitter buffer.

Bit 16

All modes (RINT) — **Receiver Interrupt.** In all modes RINT =[RBRL (input bit 21) + RHRL (input bit 13) + RABRT (input bit 14)] and RIENB (output bit 18). RINT indicates the presence of an enabled interrupt caused by a loaded receiver buffer or a loaded receiver holding register or a receiver abort (mode 1 only).

Bit 15

All modes (RIN) — **Receiver Input.** In all modes RIN indicates the status of the RIN input to the device.

Bit 14

Mode 1 (RABRT) — **Receiver Abort.** RABRT is set in mode 1 when a flag sequence (01111110) has been previously detected and seven consecutive ones are received. RABRT is reset by an output to output select bit 18 (RIENB).

Modes 5, 6 (RSBD) — **Receiver Start Bit Detect.** In modes 5 and 6 RIN is sampled one half-bit time after the one-to-zero transition of RIN. If RIN is still zero at such time, RSBD is set, indicating the start of a character. RSBD remains true until the complete character has been received. If RIN is not zero at the half-bit time, RSBD remains reset and the receiver waits for the next one-to-zero transition of RIN. This bit is normally used for testing purposes.

Modes 0, 2, 3 — Not Used, (always equals zero).

Bit 13

Mode 1 (RHRL) — **Receiver Holding Register Loaded.** RHRL is set in mode 1 when the receiver has received a complete frame. RHRL is reset by the output of a zero to output select bit 26, RHRRD (receiver holding register read).

Modes 5, 6 (RFBD) — **Receiver Full Bit Detect.** RFBD is set in modes 5 and 6 one full bit time after RSBD is set to indicate the sampling point for the first data bit of the received character. RFBD is reset when the character has been completely received. This bit is normally used for testing purposes.

Modes 0, 2, 3 — Not Used (always equals zero).

Bit 12

Mode 1 (RHROV) — **Receiver Holding Register Overrun.** RHROV is set in mode 1 when the contents of the RHR are altered before RHRL is reset. RHROV is reset by the output of a zero to output select bit 26 (RHRRD).

Modes 5, 6 (RFER) — **Receiver Framing Error.** RFER is set in modes 5 and 6 when a character is received in which the stop bit, which should be a logic one, is a logic zero. RFER should only be read when RBRL (input bit 21) is a logic one. RFER is reset when a character with the correct stop bit is received.

Modes 0, 2, 3 Not Used (always equals zero).

Bit 11

All modes (ROVER) — **Receiver Overrun.** ROVER is set in all modes when the RBR (receiver buffer register) is loaded with a new character before RBRL is reset, indicating that the CPU failed to read the previous character and reset RBRL before the present character is completely received. ROVER is reset when a character is received and RBRL = 0 when the character is transferred to the RBR or an output to output select bit 18 (RIENB).

Bit 10

Modes 0, 2, 3, 5, 6 (RPER) — **Receiver Parity Error.** RPER is set in mode 0, 2, 3, 5, and 6 when the character transferred to the RBR was received with incorrect parity. RPER is reset when a character with correct parity is transferred to the RBR.

Mode 1 (RZER) — **Receiver Zero Error.** RZER is set in mode 1 when the last five bits received prior to the FLAG character ($7E_{16}$) are all ones without being followed by a zero. RZER is reset by resetting RHRRD (Receiver Holding Register Read).

Bit 9

Modes 0, 2, 3, 5, 6 (RCVERR) — **Receiver Error.** In modes 0, 2, 3, 5, and 6 RCVERR = ROVER + RPER + RFER. RCVERR indicates the presence of an error in the most recently received character.

Mode 1 (RFLDT) — **Receiver Flag Detect.** RFLDT is set in mode 1 when the FLAG character ($7E_{16}$) is detected in the input stream. RFLDT is reset by an output to output select bit 18 (RIENB). RFLDT is also set when RABRT is set.

Bit 8 - Bit 0

All modes (RBR8-RBR0) — **Receiver Buffer Register.** The receiver buffer register contains the most recently received complete character. For received character lengths of fewer than nine bits, the character is right justified to the LSB position (RBR0), with unused most-significant bit(s) all zero(s). The presence of data in the RBR is indicated when RBRL is a logic one.

TABLE 6. CRU INPUT ADDRESS ASSIGNMENTS

ADDR	MODE 0 1 2 3 5/6	NAME
31	X X X X X	INT
30	X X X X X	FLAG
29	X X X X X	DSCH
28	X X X X X	CTS
27	X X X X X	DSR
26	X X X X X	RTSAUT
25	X X X X X	TIMELP
24	X X X X X	TIMERR
23	X X	XABRT
	X X X	
22	X X X X X	XBRE
21	X X X X X	RBRL
20	X X X X X	DSCINT
19	X X X X X	TIMINT
18	X X	XAINT
	X X X	
17	X X X X X	XBINT
16	X X X X X	RINT

ADDR	MODE 0 1 2 3 5/6	ALL FLAGS=0	LRCRC = 1	LXCRC = 1	LXBC = 1	RHRRD = 1
15	X X X X X	RIN	RCRC(15)	XCRC(15)	XCRC(15)	RHR(15)
14	X X X					
	X	RABRT				
	X	RSBD				
13	X X X					
	X	RHRL				
	X	RFBD				
12	X X X					
	X	RHROV				
	X	RFER				
11	X X X X X	ROVER				
10	X X X X	RPER				
	X	RZER				
9	X X X X	RCVERR				
	X	RFLDT				
8	X X X X X	RBR8				
7	X X X X X					
6	X X X X X					
5	X X X X X					
4	X X X X X					
3	X X X X X					
2	X X X X X					
1	X X X X X					
0	X X X X X	RBR0	RCRC(0)	XCRC(0)	XCRC(0)	RHR(0)

2.2 GENERAL TRANSMITTER DESCRIPTION

2.2.1 Transmitter Hardware Configuration

Figure 7 is a block diagram of the transmitter section of the TMS 9903 SCC. Either the XBR (transmitter buffer register), SYNC1, or SYNC2 may be loaded into the XSR (transmitter shift register). The LSB of the XSR (XSRLSB) is buffered and output as an external signal, XOUT (in mode 1 loop slave configuration RIN is retransmitted prior to synchronization). Two internal registers — XSYNCL (transmitter sync character length) and XSCL (transmitter shift register character length) are maintained to determine the number of bits per character in XBR, SYNC1, and SYNC2. Since the SYNC1 and SYNC2 registers are of the same length, but not necessarily the same length as the XBR register, the address of the last or highest order bit loaded into both registers is stored in the XSYNCL register, and XSCL contains the number of bits loaded into the XBR register. The XBR register may contain a different length character each time it is loaded. The XBCNT (transmitter bit count) register is loaded with the contents of either XSYNCL or XSCL each time a character is loaded into the XSR. The XPAR (transmitter parity) register serially accumulates the parity of each character and, when enabled, appends the correct parity bit to the transmitted character. The XOCNT (transmitter ones count) register is used in mode 1 operation to accumulate the number of consecutive ones transmitted. The SCTX signal is generated as a synchronous signal of one internal clock cycle each time a bit is to be shifted. If DRCK32 is reset, or $\overline{CTS}$ is inactive (HIGH), SCTX is generated on every one-to-zero transition of $\overline{SCT}$. In the divide-by-32 mode (DRCK32 = 1) if $\overline{CTS}$ goes from one to zero while $\overline{SCT}$ is high, transmission will begin on the second one-to-zero transition of $\overline{SCT}$. The transmitter output, XOUT, will then be updated on every 32nd one-to-zero transition of $\overline{SCT}$ thereafter. On every one-to-zero transition of $\overline{SCT}$, the $\overline{RTS}$ signal is updated by the internal, automatic request-to-send signal, (RTSAUT) unless output select bit 17 (RTS) is addressed. If RTS is selected the $\overline{RTS}$ signal is controlled by the level of output select bit 17 until either the RESET or CLRXMT (clear transmitter) command is issued.

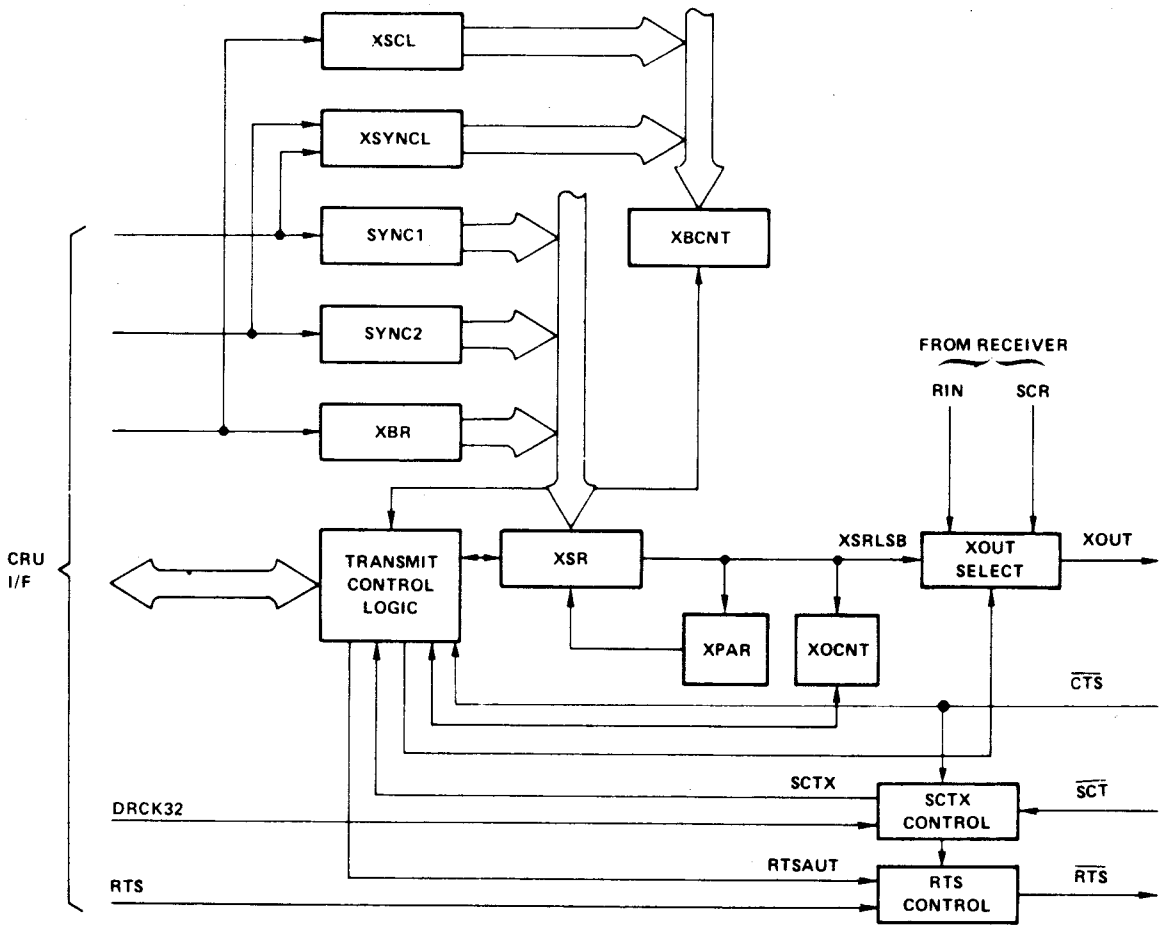


FIGURE 7. TMS 9903 SCC TRANSMITTER BLOCK DIAGRAM

A0001481

2.2.2 Transmitter Initialization

Figure 8 is the flowchart for transmitter initialization. The transmitter is reset to the inactive state when the RESET or CLRXMT commands are issued. To ensure that the control bits are properly loaded into the transmitter, issue CLRXMT after loading the control register the first time. The transmitter remains inactive until the XMTON command is set, enabling transmission and raising RTSAUT. When the CTS command is set to logic one, data transmission begins and continues until the final character is transmitted after XMTON is reset. (Refer also to Figure 13)

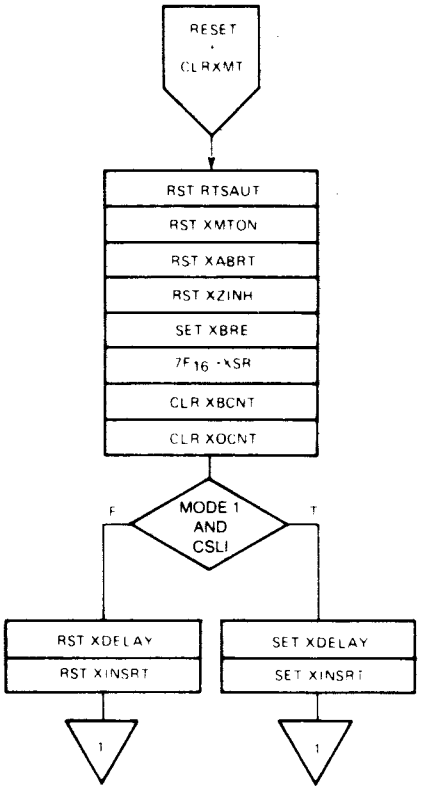


FIGURE 8. TRANSMITTER INITIALIZATION

2.3 GENERAL RECEIVER DESCRIPTION

2.3.1 Receiver Hardware Configuration

Figure 9 is a block diagram of the receiver section of the TMS 9903. The value of control register bit 11 — 32X data rate clock (DRCK32) — determines the sampling point for RIN. For DRCK32 = 0, RIN is sampled on every zero-to-one transition of SCR. For DRCK32 = 1, RIN is sampled every 32nd SCR beginning with the zero-to-one transition of the 16th SCR after syncrhronization. An internal signal, SCRX, is generated every time the 9903 is prepared to receive a bit, i.e. a zero-to-one transition of SCR. The received character is assembled in the receiver shift register (RSR) according to the length specified in control register bits 2, 1, 0 — receiver character length select (RSCL). The value of RSCL is transferred to the RBCNT (receiver bit count) register when the contents of the RSR are transferred to the receiver buffer register (RBR). This double buffering of the received character and the character length provide variable character length capability. The character length may be altered any time prior to the transfer of the next received character to the RBR. In all modes of operation except mode 1, the parity checker is updated with each bit shifted into the RSR. If parity is

enabled, the receiver compares the assembled parity bit to the received parity bit, and then sets it to zero when the character is transferred to the RBR. When the character is transferred to the RBR the receiver buffer register loaded flag RBRL is set. If RBRL was set already, the receiver overrun flag, ROVER, is set. Incorrectly received parity will set the parity error flag (RPER) in all but mode 1 operation. Note that parity generation and detection is not available in mode 1 operation. The comparator and sync character register SYNC1 are utilized in the several modes to provide flag and sync character detection. For a detailed discussion of each operation, see the discussion of the particular mode of operation desired.

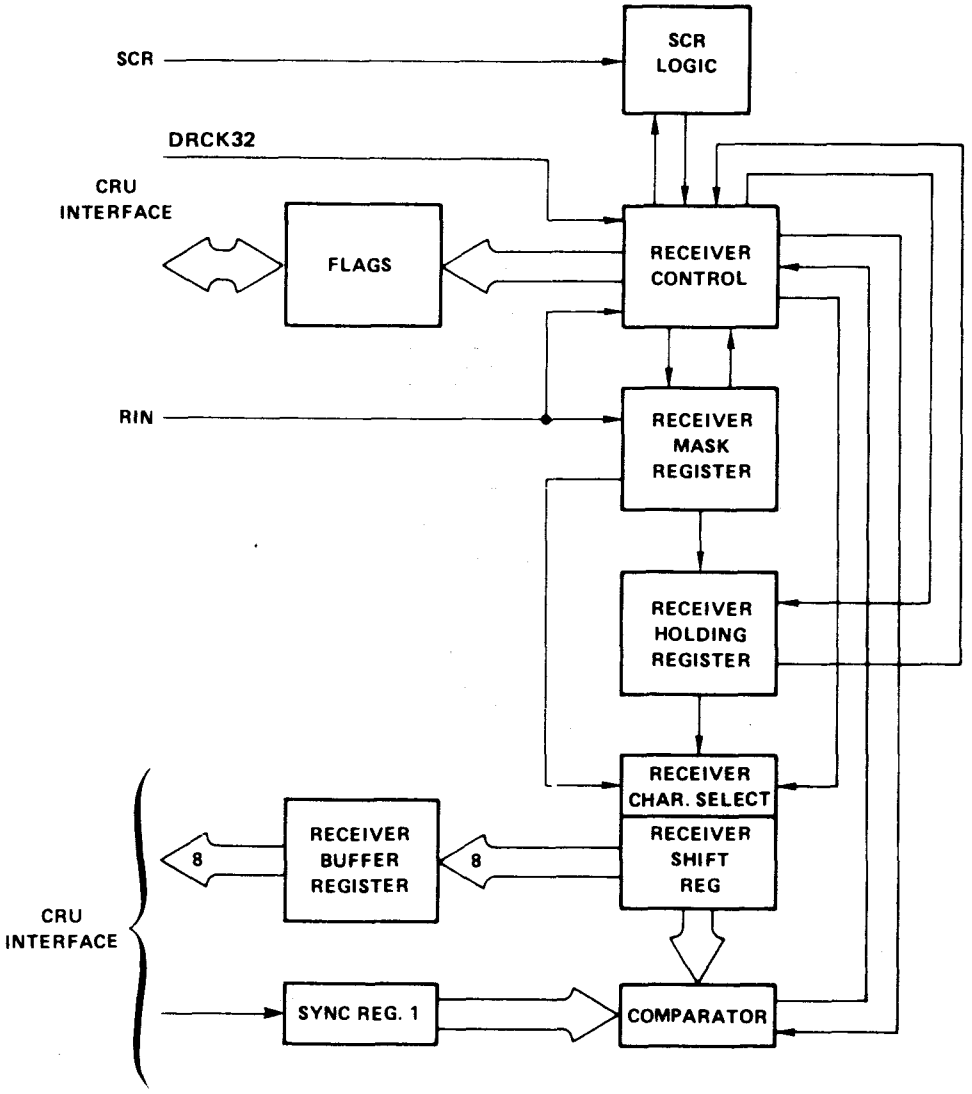


FIGURE 9. TMS 9903 RECEIVER BLOCK DIAGRAM

2.3.2 Receiver Initialization

The receiver is initialized by the RESET and CLRRCV (clear receiver) commands from the CPU. This causes the receiver mask register (used in mode 1 operation only) to be initialized to all ones. the receiver shift register and parity to be initialized, and all receiver related flags to be reset.

Initializing the RSR sets the *N-1* least-significant bits to logic one and sets the MSB (bit N) to logic zero, where *N* is the number of bits per character. The detection of the zero shifted out of the RSR signals the assembly of a complete character. For this reason the CLRRCV command should be issued after loading the control register to assure the correct assembly of the first character received after loading.

2.4 TRANSMITTER AND RECEIVER OPERATION

The TMS 9903 has six different operational modes (0, 1, 2, 3, 5, and 6). Following is a detail discussion for each mode of the transmitter and receiver operations.

2.4.1 Mode 0 Operations

2.4.1.1 Transmitter Operation

Figure 10 is a flowchart for mode 0 transmitter operation. If parity is enabled, the parity bit is appended to the transmitted character. When the character has been shifted out and no data is available (XBRE = 1), the transmitter will either abort operation or transmit the contents of SYNC2, depending on the value of XPRNT (transparent). Note that parity is not generated when SYNC2 is transmitted; therefore, if parity is desired, the correct parity bit must be appended to the sync character when it is loaded into SYNC2.

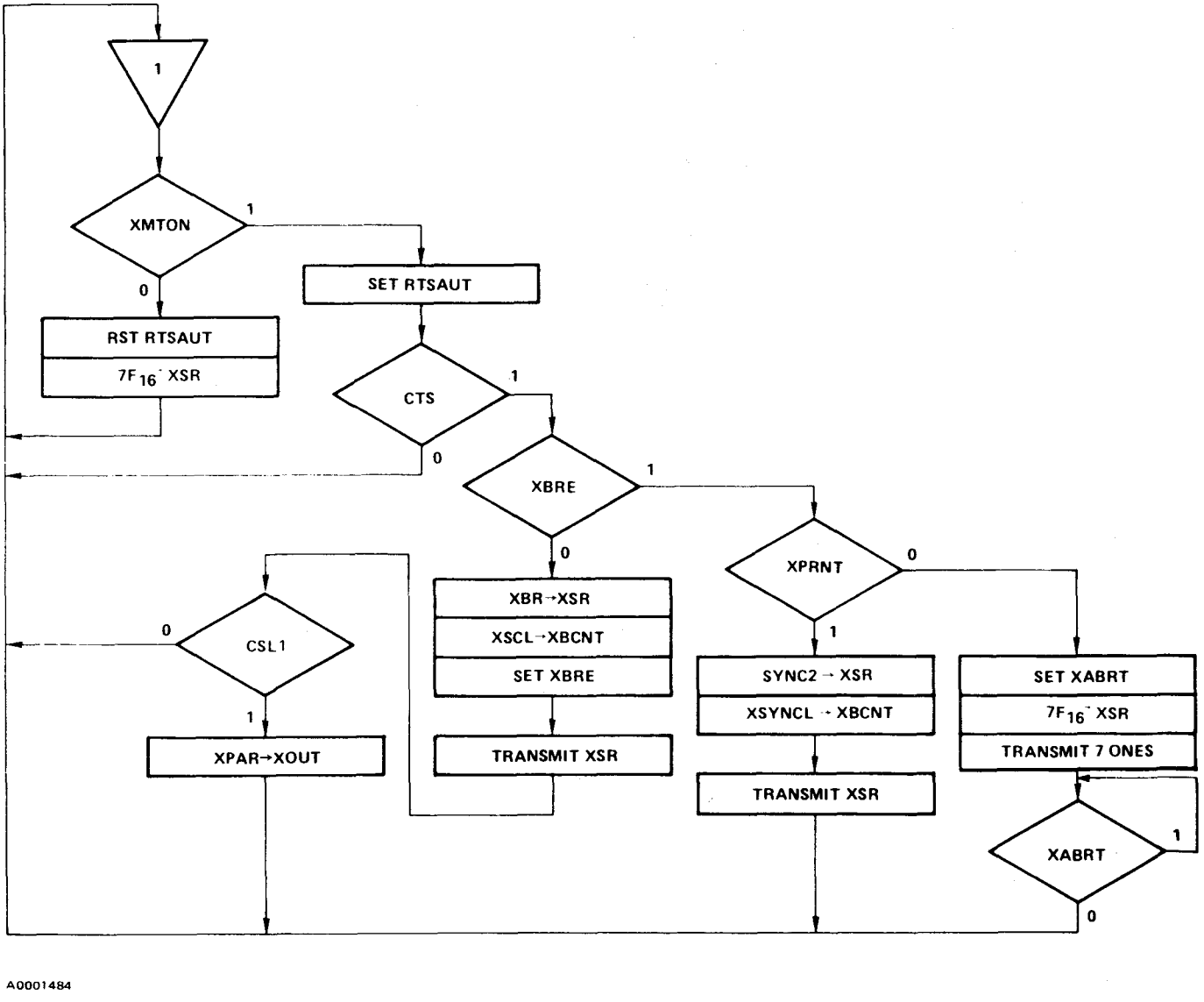


FIGURE 10. MODE 0 TRANSMITTER OPERATION

2.4.1.2 *Mode 0 Receiver Operation*

Figure 11 is a flowchart for mode 0 receiver operation. This mode is the basic subset of receiver operation for all modes. The general description of receiver operation described in Section 2.3. above applies to mode 0 operation.

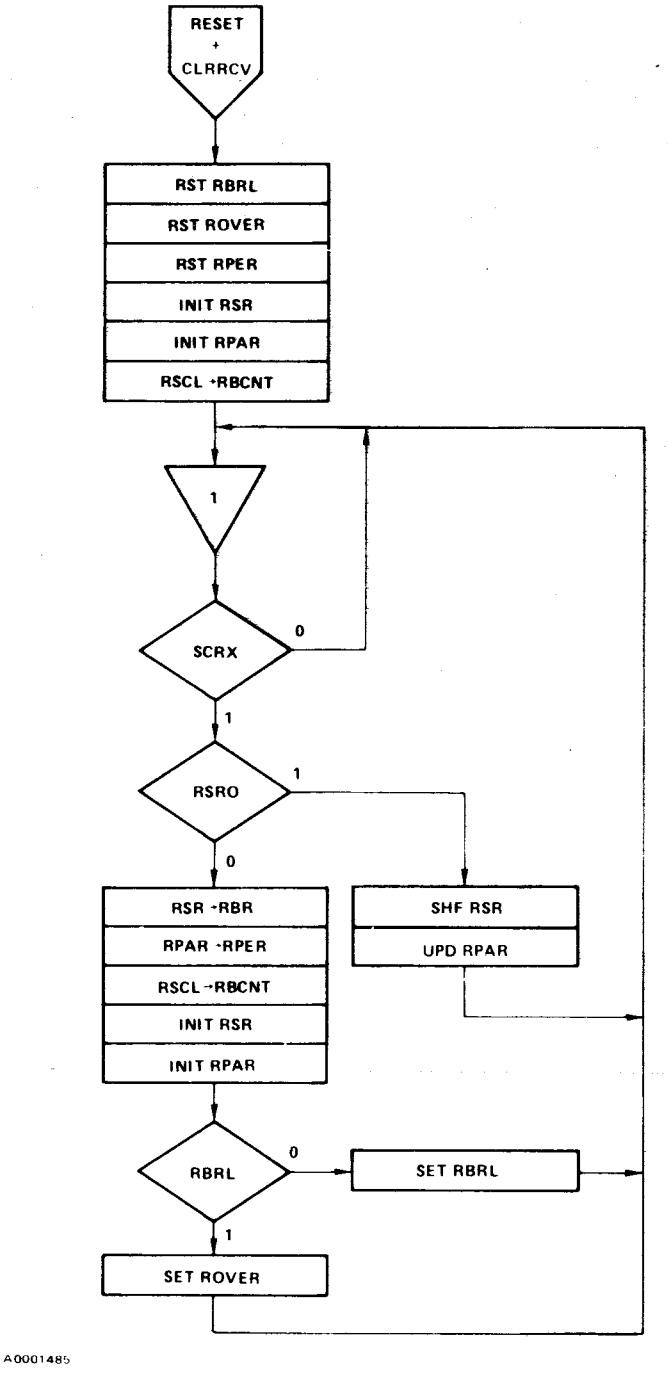


FIGURE 11. MODE 0 RECEIVER OPERATION

2.4.2 Mode 1 Operation [SDLC]

2.4.2.1 Mode 1 Transmitter Operation

Figure 12 is a flowchart of transmitter operation in mode 1. Beginning transmission varies slightly, depending on the configuration selected with control register bits 5 and 4, the configuration select (CSL1, CSL0).

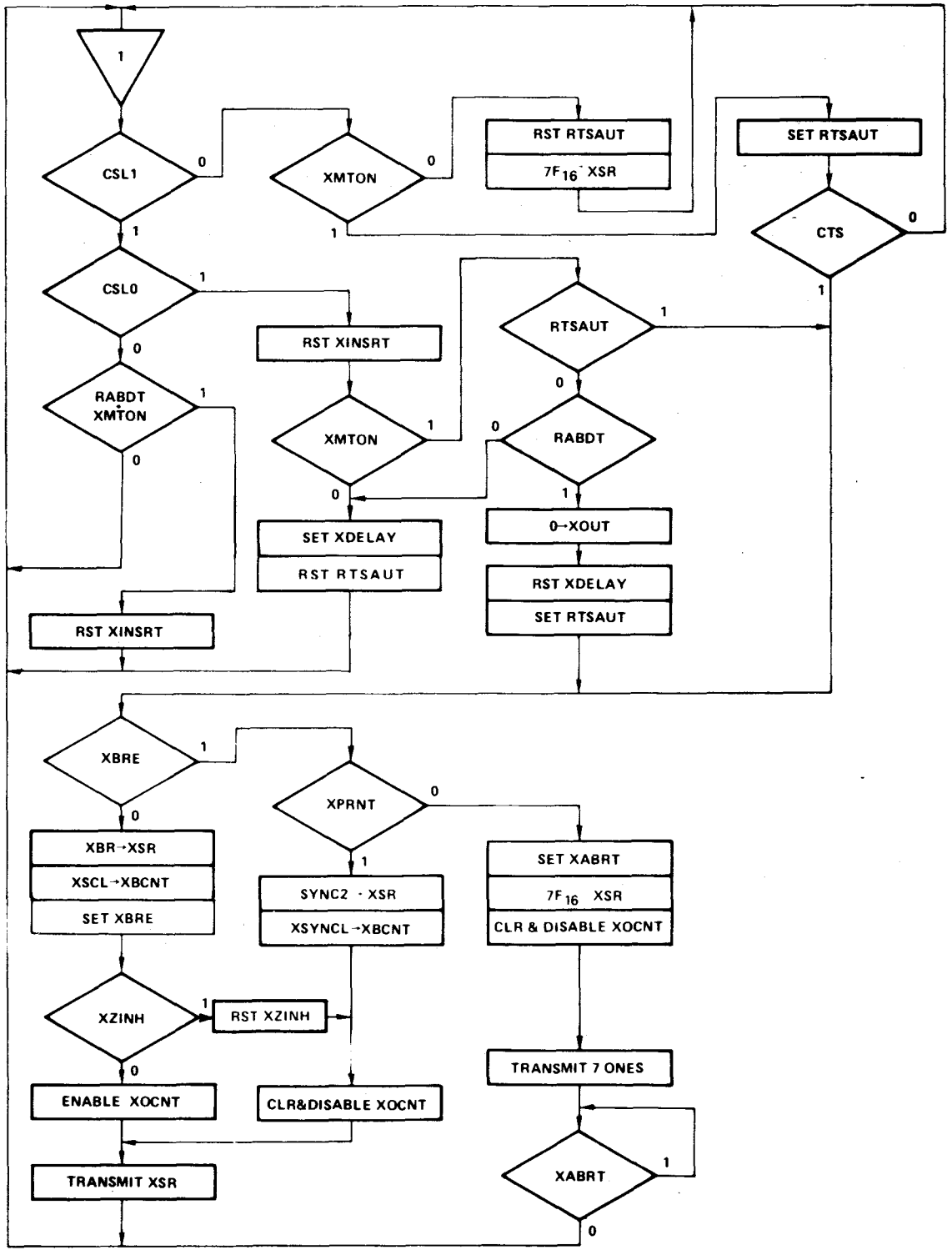


FIGURE 12. MODE 1 TRANSMITTER OPERATION

2.4.2.1.1 Normal and Loop Master (CSL1 = 0) Operation. The operation of the transmitter is the same when CSL1 = 0, regardless of the status of CSL0. When XMTON is set, RTSAUT becomes active and data transmission begins with CTS = 1. As each character is transferred from XBR to XSR, the XZINH flag is tested. If XZINH = 1, XOCNT is cleared and zero-insertion is disabled. If XZINH = 0, a zero bit will be inserted after each fifth consecutive transmitted one. If XBRE = 1 when a character is to be loaded into the XSR, the transmitter will either abort (when XPRNT = 0) or transmit the contents of SYNC2 (when XPRNT = 1). When SYNC2 is transmitted, XOCNT is cleared and disabled, prohibiting zero-insertion. If the transmitter aborts, the XABRT flag is set and a minimum of seven ones are transmitted. The transmitter will remain inactive until XABRT is cleared.

2.4.2.1.2 Loop Slave (Pending Synchronization) (CSL1 = 1, CSL0 = 0) Operation. As a loop slave the device must first synchronize itself to the communication line before actively transmitting data. Initially, the line is monitored to search for an *end-of-poll* (EOP = 11111110) character, which occurs when RABDT = 1. At this time, if XMTON = 1, the transmitter introduces a single-bit delay by retransmitting the final one, and subsequently retransmitting each received data bit. The logic associated with XOUT is shown in Figure 13. When XINSRT = 1 and XDELAY = 1 then, XOUT = RIN. When XINSRT is reset by detection of an EOP, RIN is delayed a single bit-time before being transmitted on XOUT.

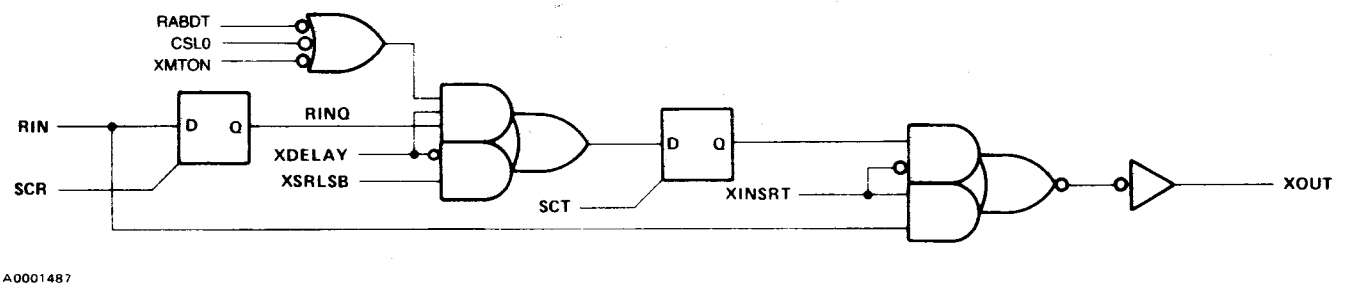


FIGURE 13. SIMPLIFIED XOUT SELECT LOGIC

2.4.2.1.3 Loop Slave (Active) (CSL1 = 1, CSL0 = 1) Operation. After loop synchronization has been achieved, transmission may begin by first detecting an EOP (11111110). The last one is inverted to provide the beginning flag of the transmitted frame, and normal data transmission begins.

2.4.2.2 Mode 1 Receiver Operation

Figure 14 is a flowchart of the mode 1 receiver operation and Figure 15 shows the register circuitry used to perform these operations. As described in Section 2.3.2 above, executing the RESET or CLRRCV commands resets all flags, initializes the receiver registers, and loads all ones into the mask register.

2.4.2.2.1 Synchronization. Each bit time (SCRX = 1) data is shifted into RMSK. When a FLAG character bit pattern of $7E_{16}$ is detected (RFLG = 1), the receiver achieves synchronization and the bit pattern 001111111_2 is loaded into the nine-bit RSR.

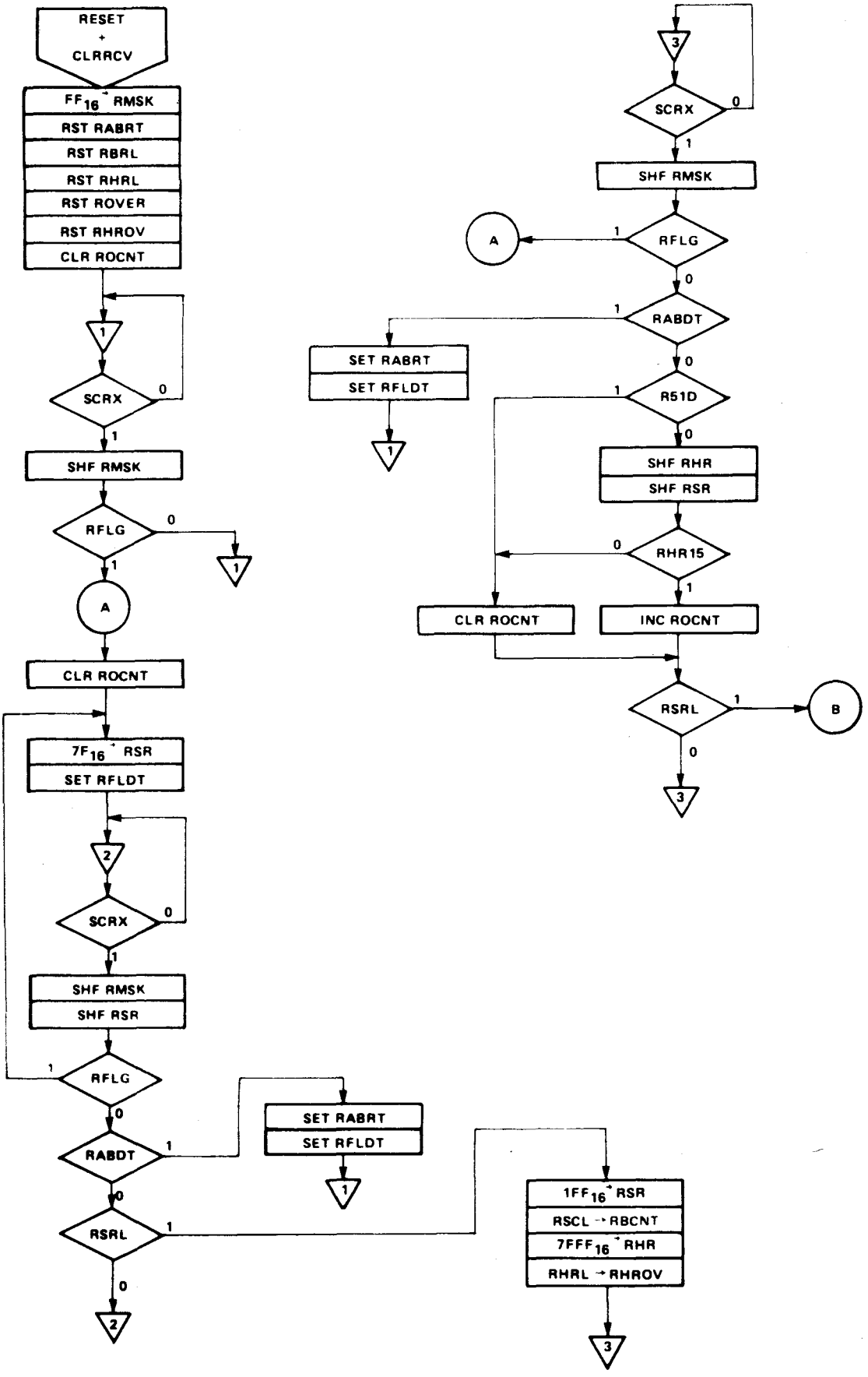


FIGURE 14. MODE 1 RECEIVER OPERATION (PAGE 1 OF 2)

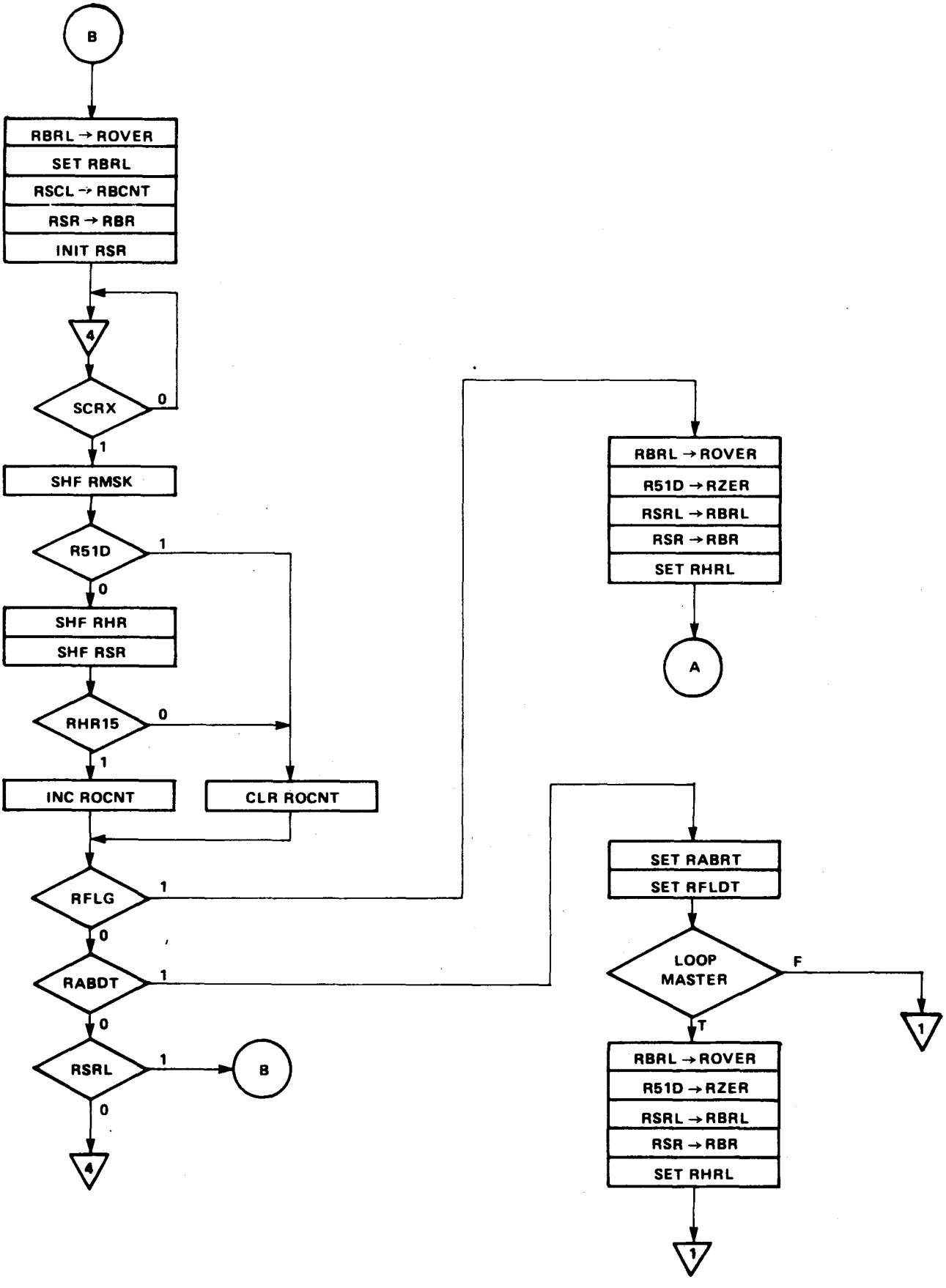


FIGURE 14. MODE 1 RECEIVER OPERATION (PAGE 2 OF 2)

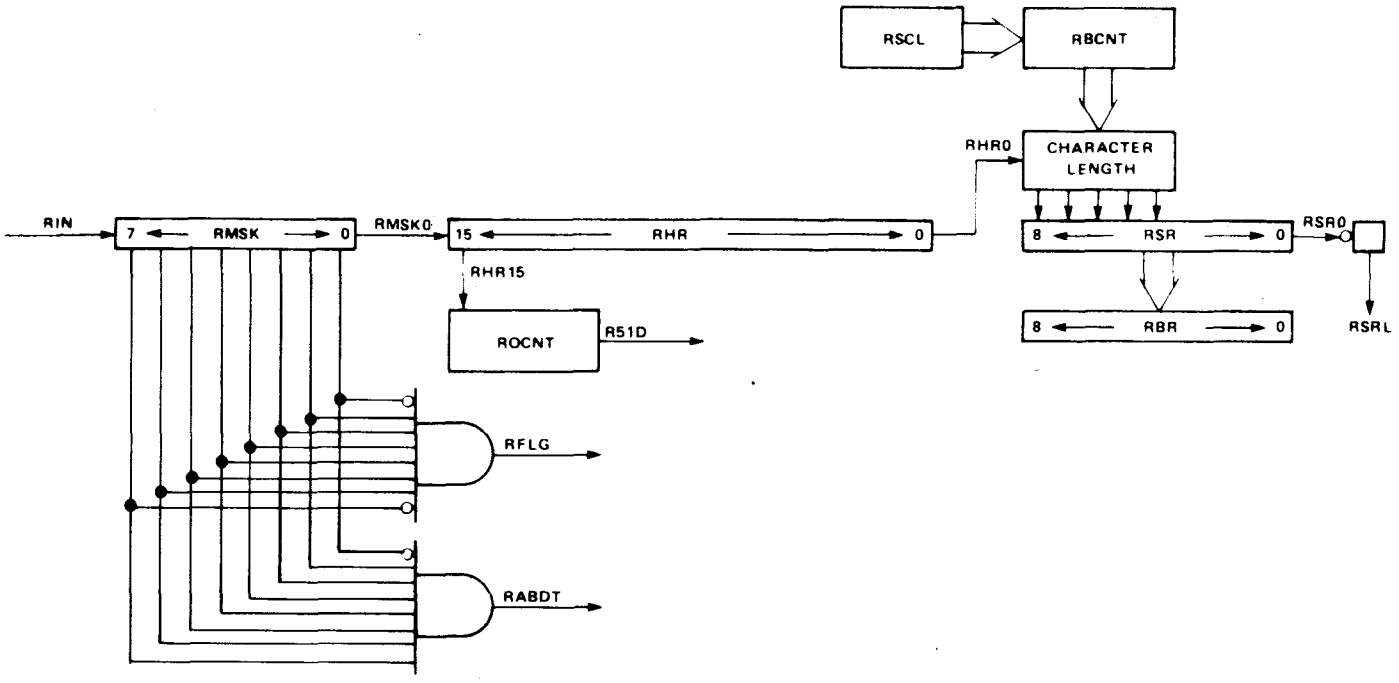


FIGURE 15. MODE 1 RECEIVER CIRCUITRY

2.4.2.2.2 Eight-Bit Delay. Each bit time, RIN is shifted into RMSK, and RSR is shifted right until RSR0 = 0. This sets RSRL, indicating eight bits have been shifted. If the FLAG pattern is detected again, the eight-bit delay is repeated. The FLAG pattern consists of six consecutive ones (01111110). If more than six consecutive ones are detected in RMSK, RABDT is set to a one and the receiver aborts. FLAG patterns, abort patterns, and zeroes generated by five-ones-zero insertion are all deleted from the serial bit stream before being shifted into RHR.

2.4.2.2.3 16 + (RSCL) Bit Delay. After the eight-bit delay, the RHR (receiver holding register) is loaded with $7FFF_{16}$ and the RSR is loaded with all ones. The contents of RSCL (receiver character length select field of the control register) are loaded into RBCNT (receiver bit count register), which selects which bit of the RSR is the MSB. Each bit time, RIN is shifted into RMSK. When R51D = 1 (five consecutive ones detected), the next bit — the inserted zero bit — is not shifted from RMSK0 to RHR15; otherwise, RMSK0 is shifted into RHR15, RHR is shifted right, RHR0 is shifted into the selected MSB of RSR, and RSR is shifted right. This operation continues until RSRL = 1, indicating that the delay has been completed, and RMSK, RHR, and RSR all contain valid data. The fully assembled character is then transferred from the RSR to the receiver buffer register (RBR) and the receiver buffer register loaded flag (RBRL) is set.

2.4.2.2.4 Character Assembly. Each time RSRL = 1, RSCL is transferred to RBCNT, RSR to RBR, RBRL is set; an RSR is initailized to all ones except for the MSB of the selected character length. That is, for a seven-bit character, RSR is loaded with 00111111_2. Data is shifted through RMSK, RHR, and RSR each bit time, performing zero-deletion until a FLAG pattern or an abort sequence is detected.

2.4.2.2.5 Receiver Abort. When the receiver detects the abort pattern (7 or more consecutive ones), RABRT is set and control returns to the initial state where the FLAG pattern is required for synchronization.

2.4.2.2.6 Flag Detection. After entry into character assembly, the receiver operation continues until a flag is detected, indicating the end of a frame. When this occurs, several operations are performed:

(1) RSR is transferred to RBR.
(2) If RBRL is set, ROVER is set.
(3) If RSRL = 1, RBRL is set.
(4) RHRL is set.
(5) Control returns to the eight-bit delay described in paragraph 2.4.2.2.2 above.

Thus, RHRL is set whenever the end of a frame is detected. If a complete character is received, RBRL is set; otherwise RBRL is not set and the number of bits received can be determined by shifting the contents of RBR right until the first zero is shifted out. After the receiver CRC register (RCRC) is updated with the most recently received data, RHR may be compared with RCRC to determine if the received CRC contained in RHR matches the expected CRC contained in RCRC. If RZER (receiver zero error) = 1, it indicates a zero was not appended to the last five consecutive ones received. This occurs only if the last 13 received bits are "0111111011111_2".

2.4.2.2.7 Variable Receive Character Length. Since the *advanced data communication control protocol* (ADCCP) permits variable length characters in the same frame, the receiver double-buffers the received character length. Each time RSR is transferred to RBR, RSCL is transferred to RBCNT. Thus, RSCL (bits 2-0 of the control register) may be altered any time before the next character is transferred into RBR as long as a minimum setup time of two internal clocks is met.

2.4.2.2.8 Loop Master Operation. When the TMS 9903 is configured to operate as a loop master (CSL1 = 0, CSL0 = 1), the EOP character (11111110_2, or RABDT = 1) is interpreted in the same manner as the FLAG character with respect to terminating frame reception. However, a FLAG must be received before synchronization occurs for the reception of the next frame.

2.4.3 Mode 2 Operation

2.4.3.1 Mode 2 Transmitter Operation

Figure 16 is a flowchart of mode 2 transmitter operation. If parity is enabled, the parity bit is appended to the transmitted character. When the character has been shifted out and no data is available (XBRE = 1), the contents of SYNC2 are transmitted without parity. If parity is required for the sync character, it must be appended to the character when it is loaded into SYNC2.

2.4.3.2 Mode 2 Receiver Operation

Figure 17 is a flowchart of mode 2 receiver operation. In mode 2 operation, after initialization, the receiver assembles a character in the RSR and compares it to the sync character contained in SYNC1. Once the RSR receives the sync character, receiver operation is similar to that of mode 0 receiver operation discussed in Section 2.4.1.2 above.

2.4.4 Mode 3 Operation

2.4.4.1 Mode 3 Transmitter Operation

Figure 18 is a flowchart of mode 3 transmitter operation. If parity generation is enabled, the correct parity bit is assembled as the character is shifted out of the XSR and appended as the last bit. When the character has been transmitted and no further data is available (XBRE = 1), and XPRNT = 0, the contents of SYNC1 are loaded and shifted out twice to give a fill sequence of SYNC1 — SYNC1. If XPRNT = 1 and XBRE = 1, the contents of SYNC2 are loaded and shifted, followed by the contents of SYNC1, giving a fill sequence of SYNC2 — SYNC1.

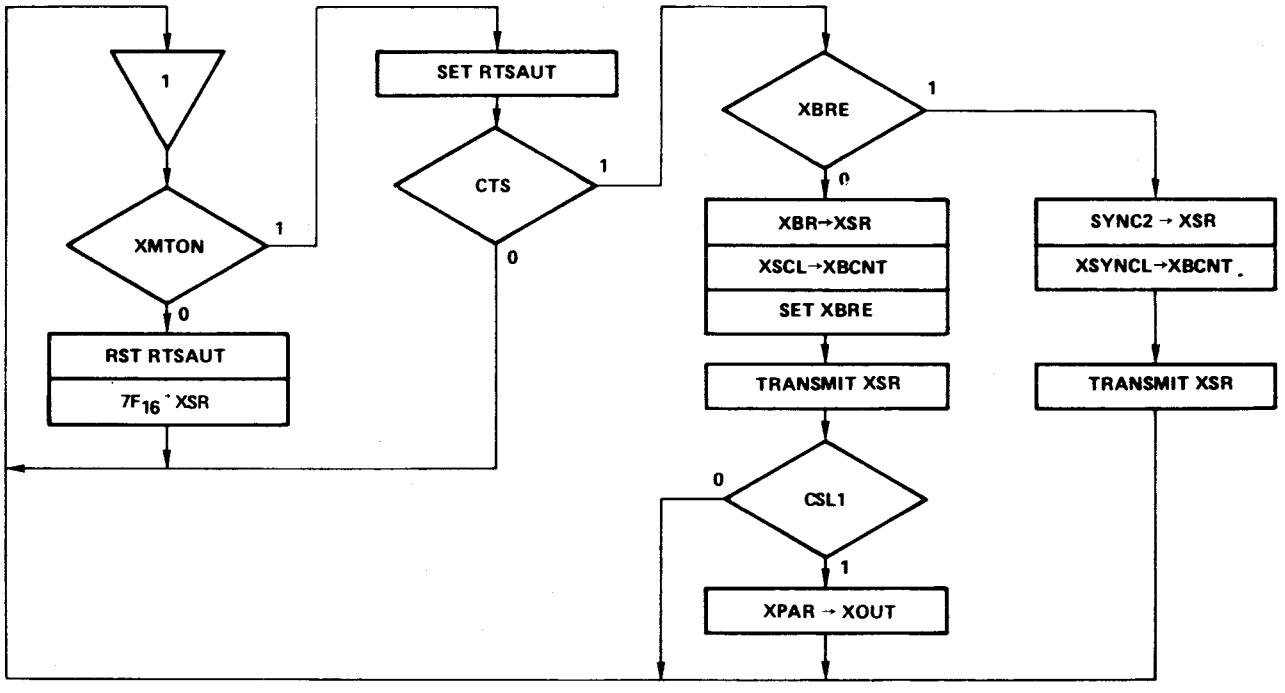


FIGURE 16. MODE 2 TRANSMITTER OPERATION

2.4.4.2 *Mode 3 Receiver Operation*

Figure 19 is a flowchart of mode 3 receiver operation. In mode 3 operation, after initialization, the receiver assembles two consecutive SYNC1 characters before returning to mode 0 operation.

2.4.5 Mode 5 and 6 Operation

Although the TMS 9903 is designed primarily for synchronous communication control, it can be used for asynchronous operation if it is set to operate in mode 5 or 6, and if external baud rate clocks are provided for both SCR and $\overline{\text{SCT}}$. Mode 5 is asynchronous operation with one start and two stop bits, while mode 6 is asynchronous operation with one start and one stop bit.

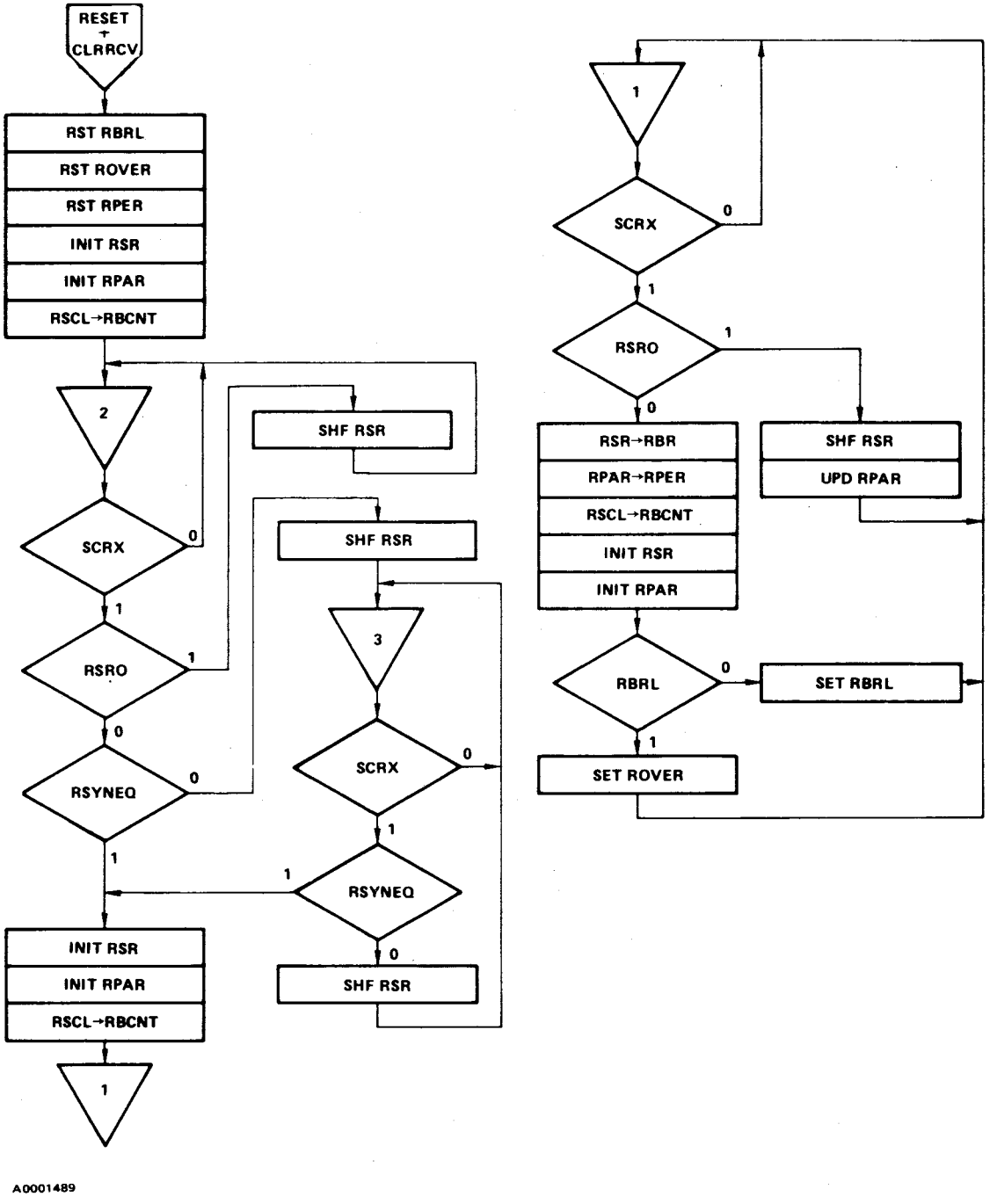


FIGURE 17. MODE 2 RECEIVER OPERATION

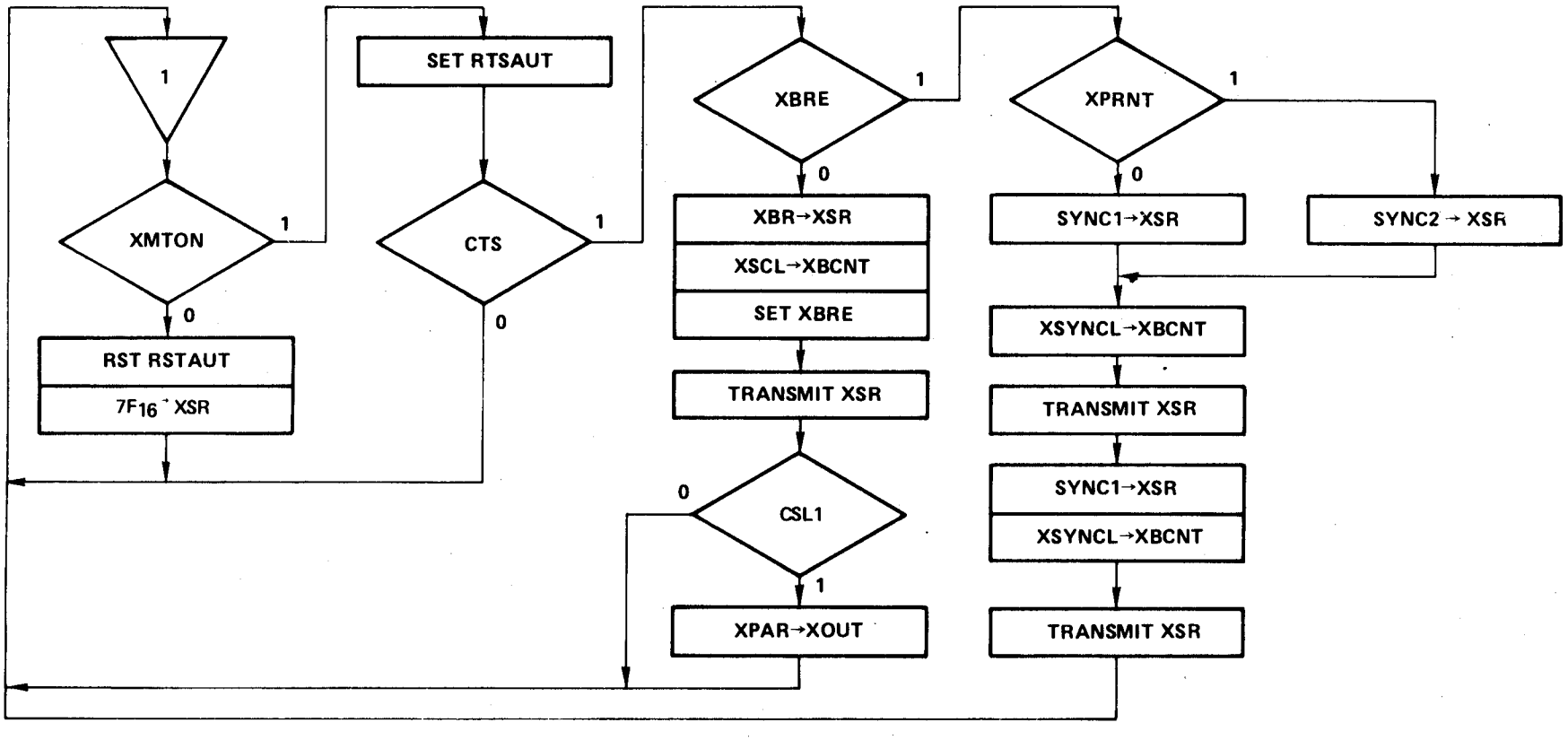


FIGURE 18. MODE 3 TRANSMITTER OPERATION

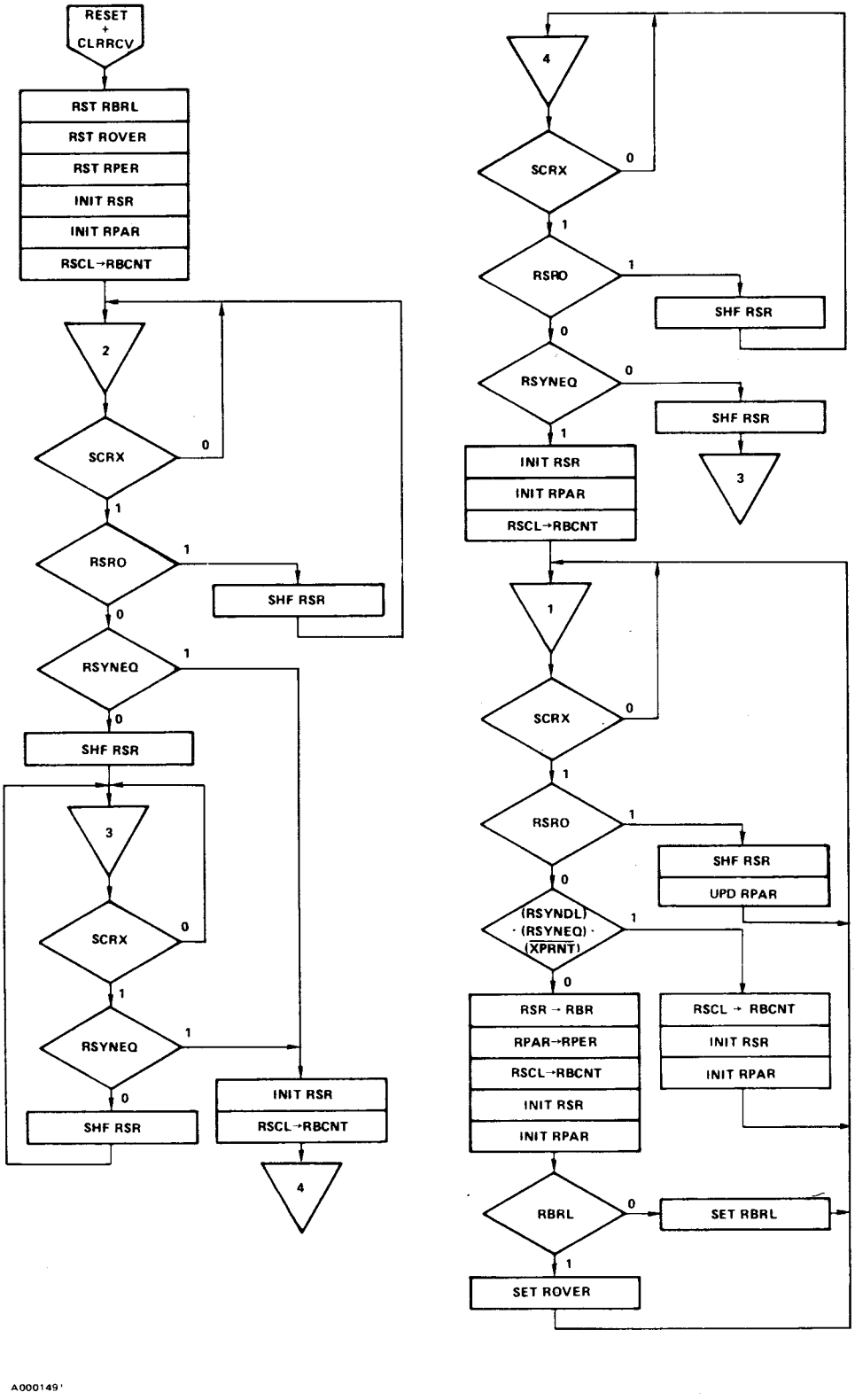


FIGURE 19. MODE 3 RECEIVER OPERATION

2.4.5.1 *Mode 5 and 6 Transmitter Operation*

Operation of the transmitter in modes 5 and 6 is described in the Figure 20 flowchart. The transmitter is initialized by issuing the RESET or CLRXMT commands, which cause the internal signals XBRE to be set and XMTON to be reset. Device outputs $\overline{RTS}$ and XOUT are set, placing the transmitter in the inactive state. When XMTON is set by the CPU, the $\overline{RTS}$ output becomes active. Transmission then begins when CTS becomes active.

If BRKON is set, the character in transmission is completed; any character in the XBR is loaded into the XSR and transmitted; and XOUT is set to zero. Further loading of XBR should be avoided until BRKON is reset. If BRKON = 0, XOUT is set to logic one when the transmitter completes the current transmission and no further data is loaded into XBR.

2.4.5.2 *Mode 5 and 6 Receiver Operation*

Figure 21 is a flowchart of mode 5 or 6 receiver operation. The receiver is initialized when the RESET or CLRRCV command is issued in mode 5 or 6. The RBRL flag is reset to indicate that no character is in the RBR, and the RSBD and RFBD flags are reset. The receiver remains inactive until a one-to-zero transition of the RIN device-input is detected which sets SBD.

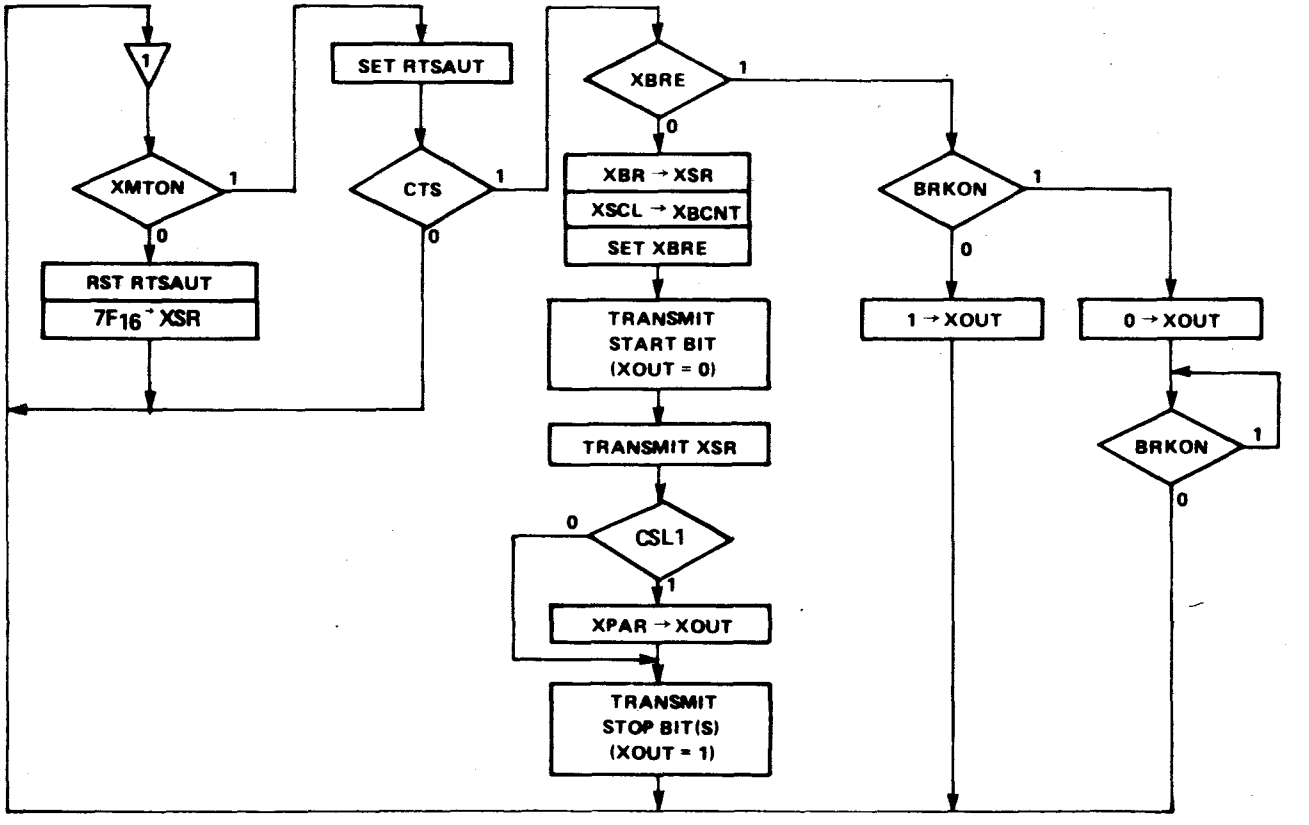


FIGURE 20. MODE 5 OR 6 TRANSMITTER OPERATION

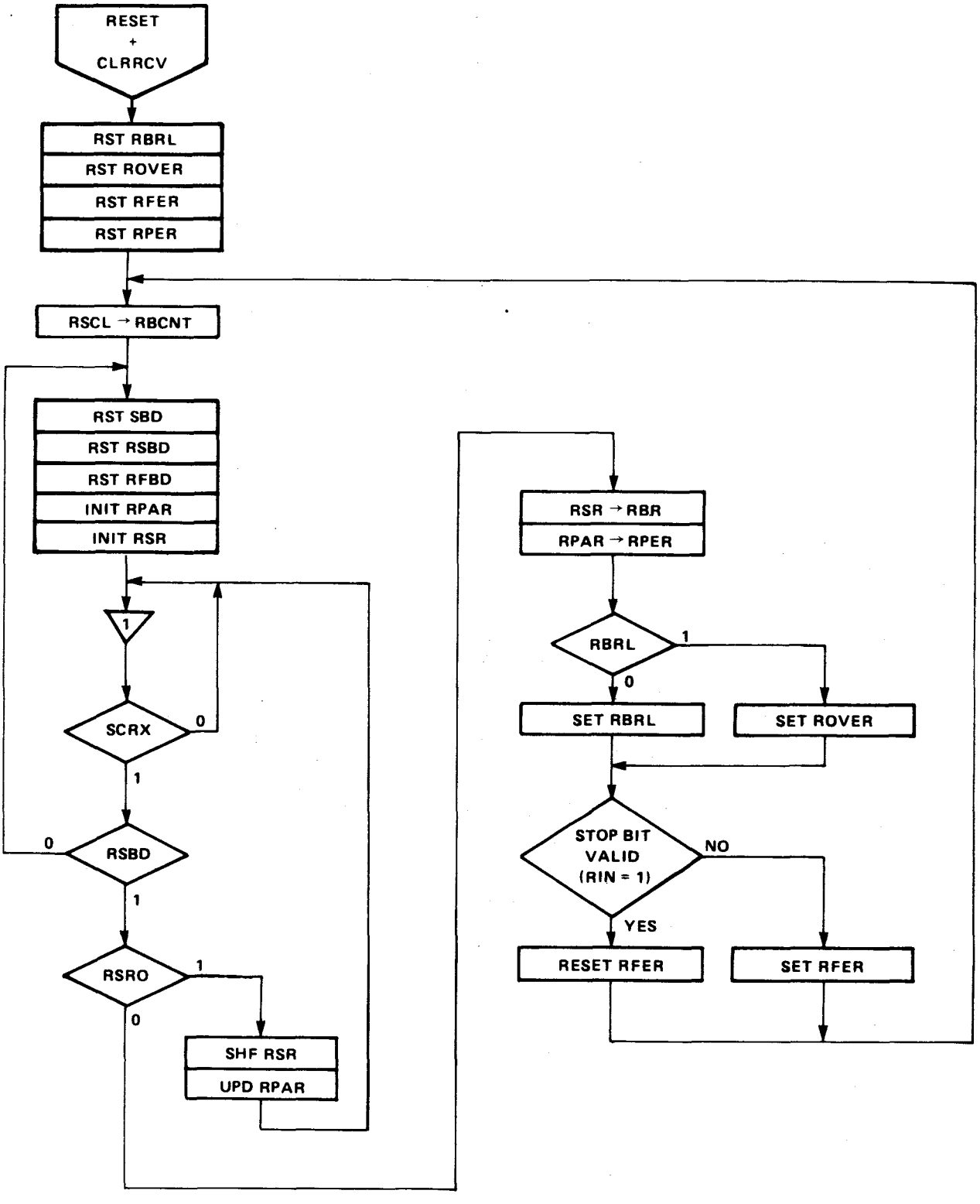


FIGURE 21. MODE 5 OR 6 RECEIVER OPERATION

2.4.5.2.1 **Start Bit Detect.** The receiver delays one-half bit time from SBD being set and again samples RIN to ensure that a valid start bit has been detected. If RIN = 0 after the half-bit delay, RSBD (receiver start bit detect) is set and data reception begins. If RIN = 1, no data reception occurs. SBD and RSBD are reset and wait for the next one-to-zero transition of RIN.

2.4.5.2.2 **Data Reception.** In addition to verifying the valid start bit, the half-bit delay after the one-to-zero transition also establishes the sample point for all subsequent data bits in this character. Theoretically, the sample point is in the center of each bit cell, thus maximizing the limits of acceptable distortion of data cells. After the first full bit delay the least significant data bit is received and RFBD is set. The receiver continues to delay one-bit intervals and samples RIN until the selected number of bits are received. If parity is enabled, one additional bit is read for parity. After an additional bit delay, the received character is transferred to the receiver buffer register, RBRL is set, ROVER and RPER are loaded with appropriate values, and RIN is tested for a valid stop bit. If RIN = 1, the stop bit is valid. RFER, RSBD, and RFBD are reset and the receiver waits for the next start bit to begin reception of the next character.

If RIN = 0 when the stop bit is sampled, RFER is set to indicate the occurrence of a framing error. RSBD and RFBD are reset, but sampling for the start bit of the next character does not begin until RIN = 1.

2.5 INTERVAL TIMER SECTION

A block diagram of the interval timer is shown in Figure 22. When the load interval register flag (LDIR) is set, output to CRU bit addresses 0-7 is loaded into the interval register. The LDIR flag is reset by a command from the CRU. After LDIR is reset, the contents of the interval register are loaded into the interval timer. The interval timer is decremented at the rate of the prescaler output. When the interval timer decrements to 0, the TIMELP flag is set and the contents of the interval register are reloaded into the interval timer. If TIMELP has not been cleared by the CPU by the time that the interval timer decrements to zero again, the TIMERR flag is set. A flowchart for interval timer operation is illustrated in Figure 23. Note the interval timer is always running.

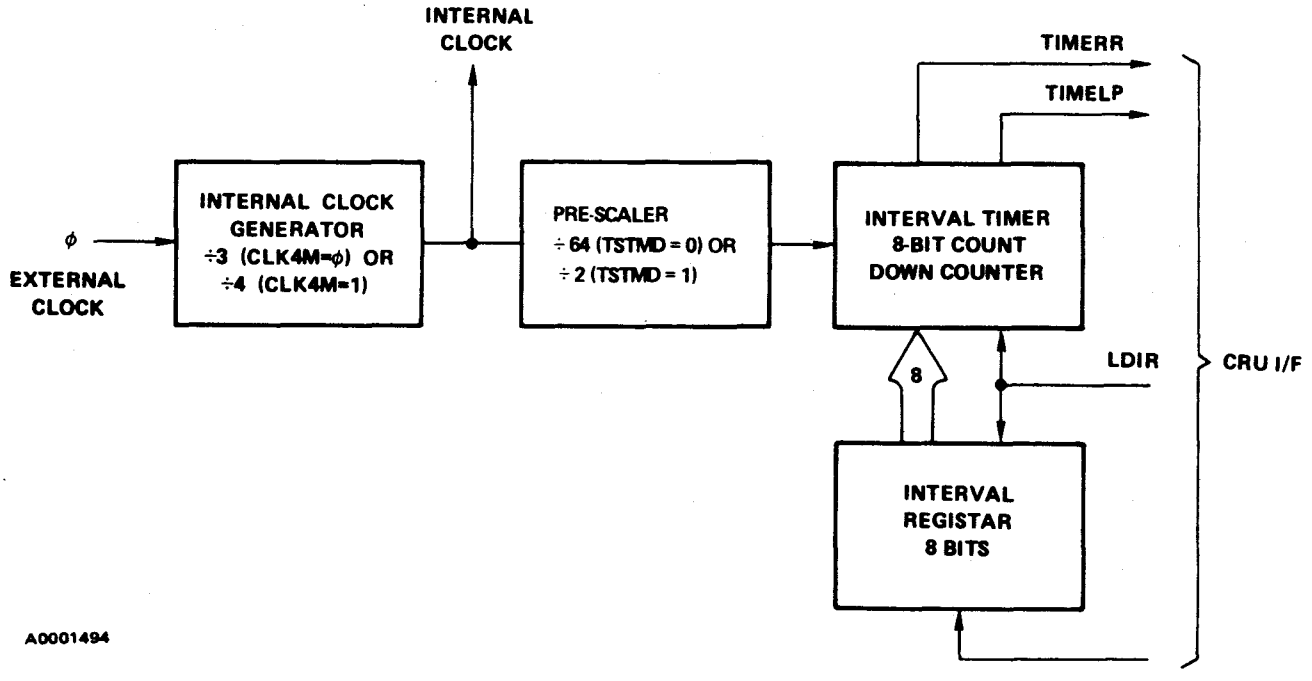
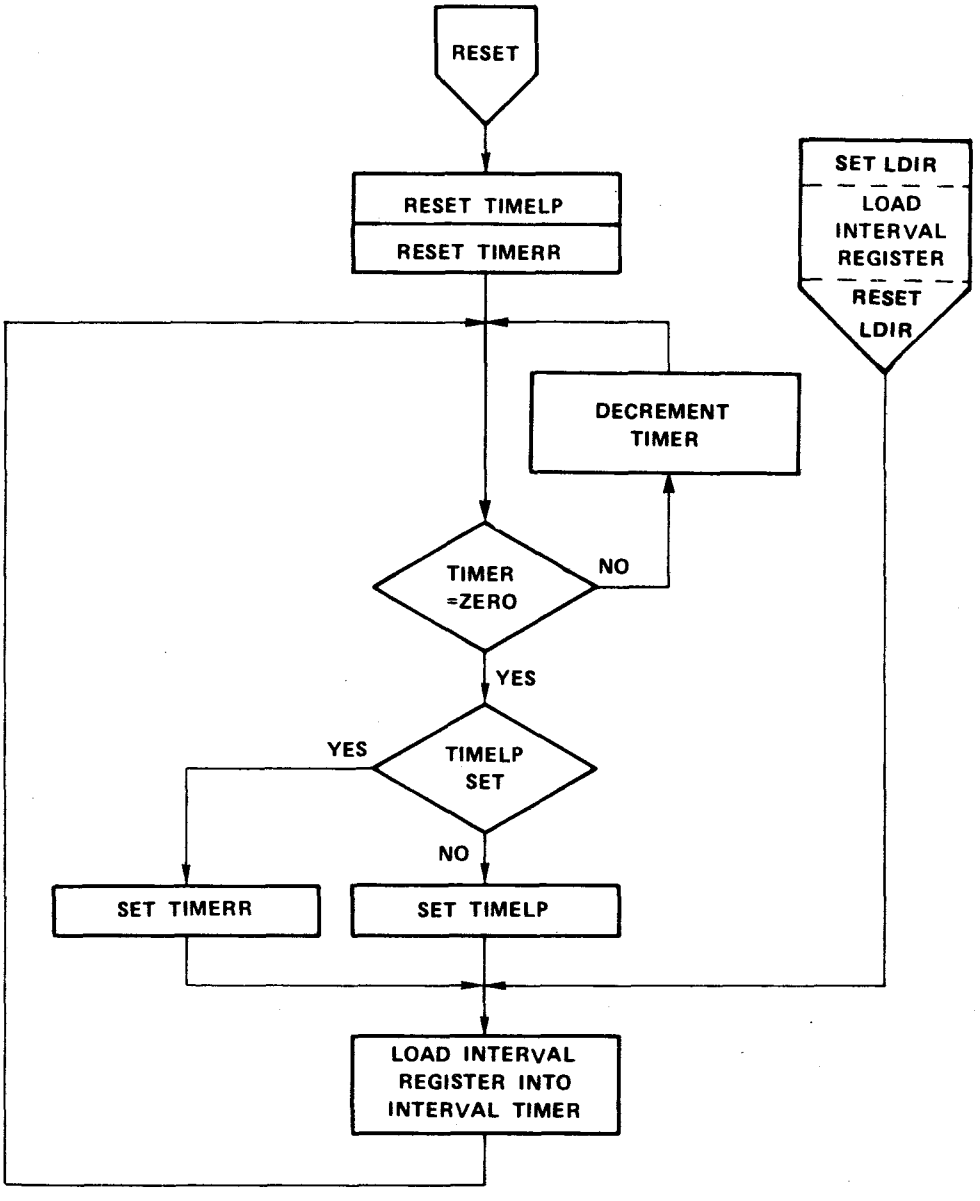


FIGURE 22. INTERVAL TIMER BLOCK DIAGRAM



A0001495

FIGURE 23. INTERVAL TIMER OPERATION

2.5.1 Time Interval Programming

The rate at which the interval timer sets TIMELP during normal operation is determined by the value loaded into the interval register. In normal operation (TSTMD = 0), the prescaler output has a frequency 1/64 of the internal system clock. Thus, when a standard 3- or 4-MHz external clock is used to generate a 1-MHz internal clock, the interval timer is decremented once every 64 microseconds. The interval register selects the number of 64-microsecond intervals contained in each interval timer period. Thus, the interval may range from 64 microseconds (interval register = 01_{16}) to 16,320 microseconds (interval register = FF_{16}) in 64-microsecond increments.

2.5.2 Test Mode Interval Timer Operation

When TSTMD = 1, the prescaler divides the internal system clock frequency by two rather than by 64, causing the interval timer to operate at 32 times the rate at which it operates when TSTMD = 0 for identical interval register contents.

2.6 INTERRUPTS

The interrupt output control line ($\overline{INT}$) is active (low) when any of the following conditions occur and the corresponding interrupt has been enabled by the CPU:

1) DSCH = 1.DSCH (data set status change) is set when $\overline{DSR}$, $\overline{CTS}$, or RTSAUT changes levels.

2) TIMELP = 1.TIMELP (timer elapsed) is set when the selected time interval has elapsed.

3) XBRE = 1.XBRE (transmitter buffer register empty) is set when the transmitter buffer register is empty.

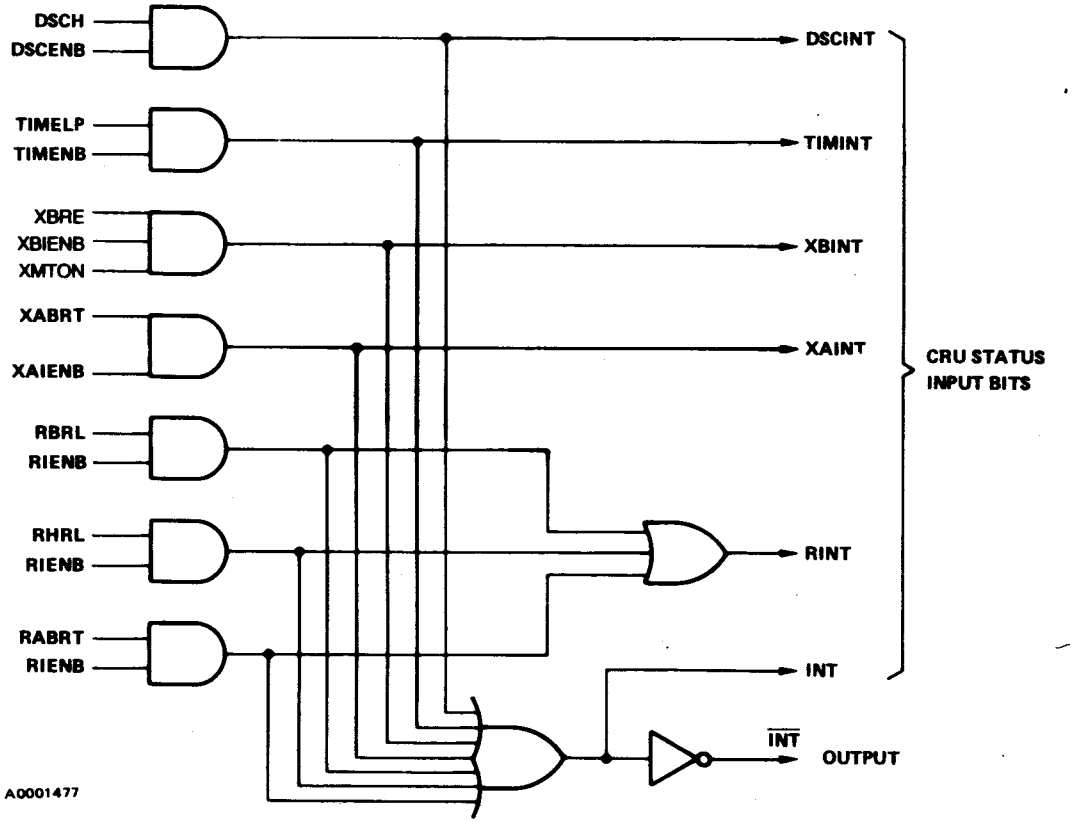
4) XABRT = 1.XABRT (transmitter abort) is set in mode 0 and 1 when no data is available for transmission, no provision is made for a fill character, and XMTON is turned ON.

5) RBRL = 1.RBRL (receiver buffer register loaded) is set when a complete character is transferred from the receiver shift register to the receiver buffer register.

6) RHRL = 1.RHRL (receiver holding register loaded) is set in mode 1 when the receiver receives a complete frame.

7) RABRT = 1.RABRT (receiver abort) is set in mode 1 when the FLAG character is detected and seven consecutive ones are received.

Interrupts are enabled in the SCC by writing a one to the associated enable bit (see Section 2.1.1). Figure 24 shows the logical equivalent of the TMS 9903 interrupt circuitry.



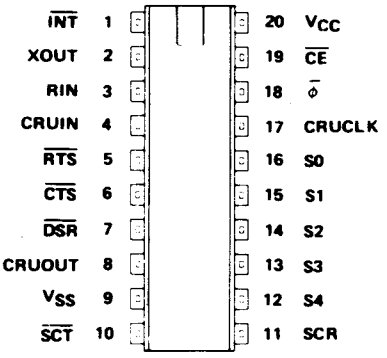
NOTE: See Tables 1 and 5 for input and output signal definitions.

FIGURE 24. INTERRUPT GENERATION LOGIC

2.7 TMS 9903 TERMINAL ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
$\overline{INT}$	1	OUT	Interrupt. When active (low), the $\overline{INT}$ output indicates that at least one of the interrupt conditions has occurred.
XOUT	2	OUT	Transmitter serial data output line.
RIN	3	IN	Receiver serial data input line.
CRUIN	4	OUT	Serial Data Output line fromTMS 9903 to CRUIN input line of the CPU.
$\overline{RTS}$	5	OUT	Request to Send output from TMS 9903 to modem. This output is enabled by the CPU and remains active (low) during data transmission from TMS 9903.
$\overline{CTS}$	6	IN	Clear-to-Send input from modem to TMS 9903. When active (low), it enables the transmitter section of the TMS 9903.
$\overline{DSR}$	7	IN	Data Set Ready input from modem to TMS 9903. This input generates an interrupt when going On or Off.
CRUOUT	8	IN	Serial data input line to TMS 9903 from CRUOUT line of the CPU.
V_{SS}	9	IN	Ground Reference Voltage.
$\overline{SCT}$	10	IN	Transmit Clock — Transmitter data is shifted out on one-to-zero transition of $\overline{SCT}$.
SCR	11	IN	Receiver Clock — Receiver serial data (RIN) is sampled at zero-to-one transition of SCR.
S4(LSB)	12	IN	Address bus S0-S4 are the lines that are addressed by the CPU to select a particular TMS 9903 function.
S3	13	IN	
S2	14	IN	
S1	15	IN	
S0(MSB)	16	IN	
CRUCLK	17	IN	CRU Clock. When active (high), TMS 9903 samples the input data on CRUOUT line.
$\overline{\phi}$	18	IN	TTL Clock.
$\overline{CE}$	19	IN	Chip Enable — When $\overline{CE}$ is inactive (high), the TMS 9903 address decoding is inhibitied. CRUIN remains at high impedance when $\overline{CE}$ is inactive (high).
V_{CC}	20	IN	Supply voltage (+5 V nominal).

TMS 9903 PIN ASSIGNMENTS
20 PIN DUAL-IN-LINE PACKAGE
(TOP VIEW)



3. DEVICE APPLICATION

This section describes the software interface between the CPU and the TMS 9903 and discusses some of the design considerations in the use of this device in synchronous and asynchronous communications applications.

3.1 DEVICE INITIALIZATION

The following discussions assume that the value to be loaded into the CRU base register (register 12) in order to point to select bit 0 of the TMS 9903 is 0040_{16}, and the $\overline{\phi}$ input to the SCC is a 4-MHz signal. The SCC divides this signal by four to generate an internal clock frequency of 1 MHz. An interrupt is generated by the interval timer every 1.6 milliseconds when timer interrupts are enabled.

When power is applied, the SCC must be initialized by the CPU with the instruction sequence shown below. The actual data (i.e., CTRL) loaded into the control register and specific initialization requirements are application-dependent and are further described in the following discussions of individual mode operations. If CRC is to be used, it is necessary to include the CRC register initialization commands CLRCRC and CLXCRC, (as in the example for mode 1 operation).

```
RESET     EQU     31
CLRRCV    EQU     30
CLRXMT    EQU     30
CTRL      DATA    >XXXX

LI        R12,>40          Initialize CRU Base.
SBO       RESET            Issue RESET command which resets the TMS 9903 and sets
                           the LDCTRL — Load Control Register — flag.
LDCR      @CTRL, 12        Load the control register, automatically resetting LDCTRL.
SBZ       CLRRCV           Initialize Receiver.
SBO       CLRXMT           Initialize Transmitter.
```

The RESET command resets all flags (other than LDCTRL), resets all output bits, and disables all interrupts. The contents of the XBR, XCRC, RCRC, RHR, RBR, SYNC1, SYNC2, and the interval register are unaffected.

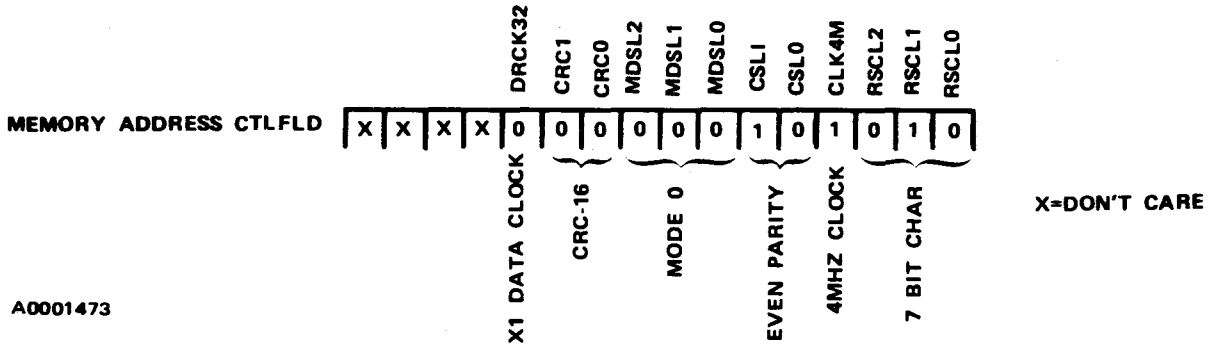
The receiver should be initialized with the CLRRCV command after the control register is loaded to ensure that the receiver logic will assemble the first received character at the proper length.

The transmitter should be initialized with the CLRXMT command after the control register is loaded to ensure that the transmitter logic will operate according to the flowchart for the selected mode.

3.1.1 Mode 0 Operation [General]

Mode 0 operation is the most unstructured of the TMS 9903 operating modes, placing all synchronization and control requirements on the CPU. It functions as the basic subset of all other modes of operation and, as such, can be used in essentially any control protocol the user desires, limited only by the ability of the user software to provide the necessary control. The following instruction sequence will set the TMS 9903 to operate in mode 0. Note that in mode 0, the receiver is continually shifting in bits as long as SCR is present.

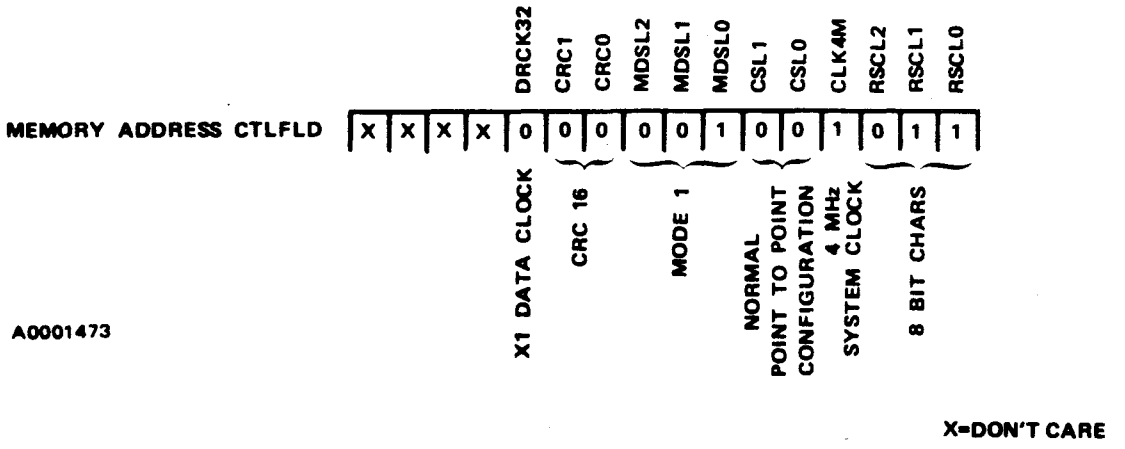
```
RESET     EQU     31
CLRRCV    EQU     30
CLRXMT    EQU     30
LDSYN2    EQU     27
XPRNT     EQU     23
          .
          .
          .
          LI      R12,>40          Initialize CRU Base.
          SBO     RESET            Reset device and set LDCTRL.
          LDCR    @ CTLFLD,12      Load Control Register and Reset LDCTRL.
          SBZ     CLRRCV           Initialize Receiver.
          SBO     CLRXMT           Initialize Transmitter
          SBO     LDSYN2   }
          LDCR    @ SYNC2,8 }      Load Sync Character Register 2.
          SBZ     LDSYN2   }
          .
          .
          .
SYNC2     BYTE    >16              ASCII Sync Character
CTLFLD    DATA    >002A
```



3.1.2 Mode 1 Operation [SDLC]

Mode 1 operation is selected to support the synchronous data link control (SDLC) protocol. SDLC supports full duplex communication links and places no constraints on the communications codes involved in information transfer. SDLC operation is initialized with the software shown below. This software sets the TMS 9903 to operate in mode 1 with eight-bit characters. The TMS 9903 further allows SDLC operation in several configurations — point-to-point, multipoint, loop master, loop slave, etc. In this case, operation is in the point-to-point configuration as set up by the configuration select bits shown. As in the case described for Bi-Sync operation (mode 3), user software will then handle message preparation, transmission, reception, and accountability, while the TMS 9903 message link handles synchronization and control.

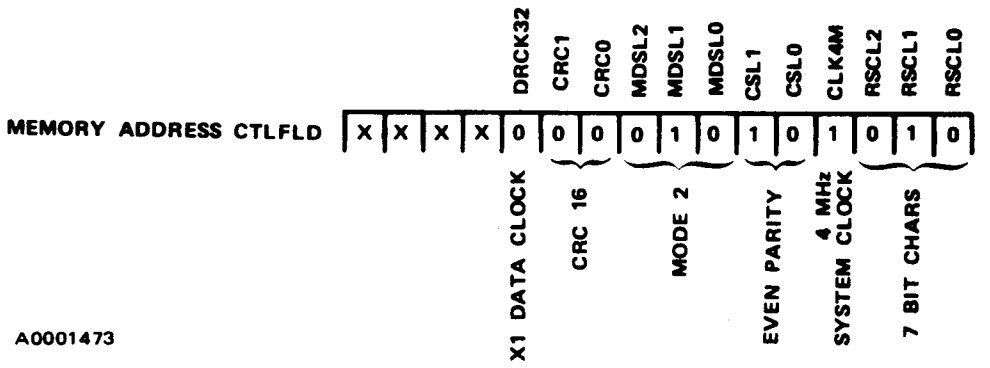
```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
LDSYN2  EQU   27
CLXCRC  EQU   29
CLRCRC  EQU   29
LXCRC   EQU   24
LRCRC   EQU   12
        :
        LI    R12,>40
        SBO   RESET       Reset Device
        LDCR  @ CTLFLD,12 Load Control Register
        SBZ   CLRRCV      Initialize Receiver
        SBO   CLRXMT      Initialize Transmitter
        SBO   LDSYN2      }
        LDCR  @ SYNC2,8   } Load Fill Character Into Sync
        SBZ   LDSYN2      } Character Register 2
        SBO   CLXCRC      Clear XMT CRC Register to all zeroes
        SBZ   CLRCRC      Clear RCV CRC Register to all zeroes
        SBO   LXCRC       }
        LDCR  @ INIB1,8   } Initialize Transmitter
        LDCR  @ INIB2,8   } CRC Registers to all Ones
        SBZ   LXCRC       }
        SBO   LRCRC       }
        LDCR  @ INIB1,8   } Initialize Receiver CRC
        LDCR  @ INIB2,8   } Registers to all Ones
        SBZ   LRCRC       }
        :
SYNC2   BYTE  >11
CTLFLD  DATA  >004B       Sync Character
INIB1   BYTE  >57
INIB2   BYTE  >15
```



3.1.3 Mode 2 Operation [General]

Mode 2 operation provides the framework for a general communication link control protocol using a character contained in SYNC1 for initial synchronization, and a character contained in SYNC2 for a fill sequence in the absence of data to be transmitted (XBRE = 1). The instruction sequence shown below will initialize the TMS 9903 to operate in mode 2.

```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
LDSYN2  EQU   27
LDSYN1  EQU   26
        .
        .
        .
        LI    R12,>40      Initialize CRU Base
        SBO   RESET        Reset SCC and set LDCTRL
        LDCR  @ CTLFLD,12  Load Control Register and Reset LDCTRL
        SBZ   CLRRCV       Initialize Receiver
        SBO   CLRXMT       Initialize Transmitter
        SBO   LDSYN2   }
        LDCR  @ SYNC2,8 } Load Fill Character in SYNC2
        SBZ   LDSYN2   }
        SBO   LDSYN1   }
        LDCR  @ SYNC1,8 } Load Sync Character in SYNC1
        SBZ   LDSYN1   }
        .
        .
        .
SYNC1   BYTE  >16
SYNC2   BYTE  >11
CTLFLD  DATA  >00AA
```

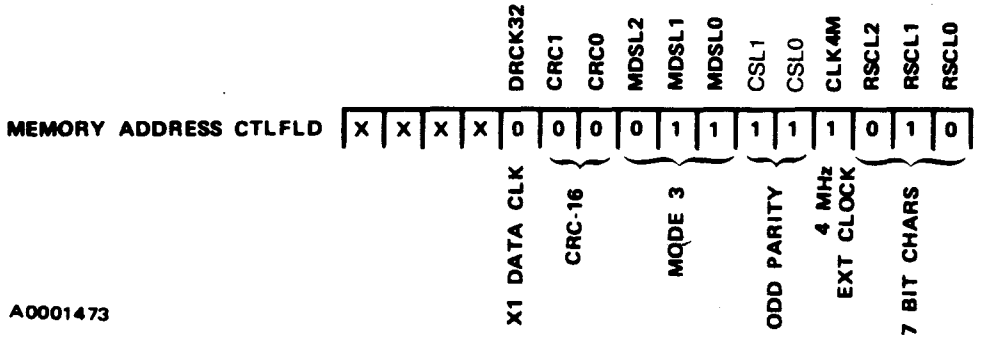


3.1.4 Mode 3 Operation [Bi-Sync]

One of the most common synchronous data link control protocols now in use is Bi-Sync, which uses a fixed character length set of data and control characters and half-duplex operation. Bi-Sync operation is invoked with the software shown below. The software instructions shown load the control register with bits set to initialize the TMS 9903 to operate in mode 3 with received character length of seven bits and odd parity.

Note that transmitted character length is determined dynamically from the length of the character loaded into the transmitter buffer. Hence, transmitting fixed seven-bit characters from the CPU to the TMS 9903, with odd parity generation selected and enabled, automatically generates the fixed length eight-bit characters required for Bi-Sync transmission. In normal operation the TMS 9903 will automatically insert SYN characters into the bit stream (from the SYNC1 register) whenever the transmitter buffer is empty and no character has been loaded by the CPU. In receive operation with RSYNDL set, the TMS 9903 will delete all SYN characters embedded in the received character stream.

```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
RSYNDL  EQU   28
LDSYN2  EQU   27
LDSYN1  EQU   26
        :
        :
        LI    R12,>40
        SBO   RESET       Issue Reset Command and Set Load
                          Control Flag LDCTRL
        LDCR  @ CTLFLD,12 Load Control Register with 12 Bits, the
                          Last of Which Resets LDCTRL
        SBZ   CLRRCV      Initialize the Receiver
        SBO   CLRXMT      Initialize the Transmitter
        SBO   LDSYN1      }
        LDCR  @ SYNC1,8   } Load SYNC1 Register
        SBZ   LDSYN1      }
        SBO   LDSYN2      }
        LDCR  @ SYNC2,8   } Load SYNC2 Register
        SBZ   LDSYN2      }
        SBO   RSYNDL      Set RCVR to Delete SYNC Characters
                          (XPRNT = 1 will Override RSYNDL)
        :
        :
SYNC1   BYTE  >16         ASCII "SYN" Character
SYNC2   BYTE  >10         ASCII "DLE" Character
CTLFLD  DATA  >00FA       Sets TMS 9903 for Mode 3, Odd Parity, 7 Bit Characters
```



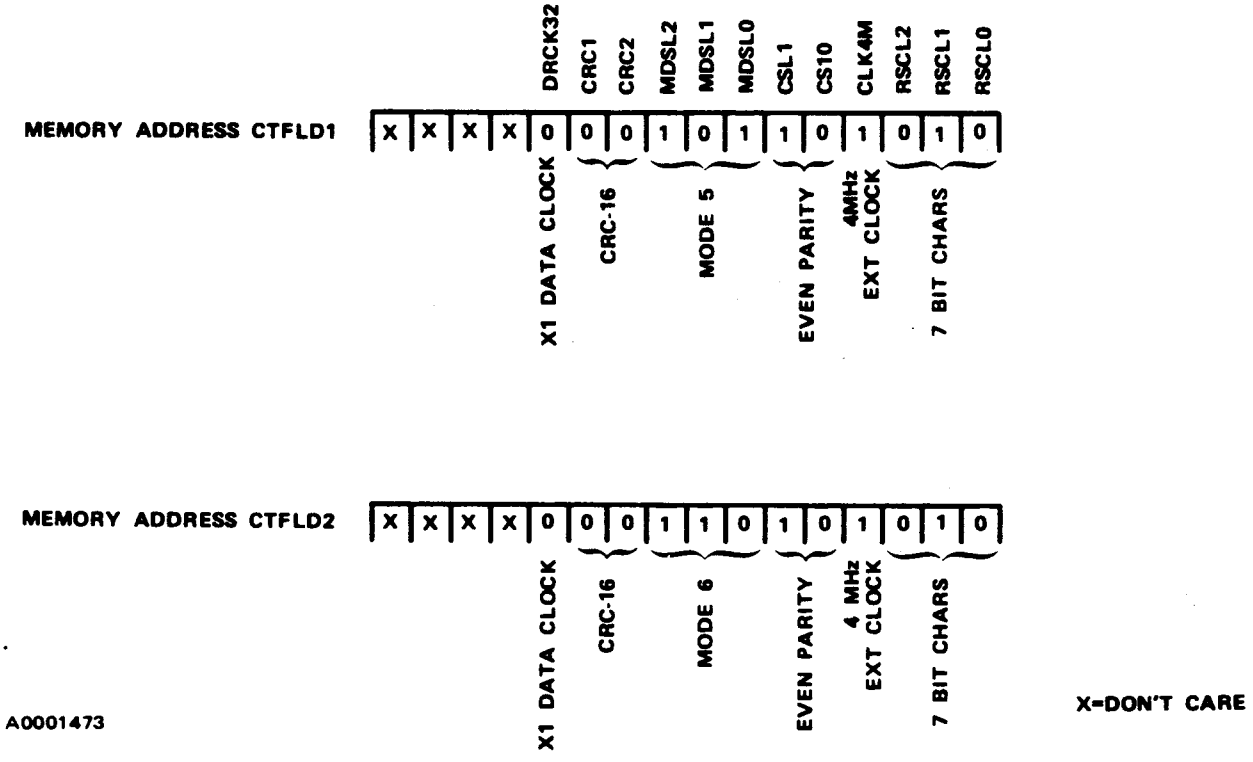
If the capability to utilize all bit combinations of the eight-bit data field is required, control bit XPRNT can be set for transparent operations. This will cause the SYNC1-SYNC1 fill sequence (i.e., normally SYN-SYN) to be replaced with SYNC2-SYNC1 (i.e., DLE SYN). Note that in transparent operation, more software is required to ensure that all data-link control commands are preceded by the DLE (data-link escape) character.

User software routines then will handle the preparation, transmission, reception, and accountability of individual messages, with link synchronization and control done by the TMS 9903.

3.1.5 Mode 5 and 6 Operation [Asynchronous]

Modes 5 and 6 are the asynchronous operation modes of the TMS 9903. Mode 5 provides operation with one start and two stop bits, and mode 6 with one start and one stop bit. The scftware shown below will initialize the TMS 9903 into mode 5 or 6 asynchronous operation mode, depending upon the mode select bits. Loading the control register with the contents of memory address CTFLD1 selects mode 5 and CTFLD2 selects mode 6.

```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
        .
        .
        .
        LI    R12,>40      Initialize CRU Base
        SBO   RESET        Reset SCC and Set LDCTRL
        LDCR  @ CTFLDX,12  Load Control Register and Reset LDCR
        SBZ   CLRRCV       Initialize Receiver
        SBO   CLRXMT       Initialize Transmitter
        .
        .
        .
CTFLD1  DATA  >016A        Two Stop Bits
CTFLD2  DATA  >01AA        One Stop Bit
```



3.1.6 Interval Timer Operation

The software shown below will set up the interval timer to generate an interrupt every 1.6 milliseconds. The value loaded into the interval register specifies the number of 64-microsecond increments in the total interval.

```
TIMENB    EQU     20
LDIR      EQU     13
INTVL     BYTE    >19          19(16) = 25(10), 25 x 64 μs = 1.6 ms
          .
          .
          .
          SBO     LDIR         Set Load Interval Register Flag
          LDCR    @ INTVL,8    Load IR with 25 Increments
          SBZ     LDIR         Reset LDIR
          SBO     TIMENB       Enable Interval Timer Interrupts
          .
          .
          .
```

3.1.7 Interval Register Loading After Initialization

The interval register may be reloaded after initialization. For example, to change the interval of the timer to 10.24 milliseconds, the instruction sequence is

```
          SBO     13           Set Load Control Flag
          LDCR    @ INTVL2,8   Load Register
          SBZ     13           Reset Flag
          .
          .
          .
INTVL2    BYTE    10240/64
```

Caution should be exercised when transmitter interrupts are enabled to ensure that the transmitter interrupt does not occur while the load control flag is set. For example, if the transmitter interrupts between execution of the "SBO 13" and the next instruction, the transmit buffer is not enabled for loading when the transmitter interrupt service routine is entered because the LDIR flag is set. This situation may be avoided by the following sequence:

```
          BLWP    @ITVCHG      Call Subroutine
          .
          .
          .
ITVCPC    LIMI    0            Mask All Interrupts
          MOV     @ 24(R13),R12 Load CRU Base Address
          SBO     13           Set Flag
          LDCR    @INTVL2,8    Load Register
          SBZ     13           Reset Flag
          RTWP                 Restore Mask and Return
          .
          .
          .
ITVCHG    DATA    ACCWP,ITVCPC
INTVL2    BYTE    10240/64
```

In this case all interrupts are masked, ensuring that all interrupts are disabled while the load control flag is set.

3.2 DATA TRANSMISSION

The software* shown below demonstrates a representative subroutine for transmitting a block of data.

```
        LI      R0,LISTAD   Initialize List Pointer
        LI      R1,COUNT    Initialize Block Count
        LI      R12,CRUBAS  Initialize CRU BASE
        SBO     XMTON       Turn on Transmitter (XMTON = 16)
XMTLP   TB      XBRE        Xmit Buffer Empty?
        JNE     XMTLP       No, Wait
        SBO     LXBC
                            Load Transmitter Buffer, Transmit CRC
                            Register, and Increment Pointer
        LDCR    *R0+,8
        SBZ     LXBC        Reset XBRE (LXBC = 25)
        DEC     R1          Decrement Counter
        JNE     XMTLP       Loop If Not Complete
        SBO     LXCRC       Set LXCRC to
        STCR    R3,0        Read Transmitter CRC
        SBZ     LXCRC       Reset LXCRC
        SWPB    R3          Get Lower Bits of CRC
        TB      XBRE        Transmit Buffer Empty?
        JNE     $-1         No, Wait
        LDCR    R3,8        Transmit Lower CRC Byte
        SBZ     LXBC        Reset LXBC
        SWPB    R3          Get Upper Half CRC
        TB      XBRE        Transmit Buffer Empty?
        JNE     $-1         No, Wait
        LDCR    R3,8        Transmit Upper CRC Byte
        SBZ     LXBC
        SBZ     XMTON       Turn Off Transmitter
```

After initializing the list pointer, block count, and CRU base address, XMTON is set, enabling data transmission. The internal autotmatic RTS signal (RTSAUT) becomes active and transmission begins when $\overline{CTS}$ becomes active. Each character to be transmitted is loaded with the LXBC flag set to load the transmitter buffer and to update simultaneously the transmitter CRC register. If the CRC register is not in use, the characters can be loaded with no flags set, which will then load only the transmitter buffer. After the last character is transmitted, the accumulated CRC is read from the SCC and transmitted, and XMTON is reset. The transmitter and $\overline{RTS}$ become inactive upon completion of transmission of the last character. Note that $\overline{RTS}$ can be CPU-controlled by setting and resetting RTS (select bit 17). This disables the RTSAUT signal until the transmitter is reset by the RESET or CLRXMT command. This routine is written for use with eight bit characters.

*The software in these examples represents generalized routines. Specific details will vary with the mode of operation selected.

3.3 DATA RECEPTION

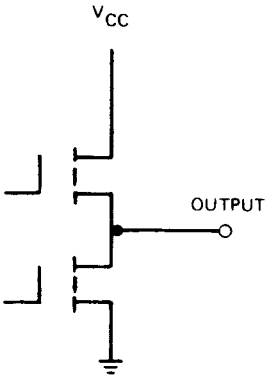
The software shown below will cause a block of data to be received and stored in memory.

```
        LI      R1,TEMPT    Initialize Working Storage
        LI      R2,RCLST    Initialize List Address
        LI      R3,MAXCNT   Initialize Max Count
        LI      R4,>0D00    Initialize End of Block Character (ASCII CR)
RCVLP   TB      21          Receiver Buffer Register Loaded?
        JNE     RCVLP       No, Wait
        STCR    *R2,8       Store Character
        SBZ     18          Reset RBRL
        SBO     12          Set LRCRC to
        LDCR    *R2,8       . Update Receiver CRC Register
        SBZ     12          Reset LRCRC
        DEC     R3          Decrement Count
        JEQ     RCVEND      Test if Count = 0
        CB      *R2+,R4     Compare to EOB Character and Increment Pointer
        JNE     RCVLP       Loop If Not Complete
RCVEND  TB      21          Character Received?
        JNE     RCVEND      No, Wait
        STCR    R1,8        Store Transmitted CRC Value
        SBZ     18          Reset RBRL
        SWPB    R1          Swap CRC Bytes
        TB      21          CRC High Byte Received?
        JNE     $-1         No, Wait
        STCR    R1,8        Store Transmitted CRC Value
        SBZ     18          Reset RBRL
        SBO     12          Set LRCRC to
        STCR    R6,0        Read Receiver CRC Register
        SBZ     12          Reset LRCRC
        C       R1,R6       If Received CRC Not Equal to
        JNE     ERR         Expected CRC, Jump to Error Routine
        RTWP                Else Return
        .
        .
        .
```

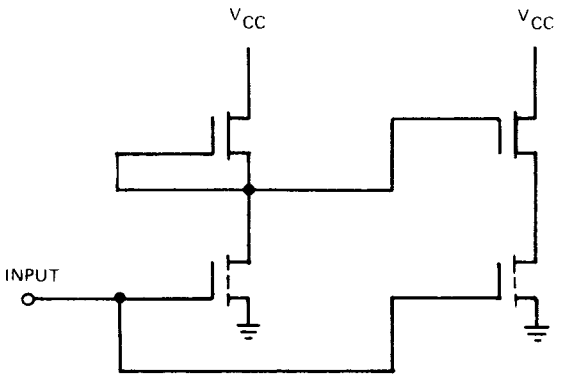
The above routine receives the block of data and compares the received CRC block check to the value accumulated in the receiver CRC register. Note that in mode 1 operation the RCVEND instructions to read the received CRC could be replaced with:

```
RCVEND  SBO     26          Set RHRRD
        STCR    R1,0        Read the Receiver Holding Register
        SBZ     26          Reset RHRRD
        SBO     12          Set LRCRC
        STCR    R6,0        Read Receiver CRC Register
        SBZ     12          Reset LRCRC
        C       R1,R6       Compare
        JNE     ERR         Jump to Error Routine If Not Equal
        .
        .
        .
```

EQUIVALENT OF OUTPUTS



EQUIVALENT OF INPUTS



4. TMS 9903 ELECTRICAL SPECIFICATIONS

4.1 ABSOLUTE MAXIMUM RATING OVER OPERATING FREE AIR TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)*

Supply voltage, V_{CC} (Note) ..... −0.3 V to 10 V
All inputs and output voltages ..... −0.3 V to 20 V
Continuous power dissipation ..... 0.7 W
Operating free-air temperature range ..... 0°C to 70°C
Storage temperature range ..... −65°C to 150°C

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute maximum rated conditions for extended periods may affect device reliability.

4.2 RECOMMENDED OPERATING CONDITIONS

PARAMETER	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5.0	5.25	V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH}	2.0	2.4	V_{CC}	V
Low-level input voltage, V_{IL}	V_{SS} − .3	0.4	0.8	V
Operating free-air temperature, T_A	0		70	°C

4.3 ELECTRICAL CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS (UNLESS OTHERWISE NOTED)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = −100 μA I_{OH} = −200 μA	2.4 2.2		V_{CC}	V
V_{OL}	Low-level output voltage	IOL = 3.2 mA	V_{SS}		0.4	V
I_I	Input current (any input)	V_I = 0 V to V_{CC}		±10		μA
$I_{CC(av)}$	Average supply current from V_{CC}	$t_{c(\phi)}$ = 330 ns, t_A = 70°C		150		mA
C_i	Capacitance, any input	f = 1 MHz, all other pins at 0 V		15		pF

NOTE: All voltages are in reference to V_{SS}.

4.4 TIMING REQUIREMENTS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

	PARAMETER	MIN	TYP	MAX	UNIT
$t_{c(\phi)}$	Clock cycle time	300	333		ns
$t_{r(\phi)}$	Clock rise time		5		ns
$t_{f(\phi)}$	Clock fall time		10		ns
$t_{w(\phi H)}$	Clock pulse width (high level)	225			ns
$t_{w(\phi L)}$	Clock pulse width (low level)	45			ns
$t_{w(CC)}$	CRUCLK pulse width	100	185		ns
t_{su1}	Setup time for CE before CRUCLK	190			ns
t_{su2}	Setup time for S0-S4 or CRUOUT before CRUCLK	220			ns
t_h	Hold time for S0-S4, CE, or CRUOUT after CRUCLK	60			ns

4.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

	PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
t_{pD1}	Propagation delay, CE to valid CRUIN	C_L = 100 pF		300		ns
t_{pD2}	Propagation delay, S0-S4 to valid CRUIN	C_L = 100 pF		320		ns

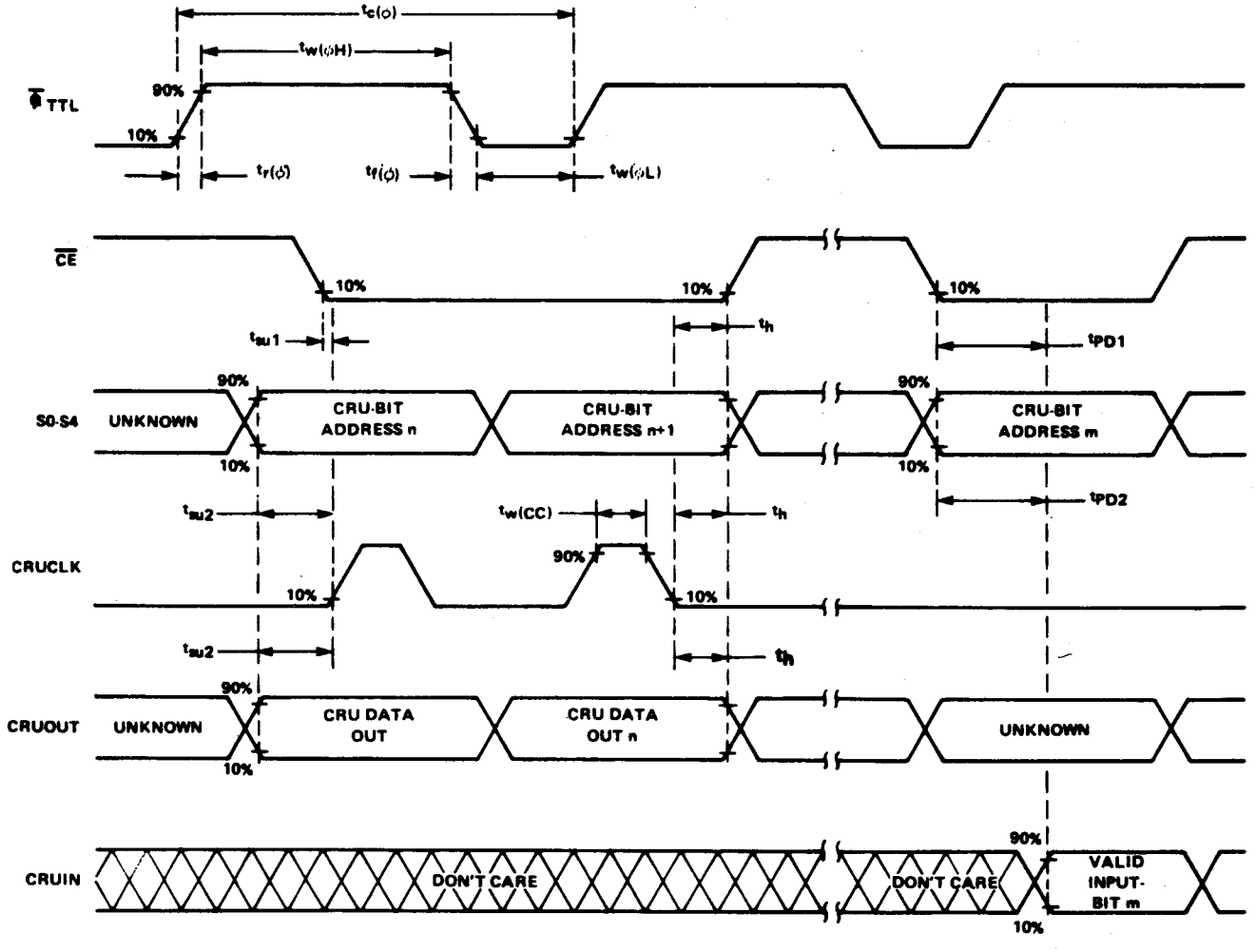
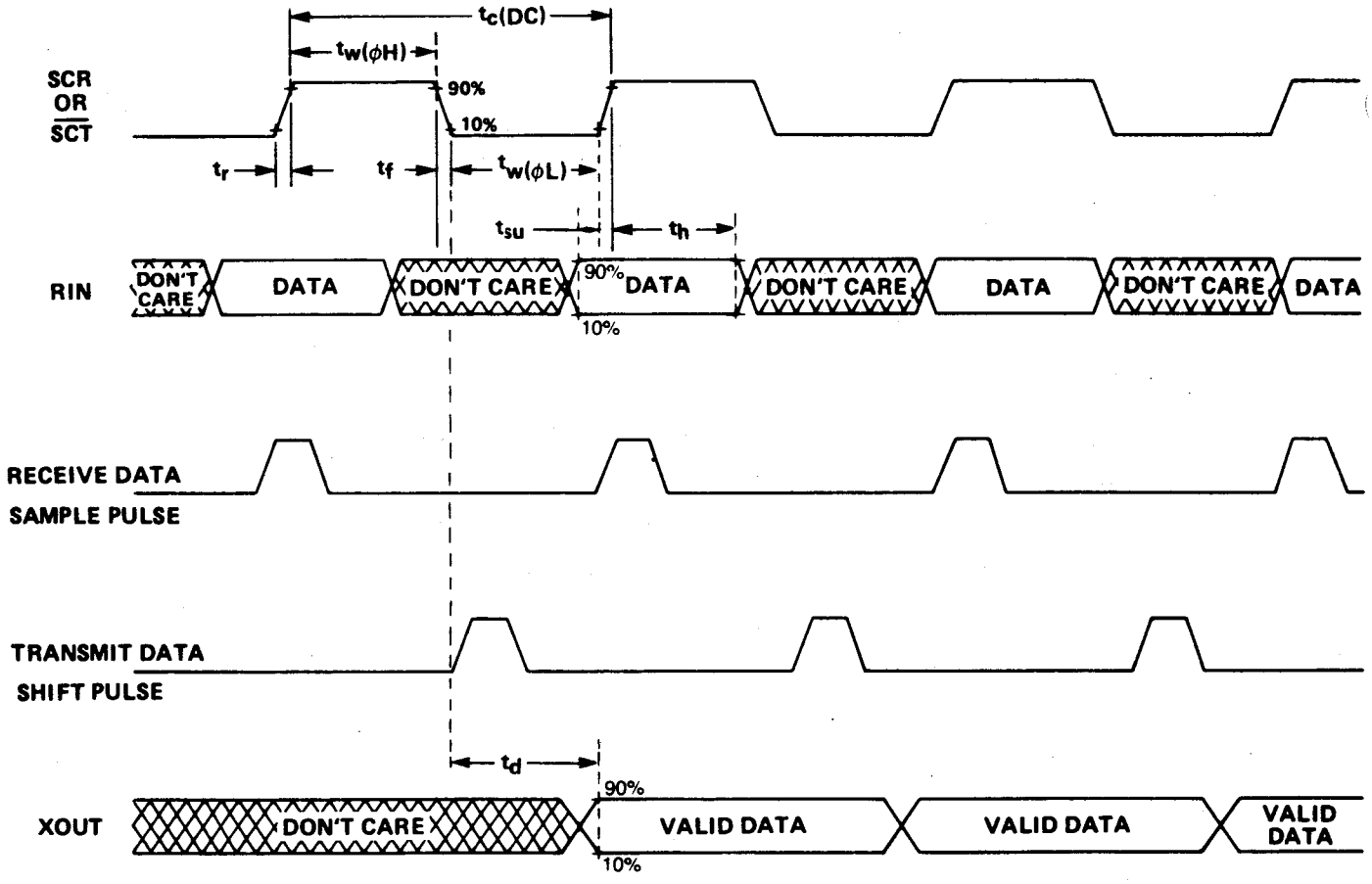


FIGURE 25. TIMING DIAGRAM



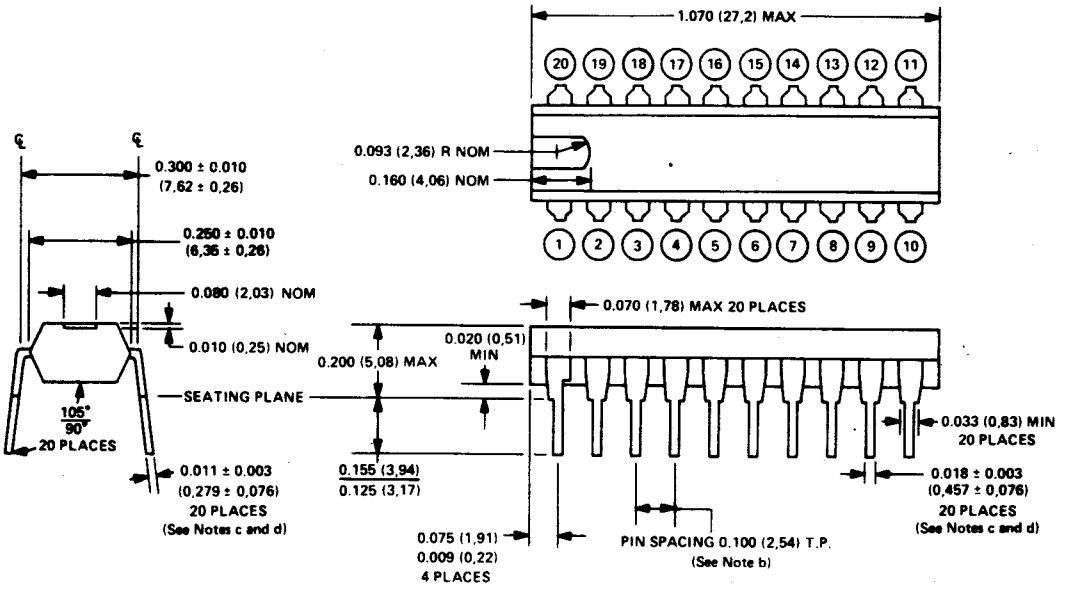
	PARAMETER	MIN	TYP	MAX	UNIT
$t_{c(DC)}$	Receiver/transmit data clock cycle time		4		μs
$t_{w(\phi H)}$	Clock pulse width (high level)		2		μs
$t_{w(\phi L)}$	Clock pulse width (low level)		2		μs
t_r	Rise time		12		ns
t_f	Fall time		12		ns
t_{su}	Setup time for RIN before SCR (DRCK32 = 0)*		250		ns
t_h	Hold time for RIN after SCR (DRCK32 = 0)*		50		ns
t_d	Delay time, SCT to valid XOUT		400		ns

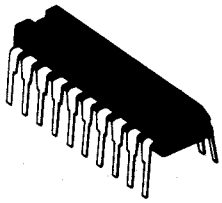
*No setup, hold, or data synchronization is required for pin in the divide-by-32 mode (DRCK32 = 1).

FIGURE 26. RECEIVE/TRANSMIT DATA CLOCK TIMING DIAGRAM

5. MECHANICAL DATA

20-PIN N PLASTIC





NOTES: a. All dimensions are shown in inches (and parenthetically in millimeters for reference only). Inch dimensions govern.

b. Each pin centerline is located within 0.010 (0,26) of its true longitudinal position.

c. This dimension does not apply for solder-dipped leads.

d. When solder-dipped leads are specified, dipped area of the lead extends from the lead tip to at least 0.020 (0,50) above the seating plane.

ABBREVIATIONS AND ACRONYMS

ADCCP	advanced data communication control protocol
ADDR	address
A0-A14	address lines
BRKON	break on
CE	chip enable, low active
CLK4M	system clock divide by four select
CLR	clear
CLRCRC	clear receiver CRC register
CLRRCV	clear receiver
CLRXMT	clear transmitter
CLXCRC	clear transmitter CRC register
CPU	central processor unit
CRC	cyclic redundancy check
CRC1 CRC2	CRC polynonical select
CRU	communications register unit
CRUCLK	CRU clock, strobe from CPU to tell when CRUOUT is valid
CRUIN	CRU input to CPU
CRUOUT	CRU output from CPU
CSL1 CSL2	configuration select
CTRL___	control register
CTS, $\overline{CTS}$	clear to send
DBIN	data bus in
DRCK32	32 times data rate clock enable
DSCENB	data set status change interrupt enable
DSCINT	data set status change interrupt
DSCH___	data set status change
DSR, $\overline{DSR}$	data set ready
EOP	end of poll
HDLC	high-level data link communication
INIT	initialize
$\overline{INT}$	interrupt, low active
IR(0) IR(7)	interval register
LDCTRL	load control register
LDIR	load interval register
LDSYN1	load sync character register 1
LDSYN2	load sync character register 2
LRCRC	load receiver CRC register
LSB	least significant bit
LXBC	load transmitter buffer and transmitter CRC register
$\underline{LXCRC}$	load transmitter CRC register
$\overline{MEMEN}$	memory enable, low active
MDSL0 - MDSL2	mode select bits of control register
MSB	most significant bit
NRZI	non-return to zero inverted
PSI	programmable system interface, TMS 9901
RABDT	receiver abort detect
RABRT	receiver abort
RBCNT	receiver bit count
RBR	receiver buffer register

RBRL	receiver buffer register loaded
RCRC	receiver CRC register
RCVERR	receiver error
RFBD	receiver full bit detect
RFER	receiver framing error
RFLDT	receiver flag detect
RFLG	receiver flag
RHR	receiver holding register
RHRL	receiver holding register loaded
RHROV	receiver holding register overrun
RHRRD	receiver holding register read
RIENB	receiver interrupt enable
RIN	serial receive data
RINT	receiver interrupt
ROCNT	receiver ones count
ROVER	receiver overrun
RMSK	receiver mask register
RPAR	receiver parity
RPER	receiver parity error
RSBD	receiver start bit detect
RSCL0 - RSCL2	receiver character length select
RSR	receiver shift register
RSYNDL	receiver sync character delete
RSYNEQ	receiver shift register/sync register equal
RTS,$\overline{\text{RTS}}$	request to send
RTSAUT	request to send automatic
RZER	receiver zero error
R51D	receiver five consecutive ones detect
SBD	start bit detect
SCC	synchronous communications controller
SCR	synchronous clock for receiver
$\overline{\text{SCT}}$	synchronous clock for transmitter
SCTX	transmitter clock negative transition
SCRX	receiver clock positive transition
SHF	shift
SDLC	synchronous data link control
SYNC1	sync character one
SYNC2	sync character two
S0 - S4	select lines
TIMELP	timer elapsed
TIMENB	timer interrupt enable
TIMERR	timer error
TIMINT	timer interrupt
TSTMD	test mode
UPD	update
$\overline{\text{WE}}$	write enable, low active
XABRT	transmitter abort
XAIENB	transmitter abort interrupt enable
XBCNT	transmitter bit count
XBIENB	transmitter buffer register empty interrupt enable
XAINT	transmitter abort interrupt
XBINT	transmitter buffer register empty interrupt
XBR	transmitter buffer register
XBRE	transmitter buffer register empty
XCRC	transmitter CRC register
XOCNT	transmitter ones count

XMTON	transmitter on
XOUT	transmitter serial data out
XPAR	transmitter parity
XPRNT	transparent
XSCL	transmitter shift register character length
XSR	transmitter shift register
XSYNCL	transmitter sync character length
XZINH	transmitter zero insertion inhibit
$\overline{\phi}3$	inverted TTL clock to TMS 9903 (usually from TIM 9904)

APPENDIX E

NOISE SUSCEPTABILITY OF LOCAL LINE MODULE

E.1 GENERAL

This appendix reports results of tests run on a compatable module, the Local Line Module (LLM) manufactured by the Digital Systems Group of Texas Instruments. The TM 990/308 follows the same design as the LLM, and the information contained herein can be applied to the TM 990/308.

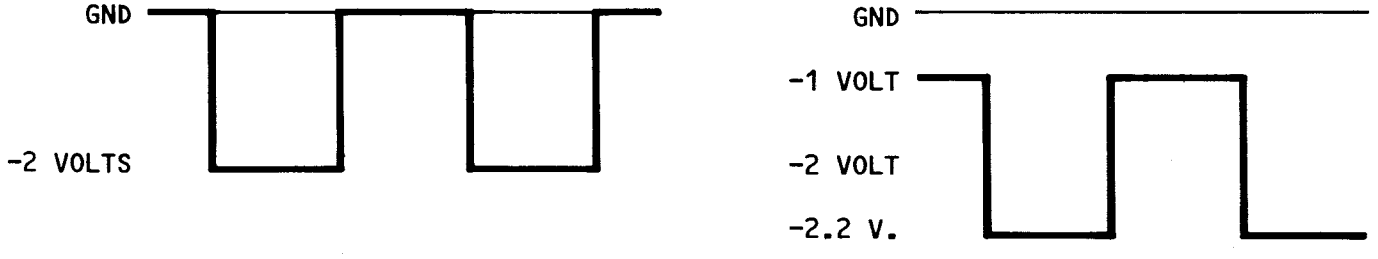
These tests were run with a Velonix Model 510 surge transient generator used to apply noise to the twisted pair and shield of the multidrop industrial interface cable. Noise spike rise time was 100 ns. Tests were run with the LLM executing diagnostic software. Noise was increased slowly and a check was made for board failure under the following conditions:

- Test 1: With and without line terminations installed on the board.
- Test 2: Noise applied to the data-line pair only with the shield attached to earth ground.
- Test 3: Noise applied to the data-line pair only with the shield open between LMs.
- Test 4: Noise applied to the data-line pair and shield with the shield isolated from earth ground by a capacitor.
- Test 5: Noise applied to the data-line pair and shield with the shield not attached to earth ground.

E.2 TEST RESULTS

Figures E-1 through E-5 show test setup and results respectively for Tests 1 to 5. Test equipment used:

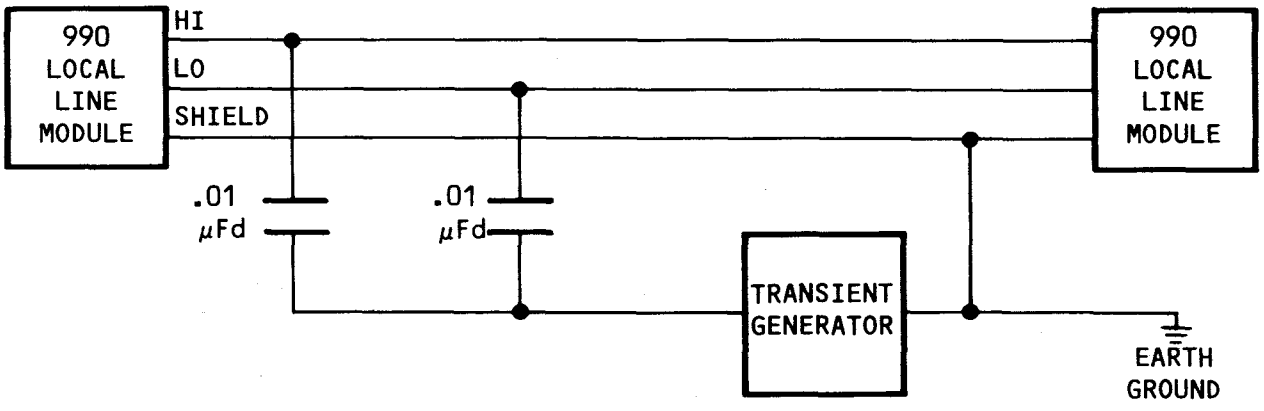
- Velonix Model 510 Surge Transient Generator, 100 ns noise rise time.
- Tektronix Model 475A Oscilloscope



(a) Terminations Installed (b) Terminations Removed

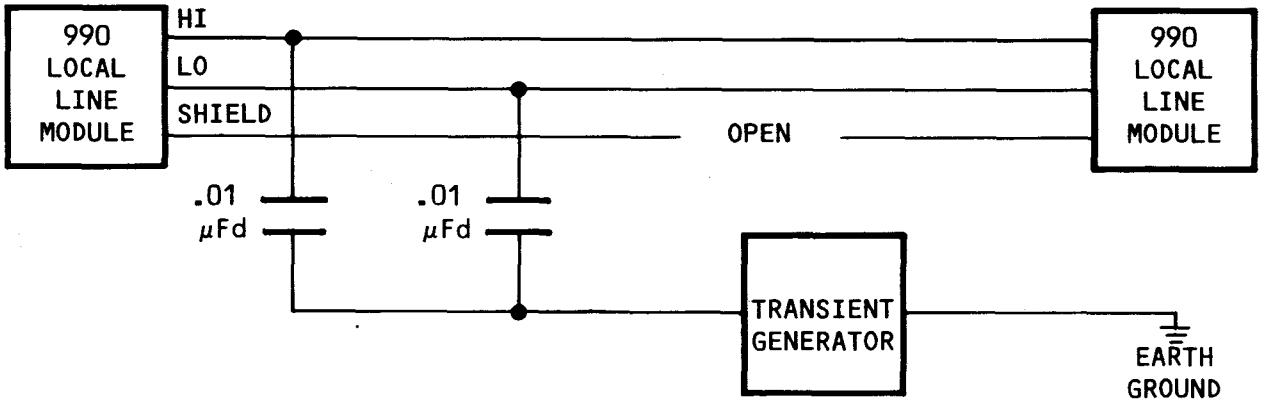
NOTE: Without line terminations, signal amplitude decreased from 2 V to 1.2 V and relative position to ground was lowered by a volt. The LLM passed all diagnostics with the terminations removed from both boards. Still, it is recommended that line terminations be installed to help maintain signal amplitude in relation to ground.

FIGURE E-1. TEST ONE: EFFECTS OF LINE TERMINATIONS INSTALLED AND REMOVED



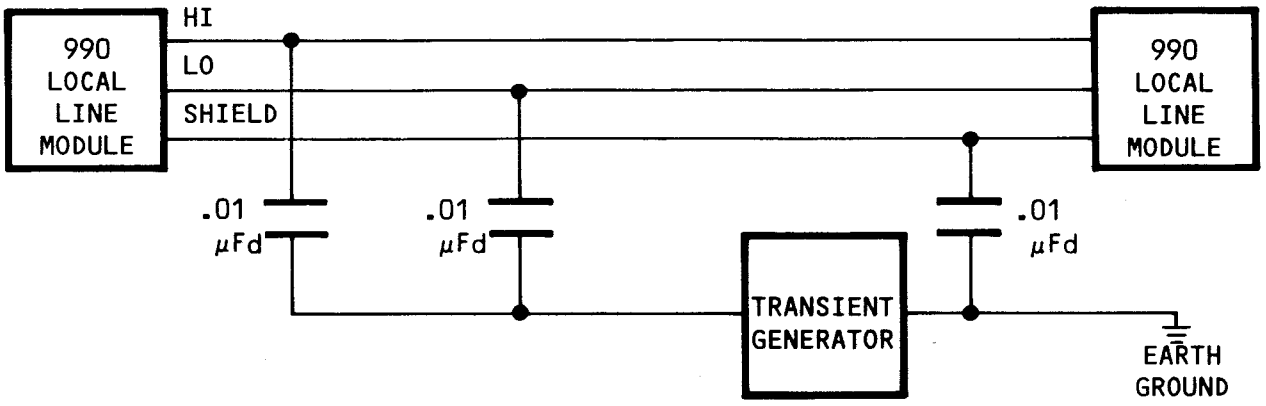
NOTE: Diagnostic error occurred around noise levels of 140 to 150 V.

FIGURE E-2. TEST TWO



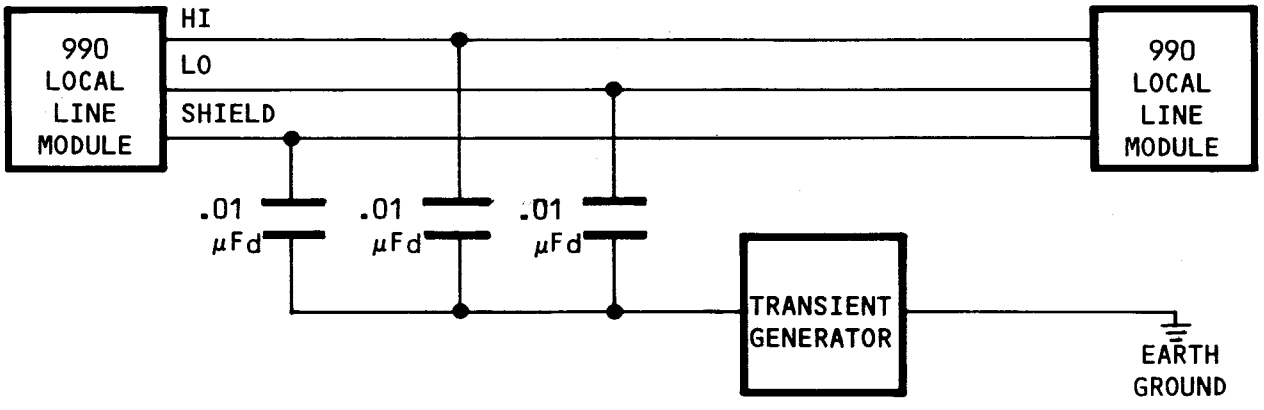
NOTE: Diagnostic error occurred around noise levels of 190 to 200 V.

FIGURE E-3. TEST THREE



NOTE: Diagnostic error occurred around noise levels of 100 to 120 V.

FIGURE E-4. TEST FOUR



NOTE: Diagnostic CRC and timeout errors occurred around noise levels of 180 to 190 V. This figure reflects the type of situation which will most likely occur -- noise being coupled onto all three lines from an outside source.

FIGURE E-5. TEST FIVE

INDEX

In the page number list of this alphabetical index, subject matter is covered by a table if a T precedes the page number or is covered by a figure if an F precedes the page number.