The Engineering Staff of
TEXAS INSTRUMENTS INCORPORATED
Semiconductor Group

TMS 9902 ASYNCHRONOUS COMMUNICATIONS CONTROLLER DATA MANUAL

JULY 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 resposibility for any circuits shown or represent that they are free from patent infringement.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1 DESCRIPTION

The TMS 9902 Asynchronous Communications Controller (ACC) is a peripheral device designed for use with the Texas Instruments 9900 family of microprocessors. The TMS 9902 is fabricated using N-channel, silicon gate, MOS technology. The TMS 9902 is TTL-compatible on all inputs and outputs, including the power supply (+5 V) and single-phase clock. The TMS 9902 ACC provides an interface between a microprocessor and a serial, asynchronous, communications channel. The ACC performs the timing and data serialization and deserialization functions, facilitating microprocessor control of the asynchronous channel. The TMS 9902 ACC accepts *EIA Standard RS-232-C* protocol.

1.2 KEY FEATURES

- Low Cost, Serial, Asynchronous Interface
- Programmable, Five- to Eight-Bit, I/O Character Length
- Programmable 1, 1½, and 2 Stop Bits
- Even, Odd, or No Parity
- Fully Programmable Data Rate Generation
- Interval Timer with Resolution from 64 to 16,320 Microseconds
- TTL-Compatibility, Including Power Supply
- Standard 18-Pin Plastic or Ceramic Package
- N-Channel, Silicon Gate Technology

1.3 TYPICAL APPLICATION

Figure 1 shows a general block diagram of a system incorporating a TMS 9902 ACC. Following is a tutorial discussion of this application. Subsequent sections of this Data Manual detail most aspects of TMS 9902 use.

The TMS 9902 interfaces with the CPU through the *communications register unit* (CRU). The CRU interface consists of five address select lines (S0-S4), chip enable ($\overline{\text{CE}}$), and three CRU lines (CRUIN, CRUOUT, CRUCLK). An additional input to the CPU is the ACC interrupt line ($\overline{\text{INT}}$). The TMS 9902 occupies 32 bits of CRU space; each of the 32 bits are selected individually by processor address lines A10-A14 which are connected to the ACC select lines S0-S4, respectively. Chip enable ($\overline{\text{CE}}$) is generated by decoding address lines A0-A9 for CRU cycles. Under certain conditions the TMS 9902 causes interrupts. The interrupt logic shown in Figure 1 can be a TMS 9901.

The ACC interfaces to the asynchronous communications channel on five lines: request to send ($\overline{\text{RTS}}$), data set ready ($\overline{\text{DSR}}$), clear to send ($\overline{\text{CTS}}$), serial transmit data (XOUT), and serial receive data (RIN). The request to send ($\overline{\text{RTS}}$) goes active (LOW) whenever the transmitter is activated. However, before data transmission begins, the clear to send ($\overline{\text{CTS}}$) input must be active. The data set ready ($\overline{\text{DSR}}$) input does not affect the receiver or transmitter. When $\overline{\text{DSR}}$ or $\overline{\text{CTS}}$ changes level, an interrupt is generated.

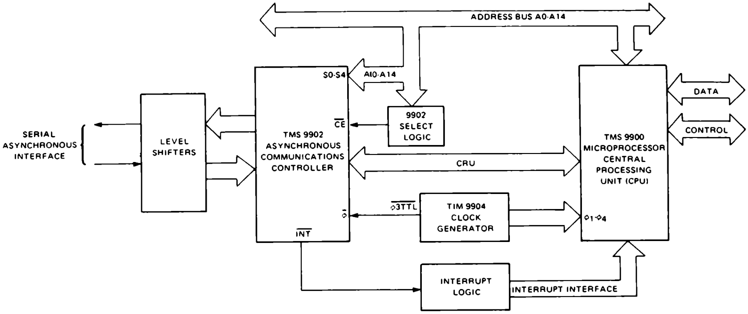


FIGURE 1. TYPICAL APPLICATION, TMS 9902 ASYNCHRONOUS COMMUNICATION CONTROLLER (ACC)

2. ARCHITECTURE

The TMS 9902 asynchronous communications controller (ACC) is designed to provide a low cost, serial, asynchronous interface to the 9900 family of microprocessors. The TMS 9902 ACC is diagrammed in Figure 2. The ACC 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 9902 ACC. The ACC occupies 32 bits of CRU read and write space. Figure 3 illustrates the CRU interface between a TMS 9902 and a TMS 9900 CPU; Figure 4 illustrates the CRU Interface for a TMS 9980A or 9981 CPU. The CRU lines are tied directly to each other as shown in Figures 3 and 4. 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 9902 via the chip enable $(\overline{CE})$ signal. When $\overline{CE}$ is inactive (HIGH), the CRU interface of the 9902 is disabled.

NOTE

> When $\overline{CE}$ is inactive (HIGH) the 9902 sets its CRUIN pin to high impedance and disables CRUCLK from coming on chip. This means the CRUIN line can be used as an OR-tied bus. The 9902 is still able to see the select lines even when $\overline{CE}$ is high.

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 multi-bit CRU transfers, the CRU address is incremented at the beginning of each CRU cycle to point to the next consecutive CRU bit.

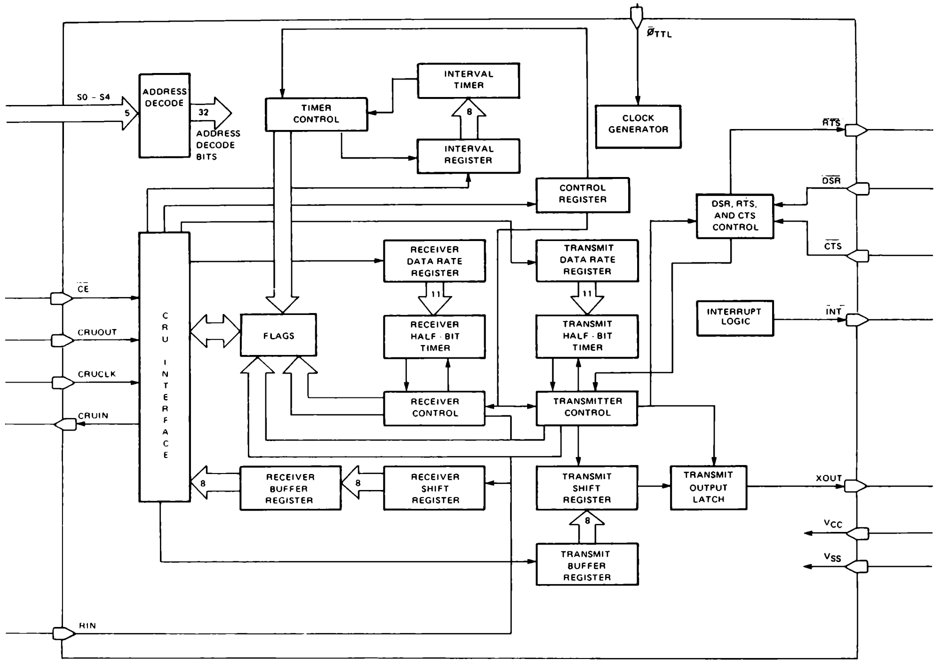


FIGURE 2. TMS 9902 ASYNCHRONOUS COMMUNICATIONS CONTROLLER (ACC) BLOCK DIAGRAM

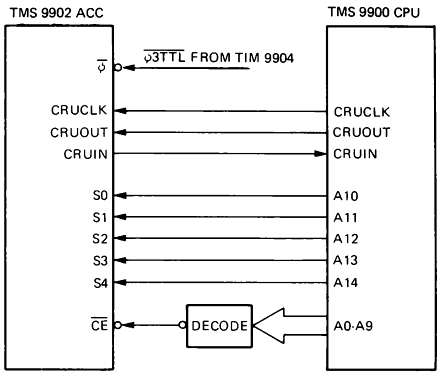


FIGURE 3. TMS 9902 – TMS 9900 CRU INTERFACE

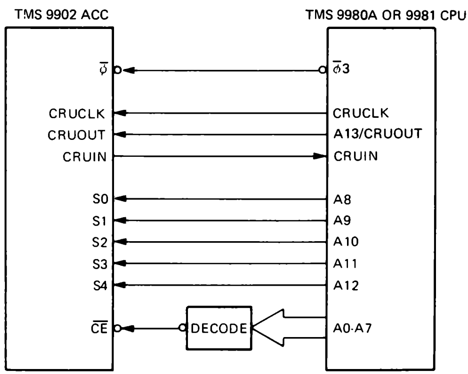


FIGURE 4. TMS 9902 – TMS 9980A OR 9981 CRU INTERFACE

When a 99XX 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.

2.1.1 CPU OUTPUT FOR CRU

The TMS 9902 ACC occupies 32 bits of output CRU space, of which 23 bits are used: 31 and 21-0. These 23 bits are employed by the CPU to communicate command and control information to the TMS 9902. Table 1 shows the mapping between CRU address select (S lines) and ACC functions. Each CRU addressable output bit on the TMS 9902 is described in detail following Table 1.

TABLE 1

TMS 9902 ACC OUTPUT SELECT BIT ASSIGNMENTS

$ADDRESS_2$ S0 S1 S2 S3 S4	$ADDRESS_{10}$	NAME	DESCRIPTION
1 1 1 1 1	31	RESET	Reset device.
	30-22		Not used.
1 0 1 0 1	21	DSCENB	Data Set Status Change Interrupt Enable.
1 0 1 0 0	20	TIMENB	Timer Interrupt Enable
1 0 0 1 1	19	XBIENB	Transmitter Interrupt Enable
1 0 0 1 0	18	RIENB	Receiver Interrupt Enable
1 0 0 0 1	17	BRKON	Break On
1 0 0 0 0	16	RTSON	Request to Send On
0 1 1 1 1	15	TSTMD	Test Mode
0 1 1 1 0	14	LDCTRL	Load Control Register
0 1 1 0 1	13	LDIR	Load Interval Register
0 1 1 0 0	12	LRDR	Load Receiver Data Rate Register
0 1 0 1 1	11	LXDR	Load Transmit Data Rate Register
	10-0		Control, Interval, Receive Data Rate, Transmit Data Rate, and Transmit Buffer Registers

Bit 31 (RESET) — **Reset.** Writing a one or zero to bit 31 causes the device to reset, consequently disabling all interrupts, initializing the transmitter and receiver, setting $\overline{RTS}$ inactive (HIGH), setting all register load control flags (LDCTRL, LDIR, LRDR, and LXDR) to a logic one level, and resetting the BREAK flag. No other input or output operations should be performed for 11 $\overline{\Phi}$ clock cycles after issuing the RESET command.

Bit 30-Bit 22 — Not used.

INTERRUPT ENABLE	SELECT BIT	INTERRUPT FLAG	INTERRUPT ENABLED
DSCENB	21	DSCH	DSCINT
TIMENB	20	TIMELP	TIMINT
XIENB	19	XBRE	XINT
RIENB	18	RBRL	RINT

Bit 21 (DSCENB) — **Data Set Change Interrupt Enable.** Writing a one to bit 21 causes the $\overline{\text{INT}}$ output to be active (LOW) whenever DSCH (Data Set Status Change) is a logic one. Writing a zero to bit 21 causes DSCH interrupts to be disabled. Writing either a one or zero to bit 21 causes DSCH to reset. (Refer also to Section 2.5).

Bit 20 (TIMENB) — **Timer Interrupt Enable.** Writing a one to bit 20 causes the $\overline{\text{INT}}$ output to be active whenever TIMELP (Timer Elapsed) is a logic one. Writing a zero to bit 20 causes TIMELP interrupts to be disabled. Writing either a one or zero to bit 20 causes TIMELP and TIMERR (Timer Error) to reset. (Refer also to Sections 2.4 and 2.5.)

Bit 19 (XBIENB) — **Transmit Buffer Interrupt Enable.** Writing a one to bit 19 causes the $\overline{\text{INT}}$ output to be active whenever XBRE (Transmit Buffer Register Empty) is a logic one. Writing a zero to bit 19 causes XBRE interrupts to be disabled. The state of XBRE is not affected by writing to bit 19. (Refer also to Sections 2.2 and 2.5.)

Bit 18 (RIENB) — **Receiver Interrupt Enable.** Writing a one to bit 18 causes the $\overline{\text{INT}}$ output to be active whenever RBRL (Receiver Buffer Register Loaded) is a logic one. Writing a zero to bit 18 disables RBRL interrupts. Writing either a one or zero to bit 18 causes RBRL to reset. (Refer also to Sections 2.3 and 2.5.)

Bit 17 (BRKON) — **Break On.** Writing a one to bit 17 causes the XOUT (Transmitter Serial Data Output) to go to a logic zero whenever the transmitter is active and the Transmit Buffer Register (XBR) and the Transmit Shift Register (XSR) are empty. While BRKON is set, loading of characters into the XBR is inhibited. Writing a zero to bit 17 causes BRKON to reset and the transmitter to resume normal operation.

Bit 16 (RTSON) — **Request To Send On.** Writing a one to bit 16 causes the $\overline{\text{RTS}}$ output to be active (LOW). Writing a zero to bit 16 causes $\overline{\text{RTS}}$ to go to a logic one after the XSR (Transmit Shift Register) and XBR (Transmit Buffer Register) are empty, and BRKON is reset. Thus, the $\overline{\text{RTS}}$ output does not become inactive (HIGH) until *after* character transmission is completed.

Bit 15 (TSTMD) — **Test Mode.** Writing a one to bit 15 causes $\overline{\text{RTS}}$ to be internally connected to $\overline{\text{CTS}}$, XOUT to be internally connected to RIN, $\overline{\text{DSR}}$ to be internally held LOW, and the Interval Timer to operate 32 times its normal rate. Writing a zero to bit 15 re-enables normal device operation. There seldom is reason to enter the test mode under normal circumstances, but this function is useful for diagnostic and inspection purposes.

Bits 14-11 — **Register Load Control Flags.** Output bits 14-11 control which of the five registers are loaded when writing to bits 10-0. The flags are prioritized as shown in Table 2.

TABLE 2

TMS 9902 ACC REGISTER LOAD SELECTION

REGISTER LOAD CONTROL FLAG STATUS				REGISTER ENABLED
LDCTRL	LDIR	LRDR	LXDR	
1	X	X	X	Control Register
0	1	X	X	Interval Register
0	0	1	X	Receive Data Rate Register *
0	0	X	1	Transmit Data Rate Register *
0	0	0	0	Transmit Buffer Register

*If both LRDR and LXDR bits are set, both registers are loaded, assuming LDCTRL and LDIR are disabled; if only one of these registers is to be loaded, only that register bit is set, and the other register bit reset.

Bit 14 (LDCTRL) — **Load Control Register.** Writing a one to bit 14 causes LDCTRL to be set to a logic one. When LDCTRL = 1, any data written to bits 0-7 is directed to the Control Register. Note that LDCTRL is also set to a logic one when a one or zero is written to bit 31 (RESET). Writing a zero to bit 14 causes LDCTRL to reset to a logic zero, disabling loading of the Control Register. LDCTRL is also automatically reset to logic zero when a datum is written to bit 7 of the Control Register, reset normally occurs as the last bit is written when loading the Control Register with a LDCR instruction.

Bit 13 (LDIR) — **Load Interval Register.** Writing a one to bit 13 causes LDIR to set to a logic one. When LDIR = 1 and LDCTRL = 0, any data written to bits 0-7 is directed to the Interval Register. Note that LDIR is also set to a logic one when a datum is written to bit 31 (RESET); however, Interval Register loading is not enabled until LDCTRL is set to a logic zero. Writing a zero to bit 13 causes LDIR to be reset to logic zero, disabling loading of the Interval Register. LDIR is also automatically reset to logic zero when a datum is written to bit 7 of the Interval Register; reset normally occurs as the last bit is written when loading the Interval Register with a LDCR instruction.

Bit 12 (LRDR) — **Load Receive Data Rate Register.** Writing a one to bit 12 causes LRDR to set to a logic one. When LRDR = 1, LDIR = 0, and LDCTRL = 0, any data written to bits 0-10 is directed to the Receive Data Rate Register. Note that LRDR is also set to a logic one when a datum is written to bit 31 (RESET); however, Receive Data Rate Register loading is not enabled until LDCTRL and LDIR are set to a logic zero. Writing a zero bit to 12 causes LRDR to reset to a logic zero, disabling loading of the Receive Data Rate Register. LRDR is also automatically reset to logic zero when a datum is written to bit 10 of the Receive Data Rate Register; reset normally occurs as the last bit is written when loading the Receive Data Rate Register with a LDCR instruction.

Bit 11 (LXDR) — **Load Transmit Data Rate Register.** Writing a one to bit 11 causes LXDR to set to a logic one. When LXDR = 1, LDIR = 0, and LDCTRL = 0, any data written to bits 0-10 is directed to the Transmit Data Rate Register. Note that loading of both the Receive and Transmit Data Rate Registers is enabled when LDCTRL = 0, LDIR = 0, LRDR = 1, and LXDR = 1; thus these two registers may be loaded simultaneously when data is received and transmitted at the same rate. LXDR is also set to a logic one when a datum is written to bit 31 (RESET); however, Transmit Data Rate Register loading is not enabled until LDCTRL and LDIR are to logic zero. Writing a zero to bit 11 causes LXDR to reset to logic zero, consequently disabling loading of the Transmit Data Rate Register. Since bit 11 is the next bit addressed after loading the Transmit Data Rate Register, the register may be loaded and the LXDR flag reset with a single LDCR instruction where 12 bits (Bits 0-11) are written and a zero is written to Bit 11.

Bits 14-11 (All Zeros) — **Load Transmit Buffer Register.** See Section 2.1.2.5.

Bits 10-0 (Data) — **Data.** Information written to bits 10-0 is loaded into the controlling registers as indicated by LDCTRL, LDIR, LRDR, and LXDR (see Table 2). The different register bits are described in Section 2.1.2 below.

2.1.2 REGISTERS

2.1.2.1 Control Register

The Control Register is loaded to select character length, device clock operation, parity, and the number of stop bits for the transmitter; control register loading occurs when LDCTRL is active (see Table 2). Table 3 shows the bit address assignments for the Control Register.

TABLE 3
CONTROL REGISTER BIT ADDRESS ASSIGNMENTS

$ADDRESS_{10}$	NAME	DESCRIPTION
7	SBS1	Stop Bit Select
6	SBS2	
5	PENB	Parity Enable
4	PODD	Odd Parity Select
3	CLK4M	$\overline{\phi}$ Input Divide Select
2	—	Not Used
1	RCL1	Character Length Select
0	RCL0	

7	6	5	4	3	2	1	0
SBSI	SBS2	PENB	PODD	CLK4M	NOT USED	RCL1	RCLO
MSB							LSB

Bits 7 and 6 (SBS1 and SBS2) — **Stop Bit Selection.** The number of stop bits to be appended to each transmitter character is selected by bits 7 and 6 of the Control Register as shown below. The receiver only tests for a single stop bit, regardless of the status of bits 7 and 6.

STOP BIT SELECTION

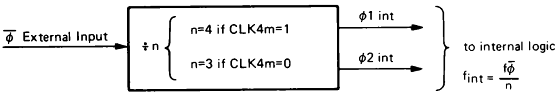
SBS1 BIT 7	SBS2 BIT 6	NUMBER OF TRANSMITTED STOP BITS
0	0	1½
0	1	2
1	0	1
1	1	1

Bits 5 and 4 (PENB and PODD) — **Parity Selection.** The type of parity generated for transmission and detected for reception is selected by bits 5 and 4 of the Control Register as shown below. When parity is enabled (PENB = 1), the parity bit is transmitted and received in addition to the number of bits selected for the character length. Odd parity is such that the total number of ones in the character and parity bit, exclusive of stop bit(s), will be odd. For even parity, the total number of ones will be even.

PARITY SELECTION

PENB BIT 5	PODD BIT 4	PARITY
0	0	None
0	1	None
1	0	Even
1	1	Odd

Bit 3 (CLK4M) — **$\overline{\phi}$ Input Divide Select.** The $\overline{\phi}$ input to the TMS 9902 ACC is used to generate internal dynamic logic clocking and to establish the time base for the Interval Timer, Transmitter, and Receiver. The $\overline{\phi}$ input is internally divided by either 3 or 4 to generate the two-phase internal clocks required for MOS logic, and to establish the basic internal operating frequency (f_{int}) and internal clock period (t_{int}). When bit 3 of the Control Register is set to a logic one (CLK4M = 1), $\overline{\phi}$ is internally divided by 4, and when CLK4M = 0, $\overline{\phi}$ is divided by 3. For example, when $f\overline{\phi}$ = 3 MHz, as in a standard 3 MHz TMS 9900 system, and CLK4M = 0, $\overline{\phi}$ is internally divided by 3 to generate an internal clock period t_{int} of 1 μs. The figure below shows the operation of the internal clock divider circuitry. The internal clock frequency should be no greater than 1.1 MHz; thus, when $f\overline{\phi} >$ 3.3 MHz, CLK4M should be set to a logic one.



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

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

INTERNAL CLOCK DIVIDER CIRCUITRY

Bits 1 and 0 (RCL1 and RCL0) — **Character Length Select.** The number of data bits in each transmitted and received character is determined by bits 1 and 0 of the Control Register as shown below:

CHARACTER LENGTH SELECTION

RCL1 BIT 1	RCL0 BIT 0	CHARACTER LENGTH
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

NOTE: $f\phi$ denotes frequency of ϕ

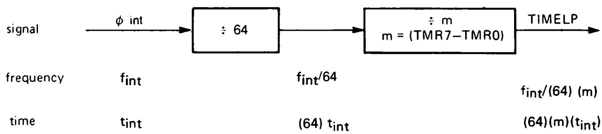
2.1.2.2 Interval Register

The Interval Register is enabled for loading when LDCTRL = 0 and LDIR = 1 (see Table 2). The Interval Register is used to select the rate at which interrupts are generated by the TMS 9902 Interval Timer. The figure below shows the bit assignments for the Interval Register when enabling for loading.

7	6	5	4	3	2	1	0
TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
MSB							LSB

INTERVAL REGISTER BIT ADDRESS ASSIGNMENTS

The figure below illustrates the establishment of the interval for the Interval Timer. For example, if the Interval Register is loaded with a value of 80_{16} (128_{10}) the interval at which Timer Interrupts are generated is t_{itvl} = $t_{int} \cdot 64 \cdot M = (1\ \mu s)(64)(128) = 8.192$ ms when $t_{int} = 1\ \mu s$. $t_{int} = n/f\bar{\phi}$ where n = 4 if CLK4M = 1, 3 if CLK4M = 0.



TIME INTERVAL SELECTION

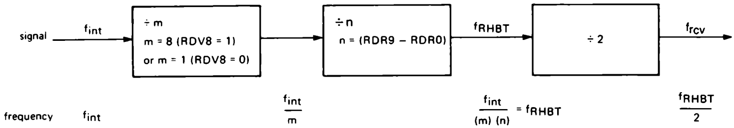
2.1.2.3 Receive Data Rate Register

The Receive Data Rate Register (RDR) is enabled for loading when LDCTRL = 0, LDIR = 0, and LRDR = 1 (see Table 2). The Receive Data Rate Register is used to select the bit rate at which data is received. The diagram shows the bit address assignments for the Receive Data Rate Register when enabled for loading.

10	9	8	7	6	5	4	3	2	1	0
RDV8	RDR9	RDR8	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
MSB										LSB

RECEIVE DATA RATE REGISTER BIT ADDRESS ASSIGNMENTS

The diagram below illustrates the manner in which the receive data rate is established. Basically, two programmable counters are used to determine the interval for half the bit period of receive data. The first counter divides the internal system clock frequency (f_{int}) by either 8 (RDV8 = 1) or 1 (RDV8 = 0). The second counter has ten stages and may be programmed to divide its input signal by any value from 1 (RDR9 – RDR0 = 0000000001) to 1023 (RDR9 – RDR0 = 1111111111). The frequency of the output of the second counter (f_{rhbt}) is double the receive-data rate. For example, assume the Receive Data Rate Register is loaded with a value of 11000111000; RDV8 = 1, and RDR9 – RDR0 = 1000111000 = 238_{16} = 568_{10}. Thus, for f_{int} = 1 MHz, (see Control Register, bit 3) the receive data rate = $f_{rcv} = [(1 \times 10^6 \div 8) \div 568] \div 2 = 110.04$ bits per second.



RECEIVE DATA RATE SELECTION

Quantitatively, the receive-data rate f_{RCV} is described by the following algebraic expression:

$$f_{rcv} = \frac{f_{RHBT}}{2} = \frac{f_{int}}{(2)(m)(n)} = \frac{f_{int}}{(2)(8^{RDV8})(RDR9 - RDR0)}$$

2.1.2.4 Transmit Data Rate Register

The Transmit Data Rate Register (XDR) is enabled for loading when LDCTRL = 0, LDIR = 0, and LXDR = 1 (see Table 2). The Transmit Data Rate Register is used to select the data for the transmitter. The figure below shows the bit address assignments for the Transmit Data Rate Register when enabled for loading.

10	9	8	7	6	5	4	3	2	1	0
XDV8	XDR9	XDR8	XDR7	XDR6	XDR5	XDR4	XDR3	XDR2	XDR1	XDR0
MSB										LSB

The transmit data rate is selected with the Transmit Data Rate Register in the same manner the receive data rate is selected with the Receive Data Rate Register. The algebraic Expression for the Transmit Data Rate f_{xmt} is

$$f_{xmt} = \frac{f_{XHBT}}{2} = \frac{f_{int}}{(2)(8^{XDV8})(XDR9\text{-}XDR0)}$$

For example, if the Transmit Data Rate Register is loaded with a value of 00110100001; XDV8 = 0, and XDR9 – XDR0 = $1A1_{16} = 417_{10}$, if f_{int} = 1 MHz the transmit data rate = f_{xmt} = $[(1 \times 10^6 \div 1) \div 417] \div 2$ = 1199.0 bits per second.

2.1.2.5 Transmit Buffer Register

The Transmit Buffer Register (XBR) is enabled for loading when LDCTRL = 0, LDIR = 0, LRDR = 0, LXDR = 0, and BRKON = 0 (see Table 2). The Transmit Buffer Register is used to store the next character to be transmitted. When the transmitter is active, the contents of the Transmit Buffer Register are transferred to the Transmit Shift Register (XSR) each time the previous character has been completely transmitted (XSR becomes empty). The bit address assignments for the Transmit Buffer Register are shown below:

7	6	5	4	3	2	1	0
XBR7	XBR6	XBR5	XBR4	XBR3	XBR2	XBR1	XBR0
MSB							LSB

TRANSMIT BUFFER REGISTER BIT ADDRESS ASSIGNMENTS

TABLE 4. CRU OUTPUT BIT ADDRESS ASSIGNMENTS

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESET	NOT USED									DSCENB	TIMENB	XBIENB	RIENB	BRKON	RTSON

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSTMD	LDCTRL	LDIR	LRDR	LXDR	CONTROL, INTERVAL, RECEIVE DATA RATE, TRANSMIT DATA RATE, AND TRANSMIT BUFFER REGISTERS										
CONTROL REGISTER		1	X	X	X				SBS1 (note 1)	SBS2	PENB	PODD	CLK4M	–	RCL1	RCL0
INTERVAL REGISTER		0	1	X	X				TMR7 (note 1)	TMR 6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
RECEIVE DATA RATE REGISTER		0	0	1	X	RDV8 (note 1)	RDR9	RDR8	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
TRANSMIT DATA RATE REGISTER		0	0	X	1	XDV8	XDR9	XDR8	XDR7	XDR6	XDR5	XDR4	XDR3	XDR2	XDR1	XDR0
TRANSMIT BUFFER REGISTER		0	0	0	0				XBR7	XBR6	XBR5	XBR4	XBR3	XBR2	XBR1	XBR0

CONTROL REGISTER

Stop Bits	
00	1-1/2
01	2
1X	1

Parity	
0X	none
10	even
11	odd

$f_{int} =$ $f\bar{\phi}/(3+CLK4M)$

Character Length	
00	5
01	6
10	7
11	8

INTERVAL REGISTER

TMR

$T_{ITVL} = t_{int} \times 64 \times TMR$

RECEIVE DATA RATE REGISTER

RDR

$f_{rcv} = f_{int} \div 8\,RDV8 \div RDR \div 2$

TRANSMIT DATA RATE REGISTER

XDR

$f_{xmt} = f_{int} \div 8\,XDV8 \div XDR \div 2$

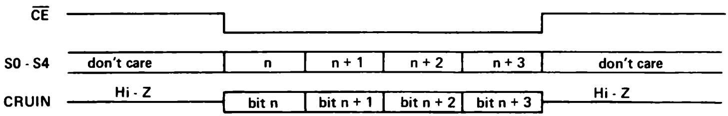
NOTE 1 LOADING OF THE BIT INDICATED BY [hatched box] CAUSES THE LOAD CONTROL FLAG FOR THAT REGISTER TO RESET AUTOMATICALLY.

All eight bits should be transferred into the register, regardless of the selected character length. The extraneous high order bits will be ignored for transmission purposes; however, loading of bit 7 is internally detected which causes the Transmit Buffer Register Empty (XBRE) status flag to reset.

2.1.3 INPUT TO CPU FOR CRU

The TMS 9902 ACC occupies 32 bits of input CRU space. The CPU reads the 32 bits from the ACC to sense the status of the device. Table 5 shows the mapping between CRU bit address and TMS 9902 read data. Each CRU addressable read bit is described following Table 5.

Status and data information is read from the ACC using $\overline{CE}$, S0-S4, and CRUIN. The following figure illustrates the relationship of the signals used to access four bits of data from the ACC.



ACC DATA ACCESS SIGNAL TIMING

TABLE 5
TMS 9902 ACC INPUT SELECT BIT ASSIGNMENTS

$ADDRESS_2$					$ADDRESS_{10}$	NAME	DESCRIPTION
S0	S1	S2	S3	S4			
1	1	1	1	1	31	INT	Interrupt
1	1	1	1	0	30	FLAG	Register Load Control Flag Set
1	1	1	0	1	29	DSCH	Data Set Status Change
1	1	1	0	0	28	CTS	Clear to Send
1	1	0	1	1	27	DSR	Data Set Ready
1	1	0	1	0	26	RTS	Request to Send
1	1	0	0	1	25	TIMELP	Timer Elapsed
1	1	0	0	0	24	TIMERR	Timer Error
1	0	1	1	1	23	XSRE	Transmit Shift Register Empty
1	0	1	1	0	22	XBRE	Transmit Buffer Register Empty
1	0	1	0	1	21	RBRL	Receive Buffer Register Loaded
1	0	1	0	0	20	DSCINT	Data Set Status Change Interrupt (DSCH · DSCENB)
1	0	0	1	1	19	TIMINT	Timer Interrupt (TIMELP · TIMENB)
1	0	0	1	0	18	–	Not Used (always = 0)
1	0	0	0	1	17	XBINT	Transmitter Interrupt (XBRE · XBIENB)
1	0	0	0	0	16	RBINT	Receiver Interrupt (RBRL · RIENB)
0	1	1	1	1	15	RIN	Receive Input
0	1	1	1	0	14	RSBD	Receive Start Bit Detect
0	1	1	0	1	13	RFBD	Receive Full Bit Detect
0	1	1	0	0	12	RFER	Receive Framing Error
0	1	0	1	1	11	ROVER	Receive Overrun Error
0	1	0	1	0	10	RPER	Receive Parity Error
0	1	0	0	1	9	RCVERR	Receive Error
0	1	0	0	0	8	–	Not Used (always = 0)
					7-0	RBR7 - RBR0	Receive Buffer Register (Received Data)

Bit 31 (INT) — INT = DSCINT (Data Set Status Change Interrupt) + TIMINT (Timer Interrupt) + XBINT (Transmitter Interrupt) + RBINT (Receiver Interrupt). The interrupt output ($\overline{\text{INT}}$) is active (LOW) when this status signal is a logic one. (Refer also to Section 2.6.)

Bit 30 (FLAG) — FLAG = LDCTRL + LDIR + LRDR + LXDR + BRKON. When any of the register load control flags or BRKON is set, FLAG = 1 (see Section 2.1.1).

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

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

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

Bit 26 (RTS) — **Request To Send.** The RTS signal indicates the inverted status of the $\overline{\text{RTS}}$ device output.

Bit 25 (TIMELP) — **Timer Elapsed.** TIMELP is set each time the Interval Timer decrements to 0. TIMELP is reset by an output to bit 20 (TIMENB).

Bit 24 (TIMERR) — **Timer Error.** TIMERR is set whenever the Interval Timer decrements to 0 and TIMELP (Timer Elapsed) is already set, indicating that the occurrence of TIMELP was not recognized and cleared by the CPU before subsequent intervals elapsed. TIMERR is reset by an output to bit 20 (TIMENB, Timer Interrupt Enable).

Bit 23 (XSRE) — **Transmit Shift Register Empty.** When XSRE = 1, no data is currently being transmitted and the XOUT output is at logic one unless BRKON (see Section 2.1.1) is set. When XSRE = 0, transmission of data is in progress.

Bit 22 (XBRE) — **Transmit Buffer Register Empty.** When XBRE = 1, the transmit buffer register does not contain the next character to be transmitted. XBRE is set each time the contents of the transmit buffer register are transferred to the transmit shift register, XBRE is reset by an output to bit 7 of the transmit buffer register (XBR7), indicating that a character has been loaded.

Bit 21 (RBRL) — **Receive Buffer Register Loaded.** RBRL is set when a complete character has been assembled in the receive shift register, and the character is transferred to the receive buffer register. RBRL is reset by an output to bit 18 (RIENB, Receiver Interrupt Enable).

Bit 20 (DSCINT) — **Data Set Status Change Interrupt.** DSCINT = DSCH (Data Set Status Change)AND DSCENB(Data Set Status Change Interrupt Enable). DSCINT indicates the presence of an enabled interrupt caused by the changing of state of $\overline{\text{DSR}}$ or $\overline{\text{CTS}}$.

Bit 19 (TIMINT) — **Timer Interrupt.** TIMINT = TIMELP (Timer Elapsed)AND TIMENB (Timer Interrupt Enable). TIMINT indicates the presence of an enabled interrupt caused by the interval timer.

Bit 17 (XBINT) — **Transmitter Interrupt.** XBINT = XBRE (Transmit Buffer Register Empty) AND XBIENB (Transmit Buffer Interrupt Enable). XBINT indicates the presence of an enabled interrupt caused by the transmitter.

Bit 16 (RBINT) — **Receiver Interrupt.** RBINT = RBRL (Receive Buffer Register Loaded) AND RIENB (Receiver Interrupt Enable). RBINT indicates the presence of an enabled interrupt caused by the receiver.

Bit 15 (RIN) — **Receive Input.** RIN indicates the status of the RIN input to the device.

Bit 14 (RSBD) — **Receive Start Bit Detect.** RSBD is set a half bit time after the 1-to-0 transition of RIN, indicating the start bit of a character. If RIN is not still 0 at such time, RSBD is reset. Otherwise, RSBD remains true until the complete character has been received. This bit is normally used only for testing purposes.

Bit 13 (RFBD) — **Receive Full Bit Detect.** RFBD is set one bit time after RSBD is set to indicate the sample point for the first data bit of the received character. RSBD is reset when the character has been completely received. This bit is normally used only for testing purposes.

Bit 12 (RFER) — **Receive Framing Error.** RFER is set 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 (Receive Buffer Register Loaded) is a one. RFER is reset when a character with the correct stop bit is received.

Bit 11 (ROVER) — **Receive Overrun Error.** ROVER is set when a new character is received before the RBRL (Receive Buffer Register Loaded) flag 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 is 0 when the character is transferred to the receive buffer register.

Bit 10 (RPER) — **Receive Parity Error.** RPER is set when a character is received in which the parity is incorrect. RPER is reset when a character with correct parity is received.

Bit 9 (RCVERR) — **Receive Error.** RCVERR = RFER (Receive Framing Error) + ROVER (Receiver Overrun Error) + RPER (Receive Parity Error). The RCVERR signal indicates the presence of an error in the most recently received character.

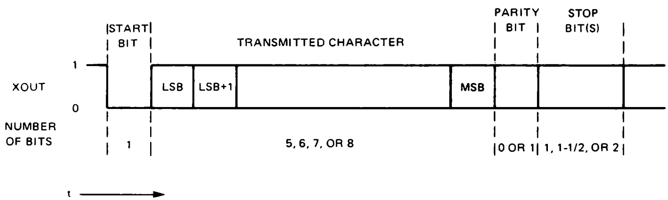
Bit 7-Bit 0
(RBR7-RBR0) — **Receive Buffer Register.** The Receive Buffer Register contains the most recently received character. For character lengths of fewer than eight bits, the character is right-justified, with unused most significant bit(s) all zero(es). The presence of valid data in the Receive Buffer Register is indicated when RBRL (Receive Buffer Register Loaded) is a logic one.

2.2 TRANSMITTER OPERATION

The operation of the transmitter is diagrammed in Figure 5. The transmitter is initialized by issuing the RESET command (output to bit 31), which causes the internal signals XSRE (Transmit Shift Register Empty) and XBRE (Transmit Buffer Register Empty) to set, and BRKON to reset. Device outputs $\overline{RTS}$ and XOUT are set, placing the transmitter in its idle state. When RTSON (Request-to-Send On) is set by the CPU, the $\overline{RTS}$ output becomes active (LOW) and the transmitter becomes active when the $\overline{CTS}$ input goes LOW.

2.2.1 Data Transmission

If the Transmit Buffer Register contains a character, transmission begins. The contents of the Transmit Buffer Register are transferred to the Transmit Shift Register, causing XSRE to reset and XBRE to set. The first bit transmitted (start bit) is always a logic zero. Subsequently, the character is shifted out, LSB first. Only the number of bits specified by RCL1 and RCL0 (character length select) of the Control Register are shifted. If parity is enabled, the correct parity bit is next transmitted. Finally the stop bit(s) selected by SBS1 and SBS0 of the Control Register are transmitted. Stop bits are always logic one. XSRE is set to indicate that no transmission is in progress, and the transmitter again tests XBRE to determine if the CPU has yet loaded the next character. The timing for a transmitted character is shown below.



TRANSMITTED CHARACTER TIMING

2.2.2 BREAK Transmission

The BREAK message is transmitted only if XBRE = 1, $\overline{CTS}$ = 0, and BRKON = 1. After transmission of the BREAK message begins, loading of the Transmit Buffer Register is inhibited and XOUT is reset. When BRKON is reset by the CPU, XOUT is set and normal operation continues. It is important to note that characters loaded into the Transmit Buffer Register are transmitted prior to the BREAK message, regardless of whether or not the character has been loaded into the Transmit Shift Register before BRKON is set. Any character to be transmitted subsequent to transmission of the BREAK message may not be loaded into the Transmit Buffer Register until after BRKON is reset.

2.2.3 Transmission Termination

Whenever XSRE = 1 and BRKON = 0 the transmitter is idle, with XOUT set to one. If RTSON is reset at this time, the $\overline{RTS}$ device output will go inactive (HIGH), disabling further data transmission until RTSON is again set. $\overline{RTS}$ will not go inactive, however, until any characters loaded into the Transmit Buffer Register prior to resetting RTSON are transmitted and BRKON = 0.

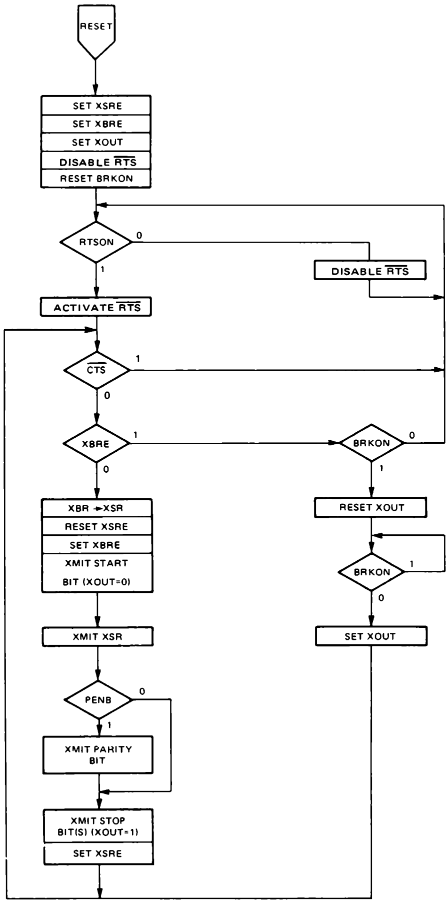


FIGURE 5. TMS 9902 TRANSMITTER OPERATION

2.3 RECEIVER OPERATION

2.3.1 Receiver Initialization

Operation of the TMS 9902 receiver is diagrammed in Figure 6. The receiver is initialized whenever the CPU issues the RESET command. The RBRL (Receive Buffer Register Loaded) flag is reset to indicate that no character is currently in the Receive Buffer Register, and the RSBD (Receive Start Bit Detect) and RFBD (Receive Full Bit Detect) flags are reset. The receiver remains in the inactive state until a one to zero transition is detected on the RIN device input.

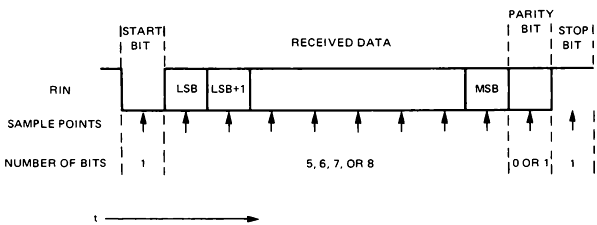
2.3.2 Start Bit Detection

The receiver delays a half bit time and again samples RIN to ensure that a valid start bit has been detected. If RIN = 0 after the half-bit delay, RSBD is set and data reception begins. If RIN = 1, no data reception occurs.

2.3.3 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 a valid received 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 sample 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 Receive Buffer Register, RBRL is set, ROVER (Receive Overrun Error) and RPER (Receive Parity Error) are loaded with appropriate values, and RIN is tested for a valid stop bit. If RIN = 1, the stop bit is valid. RFER (Receive Framing Error), 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. The timing for a received character is depicted below.



RECEIVED CHARACTER TIMING

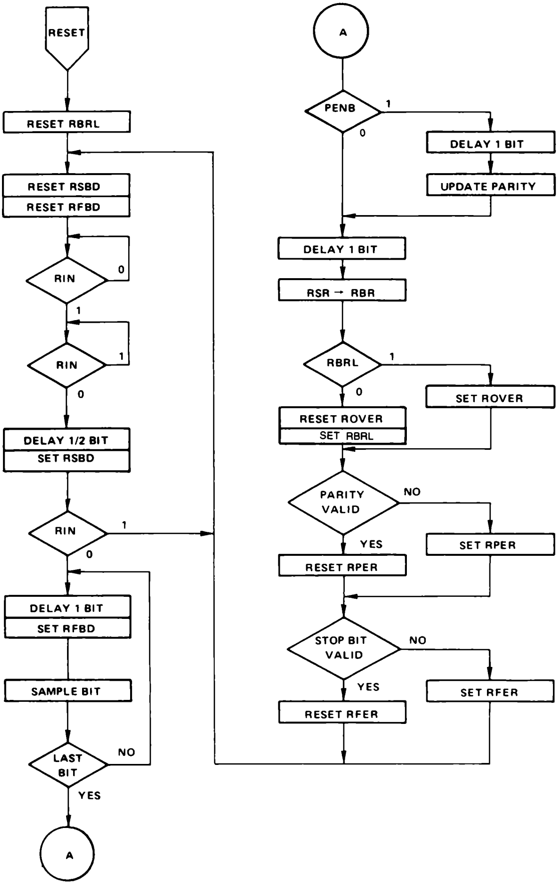


FIGURE 6. TMS 9902 RECEIVER OPERATION

2.4 INTERVAL TIMER OPERATION

A flowchart of the operation of the Interval Timer is shown in Figure 7. Execution of the RESET command by the CPU causes TIMELP (Timer Elapsed) and TIMERR (Timer Error) to reset and LDIR (Load Interval Register) to set. Resetting LDIR causes the contents of the Interval Register to be loaded into the Interval Timer, thus beginning the selected time interval. The timer is decremented every 64 internal clock cycles (every two internal clock cycles when in Test Mode) until it reaches zero, at which time the Interval Timer is reloaded by the Interval Register and TIMELP is set. If TIMELP was already set, TIMERR is set to indicate that TIMELP was not cleared by the CPU before the next time period elapsed. Each time LDIR is reset, the contents of the Interval Register are loaded into the Interval Timer, thus restarting the timer (refer also to Section 2.1.2.2).

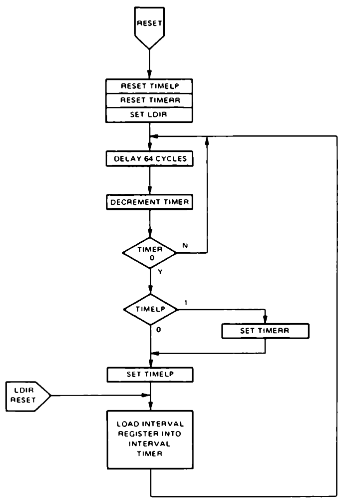


FIGURE 7. TMS 9902 INTERVAL TIMER OPERATION

INTERVAL TIMER SELECTION

2.5 INTERRUPTS

The interrupt output ($\overline{INT}$) is active (LOW) when any of the following conditions occurs and the corresponding interrupt has been enabled on the TMS 9902 by the CPU:

(1) $\overline{DSR}$ or $\overline{CTS}$ changes levels (DSCH = 1);

(2) a character has been received and stored in the Receive Buffer Register (RBRL = 1);

(3) the Transmit Buffer Register is empty (XBRE = 1); or

(4) the selected time interval has elapsed (TIMELP = 1).

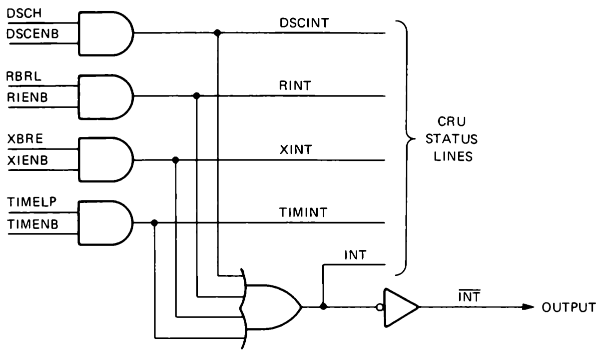


FIGURE 8. $\overline{INT}$ OUTPUT GENERATION

Figure 8 illustrates the logical equivalent of the ACC interrupt section. Table 6 lists the actions necessary to clear those conditions of the TMS 9902 that cause interrupts.

TABLE 6
TMS 9902 INTERRUPT CLEARING

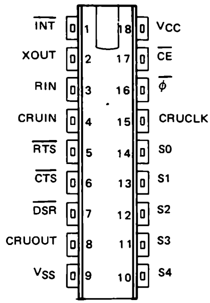
MNEMONIC	CAUSE	ACTION TO RESET
DSCINT	$\overline{CTS}$ or $\overline{DSR}$ change state	Write a bit to DSCENB (bit 21)*
RINT	Recieve Buffer Full	Write a bit to RIENB (bit 18)*
XINT	Transmit Buffer Register Empty	Load Transmit Buffer
TIMINT	Timer Elapsed	Write a bit to TIMENB (bit 20)*

*Writing a zero to clear the interrupt will clear the interrupt and disable further interrupts.

2.6 TMS 9902 TERMINAL ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
$\overline{INT}$	1	O	Interrupt — when active (LOW), the $\overline{INT}$ output indicates that at least one of the interrupt conditions has occurred.
XOUT	2	O	Transmitter Serial Data Output line — XOUT, remains inactive (HIGH) when TMS 9902 is not transmitting.
RIN	3	I	Receiver Serial Data Input Line — RCV must be held in the inactive (HIGH) state when not receiving data. A transition from HIGH to LOW activates the receiver circuitry.
CRUIN	4	O	Serial data output pin from TMS 9902 to CRUIN input pin of the CPU.
$\overline{RTS}$	5	O	Request-to-Send output from TMS 9902 to modem. $\overline{RTS}$ is enabled by the CPU and remains active (LOW) during transmission from the TMS 9902.
$\overline{CTS}$	6	I	Clear-to-Send input from modem to TMS 9902. When active (LOW), it enables the transmitter section of TMS 9902.
$\overline{DSR}$	7	I	Data Set Ready input from modem to TMS 9902. $\overline{DSR}$ generates an interrupt when it changes state.
CRUOUT	8	I	Serial data input line to TMS 9902 from CRUOUT line of the CPU.
V_{SS}	9	I	Ground reference voltage.
S4 (LSB) S3 S2 S1 S0	10 11 12 13 14	I I I I I	Address Select Lines. The data bit being accessed by the CPU interface is specified by the 5-bit code appearing on S0-S4.
CRUCLK	15	I	CRU Clock. When active (HIGH), indicates valid data on the CRUOUT line for the 9902.
$\overline{\phi}$	16	I	TTL Clock.
$\overline{CE}$	17	I	Chip Enable — when $\overline{CE}$ is inactive (HIGH), TMS 9902 CRU interface is disabled. CRUIN remains at high-impedance when $\overline{CE}$ is inactive (HIGH).
V_{CC}	18	I	Supply voltage (+5 V nominal).

TMS 9902
18-PIN PACKAGE



3. DEVICE APPLICATION

This section describes the software interface between the CPU and the TMS 9902 ACC and discusses some of the design considerations in the use of this device for asynchronous communications applications.

3.1 DEVICE INITIALIZATION

The ACC is initialized by the RESET command from the CPU (output bit 31), followed by loading the Control, Interval, Receive Data Rate, and Transmit Data Rate registers. Assume that the value to be loaded into the CRU Base Register (register 12) in order to point to bit 0 is 0040_{16}. In this application characters have seven bits of data plus even parity and one stop bit. The $\overline{\phi}$ input to the ACC is a 3 MHz signal. The ACC divides this signal frequency by three 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. The transmitter operates at a data rate of 300 bits per second, and the receiver operates at 1200 bits per second.

NOTE

> To operate both the transmitter and receiver at 300 bits per second, delete the "LDCR @RDR,11" instruction (see below), and the "LDCR @XDR,12" instruction will cause both data rate registers to be loaded and LRDR and LXDR to reset.

3.1.1 Initialization Program

The initialization program for the configuration described above is shown below. The RESET command disables all interrupts, initializes all controllers, and sets the four register load control flags (LDCTRL, LDIR, LRDR, and LXDR). Loading the last bit of each of the registers causes the load control flag to reset automatically.

```
        LI      R12,>40        INITIALIZE CRU BASE
        SBO     31             RESET COMMAND
        LDCR    @ CNTRL,8      LOAD CONTROL AND RESET LDCTRL
        LDCR    @ INTVL,8      LOAD INTERVAL AND RESET LDIR
        LDCR    @ RDR,11       LOAD RDR AND RESET LRDR
        LDCR    @ XDR,12       LOAD XDR AND RESET LXDR
          .
          .
          .
CNTRL   BYTE    >A2
INTVL   BYTE    1600/64
RDR     DATA    >1A1
XDR     DATA    >4D0
```

The RESET command initializes all subcontrollers, disables interrupts, and sets LDCTRL, LDIR, LRDR, and LXDR, enabling loading of the control register.

3.1.2 Control Register

The options listed in Table 3 in Section 2.1.2.1 are selected by loading the value shown below.

	7	6	5	4	3	2	1	0
	SBS1	SBS2	PENB	PODD	CLK4M	—	RCL1	RCL0
	MSB							LSB
VALUE	1	0	1	0	0	0	1	0

= $A2_{16}$

- 1 0 — 1 STOP BIT
- 1 0 — EVEN PARITY
- 0 — $\bar{\phi}$ DIVIDE-BY-3
- 1 0 — 7 BIT CHARACTER

3.1.3 Interval Register

To set up the interval register to generate an interrupt every 1.6 milliseconds, load the value into the interval register to specify the number of 64-microsecond increments in the total interval desired.

7	6	5	4	3	2	1	0
TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
0	0	0	1	1	0	0	1

$19_{16} = 25_{10}$

25 X 64 MICROSECONDS = 1.6 MILLISECONDS

3.1.4 Receive Data Rate Register

To set the data rate for the receiver to 1200 bits per second, load the value into the Receive Data Rate register as shown below:

10	9	8	7	6	5	4	3	2	1	0
RDV8	RDR9	RDR8	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
0	0	1	1	0	1	0	0	0	0	1

$8^{RDV8} = 1$

$1A1_{16} = 417_{10}$

$10^6 \div 1 \div 417 \div 2$ = 1199.04 BITS PER SECOND

3.1.5 Transmit Data Rate Register

To program the data rate for the transmitter for 300 bits per second, load the following value into the Transmit Data Rate register:

10	9	8	7	6	5	4	3	2	1	0
XDV8	XDR9	XDR8	XDR7	XDR6	XDR5	XDR4	XDR3	XDR2	XDR1	XDR0
1	0	0	1	1	0	1	0	0	0	0

$0D0_{16} = 208$

$_8XDV8 = 8$

$1 \times 10^6 \div 8 \div 208 \div 2 = 300.48$ BITS PER SECOND

3.2 DATA TRANSMISSION

The subroutine shown below demonstrates a simple loop 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    16              TURN ON TRANSMITTER
XMTLP   TB     22              WAIT FOR XBRE = 1
        JNE    XMTLP
        LDCR   *R0+,8          LOAD CHARACTER INCREMENT POINTER
                                  RESET XBRE
        DEC    R1              DECREMENT COUNT
        JNE    XMTLP           LOOP IF NOT COMPLETE
        SBZ    16              TURN OFF TRANSMITTER
```

After initializing the list pointer, block count, and CRU base address, RTSON is set to cause the transmitter and the $\overline{RTS}$ output to become active. Data transmission does not begin, however, until the $\overline{CTS}$ input becomes active. After the final character is loaded into the Transmit Buffer register, RTSON is reset. The transmitter and the $\overline{RTS}$ output do not become inactive until the final character is transmitted.

3.3 DATA RECEPTION

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

```
CARRET   BYTE   >0D
RCVBLK   LI     R2, RCVLST     INITIALIZE LIST COUNT
         LI     R3, MXRCNT     INITIALIZE MAX COUNT
         LI     R4, CARRET     SET UP END OF BLOCK CHARACTER
RCVLP    TB     21             WAIT FOR RBRL = 1
         JNE    RCVLP
         STCR   *R2,8          STORE CHARACTER
         SBZ    18             RESET RBRL
         DEC    R3             DECREMENT COUNT
         JEQ    RCVEND         END IF COUNT = 0
         CB     *R2+,R4        COMPARE TO EOB CHARACTER, INCREMENT
                                  POINTER
         JNE    RCVLP          LOOP IF NOT COMPLETE
RCVEND   RT                    END OF SUBROUTINE
```

3.4 REGISTER LOADING AFTER INITIALIZATION

The Control, Interval, and Data Rate registers may be reloaded after initialization. For example, it may be desirable to change the interval of the timer. Assume that the interval is to be changed to 10.24 milliseconds; the instruction sequence is:

```
         SBO    13             SET LOAD CONTROL FLAG
         LDCR   @ INTVL2,8     LOAD REGISTER, RESET FLAG
           .
           .
           .
INTVL2   BYTE   10240/64
```

When transmitter interrupts are enabled, caution should be exercised to ensure that a transmitter interrupt does not occur while the load control flag is set. For example, if a transmitter interrupt occurs 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 AND 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.

3.5 INTERFACE TO A DATA TERMINAL

Following is a discussion of the TMS 9902 interface to a TI Model 733 data terminal as implemented on the TM 990/100M microcomputer module. Figure 9 diagrams the hardware interface, and Table 7 lists the software interface. The 733 data terminal is an ASCII-code, serial, asynchronous, EIA device equipped with a keyboard, thermal printer, and digital cassette tape.

3.5.1 Hardware Interface

The hardware interface between the TMS 9902 and the 733 data terminal is shown in Figure 9. The asynchronous communication conforms to *EIA Standard RS-232-C*. The 75188 and 75189 performs the necessary level shifting between TTL levels and RS-232-C levels. The ACC chip enable ($\overline{\text{9902SEL}}$) signal comes from decode circuitry which looks at A0-A9 on CRU cycles. The interrupt output ($\overline{\text{INT}}$) of the TMS 9902 is sent to the TMS 9901 for prioritization and encoding. When the 9902 is communicating with a terminal, the $\overline{\text{RTS}}$ pin can be connected to the $\overline{\text{CTS}}$ pin because the terminal will always be in the clear-to-send ($\overline{\text{CTS}}$) condition.

3.5.2 Software

The software required to initialize, read from, and write to the TMS 9902 ACC is listed in Table 7. These routines are taken directly from TIBUG (TM 990/402-1) which is the monitor that runs on the TM 990/100M boards. The coding shown is part of a routine entered because of a power-up reset. Before this section of code was entered, not shown, R12 is set to the correct value of the TMS 9902 CRU base address. The baud rate is detected by measuring the start bit length when an "A" is entered via the keyboard. The variable COUNT is incremented every time the SPLOOP loop is executed. When a zero is seen at 9902 bit 15 (RIN) the start bits are finished being received. The value of COUNT is then compared against a table of known values in TABLE to determine the baud rate.

TIBUG assumes that all 1200-baud data terminals are TI Model 733 data terminals. The TI Model 733 communicates at 1200 baud, but prints at 300 baud; this means that bits travel the communications line at 1200 baud, but the spacing between characters is 300 baud. A wait loop is included in the write character routine to handle this spacing requirement. The TIBUG T command is used to indicate that a 1200 baud terminal is true 1200 baud; i.e., not a TI 733.

This code is taken from the middle of TIBUG; thus constructs and symbols are used which are not defned here. Lines 261 and 262 of the code contain XOP calls. The READ opcode is really a call to XOP 13 and the MESG opcode is a call to XOP 14, which in turn calls XOP 12. This can be figured out if the assembled code for these opcodes is examined. Following is a list of EQU statements that appear at the beginning of TIBUG, but are not shown here:

```
COUNT     EQU     3
POINT     EQU     7
LINK      EQU    11
CRUBAS    EQU    12
```

Once again, these values could easily be obtained by looking at the assembled code for the statement in which the symbol is used.

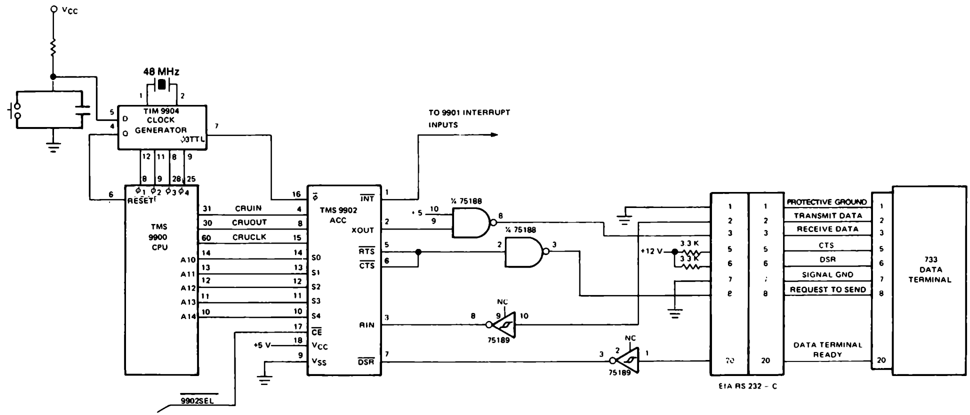


FIGURE 9. INTERFACE TO A 733 DATA TERMINAL

TABLE 7

TMS 9902 SOFTWARE

```
TIBUG
***COMMAND SEARCH AND SYSTEM INZ***

STATEMENT NO.  ADDRESS (HEX)  ASSEMBLED CODE

0231                *
0232                * INITIALIZE TMS9902 FOR: *BAUD RATE
0233                *                         *7 BITS/CHARACTER
0234                *                         *EVEN PARITY
0235                *                         *2 STOP BITS
0236                *                         *POLLED OPERATION
0237                *
0238 015E 1D1F             SBO  31            RESET TMS9902 UART

0239 0160 3220             LDCR @CR,8         INITIALIZE TMS9902 CONTROL REG
     0162 01A4'
0240 0164 1E0D             SBZ  13            DO NOT INT INTERVAL REG
0241 0166 04C3             CLR  COUNT         RESET LOOP COUNT
0242 0168 1F0F      TSTSP  TB   15            SPACE?
0243 016A 13FE             JEQ  TSTSP         NO, JUMP BACK
0244 016C 0583      SPLOOP INC  COUNT         TIME THE START BIT
0245 016E 1F0F             TB   15            FALL OUT ON A MARK
0246 0170 16FD             JNE  SPLOOP
0247                *
0248                * TABLE SEARCH FOR BAUD RATE
0249                *
0250 0172 0207             LI   POINT,TABLE   SET POINTER TO TABLE
     0174 0194'
0251 0176 8DC3      BDLOOP C    COUNT,*POINT+  MATCH?
0252 0178 1202             JLE  MATCH         YES, SET BAUD RATE
0253 017A 05C7             INCT POINT         NO, UPDATE POINTER
0254 017C 10FC             JMP  BDLOOP
0255      017E'     MATCH  EQU  $
0256 017E 3317             LDCR *POINT,12     INT. REC./XMT. DATA RATE
0257 0180 C1D7             MOV  *POINT,POINT
0258 0182 0287             CI   POINT,>1A0    1200 BAUD ?
     0184 01A0
0259 0186 1602             JNE  BANNER        LEAVE ASR FLAG ALONE
0260 0188 0720             SETO @ASR          SET 733ASR FLAG
     018A FFF4
0261 018C 2F45      BANNER READ CHAR
0262 018E 2FA0             MESG @LOGON        PRINT LOG ON MESSAGE
     0190 022B'
0263 0192 10DC             JMP  JMMONT        TO TOP OF MONITOR
0264 0194 0040      TABLE  DATA >40,>D0       2400 BAUD
     0196 00D0
0265 0198 0070             DATA >70,>1A0      1200 BAUD
     019A 01A0
0266 019C 0200             DATA >200,>4D0     300 BAUD
     019E 04D0
0267 01A0 0400             DATA >400,>638     110 BAUD
     01A2 0638
0268 01A4   62      CR     BYTE >62
```

*** WRITE CHARACTER ***

```
                    ***********************************************
                    * WRITE CHARACTER -- XOP   R,12
                    *                                ---      NORMAL RETURN
                    *
                    * TRANSMIT THE CHARACTER IN THE LEFT BYTE OF
                    * USER REGISTER R. IF THE CHARACTER IS A
                    * CARRIAGE RETURN, THE ROUTINE WAITS 200 MSEC FOR
                    * THE CARRIAGE TO RETURN. IF THE TERMINAL IS
                    * A 733ASR AS DENOTED IN THE T COMMAND, EACH
                    * CHARACTER IS PADDED WITH 25 MSEC TO REDUCE
                    * THE TRANSFER RATE TO 300 BAUD.
                    ***********************************************
01B6 020A   WENTRY LI    R10,3750
01B8 0EA6
01BA 020C          LI    CRUBAS,>80    SET CRU BASE REG.
01BC 0080
01BE 1D10          SBO   16            SET RTSON
01C0 1F16          TB    22            TRANSMIT BUFFER REG. EMPTY?
01C2 16F9          JNE   WENTRY        NO, WAIT UNTIL IT IS
01C4 321B          LDCR  *LINK,8       CHARACTER TO UART
01C6 D2DB          MOVB  *LINK,LINK
01C8 1E10          SBZ   16            RESET RTSON
01CA 098B          SRL   LINK,8
01CC 028B          CI    LINK,>000D    CARRIAGE RETURN
01CE 000D
01D0 1608          JNE   ASR733        NO, SKIP
01D2 0A3A          SLA   R10,3
01D4 1F16   WLOOP1 TB    22            WAIT FOR XMISSION TO END
01D6 16FE          JNE   WLOOP1
01D8 1F17          TB    23
01DA 16FC          JNE   WLOOP1
01DC 060A   WLOOP2 DEC   R10           WAIT LOOP
01DE 16FE          JNE   WLOOP2
01E0 0380          RTWP
01E2 C2E0   ASR733 MOV   @DUMPFG,LINK IN DUMP ROUTINE ?
01E4 FFF6
01E6 1303          JEQ   WEXIT         YES,IGNORE ASR FLAG
01E8 C2E0          MOV   @ASR,LINK     ASR733 ?
01EA FFF4
01EC 16F3          JNE   WLOOP1        YES, WAIT 3 NULLS
01EE 0380   WEXIT  RTWP
```

TIBUG

*** READ CHARACTER ***

```
                    ***********************************************
                    * READ CHARACTER -- XOP   R,13
                    *                              --      NORMAL RETURN
                    *
                    * READ WAITS FOR A CHARACTER TO BE ASSEMBLED IN
                    * THE UART. THE CHARACTER IS PLACED IN THE LEFT
                    * BYTE OF USER REGISTER R. THE RIGHT BYTE IS
                    * ZEROED. ALL ERRORS ARE IGNORED.
                    ***********************************************
                    *
01A6 020C   RENTRY LI    CRUBAS,>80    SET CRU BASE REG.
01A8 0080
01AA 1F15          TB    21            RECEIVE BUFFER REG. FULL?
01AC 16FC          JNE   RENTRY        NO, LOOP
01AE 04DB          CLR   *LINK
01B0 361B          STCR  *LINK,8
01B2 1E12          SBZ   18
01B4 0380          RTWP
```

4. TMS 9902 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings Over Operating Free Air Temperature Range (Unless Otherwise Noted) *

Supply voltage, V_{CC}	−0.3 V to 10 V
All inputs and output voltages	−0.3 V to 10 V
Continuous power dissipation	0.55 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		V_{CC}	V
Low-level input voltage, V_{IL}	V_{SS}−.3		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	$I_{OH} = -100\ \mu A$	2.4		V_{CC}	V
		$I_{OH} = -200\ \mu A$	2.2		V_{CC}	V
V_{OL}	Low level output voltage	$I_{OL} = 3.2$ 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			100	mA
C_i	Small signal input capacitance, any input	f = 1 MHz			15	pF

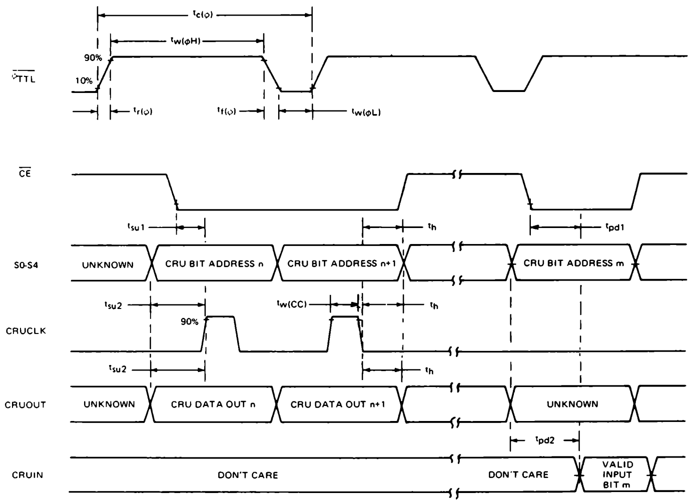
4.4 Timing Requirements Over Full Range of Operating Conditions

PARAMETER		MIN	TYP	MAX	UNIT
$t_{c(\phi)}$	Clock cycle time	300	333	667	ns
$t_{r(\phi)}$	Clock rise time	5		40	ns
$t_{f(\phi)}$	Clock fall time	10		40	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 $\overline{CE}$ before CRUCLK	150			ns
t_{su2}	Setup time for S0-S4, or CRUOUT before CRUCLK	180			ns
t_h	Hold time for $\overline{CE}$, S0-S4, or CRUOUT after CRUCLK	60			ns

*NOTE: All voltage values are referenced to V_{SS}.

4.5 Switching Characteristics Over Full Range of Recommended Operating Conditions

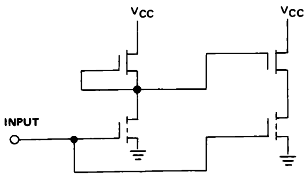
	PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
t_{pd1}	Propagation delay, $\overline{CE}$ to valid CRUIN	CL = 100pF			300	ns
t_{pd2}	Propagation delay, S0-S4 to valid CRUIN	CL = 100pF			320	ns



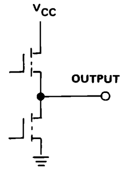
SWITCHING CHARACTERISTICS

NOTE: ALL SWITCHING TIMES ARE ASSUMED TO BE AT 10% OR 90% VALUES.

EQUIVALENT OF I/O INPUTS



EQUIVALENT OF I/O OUTPUTS

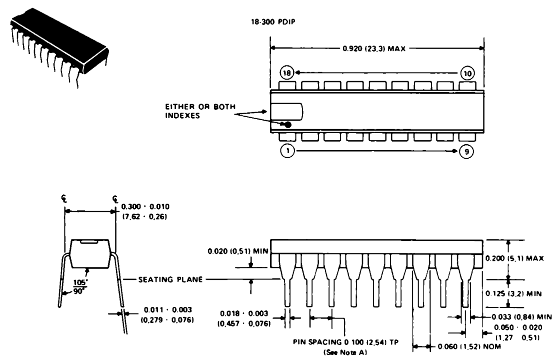


INPUT AND OUTPUT EQUIVALENTS

5. MECHANICAL SPECIFICATIONS

TMS 9902 NL

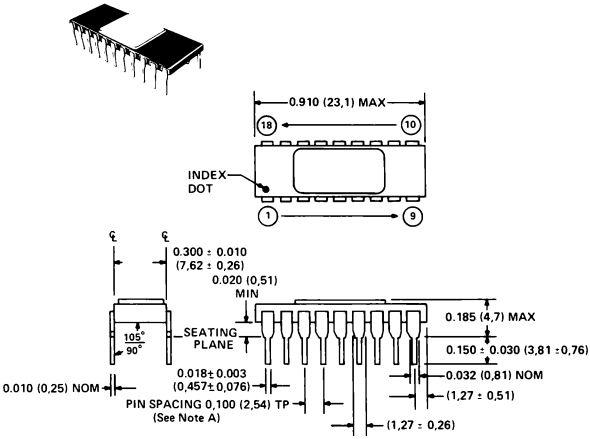
18 pin plastic packages



NOTES: A. Each pin centerline is located within 0.010 (0,26) Of its true longitudinal position
B. All linear dimensions are in inches and parenthetically in millimeters. Inch dimensions govern

TMS 9902 JL

18-pin ceramic packages with side-brazed leads and metal or epoxy or glass lid seal



NOTES: A. Each pin centerline is located within 0.010 (0,26) of its true longitudinal position.
B. All linear dimensions are in inches and parenthetically in millimeters. Inch dimensions govern.

Printed in U.S.A.