The Engineering Staff of
TEXAS INSTRUMENTS INCORPORATED
Semiconductor Group

TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER DATA MANUAL

DECEMBER 1978

TEXAS INSTRUMENTS
INCORPORATED

IMPORTANT NOTICES

Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

TI cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.

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.

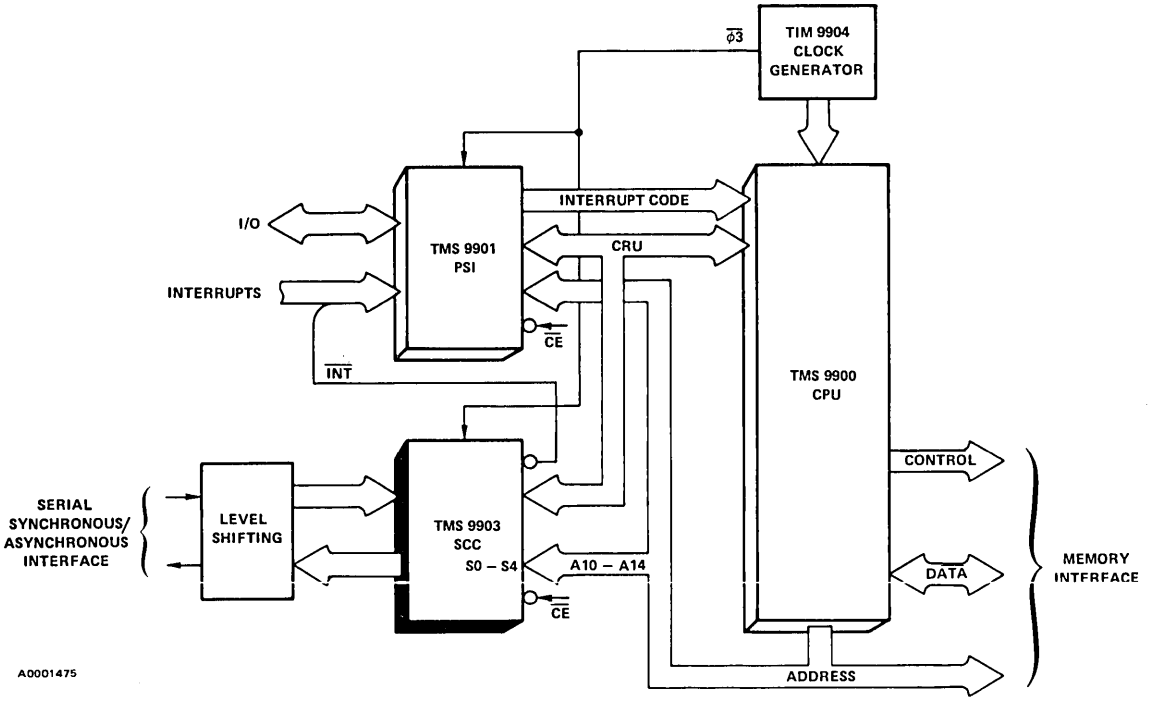


FIGURE 1. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER IN A TMS 9900 SYSTEM

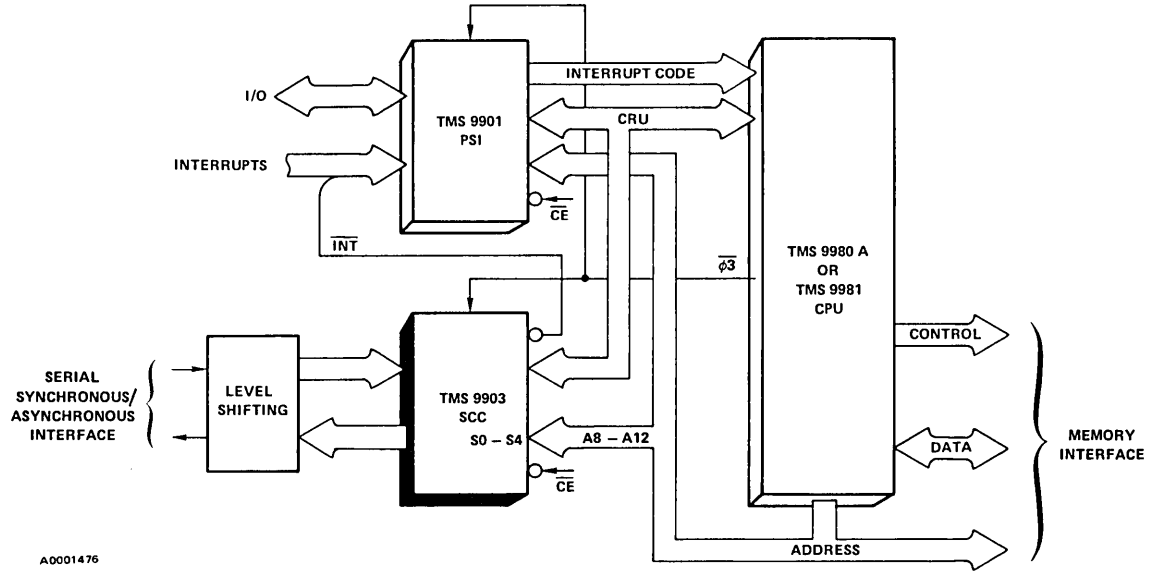


FIGURE 2. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER IN A TMS 9980 A, 9981 SYSTEM

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.

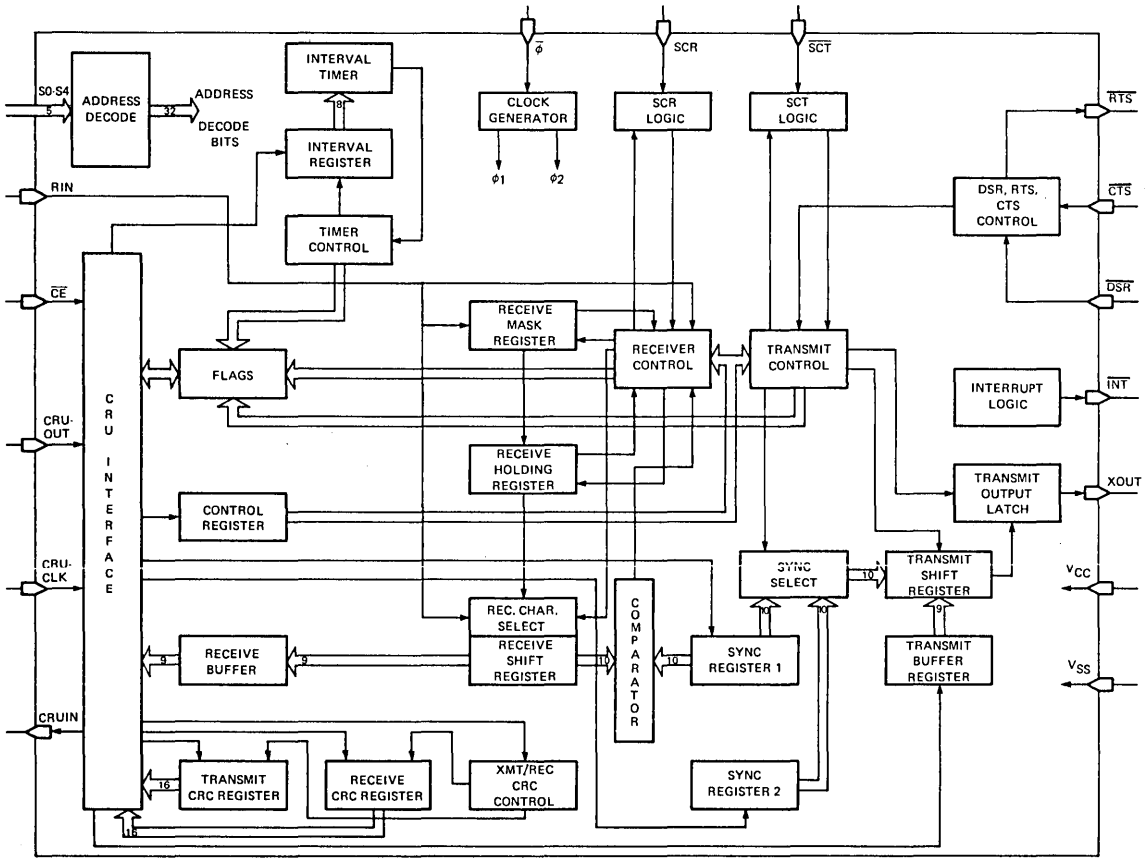
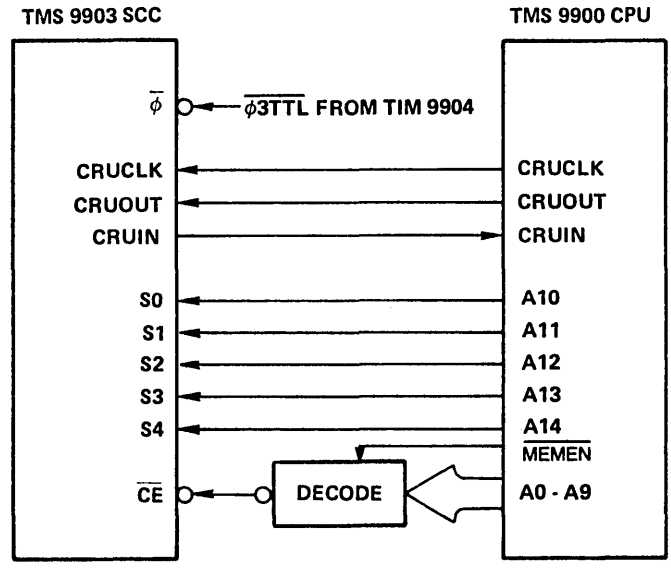
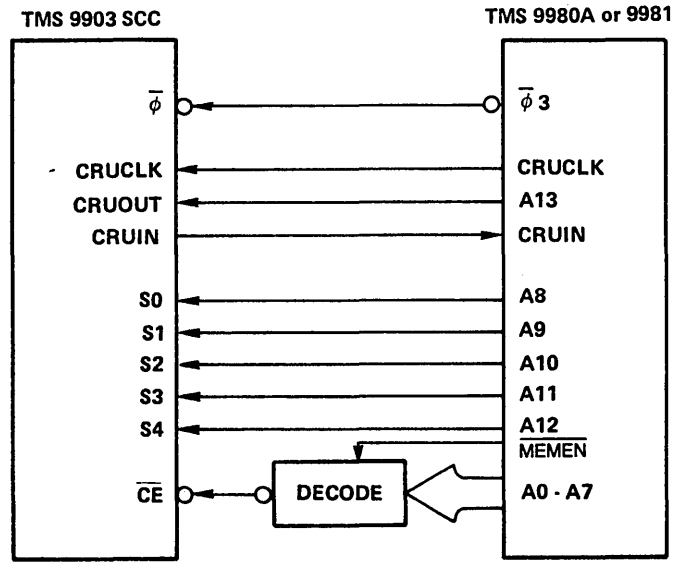


FIGURE 3. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER BLOCK DIAGRAM



A0001478

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



A0001479

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.

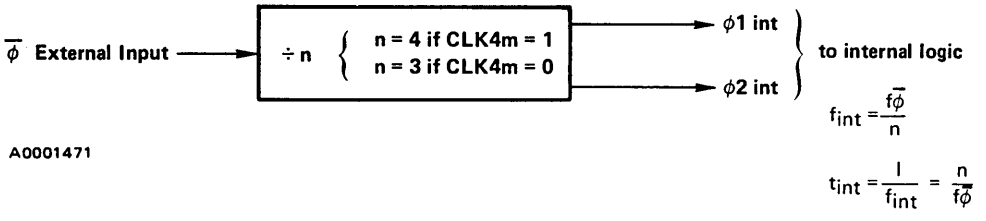
TRANSMITTER/RECEIVER CONFIGURATION BIT SELECT

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.



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.

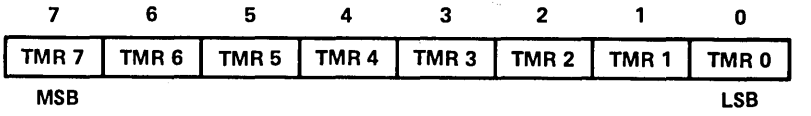
RECEIVE CHARACTER LENGTH SELECTION

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.

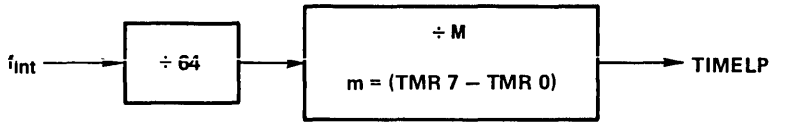


A0001472

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

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



A0001472

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.

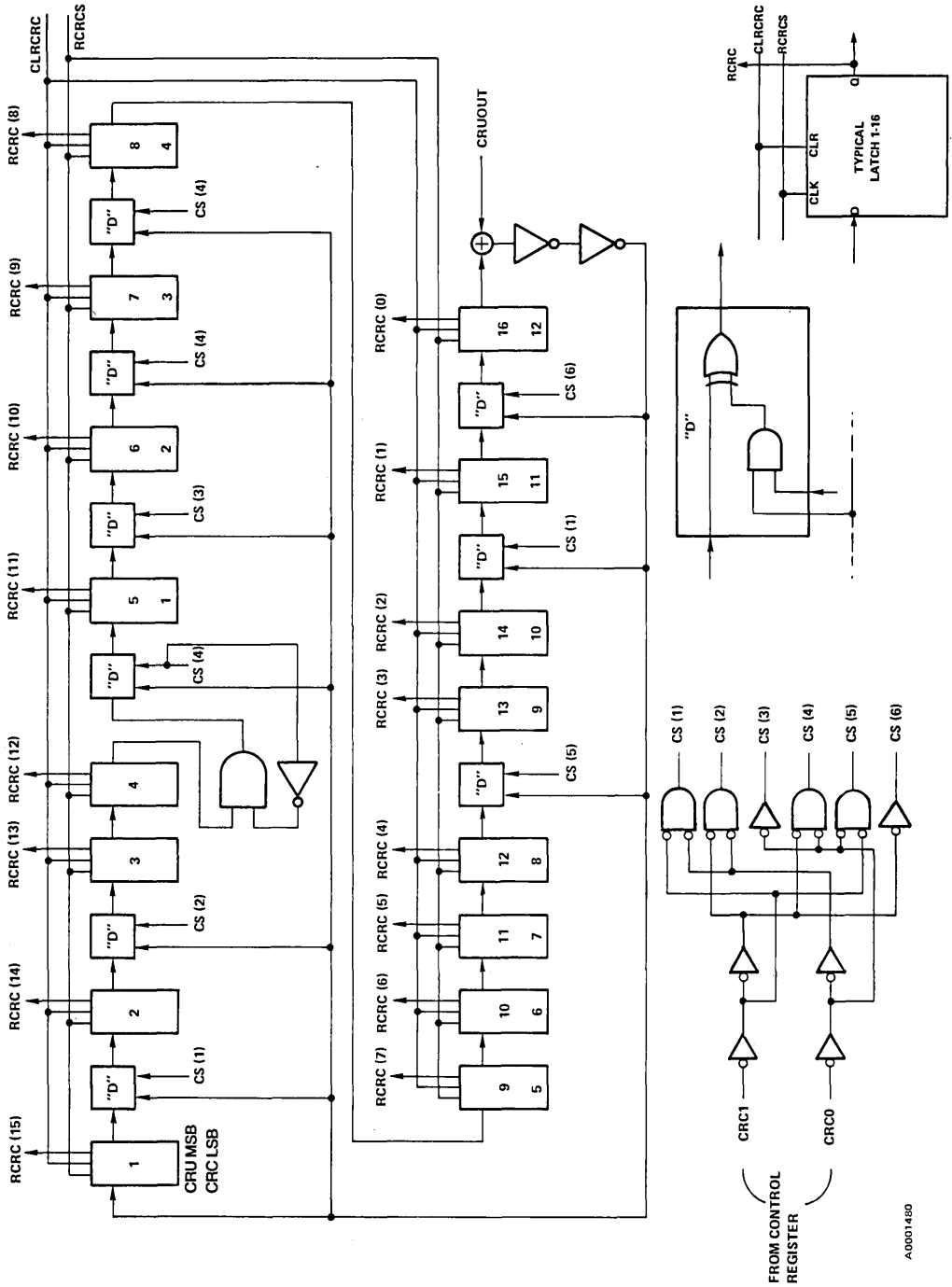


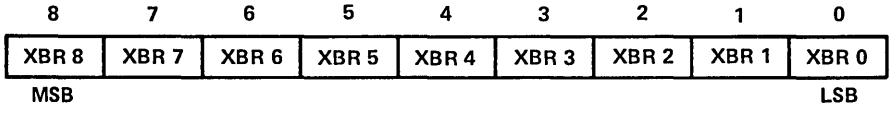
FIGURE 6. RECEIVE CRC REGISTER DIAGRAM

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.



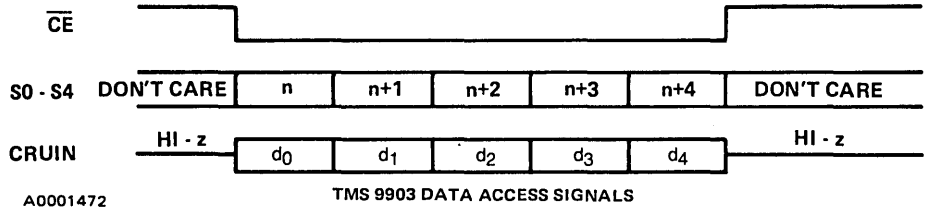
TRANSMIT BUFFER REGISTER BIT ADDRESSES

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.

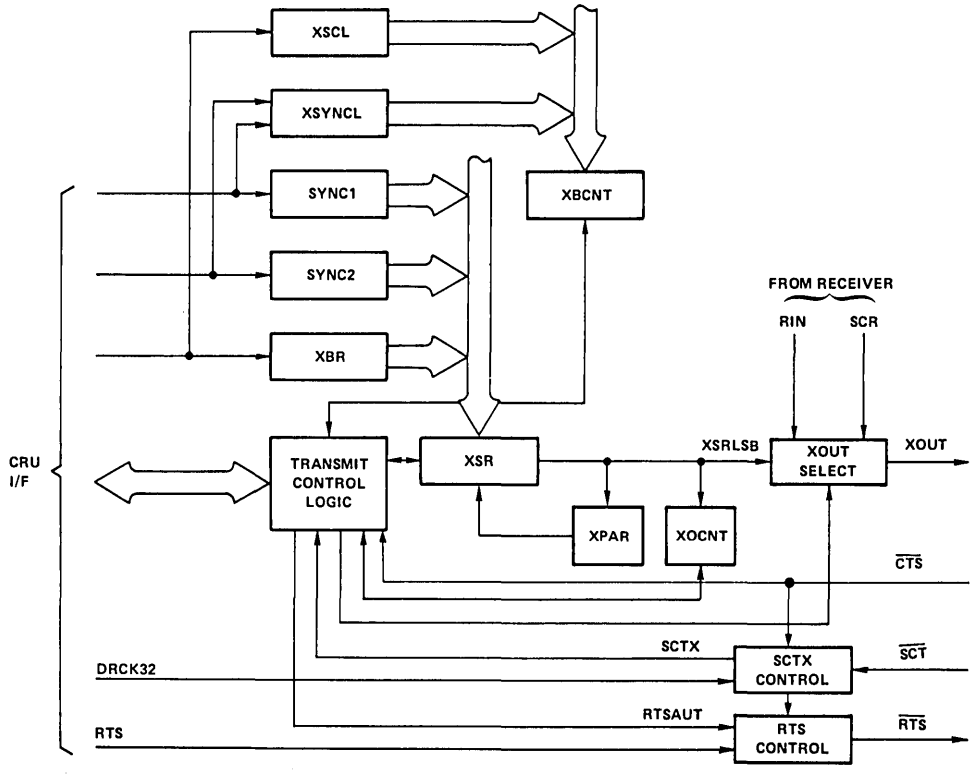


FIGURE 7. TMS 9903 SCC TRANSMITTER BLOCK DIAGRAM

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)

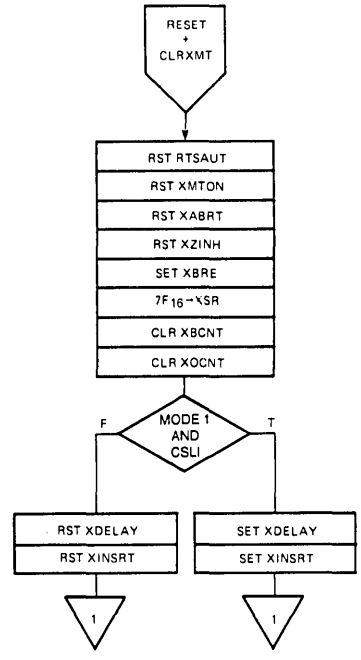


FIGURE 8. TRANSMITTER INITIALIZATION

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.

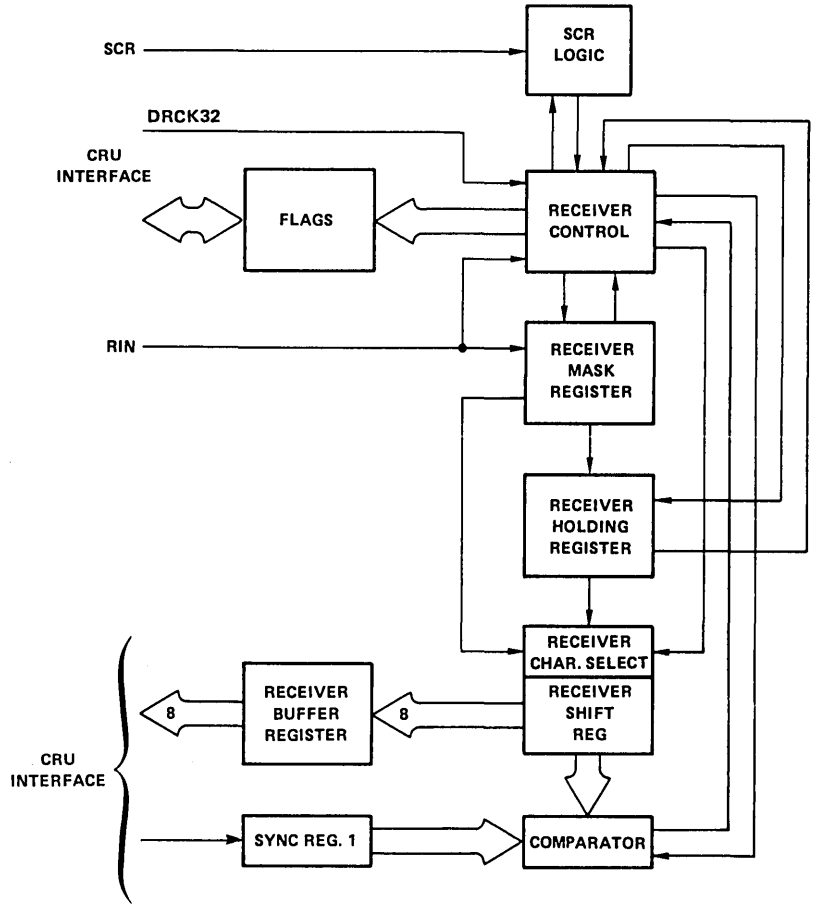


FIGURE 9. TMS 9903 RECEIVER BLOCK DIAGRAM

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.

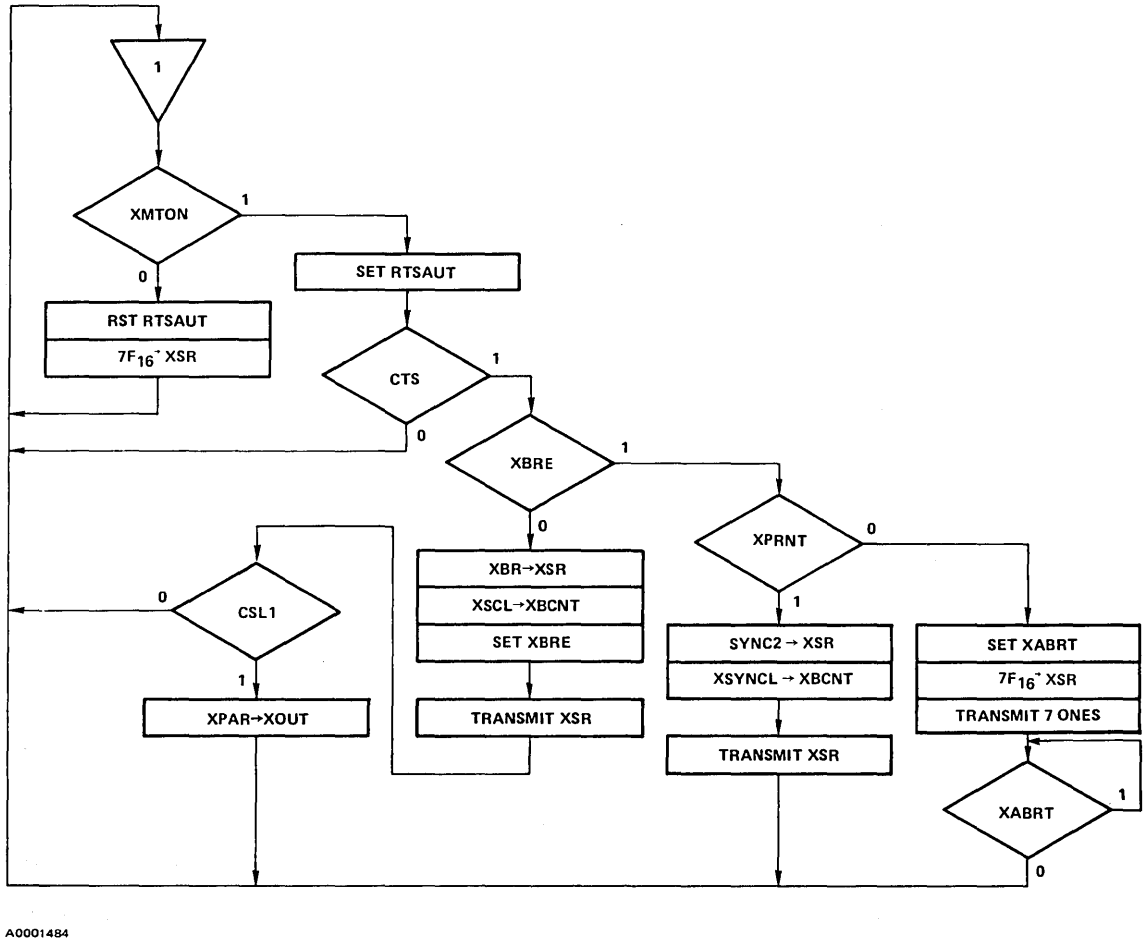
2.4 TRANSMITTER AND RECEIVER OPERATION

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.



A0001484

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.

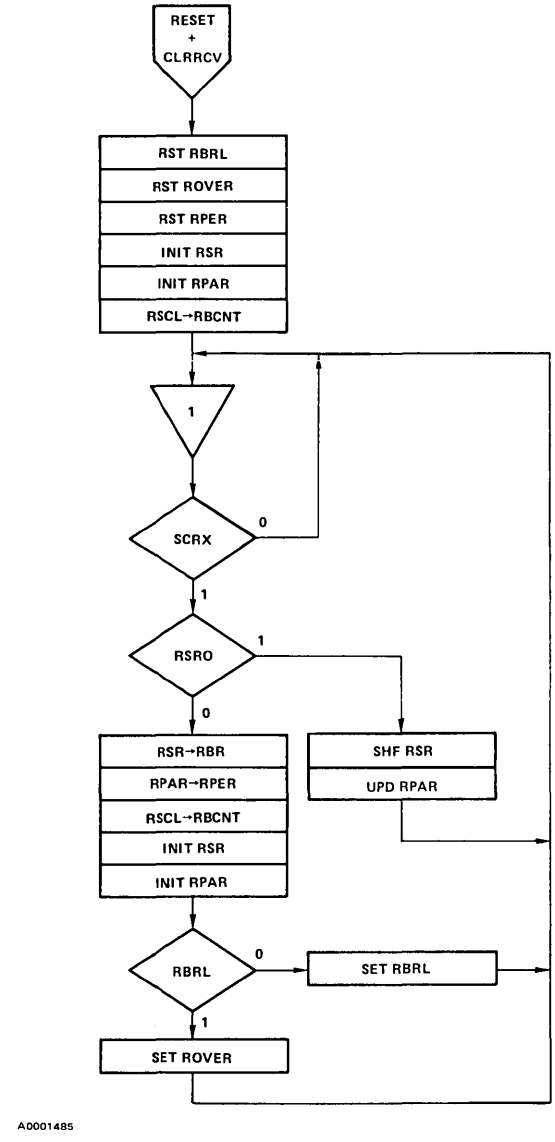


FIGURE 11. MODE 0 RECEIVER OPERATION

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).

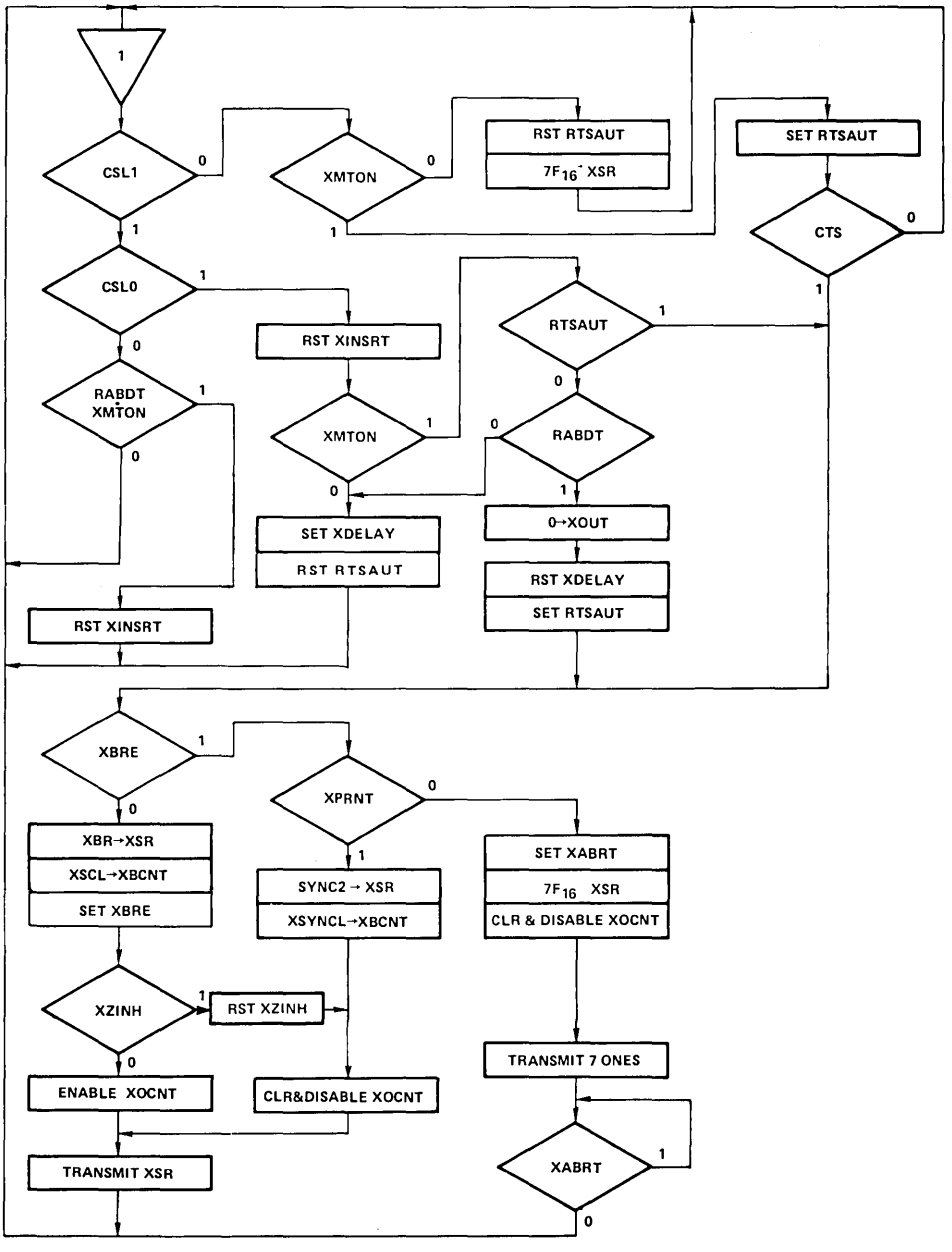


FIGURE 12. MODE 1 TRANSMITTER OPERATION

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.

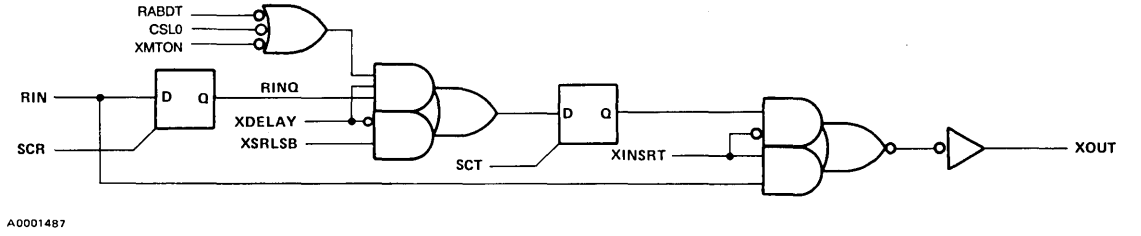


FIGURE 13. SIMPLIFIED XOUT SELECT LOGIC

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.

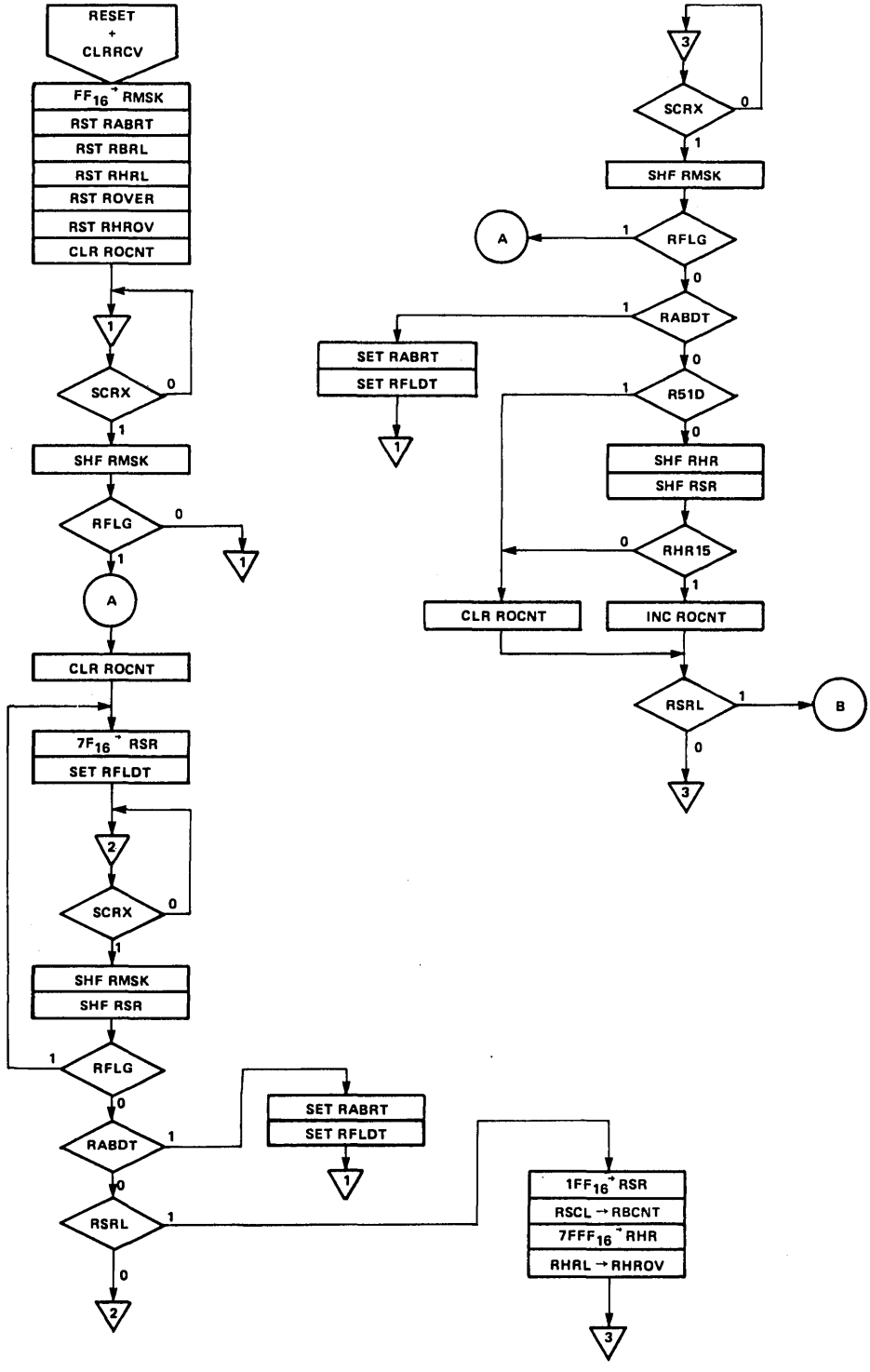


FIGURE 14. MODE 1 RECEIVER OPERATION (PAGE 1 OF 2)

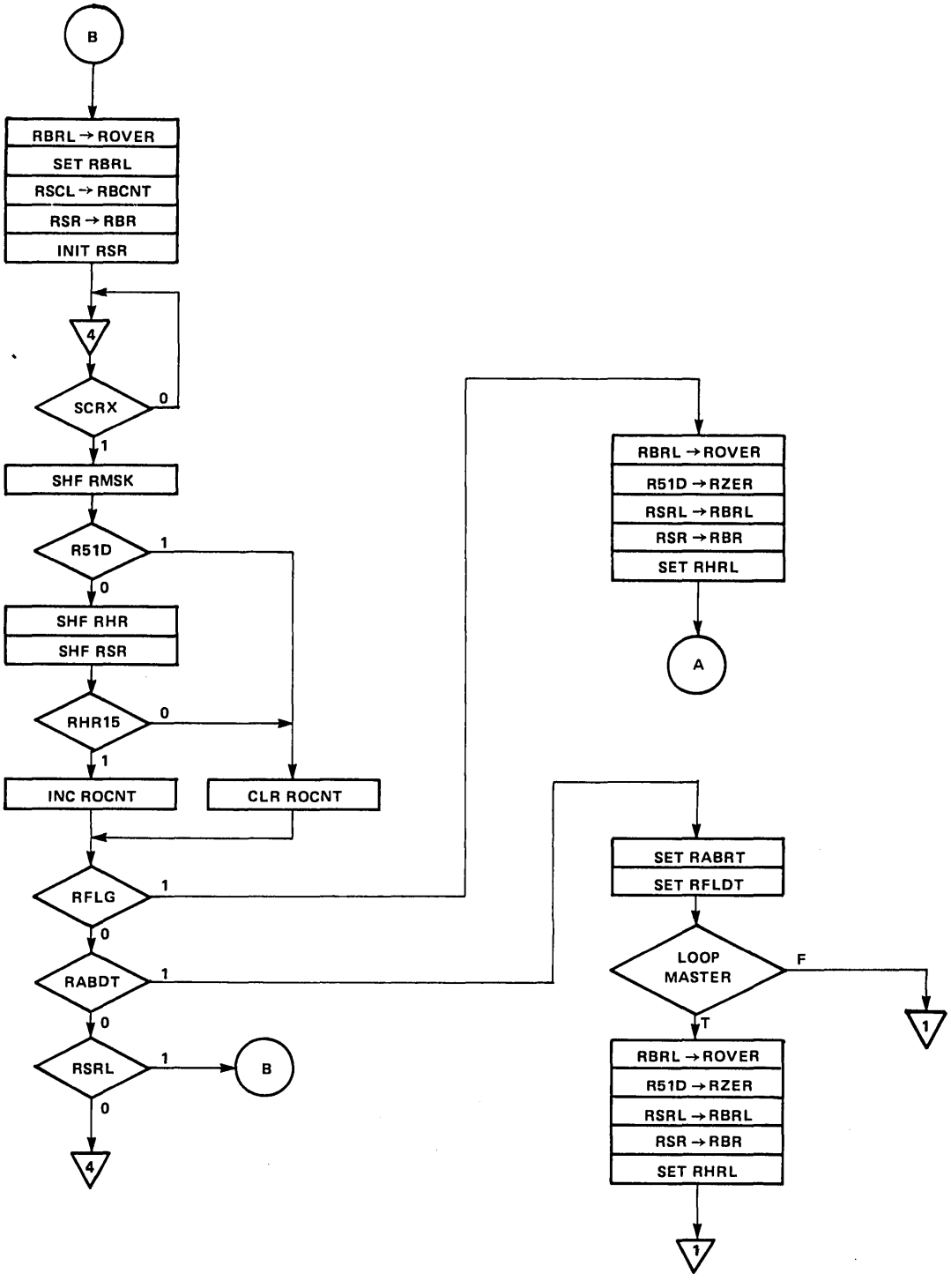


FIGURE 14. MODE 1 RECEIVER OPERATION (PAGE 2 OF 2)

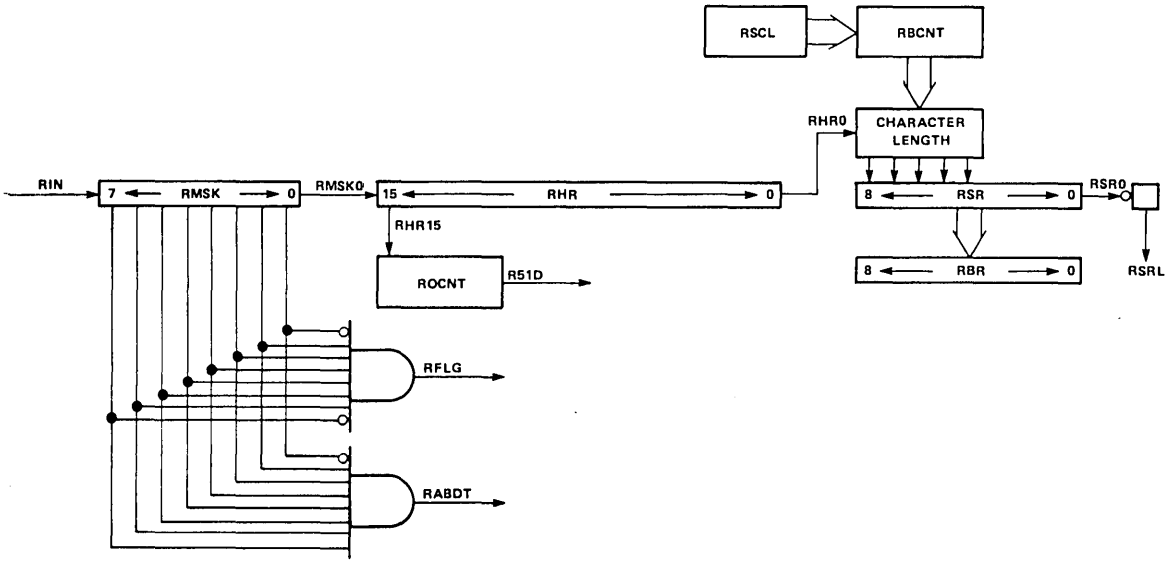


FIGURE 15. MODE 1 RECEIVER CIRCUITRY

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 000111111_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.

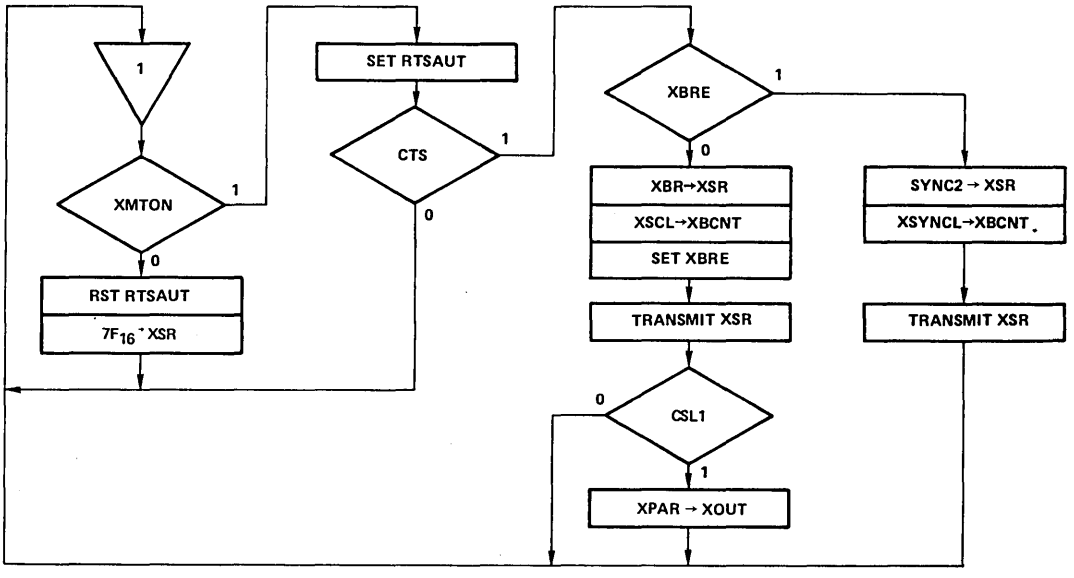


FIGURE 16. MODE 2 TRANSMITTER OPERATION

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.

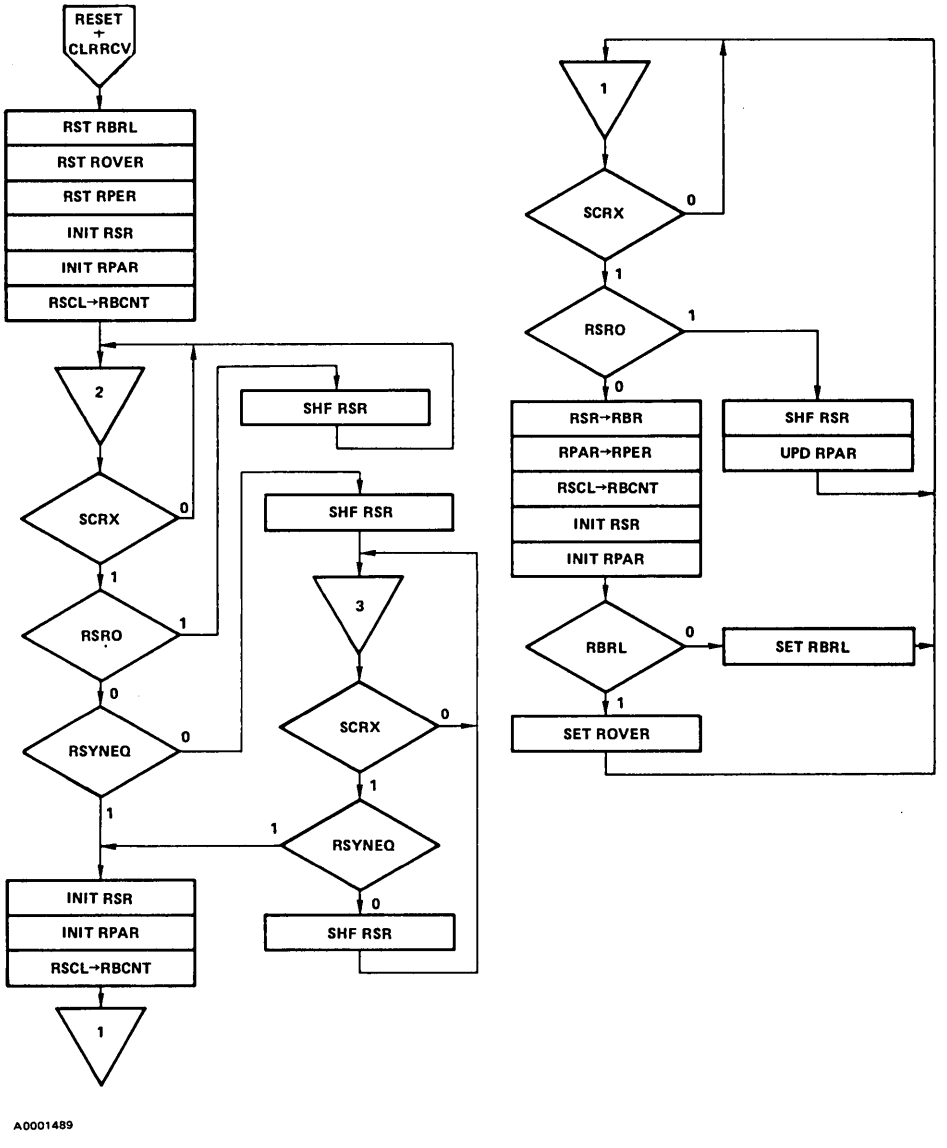
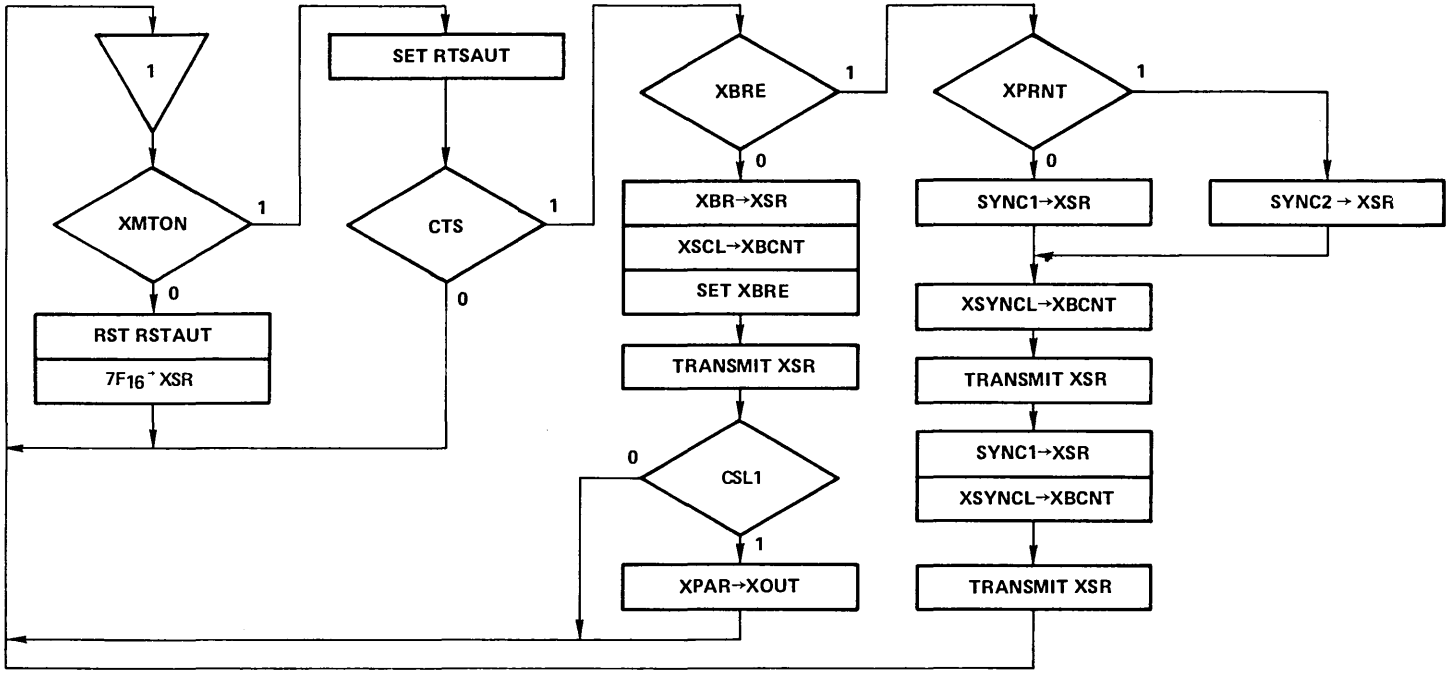


FIGURE 17. MODE 2 RECEIVER OPERATION



A0001490

FIGURE 18. MODE 3 TRANSMITTER OPERATION

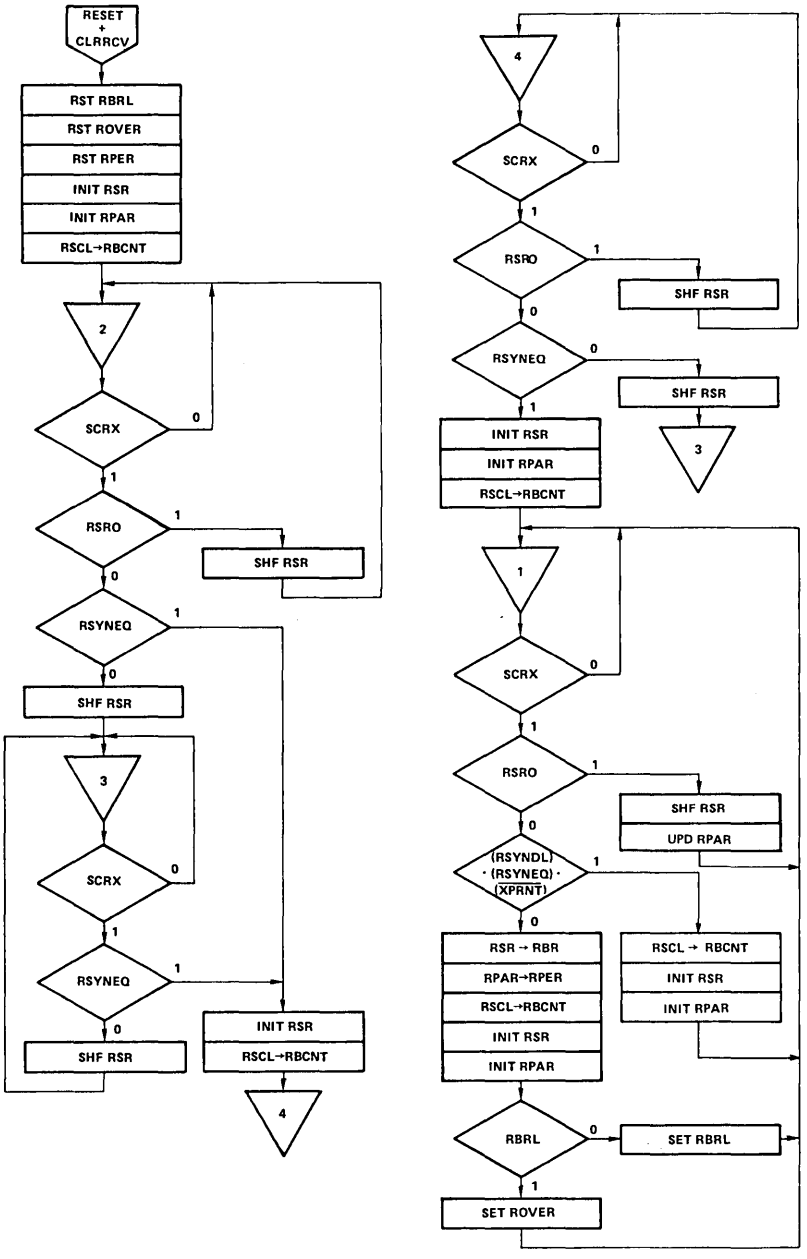


FIGURE 19. MODE 3 RECEIVER OPERATION

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.

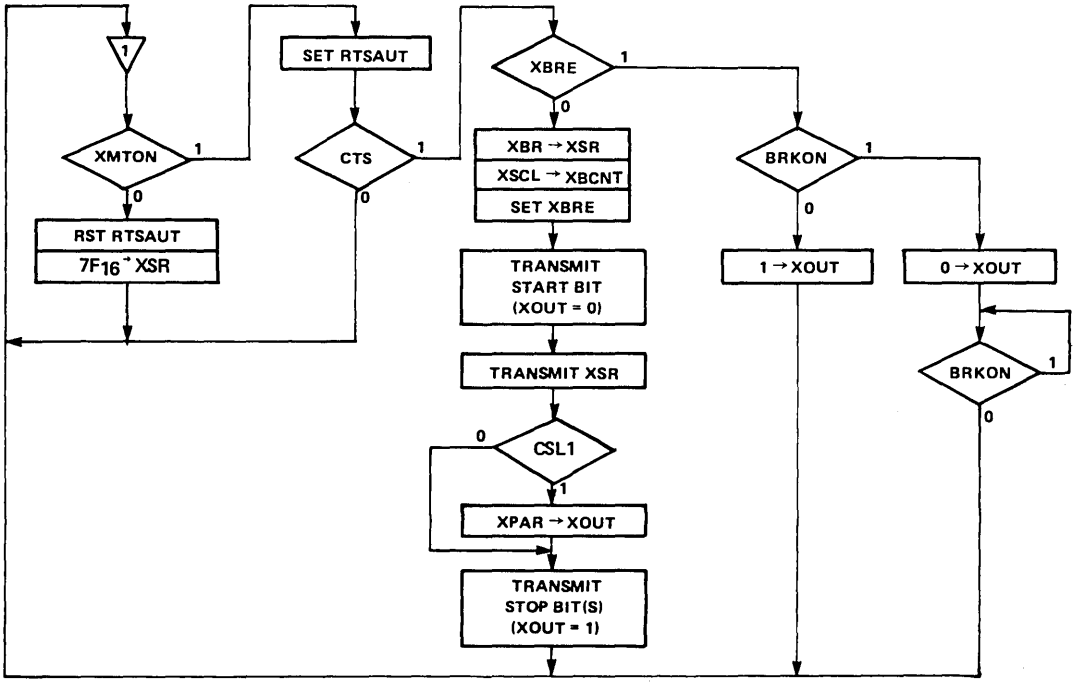


FIGURE 20. MODE 5 OR 6 TRANSMITTER OPERATION

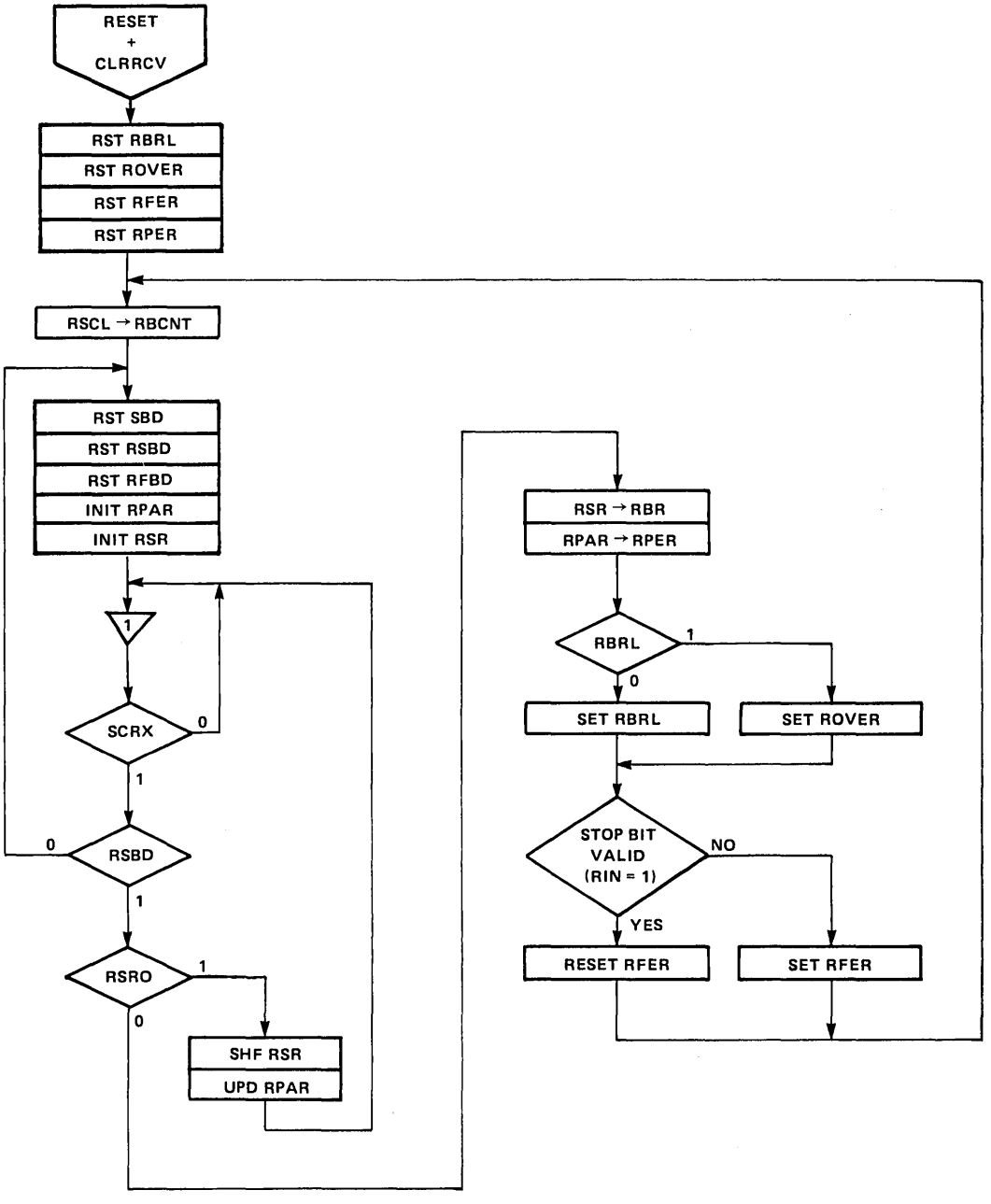


FIGURE 21. MODE 5 OR 6 RECEIVER OPERATION

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.

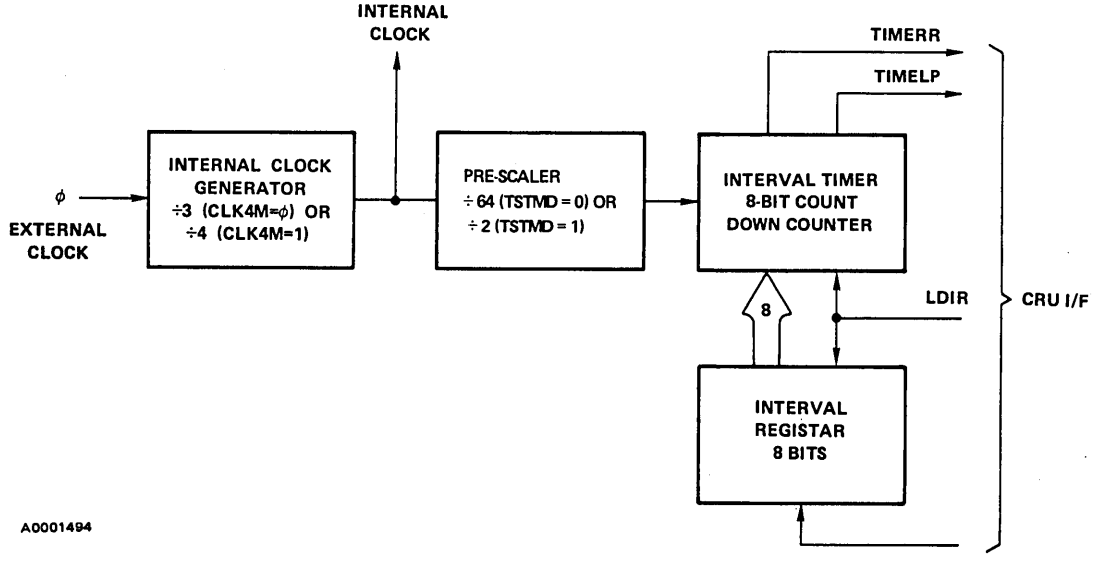
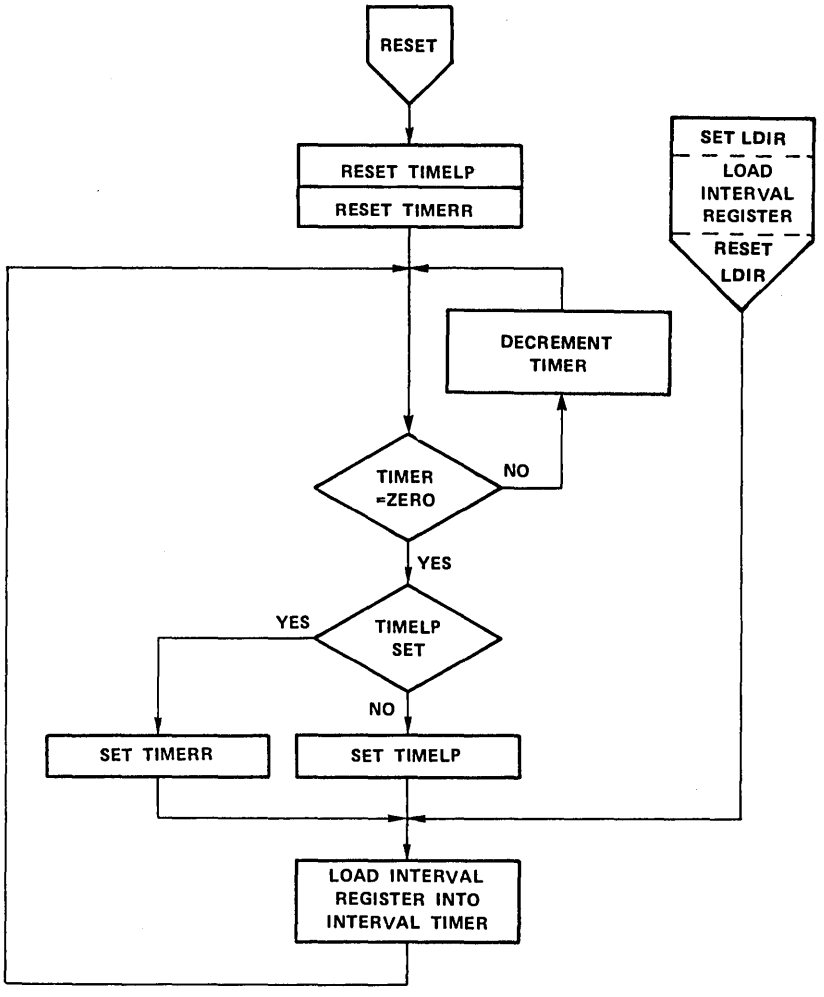


FIGURE 22. INTERVAL TIMER BLOCK DIAGRAM



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.

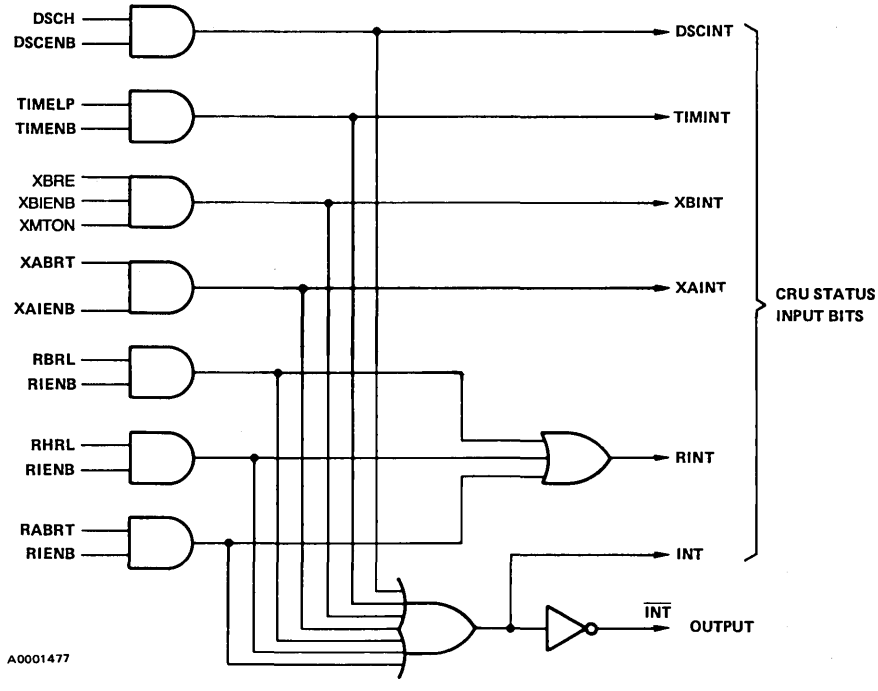
4) XABRT = 1.XABRT (transmitter abort) is set in mode 0 and 1 when no data is available for transmission, no provision is made for a fill character, and XMTON is turned ON.

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.



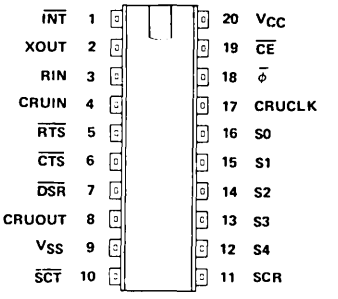
NOTE: See Tables 1 and 5 for input and output signal definitions.

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.

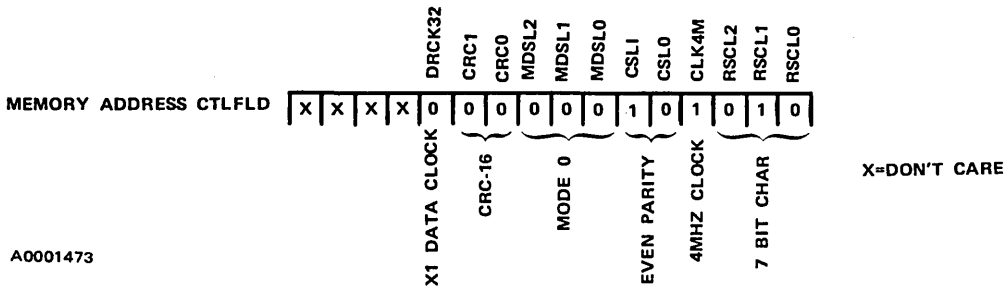
The receiver should be initialized with the CLRRCV command after the control register is loaded to ensure that the receiver logic will assemble the first received character at the proper length.

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.

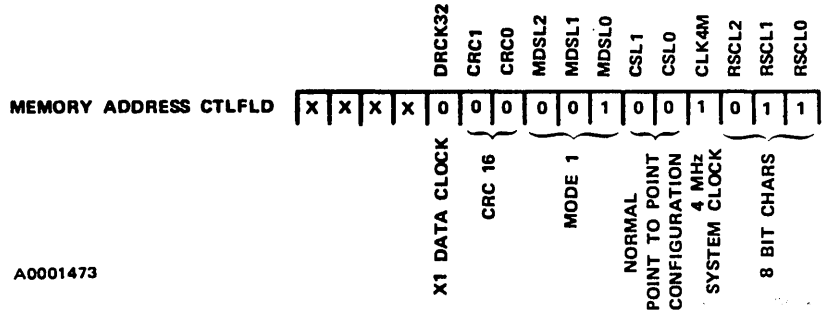
```
RESET     EQU     31
CLRRCV    EQU     30
CLRXMT    EQU     30
LDSYN2    EQU     27
XPRNT     EQU     23
          .
          .
          .
          LI      R12,>40          Initialize CRU Base.
          SBO     RESET            Reset device and set LDCTRL.
          LDCR    @ CTLFLD,12      Load 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.

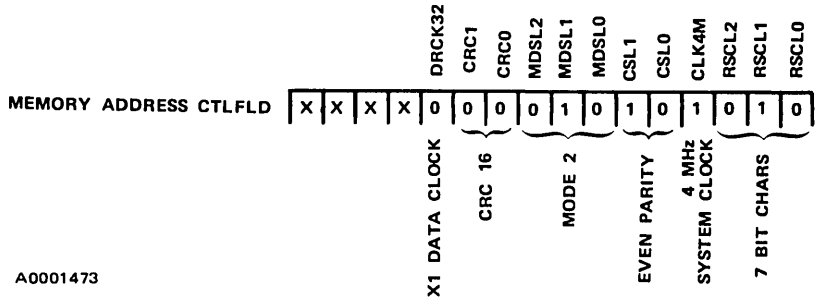
```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
LDSYN2  EQU   27
CLXCRC  EQU   29
CLRCRC  EQU   29
LXCRC   EQU   24
LRCRC   EQU   12
        :
        :
        LI    R12,>40
        SBO   RESET          Reset Device
        LDCR  @ CTLFLD,12    Load Control Register
        SBZ   CLRRCV         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
```

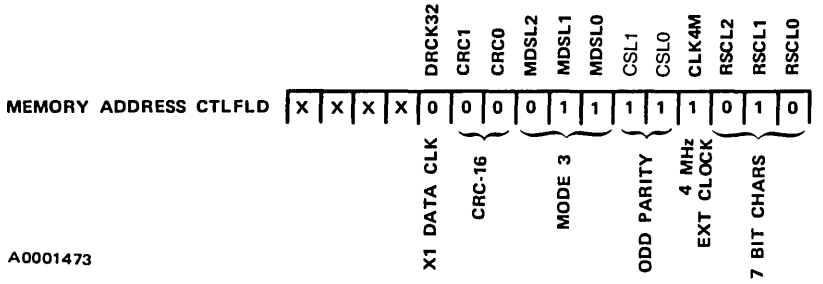


3.1.4 Mode 3 Operation [Bi-Sync]

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
```



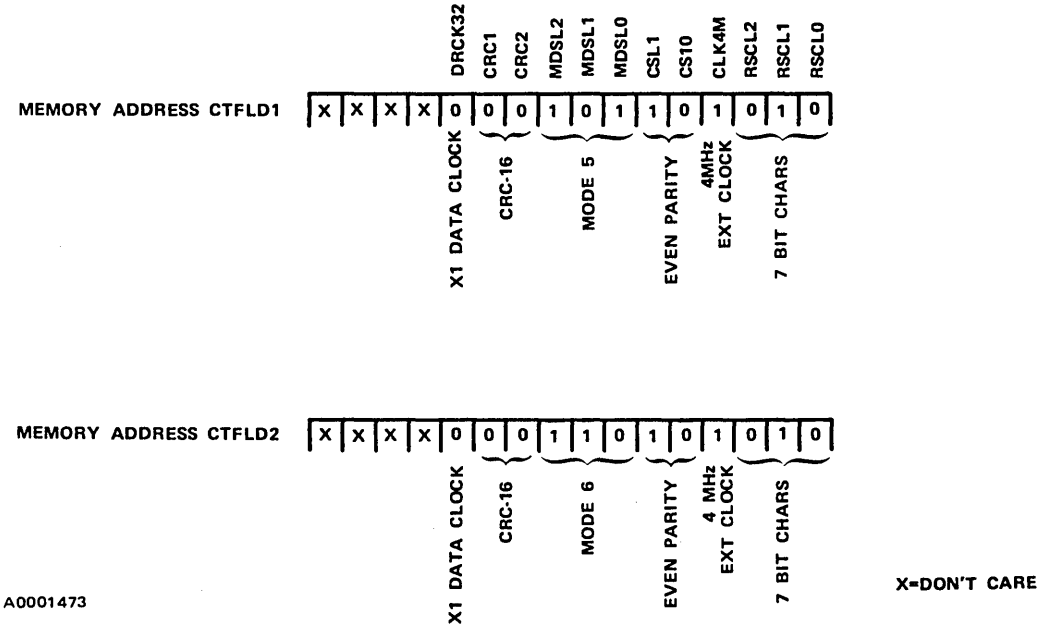
If the capability to utilize all bit combinations of the eight-bit data field is required, control bit XPRNT can be set for transparent operations. This will cause the SYNC1-SYNC1 fill sequence (i.e., normally SYN-SYN) to be replaced with SYNC2-SYNC1 (i.e., DLE SYN). Note that in transparent operation,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

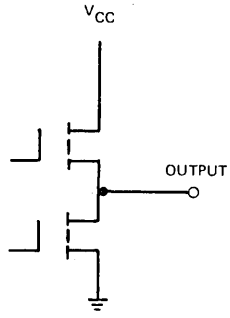
The software shown below will cause a block of data to be received and stored in memory.

```
        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
        .
        .
        .
```

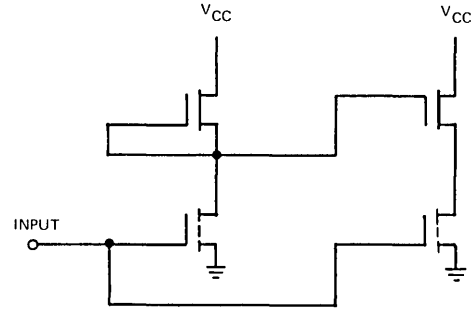
The above routine receives the block of data and compares the received CRC block check to the value accumulated in the receiver CRC register. Note that in mode 1 operation the RCVEND instructions to read the received CRC could be replaced with:

```
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	2.4		V_{CC}	V
		I_{OH} = −200 μA	2.2			
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

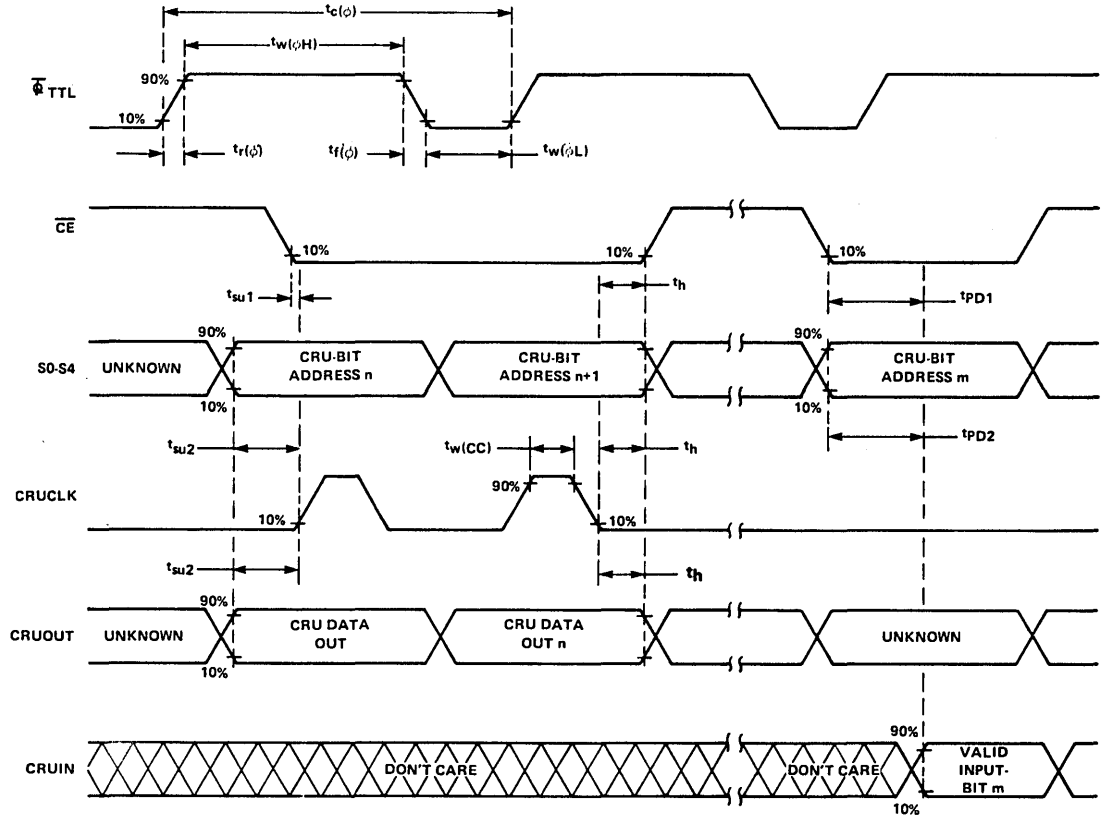
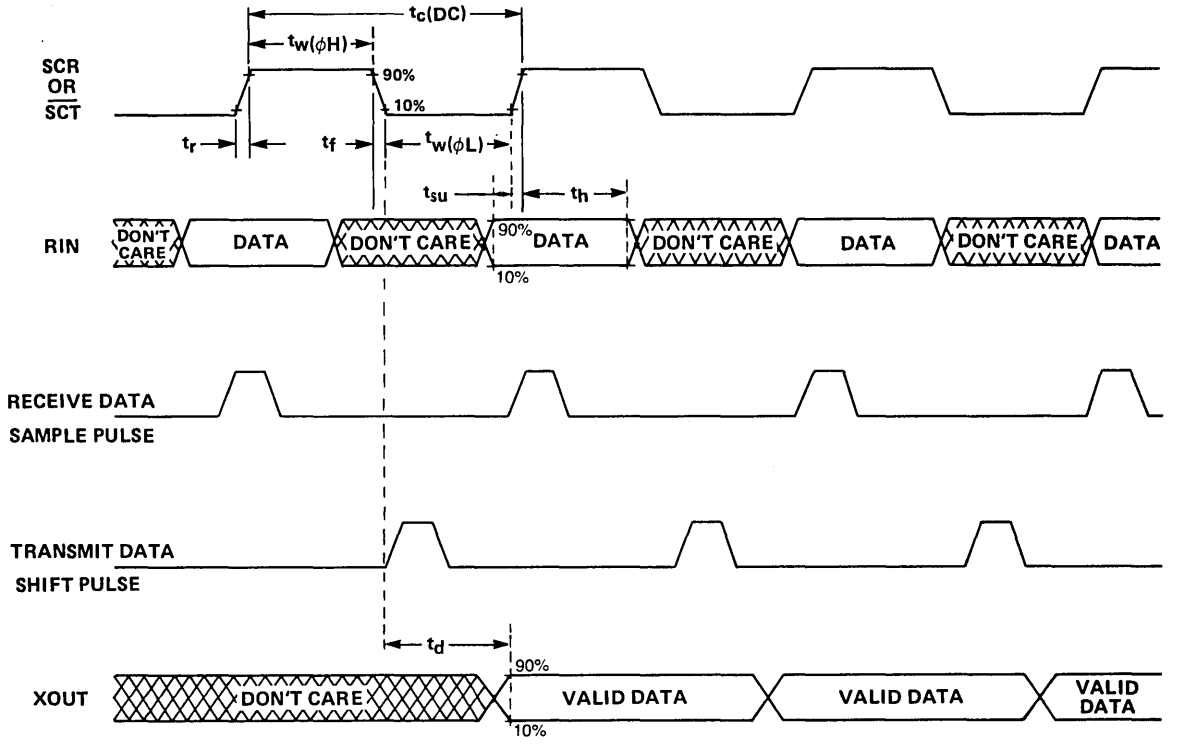


FIGURE 25. TIMING DIAGRAM



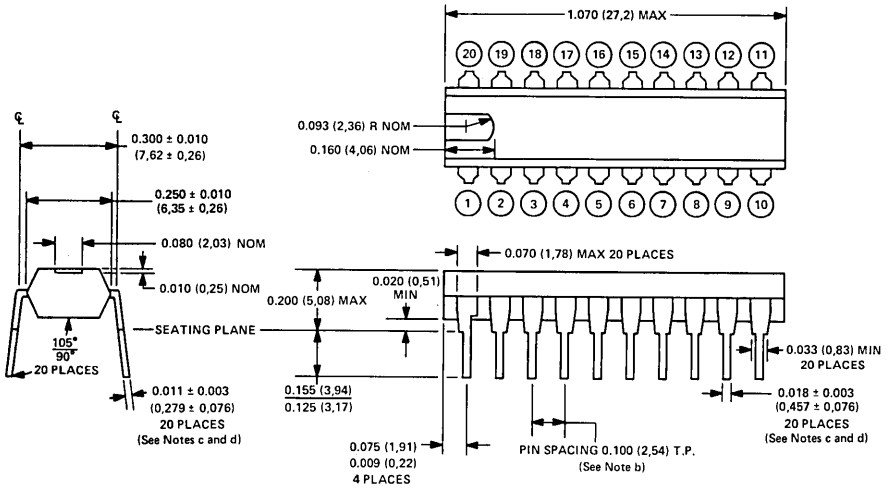
	PARAMETER		MIN	TYP	MAX	UNIT
$t_{c(DC)}$	Receiver/transmit data clock cycle time			4		μs
$t_{w(\phi H)}$	Clock pulse width (high level)			2		μs
$t_{w(\phi L)}$	Clock pulse width (low level)			2		μs
t_r	Rise time			12		ns
t_f	Fall time			12		ns
t_{su}	Setup time for RIN before SCR	(DRCK32 = 0)*		250		ns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

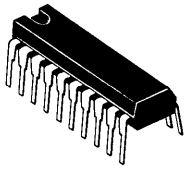
*No setup, hold, or data synchronization is required for pin in the divide-by-32 mode (DRCK32 = 1).

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, $\overline{\text{CTS}}$	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, $\overline{\text{DSR}}$	data set ready
EOP	end of poll
HDLC	high-level data link communication
INIT	initialize
$\overline{\text{INT}}$	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
$\overline{\text{LXCRC}}$	load transmitter CRC register
$\overline{\text{MEMEN}}$	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
$\overline{\text{UPD}}$	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
$\overline{\text{XZINH}}$	transmitter zero insertion inhibit
$\overline{\phi}3$	inverted TTL clock to TMS 9903 (usually from TIM 9904)

Printed in U.S.A.