CHAPTER 8

Product Data Book

8◂

Feature \ Device Type	TMS 9900	TMS 9900-40	SBP 9900A	TMS 9980A/ TMS 9981	TMS 9985	TMS 9940
Number of bytes addressable	65k		65k	16k	65k 256 RAM	2k EPROM 128 RAM
Number of Interrupts	16		16	5	5	4
Number of Pins	64		64	40	40	40
Power Supply Requirements	+5, −5, +12		500 Ma *(Note 1)*	+5,	+5	+5
Technology	N-MOS		I^2L	N-MOS	N-MOS	N-MOS
Environmental (Temperature)	0-70°C		−55-125°C	0-70°C	0-70°C	0-70°C
Clock Rate	3.3 MHz	4 MHz	3 MHz	10MHz	5 MHz	5 MHz
Relative Thruput	1.0	1.3	0.9	0.6	0.65-0.8 *(Note 2)*	1.2
Number of Address Bus Lines	15		15	14	16	*(Note 3)*
Number of Data Bus Lines	16		16	8	8	*(Note 3)*
Clock	TIM 9904		SN54LS124	On chip	On chip	On chip

Note 1: Voltage for the SBP 9900A is 1.5 to 30 volts with a series resistor.
Note 2: The relative thruput is 0.65 with an off-chip RAM and 0.8 with an on-chip RAM.
Note 3: There are 32 general purpose pins which can be programmed for I/O. While the memory and data buses are not available, 8 address bits are accessible for CRU I/O expansion.

FAMILY DESCRIPTION

The TMS 9900 micrprocessor is a single-chip 16-bit central processing unit (CPU) produced using N-channel silicon-gate MOS technology. The instruction set of the TMS 9900 includes the capabilities offered by full minicomputers. The unique memory-to-memory architecture features multiple register files, resident in memory, which allow faster response to interrupts and increased programming flexibility. The separate bus structure simplifies the system design effort. Texas Instruments provides a compatible set of MOS and TTL memory and logic function circuits to be used with a TMS 9900 system. The system is fully supported by software and a complete prototyping system.

▶8

There is a TMS 9900-40 part designed for 4-MHz operation. Refer to the separate "−40" (dash forty) electrical specification tables for exact characteristics.

The SBP 9900A is basically the same as the TMS 9900 but it employs I^2L technology to enhance environmental specifications. It is a static, bipolar microprocessor operating from a single phase clock over the frequency range from 0 to 3 MHz. The I/O is fully TTL compatible so that no special peripheral circuits are required. The power supply is specified as a single input injector current which may be varied over the range from 10 mA to 1 A with a corresponding change in speed (described in the specific SBP 9900A section). The architecture is the same for the SBP 9900A as the TMS 9900 with minor differences in clock and control lines.

Software compatibility with other 9900 microprocessor family members provides a common body of hardware/software within Texas Instruments 990 minicomputer family.
The TMS 9980A/TMS 9981 is another software-compatible member of TI's 9900 family of microprocessors. Designed to minimize the system cost for smaller systems, the TMS 9980A/TMS 9981 is a single-chip 16-bit central processing unit (CPU) which has an 8-bit data bus, on-chip clock, and is packaged in a 40-pin package. The instruction set of the TMS 9980A/TMS 9981 includes the capabilities offered by full minicomputers and is exactly the same as the 9900's.

The TMS 9940 is a single-chip, 16-bit microprocessor containing a CPU, memory (RAM and EPROM/ROM), and extensive I/O. Except for four instructions which do not apply to the TMS 9940 microcomputer configuration, the TMS 9940 instruction set matches that of the TMS 9900 and includes capabilities offered by *minicomputers*. In addition, the TMS 9940 instruction set includes two instructions which facilitate manipulation of binary coded decimal (BCD) data, and a single-word load-interrupt-mask (LIIM) instruction.
The unique memory-to-memory architecture features multiple register files, resident in the RAM, which allow faster response to interrupts and increased programming flexibility. The memory consists of 128 bytes of RAM and 2048 bytes of EPROM. The TMS 9940 implements four levels of interrupts including an internal decrementer which can be programmed as a timer or an event counter. All members of the TMS 9900 family of peripheral circuits are compatible with the TMS 9940. The TMS 9940 is fully supported by software and hardware development systems and by factory applications engineers and technical answering services.

The TMS 9985 is a software compatible member of TI's 9900 family of microprocessors and microcomputers and contains a 16-bit CPU, 256 bytes of RAM, on chip timer/event counter, external 16-bit address bus and 8-bit data bus, and is in a 40-pin package. The instruction set of the TMS 9985 includes the capabilities offered by full minicomputers and is exactly the same as the TMS 9940 microcomputer's. The unique memory-to-memory architecture features multiple register files, resident in memory, which allows faster response to interrupts and increased programming flexibility. In addition, the TMS 9985 has excellent I/O flexibility with CRU, memory mapped I/O and direct memory access.

COMMON KEY FEATURES

- 16-Bit Architecture
- 69 Powerful Instructions Include:
  - Multiply and Divide
  - 5 Addressing Modes
  - Bit, Byte, and Word Addressing
  - One, Two and Three Word Instructions
- Rapid Hardware Context Switching
- Multiple 16-Word Register Files (Workspaces) Reside in Memory
- Separate I/O, Memory and Interrupt Bus Structures
- Programmed and DMA I/O Capability
- Communications Register Unit (CRU) for Low and Medium Speed Devices
- Efficient Memory-to-Memory Architecture
- Extended Operations (XOP) Feature Allows Users to Augment the Instruction Set
- Maskable Vectored Priority Interrupts for Multiprogramming Requirements
- Software Compatible with 990 Minicomputer Family

KEY FEATURES OF SPECIFIC DEVICES

TMS 9900

- 3.3-MHz Speed (4.0 MHz Speed for the TMS 9900-40)
- Up to 65,536 Bytes of Memory
- 16 Prioritized Interrupts
- 64-Pin Package
- N-Channel Silicon-Gate Technology
- 0-70°C Ambient Temperature Range
- Directly TTL Compatible I/O

SBP 9900A

- 3-MHz Speed
- Up to 65,536 Bytes of Memory
- 16 Prioritized Interrupts
- 64-Pin Package
- I^2L Technology
- −55 to 125°C Ambient Temperature Range
- Single dc Power Supply
- Directly TTL Compatible I/O

TMS 9980A/TMS9981

- 10-MHz Speed
- Up to 16,384 Bytes of Memory
- 8-Bit Memory Data Bus
- 4 Prioritized Interrupts
- On-Chip 4-Phase Clock Generator
- 40-Pin Package
- N-Channel Silicon-Gate Technology
- 0-70°C Ambient Temperature Range

TMS 9980A/TMS 9981 Differences

The TMS 9980A and the TMS 9981, although very similar, have several differences of which the user should be aware.

1. The TMS 9980A requires a V_{BB} supply (pin 21) while the TMS 9981 has an internal charge pump to generate V_{BB} from V_{CC} and V_{DD}.
2. The TMS 9981 has an optional on-chip crystal oscillator in addition to the external clock mode of the TMS 9980A.
3. The pin-outs are not the same for D0-D7, INT0-INT2, and $\overline{\phi 3}$.

TMS 9985

- 5-MHz Speed
- Up to 65,536 Bytes of Memory
- 8-Bit Memory Data Bus
- 5 Prioritized Interrupts
- 40-Pin Package
- N-Channel Silicon-Gate Technology
- 0-70°C Ambient Temperature Range
- On Chip Timer/Event Counter
- 256 Bits of RAM on Chip
- Separate Memory, I/O and Interrupt Bus Structures
- On Chip Programmable Flags (16)
- Multiprocessor System Interface
- Single 5-Volt Supply
- Speed Selected Versions

TMS 9940

- 5-MHz Speed
- 2048 Bytes of EPROM or ROM
- 128 Bytes of RAM
- 4 Prioritized Interrupts
- 40-Pin Package
- N-Channel Silicon-Gate Technology
- 0-70°C Ambient Temperature Range
- On-Chip Timer/Event Counter
- 32 Bits General Purpose I/O
- 256 Bits I/O Expansion
- Multiprocessor System Interface
- Single 5-Volt Power Supply
- Power Down Capability for Low Standby Power
- Easy Test Function
- Offered as either an EPROM device as a mask ROM device
- Speed Selected Versions

Organization of CPU Data Manuals and Instruction Set

Data manuals for the five CPU's in the 9900 family are reproduced in this section with the TMS 9900 first, followed by the SBP 9900A, TMS 9980A/81, and TMS 9940 data manuals. Following this there is an abbreviated version of the TMS 9985 manual. Since the information regarding the instruction set is common to all of the CPU's, it has been removed from the individual manuals and is printed at the end of this section.

8◀

TMS 9900

1. INTRODUCTION

1.1 DESCRIPTION

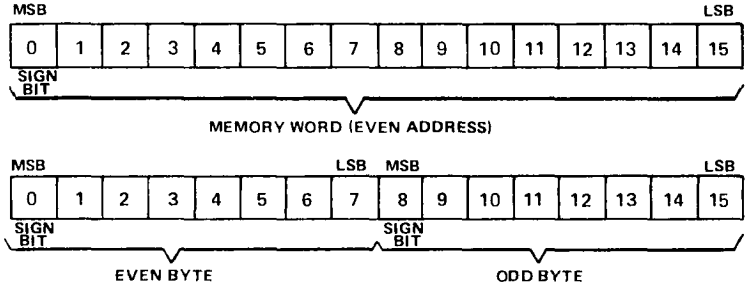
The TMS 9900 microprocessor is a single-chip 16-bit central processing unit (CPU) produced using N-channel silicon-gate MOS technology (see Figure 1). The instruction set of the TMS 9900 includes the capabilities offered by full minicomputers. The unique memory-to-memory architecture features multiple register files, resident in memory, which allow faster response to interrupts and increased programming flexibility. The separate bus structure simplifies the system design effort. Texas Instruments provides a compatible set of MOS and TTL memory and logic function circuits to be used with a TMS 9900 system. The system is fully supported by software and a complete prototyping system.

1.2 KEY FEATURES

- 16-Bit Instruction Word
- Full Minicomputer Instruction Set Capability Including Multiply and Divide
- Up to 65,536 Bytes of Memory
- 3.3 – MHz Speed
- Advanced Memory-to-Memory Architecture
- Separate Memory, I/O, and Interrupt-Bus Structures
- 16 General Registers
- 16 Prioritized Interrupts
- Programmed and DMA I/O Capability
- N-Channel Silicon-Gate Technology

2. ARCHITECTURE

The memory word of the TMS 9900 is 16 bits long. Each word is also defined as 2 bytes of 8 bits. The instruction set of the TMS 9900 allows both word and byte operands. Thus, all memory locations are on even address boundaries and byte instructions can address either the even or odd byte. The memory space is 65,536 bytes or 32,768 words. The word and byte formats are shown below.



2.1 REGISTERS AND MEMORY

The TMS 9900 employs an advanced memory-to-memory architecture. Blocks of memory designated as workspace replace internal-hardware registers with program-data registers. The TMS 9900 memory map is shown in Figure 2. The first 32 words are used for interrupt trap vectors. The next contiguous block of 32 memory words is used by the extended operation (XOP) instruction for trap vectors. The last two memory words, $FFFC_{16}$ and $FFFE_{16}$, are used for the trap vector of the LOAD signal. The remaining memory is then available for programs, data, and workspace registers. If desired, any of the special areas may also be used as general memory.

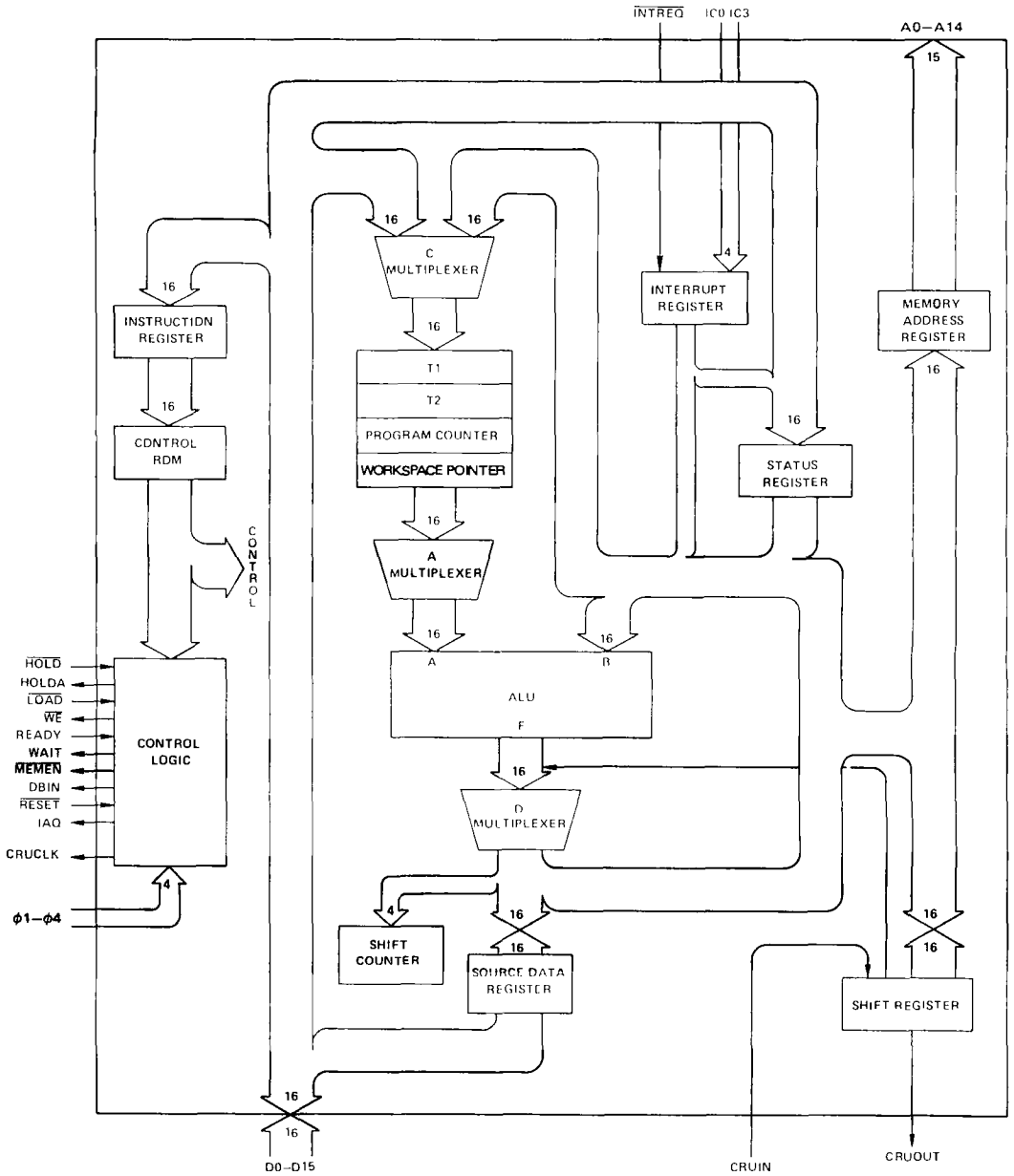


FIGURE 1 – ARCHITECTURE

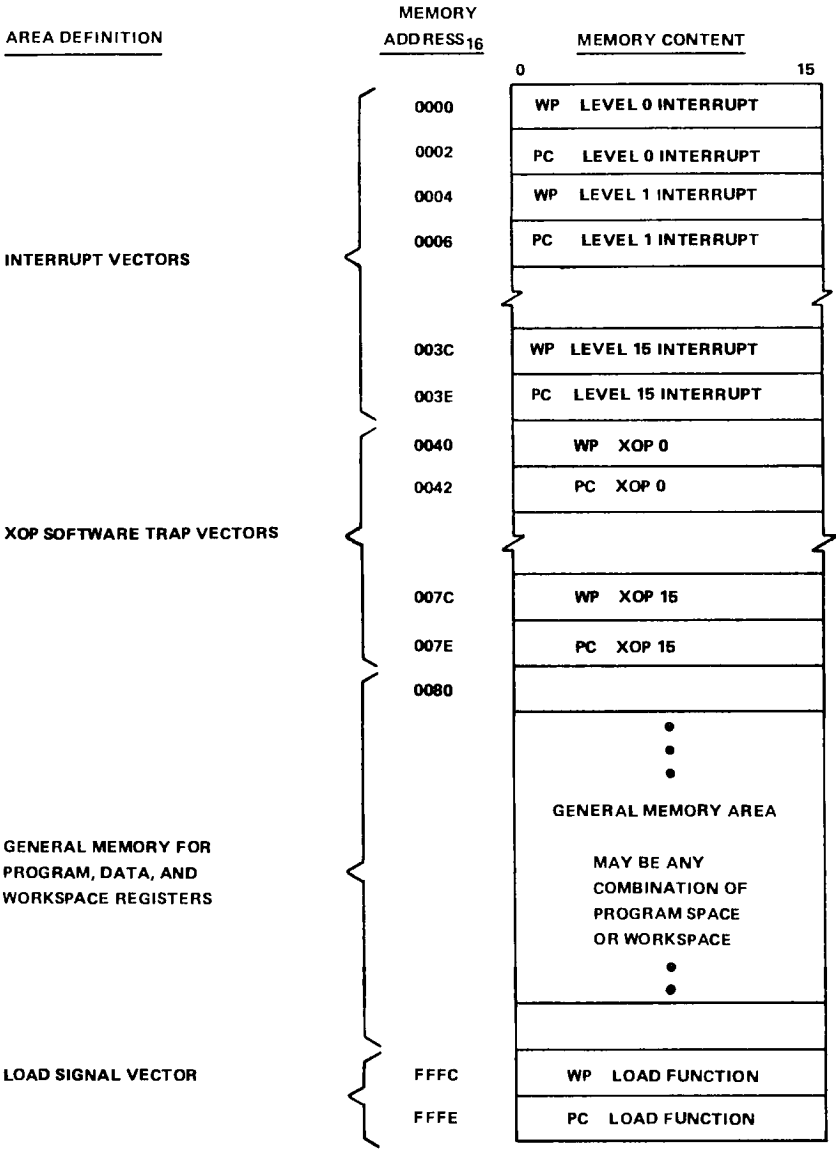
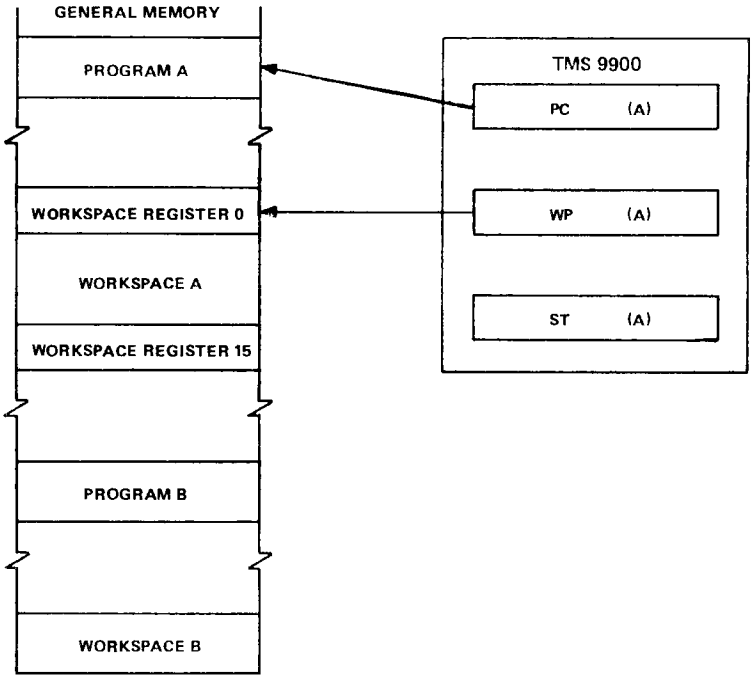


FIGURE 2 – MEMORY MAP

Three internal registers are accessible to the user. The program counter (PC) contains the address of the instruction following the current instruction being executed. This address is referenced by the processor to fetch the next instruction from memory and is then automatically incremented. The status register (ST) contains the present state of the processor and will be further defined in the Instruction Set section. The workspace pointer (WP) contains the address of the first word in the currently active set of workspace registers.

A workspace-register file occupies 16 contiguous memory words in the general memory area (see Figure 2). Each workspace register may hold data or addresses and function as operand registers, accumulators, address registers, or

index registers. During instruction execution, the processor addresses any register in the workspace by adding the register number to the contents of the workspace pointer and initiating a memory request for the word. The relationship between the workspace pointer and its corresponding workspace is shown below.



The workspace concept is particularly valuable during operations that require a context switch, which is a change from one program environment to another (as in the case of an interrupt) or to a subroutine. Such an operation, using a conventional multi-register arrangement, requires that at least part of the contents of the register file be stored and reloaded. A memory cycle is required to store or fetch each word. By exchanging the program counter, status register, and workspace pointer, the TMS 9900 accomplishes a complete context switch with only three store cycles and three fetch cycles. After the switch the workspace pointer contains the starting address of a new 16-word workspace in memory for use in the new routine. A corresponding time saving occurs when the original context is restored. Instructions in the TMS 9900 that result in a context switch include:

1. Branch and Load Workspace Pointer (BLWP)
2. Return from Subroutine (RTWP)
3. Extended Operation (XOP).

Device interrupts, $\overline{\text{RESET}}$, and $\overline{\text{LOAD}}$ also cause a context switch by forcing the processor to trap to a service subroutine.

2.2 INTERRUPTS

The TMS 9900 employs 16 interrupt levels with the highest priority level 0 and lowest level 15. Level 0 is reserved for the $\overline{\text{RESET}}$ function and all other levels may be used for external devices. The external levels may also be shared by several device interrupts, depending upon system requirements.

The TMS 9900 continuously compares the interrupt code (IC0 through IC3) with the interrupt mask contained in status-register bits 12 through 15. When the level of the pending interrupt is less than or equal to the enabling mask level (higher or equal priority interrupt), the processor recognizes the interrupt and initiates a context switch following

completion of the currently executing instruction. The processor fetches the new context WP and PC from the interrupt vector locations. Then, the previous context WP, PC, and ST are stored in workspace registers 13, 14, and 15, respectively, of the new workspace. The TMS 9900 then forces the interrupt mask to a value that is one less than the level of the interrupt being serviced, except for level-zero interrupt, which loads zero into the mask. This allows only interrupts of higher priority to interrupt a service routine. The processor also inhibits interrupts until the first instruction of the service routine has been executed to preserve program linkage should a higher priority interrupt occur. All interrupt requests should remain active until recognized by the processor in the device-service routine. The individual service routines must reset the interrupt requests before the routine is complete.

If a higher priority interrupt occurs, a second context switch occurs to service the higher priority interrupt. When that routine is complete, a return instruction (RTWP) restores the first service routine parameters to the processor to complete processing of the lower-priority interrupt. All interrupt subroutines should terminate with the return instruction to restore original program parameters. The interrupt-vector locations, device assignment, enabling-mask value, and the interrupt code are shown in Table 1.

TABLE 1
INTERRUPT LEVEL DATA

Interrupt Level	Vector Location (Memory Address In Hex)	Device Assignment	Interrupt Mask Values To Enable Respective Interrupts (ST12 thru ST15)	Interrupt Codes IC0 thru IC3
(Highest priority) 0	00	Reset	0 through F*	0000
1	04	External device	1 through F	0001
2	08	↓	2 through F	0010
3	0C	↓	3 through F	0011
4	10	↓	4 through F	0100
5	14	↓	5 through F	0101
6	18	↓	6 through F	0110
7	1C	↓	7 through F	0111
8	20	↓	8 through F	1000
9	24	↓	9 through F	1001
10	28	↓	A through F	1010
11	2C	↓	B through F	1011
12	30	↓	C through F	1100
13	34	↓	D through F	1101
14	38	↓	E and F	1110
(Lowest priority) 15	3C	External device	F only	1111

*Level 0 can not be disabled.

The TMS 9900 interrupt interface utilizes standard TTL components as shown in Figure 3. Note that for eight or less external interrupts a single SN74148 is required and for one external interrupt $\overline{INT}$ is used as the interrupt signal with a hard-wired code IC0 through IC3.

2.3 INPUT/OUTPUT

The TMS 9900 utilizes a versatile direct command-driven I/O interface designated as the communications-register unit (CRU). The CRU provides up to 4096 directly addressable input bits and 4096 directly addressable output bits. Both input and output bits can be addressed individually or in fields of from 1 to 16 bits. The TMS 9900 employs three dedicated I/O pins (CRUIN, CRUOUT, and CRUCLK) and 12 bits (A3 through A14) of the address bus to interface with the CRU system. The processor instructions that drive the CRU interface can set, reset, or test any bit in the CRU array or move between memory and CRU data fields.

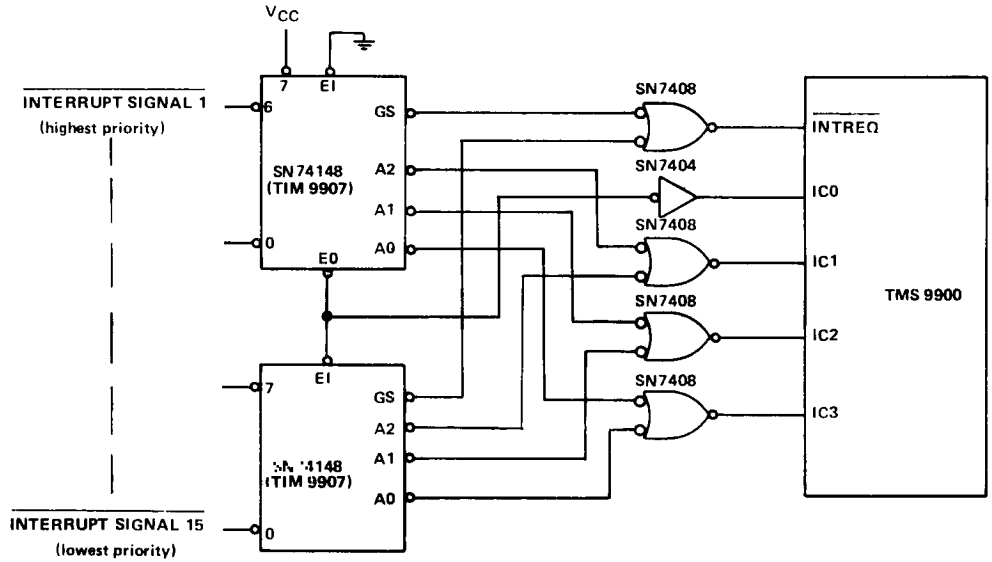


FIGURE 3 – TMS 9900 INTERRUPT INTERFACE

2.4 SINGLE-BIT CRU OPERATIONS

The TMS 9900 performs three single-bit CRU functions: test bit (TB), set bit to one (SBO), and set bit to zero (SBZ) To identify the bit to be operated upon, the TMS 9900 develops a CRU-bit address and places it on the address bus, A3 to A14.

For the two output operations (SBO and SBZ), the processor also generates a CRUCLK pulse, indicating an output operation to the CRU device, and places bit 7 of the instruction word on the CRUOUT line to accomplish the specified operation (bit 7 is a one for SBO and a zero for SBZ). A test-bit instruction transfers the addressed CRU bit from the CRUIN input line to bit 2 of the status register (EQUAL).

The TMS 9900 develops a CRU-bit address for the single-bit operations from the software base address contained in workspace register 12 and the signed displacement count contained in bits 8 through 15 of the instruction. The displacement allows two's complement addressing from base minus 128 bits through base plus 127 bits. The hardware base address, bits 3 through 14 of W12, is added to the signed displacement specified in the instruction and the result is loaded onto the address bus. *Figure 4* illustrates the development of a single-bit CRU address.

2.5 MULTIPLE-BIT CRU OPERATIONS

The TMS 9900 performs two multiple-bit CRU operations: store communications register (STCR) and load communications register (LDCR). Both operations perform a data transfer from the CRU-to-memory or from memory-to-CRU as illustrated in Figure 5. Although the figure illustrates a full 16-bit transfer operation, any number of bits from 1 through 16 may be involved. The LDCR instruction fetches a word from memory and right-shifts it to serially transfer it to CRU output bits. If the LDCR involves eight or fewer bits, those bits come from the right-justified field within the addressed byte of the memory word. If the LDCR involves nine or more bits, those bits come from the right-justified field within the whole memory word. When transferred to the CRU interface, each successive bit receives an address that is sequentially greater than the address for the previous bit. This addressing mechanism results in an order reversal of the bits; that is, bit 15 of the memory word (or bit 7) becomes the lowest addressed bit in the CRU and bit 0 becomes the highest addressed bit in the CRU field.

An STCR instruction transfers data from the CRU to memory. If the operation involves a byte or less transfer, the transferred data will be stored right-justified in the memory byte with leading bits set to zero. If the operation involves from nine to 16 bits, the transferred data is stored right-justified in the memory word with leading bits set to zero.

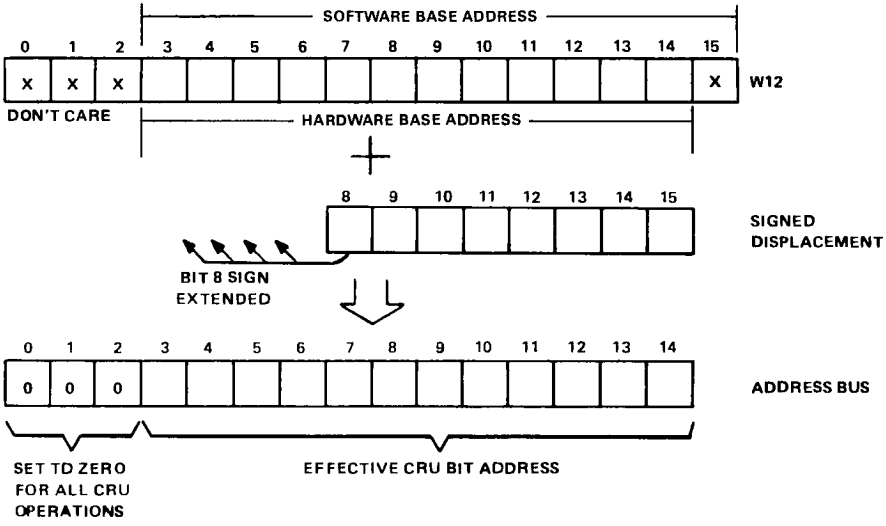


FIGURE 4 – TMS 9900 SINGLE-BIT CRU ADDRESS DEVELOPMENT

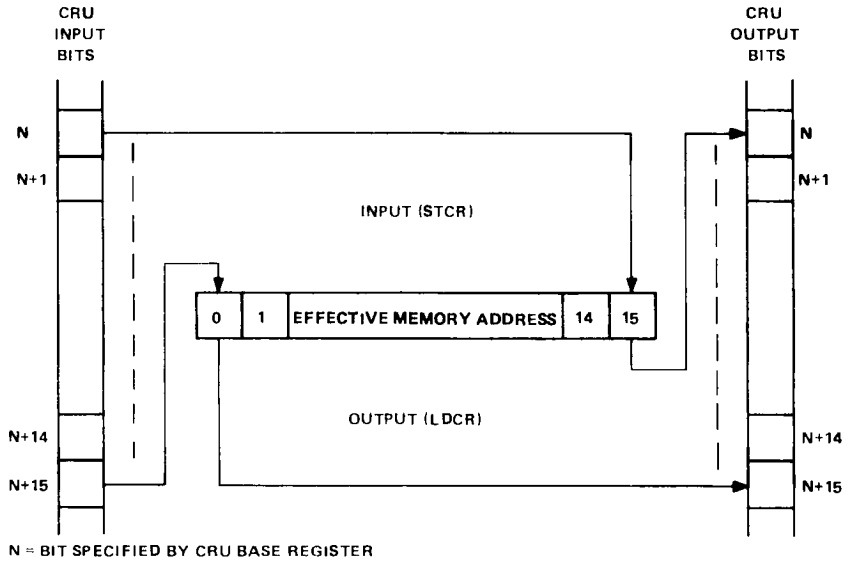


FIGURE 5 – TMS 9900 LDCR/STCR DATA TRANSFERS

When the input from the CRU device is complete, the first bit from the CRU is the least-significant-bit position in the memory word or byte.

Figure 6 illustrates how to implement a 16-bit input and a 16-bit output register in the CRU interface. CRU addresses are decoded as needed to implement up to 256 such 16-bit interface registers. In system application, however, only the exact number of interface bits needed to interface specific peripheral devices are implemented. It is not necessary to have a 16-bit interface register to interface an 8-bit device.

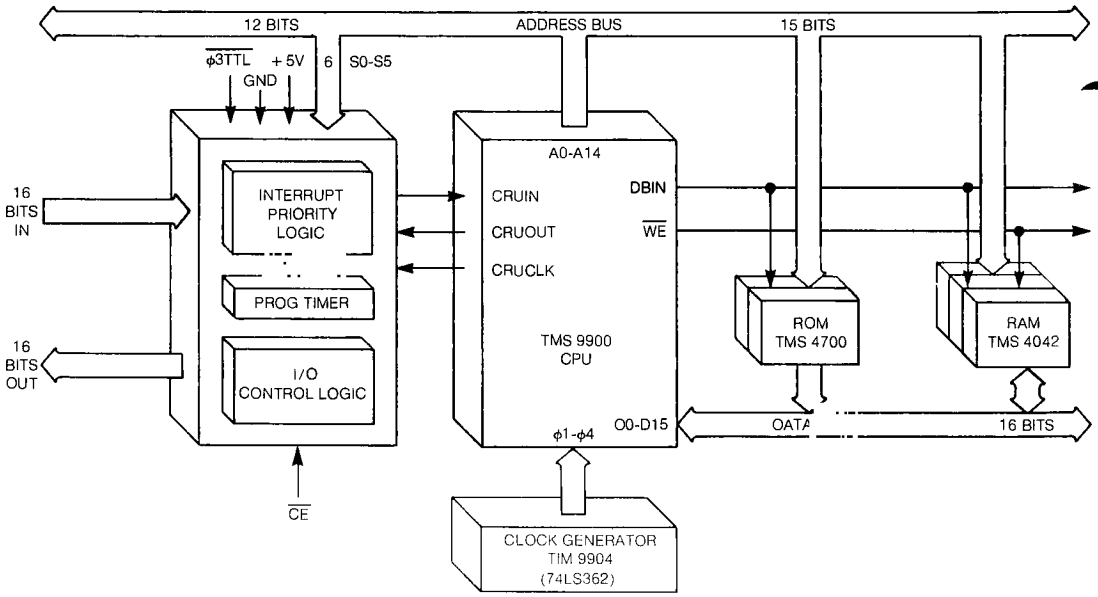


FIGURE 6 – TMS 9900 16-BIT INPUT/OUTPUT INTERFACE

2.6 EXTERNAL INSTRUCTIONS

The TMS 9900 has five external instructions that allow user-defined external functions to be initiated under program control. These instructions are CKON, CKOF, RSET, IDLE, and LREX. These mnemonics, except for IDLE, relate to functions implemented in the 990 minicomputer and do not restrict use of the instructions to initiate various user-defined functions. IDLE also causes the TMS 9900 to enter the idle state and remain until an interrupt, $\overline{\text{RESET}}$, or $\overline{\text{LOAD}}$ occurs. When any of these five instructions are executed by the TMS 9900, a unique 3-bit code appears on the most-significant 3 bits of the address bus (A0 through A2) along with a CRUCLK pulse. When the TMS 9900 is in an idle state, the 3-bit code and CRUCLK pulses occur repeatedly until the idle state is terminated. The codes are:

EXTERNAL INSTRUCTION	A0	A1	A2
LREX	H	H	H
CKOF	H	H	L
CKON	H	L	H
RSET	L	H	H
IOLE	L	H	L

Figure 7 illustrates typical external decode logic to implement these instructions. Note that a signal is generated to inhibit CRU decodes during external instructions.

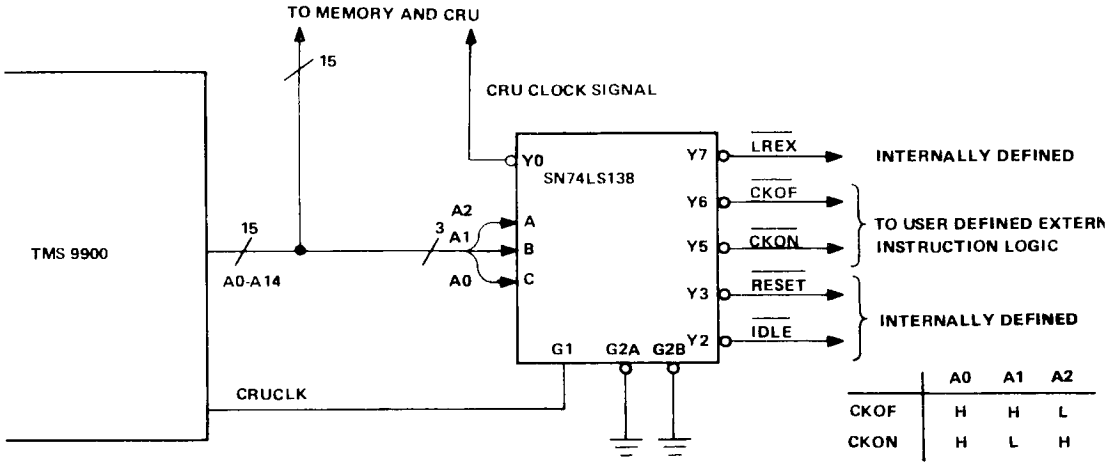


FIGURE 7 – EXTERNAL INSTRUCTION DECODE LOGIC

2.7 LOAD FUNCTION

The $\overline{\text{LOAD}}$ signal allows cold-start ROM loaders and front panels to be implemented for the TMS 9900. When active, $\overline{\text{LOAD}}$ causes the TMS 9900 to initiate an interrupt sequence immediately following the instruction being executed. Memory location FFFC is used to obtain the vector (WP and PC). The old PC, WP and ST are loaded into the new workspace and the interrupt mask is set to 0000. Then, program execution resumes using the new PC and WP.

8◀

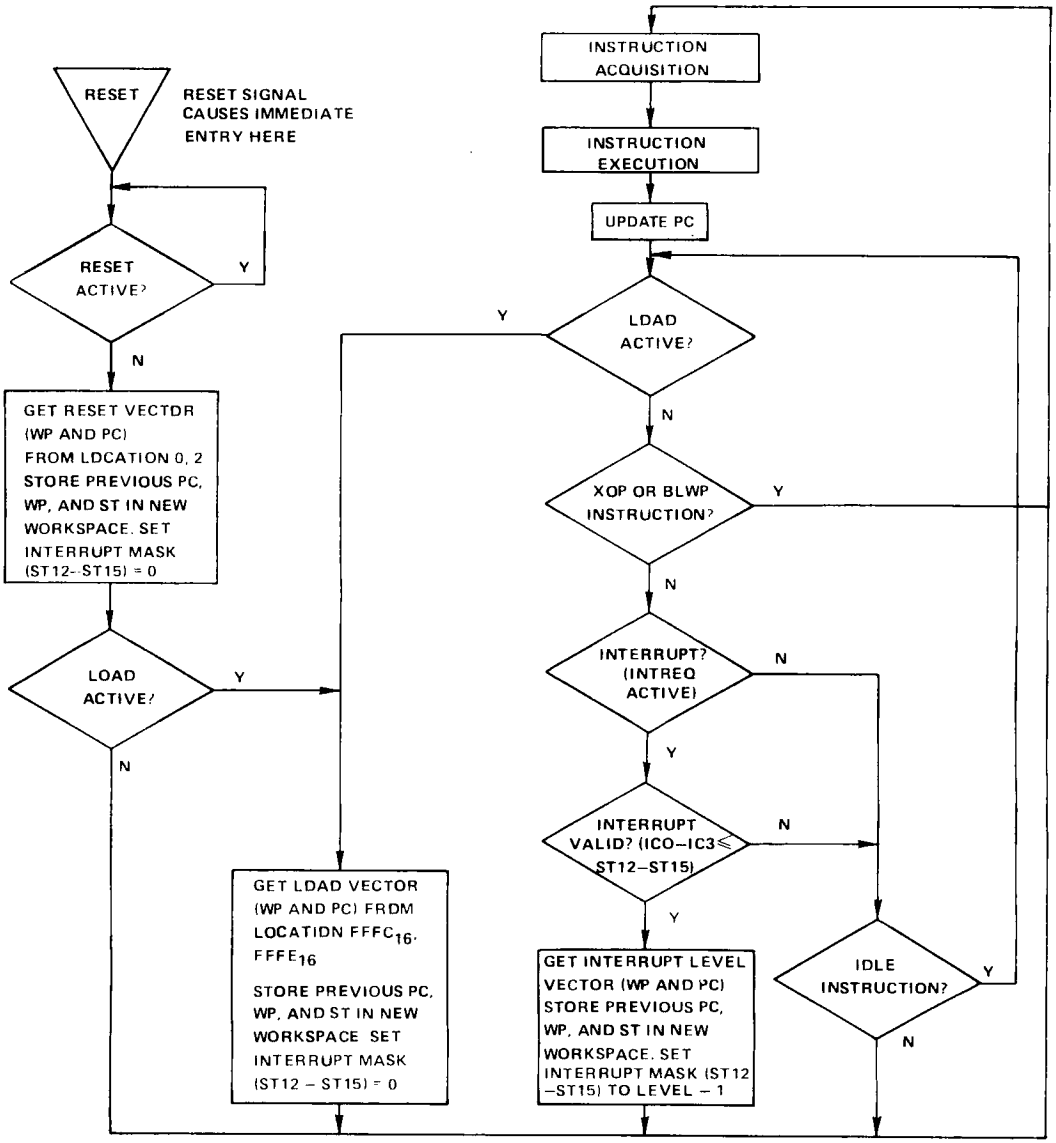


FIGURE 8 - TMS 9900 CPU FLOW CHART

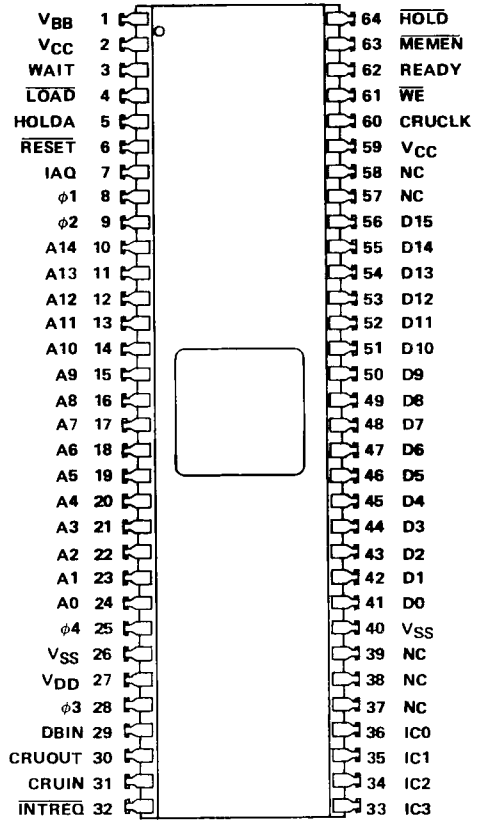
2.8 TMS 9900 PIN DESCRIPTION

Table 2 defines the TMS 9900 pin assignments and describes the function of each pin.

TABLE 2
TMS 9900 PIN ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
			ADDRESS BUS
A0 (MSB)	24	OUT	A0 through A14 comprise the address bus.
A1	23	OUT	This 3-state bus provides the memory-
A2	22	OUT	address vector to the external-memory
A3	21	OUT	system when $\overline{\text{MEMEN}}$ is active and I/O-bit
A4	20	OUT	addresses and external-instruction addresses
A5	19	OUT	to the I/O system when $\overline{\text{MEMEN}}$ is inactive.
A6	18	OUT	The address bus assumes the high-impedance
A7	17	OUT	state when HOLDA is active.
A8	16	OUT	
A9	15	OUT	
A10	14	OUT	
A11	13	OUT	
A12	12	OUT	
A13	11	OUT	
A14 (LSB)	10	OUT	
			DATA BUS
D0 (MSB)	41	I/O	D0 through D15 comprise the bidirectional
D1	42	I/O	3-state data bus. This bus transfers memory
D2	43	I/O	data to (when writing) and from (when
D3	44	I/O	reading) the external-memory system when
D4	45	I/O	$\overline{\text{MEMEN}}$ is active. The data bus assumes the
D5	46	I/O	high-impedance state when HOLDA is
D6	47	I/O	active.
D7	48	I/O	
D8	49	I/O	
D9	50	I/O	
D10	51	I/O	
D11	52	I/O	
D12	53	I/O	
D13	54	I/O	
D14	55	I/O	
D15 (LSB)	56	I/O	
			POWER SUPPLIES
V_{BB}	1		Supply voltage (−5 V NOM)
V_{CC}	2,59		Supply voltage (5 V NOM). Pins 2 and 59 must be connected in parallel.
V_{DD}	27		Supply voltage (12 V NOM)
V_{SS}	26,40		Ground reference. Pins 26 and 40 must be connected in parallel.
			CLOCKS
ϕ1	8	IN	Phase-1 clock
ϕ2	9	IN	Phase-2 clock
ϕ3	28	IN	Phase-3 clock
ϕ4	25	IN	Phase-4 clock

TMS 9900 PIN ASSIGNMENTS



NC — No Internal connection

TABLE 2 (CONTINUED)

SIGNATURE	PIN	I/O	DESCRIPTION
			BUS CONTROL
DBIN	29	OUT	Data bus in. When active (high), DBIN indicates that the TMS 9900 has disabled its output buffers to allow the memory to place memory-read data on the data bus during MEMEN. DBIN remains low in all other cases except when HOLDA is active.
MEMEN	63	OUT	Memory enable. When active (low), MEMEN indicates that the address bus contains a memory address.
WE	61	OUT	Write enable. When active (low), WE indicates that memory-write data is available from the TMS 9900 to be written into memory.
CRUCLK	60	OUT	CRU clock. When active (high), CRUCLK indicates that external interface logic should sample the output data on CRUOUT or should decode external instructions on A0 through A2.
CRUIN	31	IN	CRU data in. CRUIN, normally driven by 3-state or open-collector devices, receives input data from external interface logic. When the processor executes a STCR or TB instruction, it samples CRUIN for the level of the CRU input bit specified by the address bus (A3 through A14).
CRUOUT	30	OUT	CRU data out. Serial I/O data appears on the CRUOUT line when an LDCR, SBZ, or SBO instruction is executed. The data on CRUOUT should be sampled by external I/O interface logic when CRUCLK goes active (high).
			INTERRUPT CONTROL
INTREQ	32	IN	Interrupt request. When active (low), INTREQ indicates that an external interrupt is requested. If INTREQ is active, the processor loads the data on the interrupt-code-input lines IC0 through IC3 into the internal interrupt-code-storage register. The code is compared to the interrupt mask bits of the status register. If equal or higher priority than the enabled interrupt level (interrupt code equal or less than status register bits 12 through 15) the TMS 9900 interrupt sequence is initiated. If the comparison fails, the processor ignores the request. INTREQ should remain active and the processor will continue to sample IC0 through IC3 until the program enables a sufficiently low priority to accept the request interrupt.
IC0 (MSB)	36	IN	Interrupt codes. IC0 is the MSB of the interrupt code, which is sampled when INTREQ is active. When IC0 through IC3 are LLLH, the highest external-priority interrupt is being requested and when HHHH, the lowest-priority interrupt is being requested.
IC1	35	IN	
IC2	34	IN	
IC3 (LSB)	33	IN	
			MEMORY CONTROL
HOLD	64	IN	Hold. When active (low), HOLD indicates to the processor that an external controller (e.g., DMA device) desires to utilize the address and data buses to transfer data to or from memory. The TMS 9900 enters the hold state following a hold signal when it has completed its present memory cycle.* The processor then places the address and data buses in the high-impedance state (along with WE, MEMEN, and DBIN) and responds with a hold-acknowledge signal (HOLDA). When HOLD is removed, the processor returns to normal operation.
HOLDA	5	OUT	Hold acknowledge. When active (high), HOLDA indicates that the processor is in the hold state and the address and data buses and memory control outputs (WE, MEMEN, and DBIN) are in the high-impedance state.
READY	62	IN	Ready. When active (high), READY indicates that memory will be ready to read or write during the next clock cycle. When not-ready is indicated during a memory operation, the TMS 9900 enters a wait state and suspends internal operation until the memory systems indicate ready.
WAIT	3	OUT	Wait. When active (high), WAIT indicates that the TMS 9900 has entered a wait state because of a not-ready condition from memory.

*If the cycle following the present memory cycle is also a memory cycle, it, too, is completed before the TMS9900 enters the hold state. The maximum number of consecutive memory cycles is three.

TABLE 2 (CONCLUDED)

SIGNATURE	PIN	I/O	DESCRIPTION
			TIMING AND CONTROL
IAQ	7	OUT	Instruction acquisition. IAQ is active (high) during any memory cycle when the TMS 9900 is acquiring an instruction. IAQ can be used to detect illegal op codes.
$\overline{\text{LOAD}}$	4	IN	Load. When active (low), $\overline{\text{LOAD}}$ causes the TMS 9900 to execute a nonmaskable interrupt with memory address $FFFC_{16}$ containing the trap vector (WP and PC). The load sequence begins after the instruction being executed is completed. $\overline{\text{LOAD}}$ will also terminate an idle state. If $\overline{\text{LOAD}}$ is active during the time $\overline{\text{RESET}}$ is released, then the $\overline{\text{LOAD}}$ trap will occur after the $\overline{\text{RESET}}$ function is completed. $\overline{\text{LOAD}}$ should remain active for one instruction period. IAQ can be used to determine instruction boundaries. This signal can be used to implement cold-start ROM loaders. Additionally, front-panel routines can be implemented using CRU bits as front-panel-interface signals and software-control routines to control the panel operations.
$\overline{\text{RESET}}$	6	IN	Reset. When active (low), $\overline{\text{RESET}}$ causes the processor to be reset and inhibits $\overline{\text{WE}}$ and CRUCLK. When $\overline{\text{RESET}}$ is released, the TMS 9900 then initiates a level-zero interrupt sequence that acquires WP and PC from locations 0000 and 0002, sets all status register bits to zero, and starts execution. $\overline{\text{RESET}}$ will also terminate an idle state. $\overline{\text{RESET}}$ must be held active for a minimum of three clock cycles.

2.9 TIMING

2.9.1 MEMORY

A basic memory read and write cycle is shown in Figure 9. The read cycle is shown with no wait states and the write cycle is shown with one wait state.

$\overline{\text{MEMEN}}$ goes active (low) during each memory cycle. At the same time that $\overline{\text{MEMEN}}$ is active, the memory address appears on the address bus bits A0 through A14. If the cycle is a memory-read cycle, DBIN will go active (high) at the same time $\overline{\text{MEMEN}}$ and A0 through A14 become valid. The memory-write signal $\overline{\text{WE}}$ will remain inactive (high) during a read cycle. If the read cycle is also an instruction acquisition cycle, IAQ will go active (high) during the cycle.

The READY signal, which allows extended memory cycles, is shown high during $\phi 1$ of the second clock cycle of the read operation. This indicates to the TMS 9900 that memory-read data will be valid during $\phi 1$ of the next clock cycle. If READY is low during $\phi 1$, then the TMS 9900 enters a wait state suspending internal operation until a READY is sensed during a subsequent $\phi 1$. The memory read data is then sampled by the TMS 9900 during the next $\phi 1$, which completes the memory-read cycle.

At the end of the read cycle, $\overline{\text{MEMEN}}$ and DBIN go inactive (high and low, respectively). The address bus may also change at this time, however, the data bus remains in the input mode for one clock cycle after the read cycle.

A write cycle is similar to the read cycle with the exception that $\overline{\text{WE}}$ goes active (low) as shown and valid write data appears on the data bus at the same time the address appears. The write cycle is shown as an example of a one-wait-state memory cycle. READY is low during $\phi 1$ resulting in the WAIT signal shown.

8◀

2.9.2 HOLD

Other interfaces may utilize the TMS 9900 memory bus by using the hold operation (illustrated in Figure 10) of the TMS 9900. When $\overline{\text{HOLD}}$ is active (low), the TMS 9900 enters the hold state at the next available non-memory cycle. Considering that there can be a maximum of three consecutive memory cycles, the maximum delay between $\overline{\text{HOLD}}$ going active to HOLDA going active (high) could be $t_{c(\phi)}$ (for setup) + (6 + 3W) $t_{c(\phi)}$ + $t_{c(\phi)}$ (delay for HOLDA), where W is the number of wait states per memory cycle and $t_{c(\phi)}$ is the clock cycle time. When the TMS 9900 has entered the hold state, HOLDA goes active (high) and A0 through A15, D0 through D15 DBIN, $\overline{\text{MEMEN}}$, and $\overline{\text{WE}}$ go into a high-impedance state to allow other devices to use the memory buses. When $\overline{\text{HOLD}}$ goes inactive (high), the TMS 9900 resumes processing as shown. If hold occurs during a CRU operation, the TMS 9900 uses an extra clock cycle (after the removal of the $\overline{\text{HOLD}}$ signal) to reassert the CRU address providing the normal setup times for the CRU bit transfer that was interrupted.

2.9.3 CRU

CRU interface timing is shown in Figure 11. The timing for transferring two bits out and one bit in is shown. These transfers would occur during the execution of a CRU instruction. The other cycles of the instruction execution are not illustrated. To output a CRU bit, the CRU-bit address is placed on the address bus A0 through A14 and the actual bit data on CRUOUT. During the second clock cycle a CRU pulse is supplied by CRUCLK. This process is repeated until the number of bits specified by the instruction are completed.

The CRU input operation is similar in that the bit address appears on A0 through A14. During the subsequent cycle the TMS 9900 accepts the bit input data as shown. No CRUCLK pulses occur during a CRU input operation.

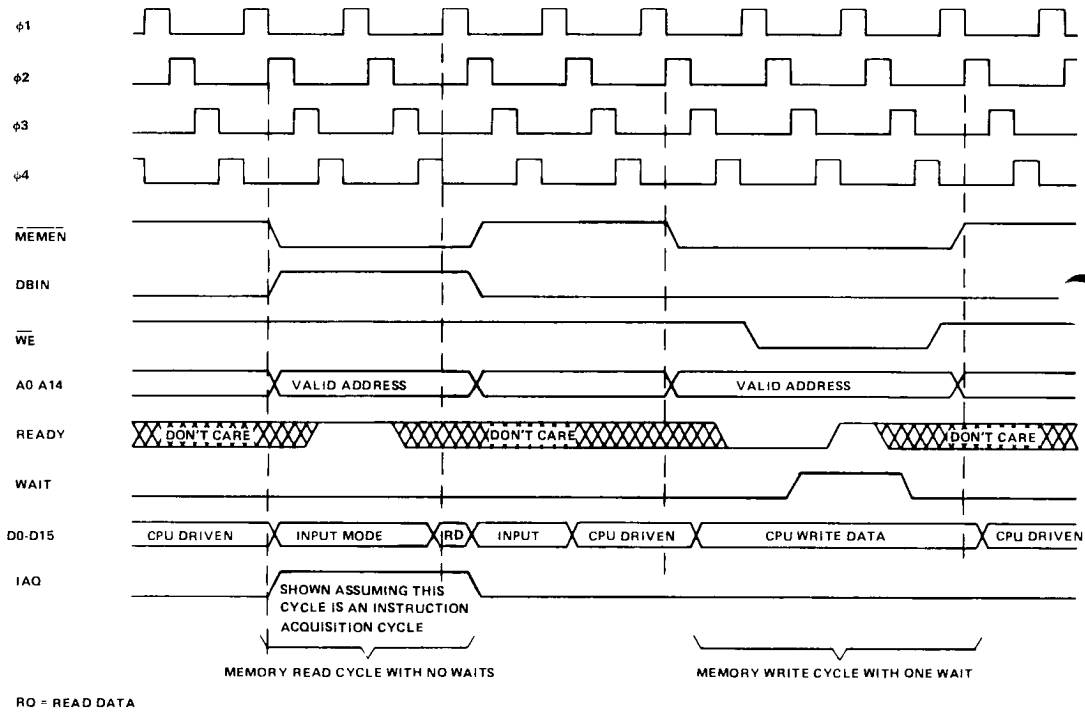


FIGURE 9 – TMS 9900 MEMORY BUS TIMING

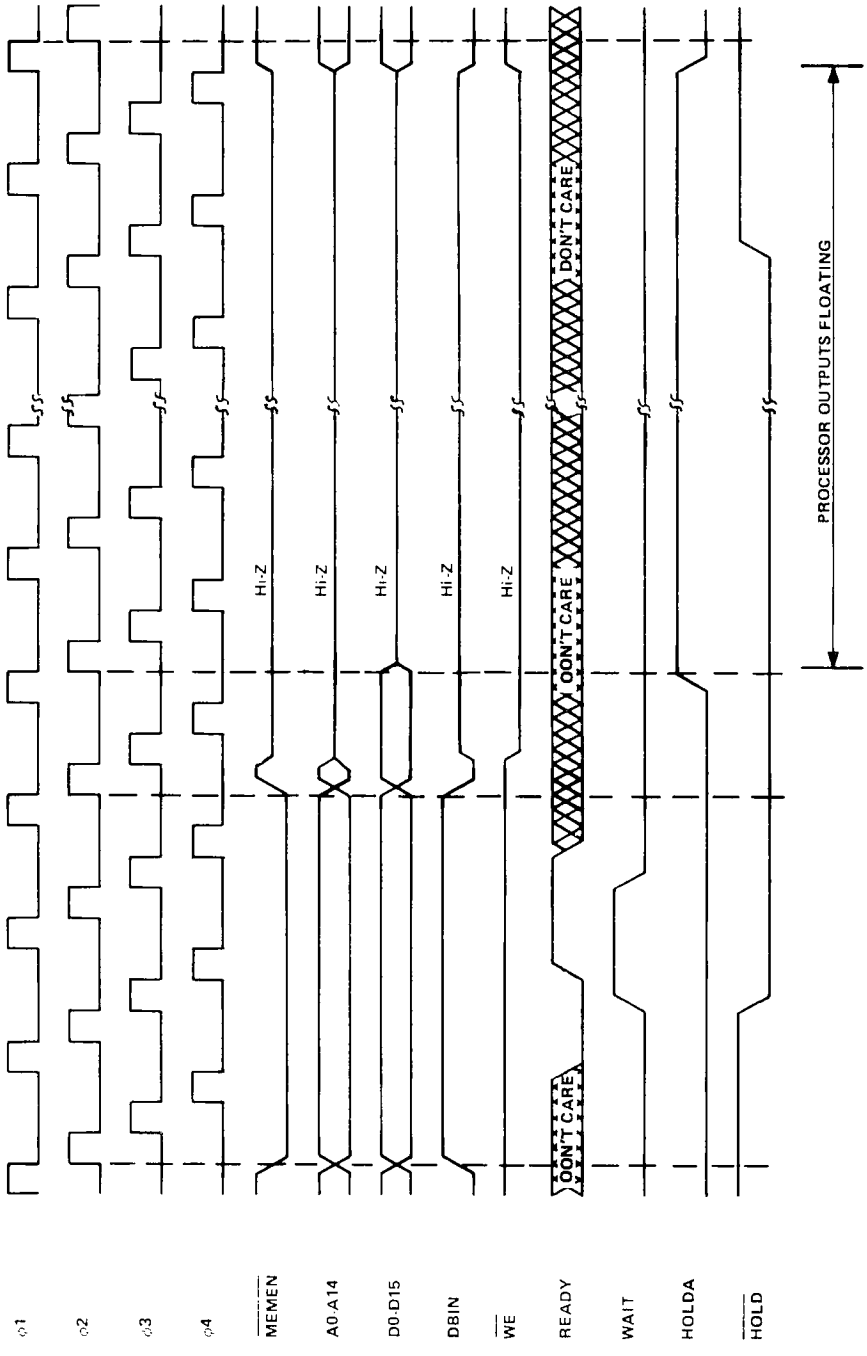


FIGURE 10 – TMS 9900 HOLD TIMING

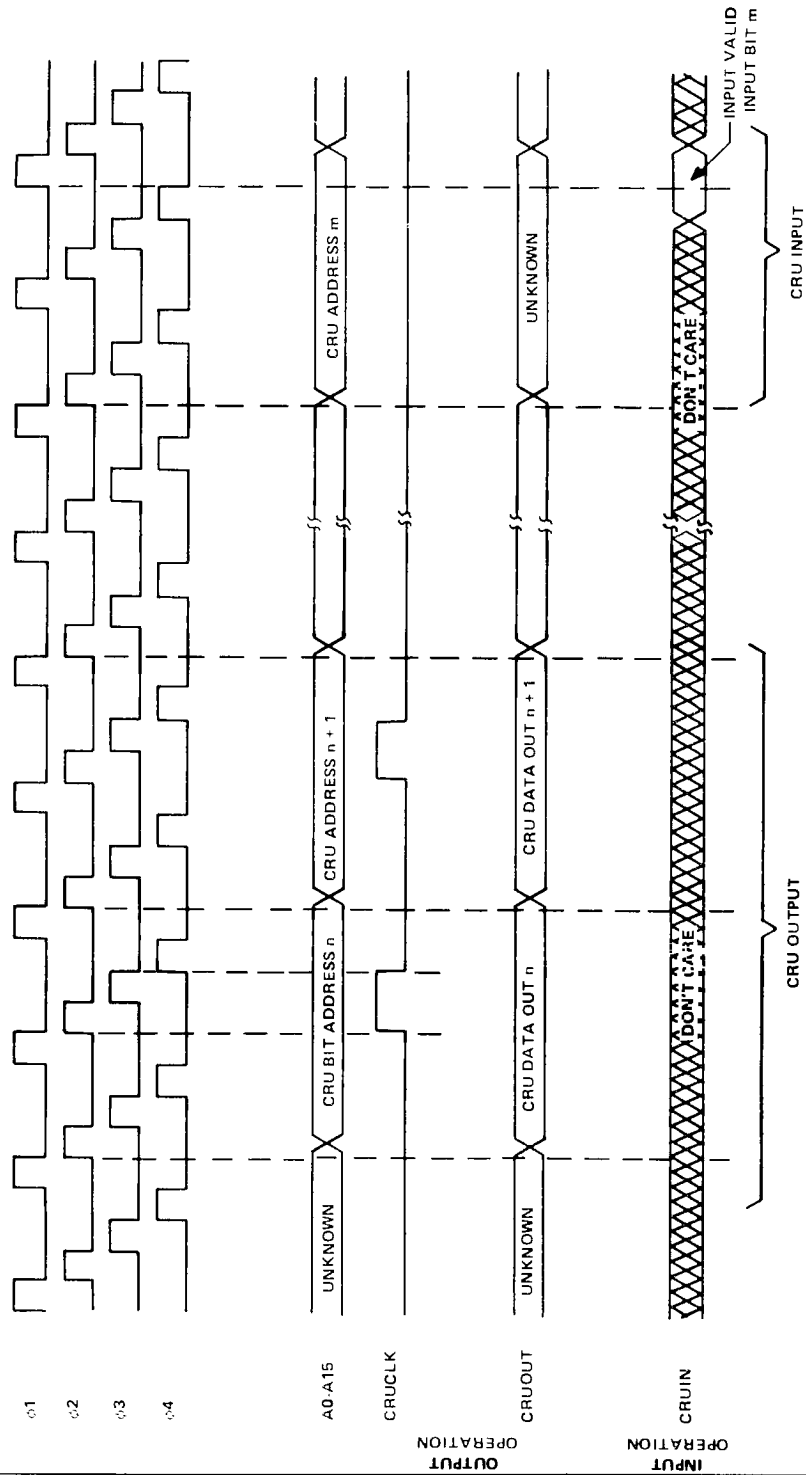


FIGURE 11 – TMS 9900 CRU INTERFACE TIMING

3.6 TMS 9900 INSTRUCTION EXECUTION TIMES

Instruction execution times for the TMS 9900 are a function of:

1) Clock cycle time, $t_{c(\phi)}$

2) Addressing mode used where operands have multiple addressing mode capability

3) Number of wait states required per memory access.

Table 3 lists the number of clock cycles and memory accesses required to execute each TMS 9900 instruction. For instructions with multiple addressing modes for either or both operands, the table lists the number of clock cycles and memory accesses with all operands addressed in the workspace-register mode. To determine the additional number of clock cycles and memory accesses required for modified addressing, add the appropriate values from the referenced tables. The total instruction-execution time for an instruction is:

$$T = t_{c(\phi)}(C+(W*M))$$

where:

T = total instruction execution time;

$t_{c(\phi)}$ = clock cycle time;

C = number of clock cycles for instruction execution plus address modification;

W = number of required wait states per memory access for instruction execution plus address modification;

M = number of memory accesses.

TABLE 3
INSTRUCTION EXECUTION TIMES

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS Modified Addressing	
A	14	4	B	B
AB	14	4	B	B
ABS (MSB = 0)	12	2	A	–
(MSB = 1)	14	3	A	
AI	14	4	–	
ANDI	14	4		
B	8	2	A	
BL	12	3	A	–
BLWP	26	6	A	
C	14	3	A	A
CB	14	3	B	B
CI	14	3	–	
CKOF	12	1	–	–
CKON	12	1	–	
CLR	10	3	A	–
COC	14	3	A	
CZC	14	3	A	
DEC	10	3	A	–
DECT	10	3	A	
DIV (ST4 is set)	16	3	A	
DIV (ST4 is reset)*	92-124	6	A	
IDLE	12	1	–	
INC	10	3	A	
INCT	10	3	A	
INV	10	3	A	
Jump (PC is changed)	10	1	–	–
(PC is not changed)	8	1	–	
LDCR (C = 0)	52	3	A	
(1 ≤ C ≤ 8)	20+2C	3	B	–
(9 ≤ C ≤ 15)	20+2C	3	A	
LI	12	3	–	–
LIMI	16	2	–	
LREX	12	1	–	
RESET function	26	5	–	–
LOAD function	22	5	–	–
Interrupt context switch	22	5	–	–

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS Modified Addressing	
LWPI	10	2		
MOV	14	4	A	A
MOVB	14	4	B	B
MPY	52	5	A	–
NEG	12	3	A	–
ORI	14	4	–	–
RSET	12	1	–	–
RTWP	14	4	–	–
S	14	4	A	A
SB	14	4	B	B
SBO	12	2	–	–
SBZ	12	2	–	–
SETO	10	3	A	–
Shift (C ≠ 0)	12+2C	3	–	–
(C = 0, Bits 12-15 of WR0 = 0)	52	4	–	–
(C = 0, Bits 12-15 of WR0 ≠ 0)	20+2N	4	–	–
SOC	14	4	A	A
SOCB	14	4	B	B
STCR (C = 0)	60	4	A	–
(1 ≤ C ≤ 7)	42	4	B	–
(C = 8)	44	4	B	–
(9 ≤ C ≤ 15)	58	4	A	–
STST	8	2	–	–
STWP	8	2	–	–
SWPB	10	3	A	–
SZC	14	4	A	A
SZCB	14	4	B	B
TB	12	2	–	–
X **	8	2	A	–
XOP	36	8	A	–
XOR	14	4	A	–
Undefined op codes 0000-01FF, 0320-033F, 0C00-0FFF, 0780-07FF	6	1	–	–

*Execution time is dependent upon the partial quotient after each clock cycle during execution.

**Execution time is added to the execution time of the instruction located at the source address minus 4 clock cycles and 1 memory access time.

†The letters A and B refer to the respective tables that follow.

ADDRESS MODIFICATION – TABLE A

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	4	1
WR indirect auto-increment (T_S or T_D = 11)	8	2
Symbolic (T_S or T_D = 10, S or D = 0)	8	1
Indexed (T_S or T_D = 10, S or D ≠ 0)	8	2

ADDRESS MODIFICATION – TABLE B

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	4	1
WR indirect auto-increment (T_S or T_D = 11)	6	2
Symbolic (T_S or T_D = 10, S or D = 0)	8	1
Indexed (T_S or T_D = 10, S or D ≠ 0)	8	2

As an example, the instruction MOVB is used in a system with $t_{c(\phi)} = 0.333\ \mu s$ and no wait states are required to access memory. Both operands are addressed in the workspace register mode:

$$T = t_c(\phi)(C+(W*M)) = 0.333\ (14+(0*4)) = 4.662\ \mu s$$

If two wait states per memory access were required, the execution time is:

$$T = 0.333\ (14+(2*4))\ \mu s = 7.326\ \mu s.$$

If the source operand was addressed in the symbolic mode and two wait states were required:

$$T = t_c(\phi)(C+(W*M))$$
$$C = 14 + 8 = 22$$
$$M = 4 + 1 = 5$$
$$T = 0.333\ (22+(2*5))\ \mu s = 10.656\ \mu s.$$

4. TMS 9900 ELECTRICAL SPECIFICATIONS

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

Supply voltage, V_{CC} (see Note 1) . . . . . −0.3 to 20 V
Supply voltage, V_{DD} (see Note 1) . . . . . −0.3 to 20 V
Supply voltage, V_{SS} (see Note 1) . . . . . −0.3 to 20 V
All input voltages (see Note 1) . . . . . −0.3 to 20 V
Output voltage (with respect to V_{SS}) . . . . . −2 V to 7 V
Continuous power dissipation . . . . . 1.2 W
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −55°C to 150°C

8

*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.

NOTE 1 Under absolute maximum ratings voltage values are with respect to the most negative supply, V_{BB} (substrate), unless otherwise noted. Throughout the remainder of this section, voltage values are with respect to V_{SS}.

4.2 RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{BB}	−5.25	−5	−4.75	V
Supply voltage, V_{CC}	4.75	5	5.25	V
Supply voltage, V_{DD}	11.4	12	12.6	V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH} (all inputs except clocks)	2.2	2.4	$V_{CC}+1$	V
High-level clock input voltage, $V_{IH(\phi)}$	$V_{DD}-2$		$V_{DD}+1$	V
Low-level input voltage, V_{IL} (all inputs except clocks)	−1	0.4	0.8	V
Low-level clock input voltage, $V_{IL(\phi)}$	−0.3	0.3	0.6	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
I_I	Input current	Data bus during DBIN	$V_I = V_{SS}$ to V_{CC}		±50	±100	μA
		$\overline{WE}$, $\overline{MEMEN}$, DBIN, Address bus, Data bus during HOLDA	$V_I = V_{SS}$ to V_{CC}		±50	±100	
		Clock	V_I = −0.3 to 12.6 V		±25	±75	
		Any other inputs	$V_I = V_{SS}$ to V_{CC}		±1	±10	
V_{OH}	High-level output voltage		I_O = −0.4 mA	2.4		V_{CC}	V
V_{OL}	Low-level output voltage		I_O = 3.2 mA			0.65	V
			I_O = 2 mA			0.50	
$I_{BB(av)}$	Supply current from V_{BB}				0.1	1	mA*
$I_{CC(av)}$	Supply current from V_{CC}				50	75	mA*
$I_{DD(av)}$	Supply current from V_{DD}				25	45	mA*
C_i	Input capacitance (any inputs except clock and data bus)		V_{BB} = −5, f = 1MHz, unmeasured pins at V_{SS}		10	15	pF
$C_{i(\phi1)}$	Clock-1 input capacitance		V_{BB} = −5, f = 1MHz, unmeasured pins at V_{SS}		100	150	pF
$C_{i(\phi2)}$	Clock-2 input capacitance		V_{BB} = −5, f = 1MHz, unmeasured pins at V_{SS}		150	200	pF
$C_{i(\phi3)}$	Clock-3 input capacitance		V_{BB} = −5, f = 1MHz, unmeasured pins at V_{SS}		100	150	pF
$C_{i(\phi4)}$	Clock-4 input capacitance		V_{BB} = −5, f = 1MHz, unmeasured pins at V_{SS}		100	150	pF
C_{DB}	Data bus capacitance		V_{BB} = −5, f = 1MHz, unmeasured pins at V_{SS}		15	25	pF

8

†All typical values are at T_A = 25°C and nominal voltages.
*D.C. Component of Operating Clock

4.4 TIMING REQUIREMENTS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

	PARAMETER	MIN	NOM	MAX	UNIT
$t_{c(\phi)}$	Clock cycle time	300	333	500	ns
$t_{r(\phi)}$	Clock rise time	5	12		ns
$t_{f(\phi)}$	Clock fall time	10	12		ns
$t_{w(\phi)}$	Clock pulse width, high level	40	45	100	ns
$t_{s(\phi)}$	Clock spacing, time between any two adjacent clock pulses	0	5		ns
$t_{d(\phi)}$	Time between rising edge valid any two adjacent clock pulses	73	83		ns
t_{su}	Data or control setup time before clock 1	30			ns
t_h	Data hold time after clock 1	10			ns

4.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{PLH\,(B)}$ or $t_{PHL\,(B)}$	All other outputs	C_L = 200 pF		20	40	ns
$t_{PLH\,(C)}$ or $t_{PHL\,(C)}$	Propagation delay CRUCLK, WE, MEMEN, WAIT, DBIN				30	ns

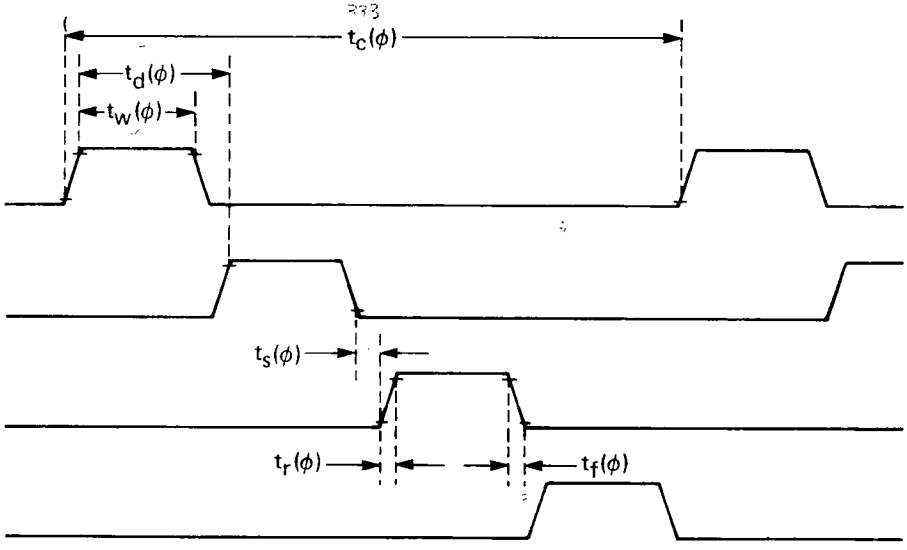


FIGURE 12 – CLOCK TIMING

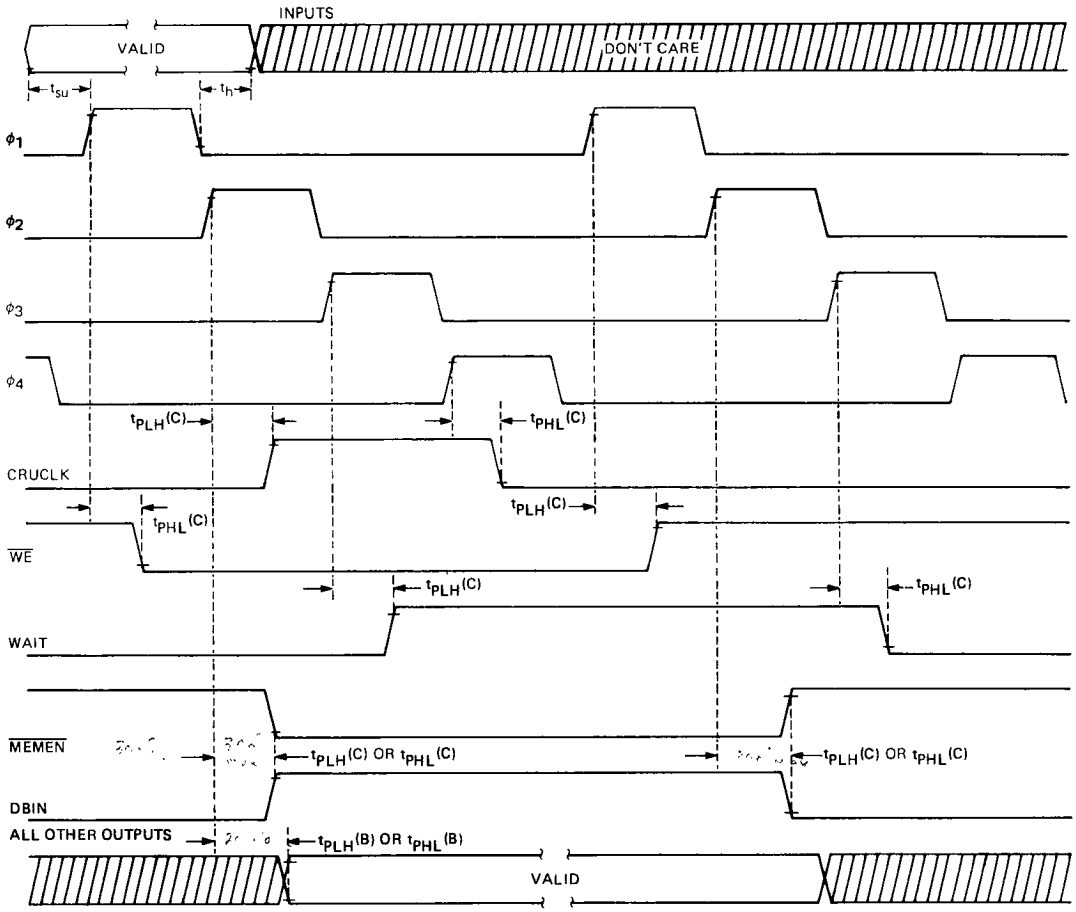


FIGURE 13–SIGNAL TIMING

8

TMS 9900-40 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)*

Supply voltage, V_{CC} (see Note 1) . . . . . −0.3 to 20 V
Supply voltage, V_{DD} (see Note 1) . . . . . −0.3 to 20 V
Supply voltage, V_{SS} (see Note 1) . . . . . −0.3 to 20 V
All input voltages (see Note 1) . . . . . −0.3 to 20 V
Output voltage (with respect to V_{SS}) . . . . . −2 V to 7 V
Continuous power dissipation . . . . . 1.2 W
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −55°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.

NOTE 1: Under absolute maximum ratings voltage values are with respect to the most negative supply, V_{BB} (substrate), unless otherwise noted. Throughout the remainder of this section, voltage values are with respect to V_{SS}

RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{BB}	−5.25	−5	−4.75	V
Supply voltage, V_{CC}	4.75	5	5.25	V
Supply Voltage, V_{DD}	11.4	12	12.6	V
Supply voltage, V_{SS}		0		V
High-level input volrage, V_{IH} (all inputs except clocks)	2.2	2.4	$V_{CC}+1$	V
High-level clock input voltage, $V_{IH(\phi)}$	$V_{DD}-2$		V_{DD}	V
Low-level input voltage, V_{IL} (all inputs except clocks)	−1	0.4	0.8	V
Low-level clock input voltage, $V_{IL(\phi)}$	−0.3	0.3	0.6	V
Operating free-air temperature, T_A	0		70	°C

▶8

DESIGN GOAL

This document describes the design specifications for a product under development Texas Instruments reserves the right to change these specifications in any manner, without notice.

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

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
I_I	Input current	Data bus during DBIN	$V_I = V_{SS}$ to V_{CC}		±50	±100	μA
		WE, MEMEN, DBIN, Address bus, Data bus during HOLDA	$V_I = V_{SS}$ to V_{CC}		±50	±100	μA
		Clock*	$V_I = -0.3$ to 12.6 V		±25	±75	μA
		Any other inputs	$V_I = V_{SS}$ to V_{CC}		±1	±10	μA
V_{OH}	High-level output voltage		$I_O = -0.4$ mA	2.4		V_{CC}	mA
V_{OL}	Low-level output voltage		$I_O = 3.2$ mA			0.65	mA
			$I_O = 2$ mA			0.50	mA
I_{BB}(av)	Supply current from V_{BB}				0.1	1	pF
I_{CC}(av)	Supply current from V_{CC}				50	75	
I_{DD}(av)	Supply current from V_{DD}				25	45	pF
C_I	Input Capacitance (any inputs except clock and data bus)		f = 1MHz, V_{BB} = -5 V, unmeasured pins at V_{SS}		10	15	pF
$C_{i(\phi_1)}$	Clock-1 input capacitance		f = 1 MHz, V_{BB} = -5 V, unmeasured pins at V_{SS}		100	150	pF
$C_{i(\phi_2)}$	Clock-2 input capacitance		f = 1 MHz, V_{BB} = -5 V, unmeasured pins at V_{SS}		150	200	pF
$C_{i(\phi_3)}$	Clock-3 input capacitance		f = 1 MHz, V_{BB} = -5 V, unmeasured pins at V_{SS}		100	150	pF
$C_{i(\phi_4)}$	Clock-4 input capacitance		f = 1 MHz, V_{BB} = -5 V, unmeasured pins at V_{SS}		100	150	
C_{DB}	Data bus capacitance		f = 1 MHz, V_{BB} = -5 V, unmeasured pins at V_{SS}		15	25	pF

†All typical values are at $T_A = 25°C$ and nominal voltages.
*D.C. component of operating clock.

TIMING REQUIREMENTS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

PARAMETER		MIN	NOM	MAX	UNIT
$t_c(\phi)$	Clock cycle time	2.40	2.50		ns
$t_r(\phi)$	Clock rise time	5	12		ns
$t_f(\phi)$	Clock fall time	10	12		ns
$t_w(\phi)$	Pulse width, high level	33			ns
$t_s(\phi)$	Clock spacing, time between any two adjacent clock pulses	0	45		ns
$t_D(o)$	Time between rising edges, valid between any two adjacent clock pulses	55	63		ns
t_{su}	Data or control setup time before clock $\phi 1$	25			ns
t_h	Data hold time after clock $\phi 1$	10			ns

8◀

SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}(c) or t_{PHL}(c) propagation delay CRUCLK, $\overline{WE}$, $\overline{MEMEN}$, WAIT, DBIN	$C_L = 200$pF		20	20	ns
t_{PLH}(B) or t_{PHL}(B) all other outputs			30	30	ns

DESIGN GOAL
See page 8-28 for Design Goal

SBP 9900A

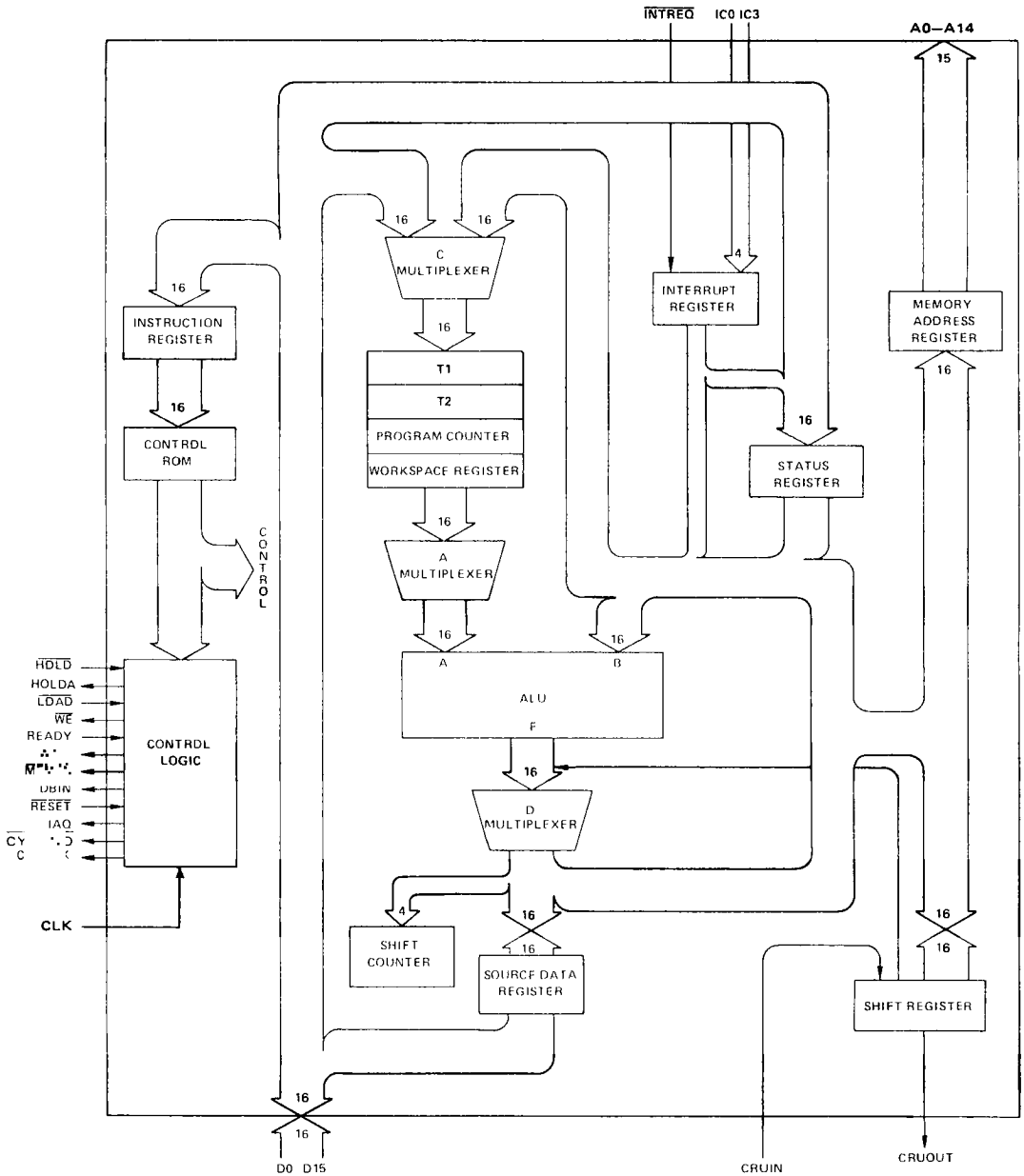


Figure 1. SBP 9900A Architecture.

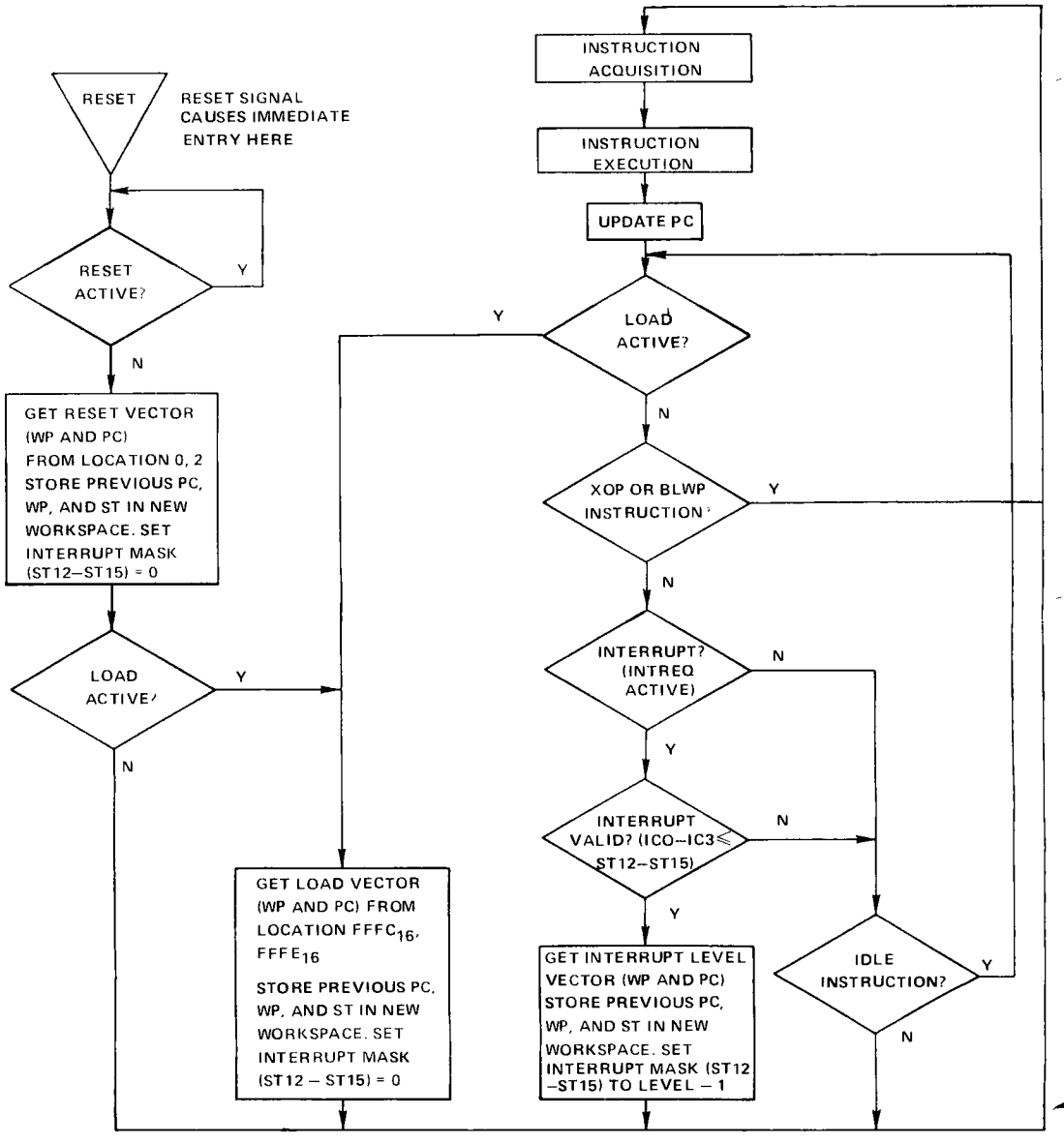
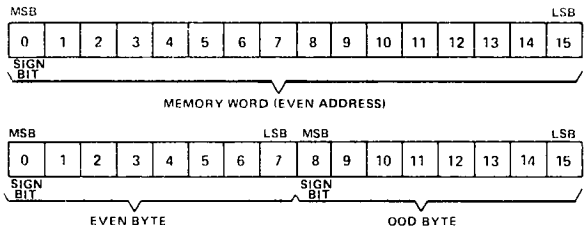


Figure 2. 9900 CPU Flow Chart

ARCHITECTURE

The Memory word of the 9900 is 16 bits long. Each word is also defined as 2 bytes of 8 bits. The instruction set of the 9900 allows both word and byte operands. Thus, all memory locations are on even address boundaries and byte instructions can address either the even or odd byte. The memory space is 65,536 bytes or 32,768 words. The word and byte formats are shown below.

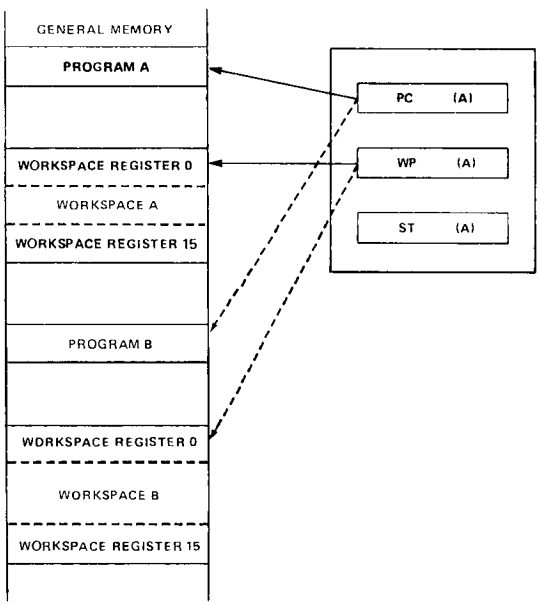


Registers and Memory

The 9900 family employs an advanced memory-to-memory architecture. Blocks of memory designated as workspace replace internal-hardware registers with program-data registers.

Three internal registers are accessible to the user. The program counter (PC) contains the address of the instruction following the current instruction being executed. This address is referenced by the processor to fetch the next instruction from memory and is then automatically incremented. The status register (ST) contains the present state of the processor. The workspace pointer (WP) contains the address of the first word in the currently active set of workspace registers.

A workspace-register file occupies 16 contiguous memory words in the general memory area (see *Figure 3).* Each workspace register may hold data or addresses and function as operand registers, accumulators, address registers, or index registers. During instruction execution, the processor addresses any given register in the workspace by adding the register number to the contents of the workspace pointer and initiating a memory request for the word. The relationship between the workspace pointer and its corresponding workspace is shown below.



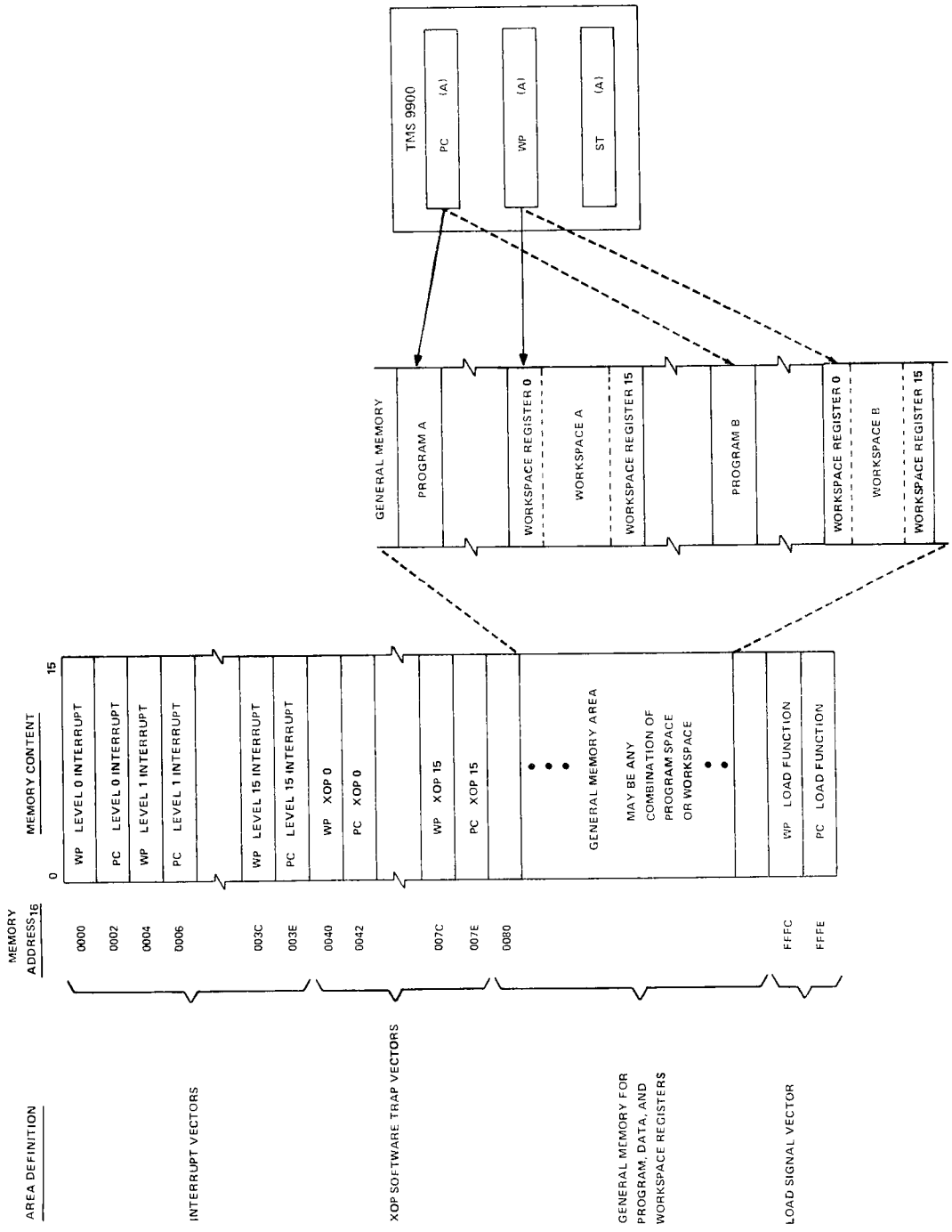


Figure 3. Memory Map

INTERRUPTS

The architecture of the 9900 family allows vectoring of 16 interrupts. These interrupts are assigned levels from 0 to 15. The interrupt at level 0 has the highest priority and the interrupt at level 15 has the lowest priority. The 9900 implements all 16 interrupt levels. Level 0 is reserved for $\overline{\text{RESET}}$ function.

The 9900 continuously compares the interrupt code with the interrupt mask contained in the status-register. When the level of the pending interrupt is less than or equal to the enabling mask level (higher or equal priority interrupt), the processor recognizes the interrupt and initiates a context switch following completion of the currently executing instruction. The processor fetches the new context WP and PC from the interrupt vector locations. Then, the previous context WP, PC, and ST are stored in workspace registers 13, 14, and 15, respectively, of the new workspace. The 9900 then forces the interrupt mask to a value that is one less than the level of the interrupt being serviced, except for level-zero interrupt, which loads zero into the mask. This allows only interrupts of higher priority to interrupt a service routine. The processor also inhibits interrupts until the first instruction of the service routine has been executed to preserve program linkage should a higher priority interrupt occur. All interrupt requests should remain active until recognized by the processor in the device-service routine. The individual service routines must reset the interrupt requests before the routine is complete.

If a higher priority interrupt occurs, a second context switch occurs to service the higher priority interrupt. When that routine is complete, a return instruction (RTWP) restores the first service routine parameters to the processor to complete processing of the lower-priority interrupt. All interrupt subroutines should terminate with the return instruction to restore original program parameters. The interrupt-vector locations, device assignment, enabling-mask value, and the interrupt code are shown in *Table 1.*

Table 1. Interrupt Level Data

Interrupt Level	Vector Location (Memory Address In Hex)	Device Assignment	Interrupt Mask Values To Enable Respective Interrupts (ST12 thru ST15)	Interrupt Codes IC0 thru IC3
(Highest priority) 0	00	Reset	0 through F*	0000
1	04	External device	1 through F	0001
2	08	↓	2 through F	0010
3	0C	↓	3 through F	0011
4	10	↓	4 through F	0100
5	14	↓	5 through F	0101
6	18	↓	6 through F	0110
7	1C	↓	7 through F	0111
8	20	↓	8 through F	1000
9	24	↓	9 through F	1001
10	28	↓	A through F	1010
11	2C	↓	B through F	1011
12	30	↓	C through F	1100
13	34	↓	D through F	1101
14	38	↓	E and F	1110
(Lowest priority) 15	3C	External device	F only	1111

*Level 0 can not be disabled.

The 9900 interrupt interface utilizes the TMS 9901 Programmable Systems Interface as shown in *Figure 4.*

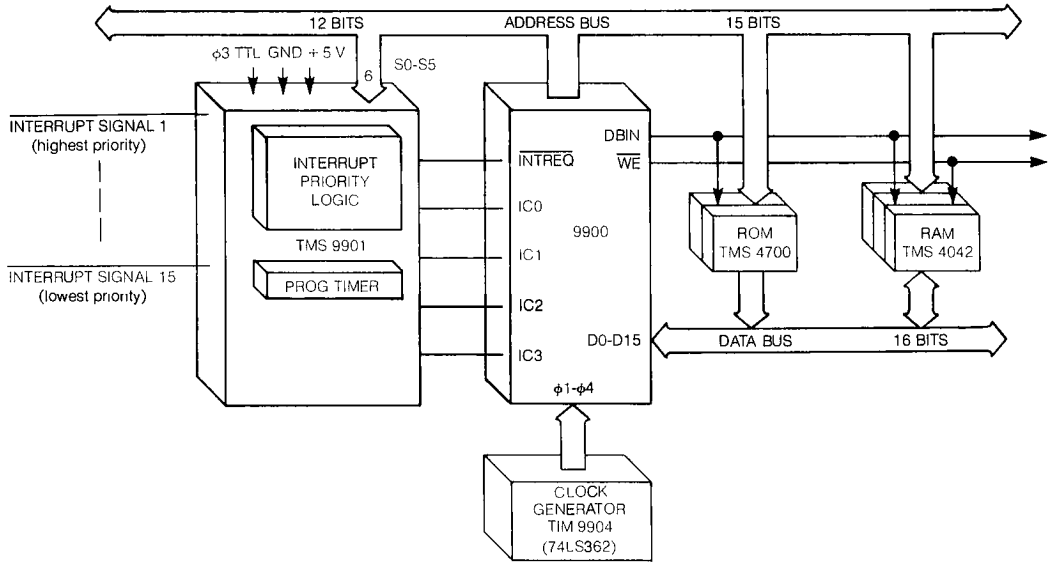


Figure 4. 9900 Interrupt Interface

INPUT/OUTPUT

The 9900 Utilizes a versatile direct command-driven I/O interface designated as the communications-register unit (CRU). The CRU provides up to 4096 directly addressable input bits and 4096 directly addressable output bits. Both input and output bits can be addressed individually or in fields of from 1 to 16 bits. The 9900 employs three dedicated I/O pins (CRUIN, CRUOUT, and CRUCLK) and certain bits of the address bus to interface with the CRU system. The processor instructions that drive the CRU interface can set, reset, or test any bit in the CRU array or move between memory and CRU data fields.

▶8

Single-Bit CRU Operations

The 9900 performs three single-bit CRU functions: test bit (TB), set bit to one (SBO), and set bit to zero (SBZ). To identify the bit to be operated upon, the 9900 develops a CRU-bit address and places it on the address bus, A3 to A14.

For the two output operations (SBO and SBZ), the processor also generates a CRUCLK pulse, indicating an output operation to the CRU device, and places bit 7 of the instruction word on the CRUOUT line to accomplish the specified operation (bit 7 is a one for SBO and a zero for SBZ). A test-bit instruction transfers the addressed CRU bit from the CRUIN input line to bit 2 of the status register (EQUAL).

The 9900 develops a CRU-bit address for the single-bit operations from the software base address contained in workspace register 12 and the signed displacement count contained in bits 8 through 15 of the instruction. The displacement allows two's complement addressing from base minus 128 bits through base plus 127 bits. The hardware base address, bits 3 through 14 of WR12, is added to the signed displacement specified in the instruction and the result is loaded onto the address bus. *Figure 5* illustrates the development of a single-bit CRU address.

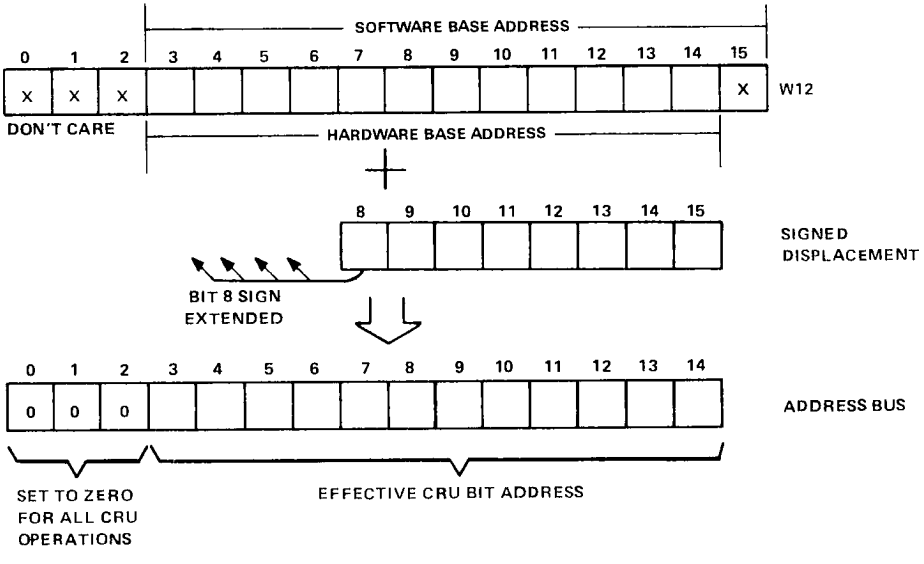


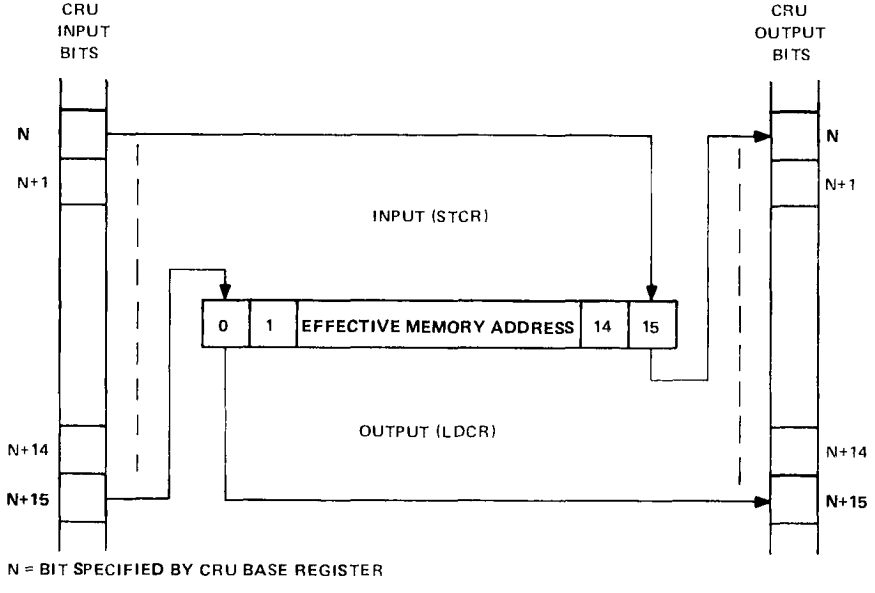
Figure 5. Single-Bit CRU Address Development

8◀

Multiple-Bit CRU Operations

The 9900 performs two multiple-bit CRU operations: store communications register (STCR) and load communications register (LDCR). Both operations perform a data transfer from the CRU-to-memory or from memory-to-CRU as illustrated in *Figure 6*. Although the figure illustrates a full 16-bit transfer operation, any number of bits from 1 to 16 may be involved. The LDCR instruction fetches a word from memory and right-shifts it to serially transfer it to CRU output bits. If the LDCR involves eight or fewer bits, those bits come from the right-justified field within the addressed byte of the memory word. If the LDCR involves nine or more bits, those bits come from the right-justified field within the whole memory. When transferred to the CRU interface, each successive bit receives an address that is sequentially greater than the address for the previous bit. This addressing mechanism results in an order reversal of the bits, that is, bit 15 of the memory word (or bit 7) becomes the lowest addressed bit in the CRU and bit 0 becomes the highest addressed bit in the CRU field.

An STRC instruction transfers data from the CRU to memory. If the operation involves a byte or less transfer, the transferred data will be stored right-justified in the memory byte with leading bits set to zero. If the operation involves from nine to 16 bits, the transferred data is stored right-justified in the memory word with leading bits set to zero.



▶8

Figure 6. 9900 LDCR/STRC Data Transfers

When the input from the CRU device is complete, the first bit from the CRU is the least-significant-bit position in the memory word or byte.

Figure 7 illustrates how to implement a 16-bit input and a 16-bit output register in the CRU interface using the TMS 9901. CRU addresses are decoded as needed to implement up to 256 such 16-bit interface registers. In system application, however, only the exact number of interface bits needed to interface specific peripheral devices are implemented. It is not necessary to have a 16-bit interface register to interface an 8-bit device.

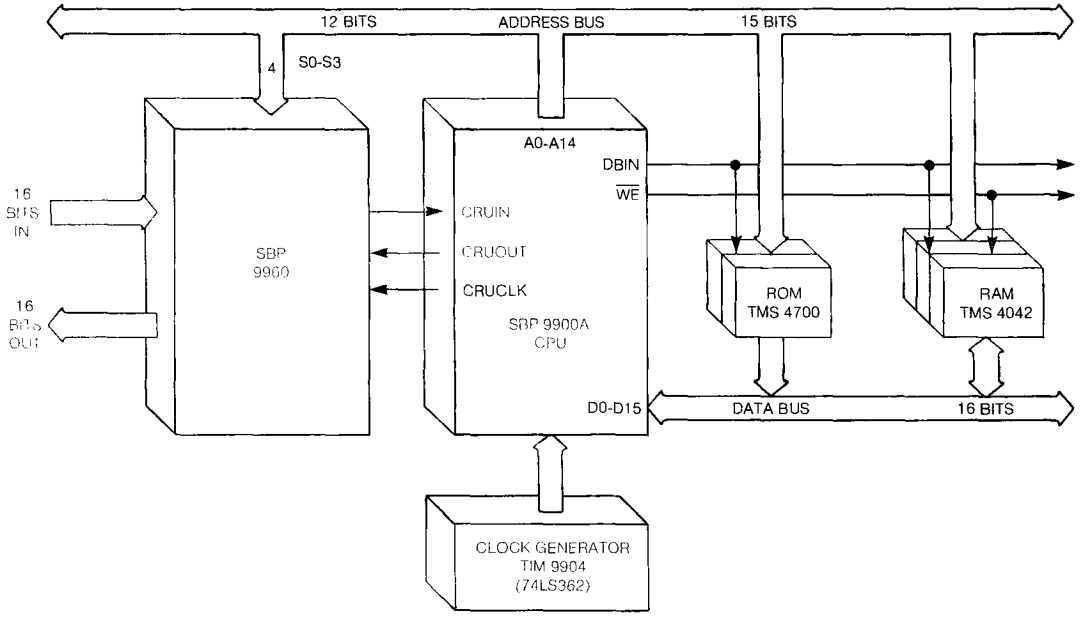


Figure 7. SBP 9900A 16-bit Input/Output Interface

External Instructions

The 9900 has five external instructions that allow user-defined external functions to be initiated under program control. These instructions are CKON, CKOF, RSET, IDLE, and LREX. These mnemonics, except for IDLE, relate to functions implemented in the 9900 minicomputer and do not restrict use of the instructions to initiate various user-defined functions. IDLE also causes the 9900 to enter the idle state and remain until an interrupt, $\overline{\text{RESET}}$, or $\overline{\text{LOAD}}$ occurs. When any of these five instructions are executed by the 9900, a unique 3-bit code appears on the address bus along with a CRUCLK pulse. The user must provide external hardware to decode this 3 bit code and implement this external function. When the 9900 is in an idle state, the 3-bit code and CRUCLK pulses occur repeatedly until the idle state is terminated. The codes are:

8◀

EXTERNAL INSTRUCTION	A0	A1	A2
LREX	H	H	H
CKOF	H	H	L
CKON	H	L	H
RSET	L	H	H
IDLE	L	H	L

Figure 8 illustrates typical external decode logic to implement these instructions. Note CRUCLK to the CRU is inhibited during external instructions.

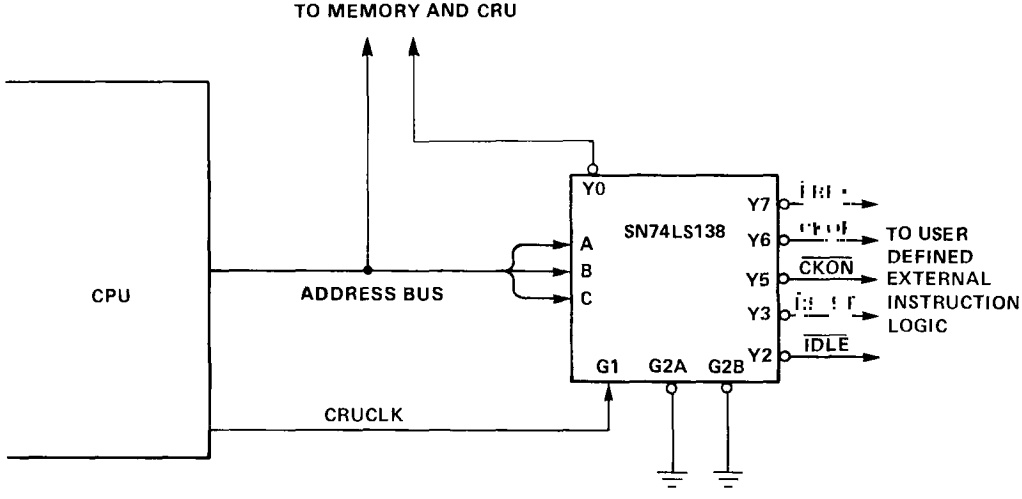


Figure 8. External Instruction Decode Logic

LOAD FUNCTION

The $\overline{\text{LOAD}}$ signal allows cold-start ROM loaders and front panels to be implemented for the 9900. When active, $\overline{\text{LOAD}}$ causes the 9900 to initiate an interrupt sequence immediately following the instruction being executed. A memory location is used to obtain the vector (WP and PC). The old PC, WP and ST are loaded into the new workspace and the interrupt mask is set to 0000. Then, program execution resumes using the new PC and WP.

▶8

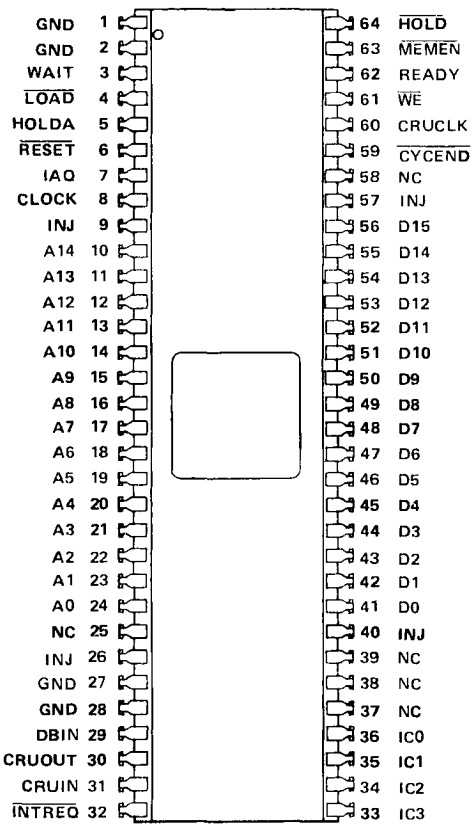
SBP 9900A PIN DESCRIPTION

Table 2 describes the function of each SBP 9900A pin, and *Figure 9* illustrates their assigned locations.

Table 2. 9900 Pin Assignments and Functions

SIGNATURE	PIN	I/O	DESCRIPTION
			ADDRESS BUS
A0 (MSB)	24	OUT	A0 (MSB) through A14 (LSB) comprise the address bus. This open-collector bus provides the memory-address vector to the external-memory system when $\overline{\text{MEMEN}}$ is active, and I/O-bit addresses to the I/O system when $\overline{\text{MEMEN}}$ is inactive. When HOLDA is active, the address bus is pulled to the logic level HIGH state by the individual pull-up resistors tied to each respective open-collector output.
A14 (LSB)	10	OUT	
			DATA BUS
D0 (MSB)	41	I/O	D0 (MSB) through D15 (LSB) comprise the bidirectional open-collector data bus. This bus transfers memory data to (when writing) and from (when r[illegible] the external-memory system when [illegible] $\overline{\text{N}}$ is active. When HOLDA is active, the data bus is pulled to the logic level HIGH state by the individual pull-up resistors tied to each respective open-collector output.
D15 (LSB)	56	I/O	
			POWER SUPPLY
INJ	9		Injector-Supply-Current
INJ	26		Injector-Supply-Current
INJ	40		Injector-Supply-Current
INJ	57		Injector-Supply-Current
GND	1		Ground Reference
GND	2		Ground Reference
GND	27		Ground Reference
GND	28		Ground Reference
			CLOCK
CLOCK	8	IN	CLOCK
			BUS CONTROL
DBIN	29	OUT	DATA BUS IN. When active (pulled to logic level HIGH), DBIN indicates that the SBP 9900A has disabled its output buffers to allow the memory to place [illegible]-read data on the data bus during [illegible]. DBIN remains at logic level LOW in all other cases except when HOLDA is active (pulled to logic level HIGH).
$\overline{\text{MEMEN}}$	63	OUT	MEMORY ENABLE. When active (logic level LOW), $\overline{\text{MEMEN}}$ indicates that the address bus contains a memory address.
$\overline{\text{WE}}$	61	OUT	WRITE ENABLE. When active (logic level LOW), $\overline{\text{WE}}$ indicates that the SBP 9900A data bus is outputting data to be written into memory.

Figure 9. SBP 9900A Pin Assignments.



NC—No internal connection

8◀

Table 2. (Continued)

SIGNATURE	PIN	I/O	DESCRIPTION
CRUCLK	60	OUT	COMMUNICATIONS-REGISTER-UNIT (CRU) CLOCK. When active (pulled to logic level HIGH), CRUCLK indicates to the external interface logic the presence of output data on CRUOUT, or the presence of an encoded external instruction on A0 through A2.
CRUIN	31	IN	CRU DATA IN. CRUIN, normally driven by 3-state or open-collector devices, receives input data from the external interface logic. When the SBP 9900A executes a STCR or TB instruction, it samples CRUIN for the level of the CRU input bit specified by the address bus (A3 through A14).
CRUOUT	30	OUT	CRU DATA OUT. CRUOUT outputs serial data when the SBP 9900A executes a LDCR, SBZ, SBO instruction. The data on CRUOUT should be sampled by the external interface logic when CRUCLK goes active (pulled to logic level HIGH).
			INTERRUPT CONTROL
$\overline{\text{INTREQ}}$	32	IN	INTERRUPT REQUEST. When active (logic level LOW), $\overline{\text{INTREQ}}$ indicates that an external interrupt is requesting service. If $\overline{\text{INTREQ}}$ is active, the SBP 9900A loads the data on the interrupt-code input-lines IC0 through IC3 into the internal interrupt-code storage register. The code is then compared to the interrupt mask bits of the status register. If equal or higher priority than the enabled interrupt level (interrupt code equal or less than status register bits 12 through 15), the SBP 9900A initiates the interrupt sequence. If the comparison fails, the SBP 9900A ignores the interrupt request. In that case, $\overline{\text{INTREQ}}$ should be held active. The SBP 9900A will continue to sample IC0 through IC3 until the program enables a sufficiently low interrupt-level to accept the requesting interrupt.
IC0 (MSB)	36	IN	INTERRUPT CODES IC0 (MSB) through IC3 (LSB), receiving an interrupt identify code, are sampled by the SBP 9900A when $\overline{\text{INTREQ}}$ is active (logic level LOW). When IC0 through IC3 are LLLH, the higher priority *external* interrupt is requesting service, when HHHH, the lowest priority external interrupt is requesting service.
IC0 (LSB)	33	IN	
			MEMORY CONTROL
$\overline{\text{HOLD}}$	64	IN	When active (logic level LOW), $\overline{\text{HOLD}}$ indicates to the SBP 9900A that an external controller (e.g., DMA device) desires to use both the address bus and data bus to transfer data to or from memory. In response, the SBP 9900A enters the hold state after completion of its present memory cycle. The SBP 9900A then allows its address bus, data bus, $\overline{\text{WE}}$, $\overline{\text{MEMEN}}$, DBIN, and HOLDA facilities to be pulled to the logic level HIGH state. When $\overline{\text{HOLD}}$ is deactivated, the SBB 9900A returns to normal operation from the point at which it was stopped.
HOLDA	5	OUT	HOLD ACKNOWLEDGE. When active (pulled to logic level HIGH), HOLDA indicates that the SBP 9900A is in the hold state and that its address bus, data bus, $\overline{\text{WE}}$, $\overline{\text{MEMEN}}$, and DBIN facilities are pulled to the logic level HIGH state.
READY	62	IN	When active (logic level HIGH), READY indicates that the memory will be ready to read or write during the next clock cycle. When not-ready is indicated during a memory operation, the SBP 9900A enters a wait state and suspends internal operation until the memory systems activate READY.
WAIT	3	OUT	When active (pulled to logic level HIGH), WAIT indicates that the SBP 9900A has entered a wait state in response to a not-ready condition from memory.
			TIMING AND CONTROL
IAQ	7	IN	INSTRUCTION ACQUISITION. IAQ is active (pulled to logic level HIGH) during any SBP 9900A initiated instruction acquisition memory cycle. Consequently, IAQ may be used to facilitate detection of illegal op codes.
$\overline{\text{CYCEND}}$	59	OUT	CYCLE END. When active (logic level LOW), $\overline{\text{CYCEND}}$ indicates that the SBP 9900A will initiate a new microinstruction cycle on the low-to-high transition of the next CLOCK.
$\overline{\text{LOAD}}$	4	IN	When active (logic level LOW), $\overline{\text{LOAD}}$ causes the SBP 9900A to execute a nonmaskable interrupt with memory addresses $FFFC_{16}$ and $FFFE_{16}$ containing the associated trap vectors (WP and PC). The load sequence is initiated after the instruction being executed is completed. $\overline{\text{LOAD}}$ will also terminate an idle state. If $\overline{\text{LOAD}}$ is active during the time $\overline{\text{RESET}}$ is active, the $\overline{\text{LOAD}}$ trap will occur after the $\overline{\text{RESET}}$ function is completed. $\overline{\text{LOAD}}$ should remain active for one instruction execution period (IAQ may be used to monitor instruction boundaries). $\overline{\text{LOAD}}$ may be

Table 2. (Continued)

SIGNATURE	PIN	I/O	DESCRIPTION
$\overline{\text{LOAD}}$ (Cont.)			used to implement cold-start RDM loaders. Additionally, front-panel routines may be implemented using CRU bits as front-panel-interface signals, and software-control routines to direct the panel operations.
$\overline{\text{RESET}}$	6	IN	When active (logic level LOW), $\overline{\text{RESET}}$ causes the SBP 9900A to reset itself and inhibit $\overline{\text{WE}}$ and CRUCLK. When $\overline{\text{RESET}}$ is released, the SBP 9900A initiates a level-zero interrupt sequence acquiring the WP and PC trap vectors from memory locations 0000_{16} and 0002_{16}, sets all status register bits to logic level LOW, and then fetches the first instruction of the reset program environment. $\overline{\text{RESET}}$ must be held active for a minimum of three CLOCK cycles

TIMING

SBP 9900A Memory

The SBP 9900A basic memory timing for a memory-read cycle with no wait states, and a memory-write cycle with one wait state, is as shown in *Figure 10.* During each memory-read or memory-write cycle, $\overline{\text{MEMEN}}$ becomes active (logic level LOW) along with valid memory-address data appearing on the address bus (A0 through A14).

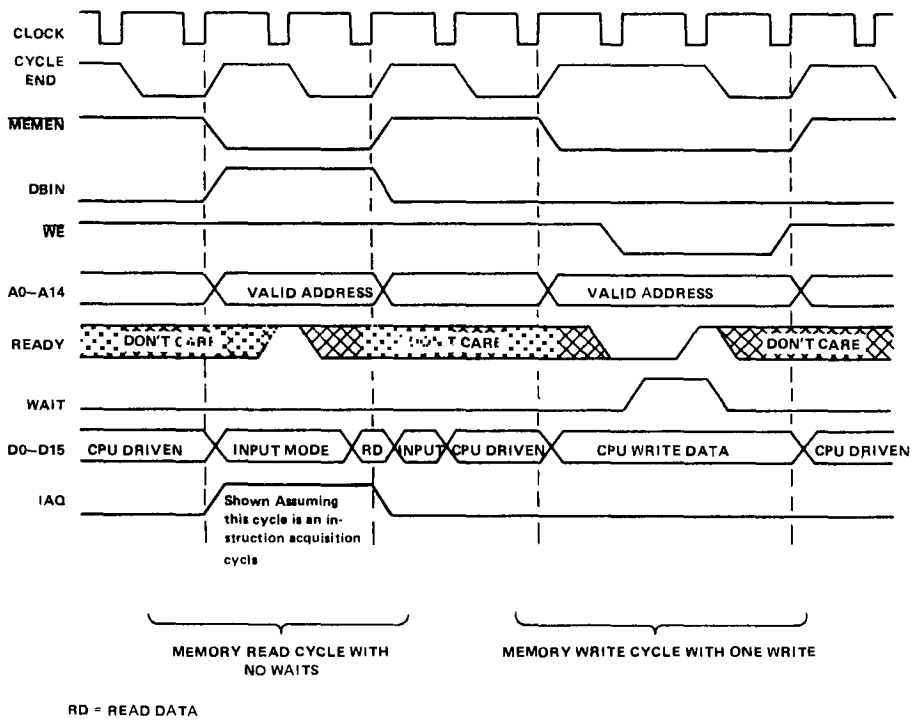


Figure 10. SBP 9900A Memory Bus Timing

In the case of a memory-read cycle, DBIN becomes active (pulled to logic level HIGH) at the same time memory-address data becomes valid; the memory write strobe $\overline{\text{WE}}$ remains inactive (pulled to logic level HIGH). If the memory-read cycle is initiated for acquisition of an instruction, IAQ becomes active (pulled to logic level HIGH) at the same time $\overline{\text{MEMEN}}$ becomes active. At the end of a memory-read cycle, $\overline{\text{MEMEN}}$ and DBIN together become inactive. At this time, though the address may change, the data bus remains in the input mode until terminated by the next high-to-low transition of the clock.

In the case of a memory-write cycle, $\overline{WE}$ becomes active (logic level LOW) with the first high-to-low transition of the clock after $\overline{MEMEN}$ becomes active; DBIN remains inactive. At the end of a memory-write cycle, $\overline{WE}$ and $\overline{MEMEN}$ together become inactive.

During either a memory-read or a memory-write operation, READY may be used to extend the duration of the associated memory cycle such that the speed of the memory system may be coordinated with the speed of the SBP 9900A. If READY is inactive (logic level LOW) during the first low-to-high transition of the clock after $\overline{MEMEN}$ becomes active, the SBP 9900A will enter a wait state suspending further progress of the memory cycle. The first low-to-high transition of the clock after READY becomes active terminates the wait state and allows normal completion of the memory cycle.

SBP 9900A Hold

The SBP 9900A hold facilities allow both the '9900A and external devices to share a common memory. To gain memory-bus control, an external device requiring direct memory access (DMA) sends a hold request ($\overline{HOLD}$) to the SBP 9900A. When the next available non-memory cycle occurs, the SBP 9900A enters a hold state and signals its surrender of the memory-bus to the external device via a hold acknowledge (HOLDA). Receiving the hold acknowledgement, the external device proceeds to utilize the common memory. After its memory requirements have been satisfied, the external device returns memory-bus control to the SBP 9900A by releasing $\overline{HOLD}$.

When $\overline{HOLD}$ becomes active (logic level LOW), the SBP 9900A enters a hold state at the beginning of the next available non-memory cycle as shown below. Upon entering a hold state, HOLDA becomes active (pulled to logic level HIGH) with the following signals pulled to a HIGH logic level by the individual pull-up resistors tied to each respective open-collector output: DBIN, $\overline{MEMEN}$, $\overline{WE}$, A0 through A14, and D0 through D15. When $\overline{HOLD}$ becomes inactive, the SPB 9900A exits the hold state and regains memory-bus control. If $\overline{HOLD}$ becomes active during a CRU operation, the SBP 9900A uses an extra clock cycle after the deactivation of $\overline{HOLD}$ to reassert the CRU address thereby providing the normal setup time for the CRU-bit transfer.

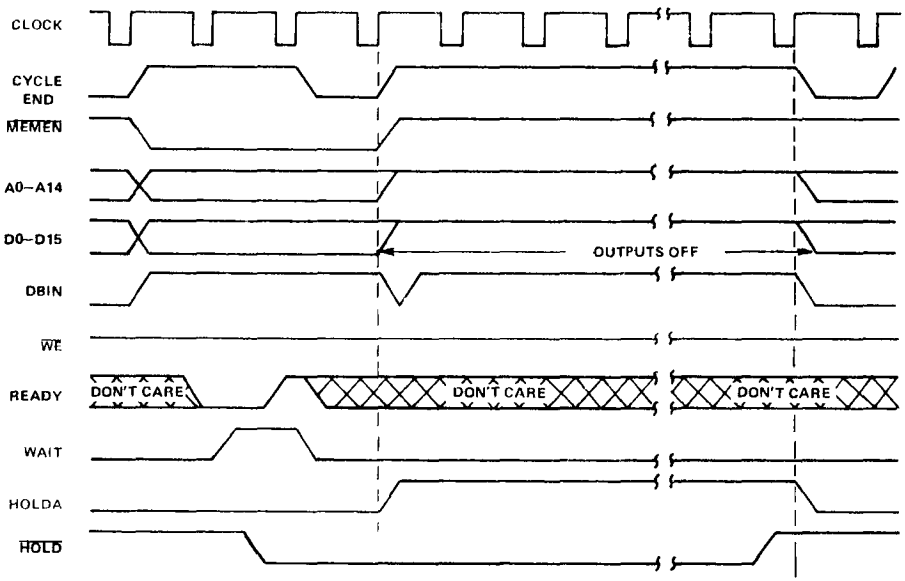


Figure 11. SBP 9900A Hold Timing

SBP 9900A CRU

The transfer of two data-bits from memory to a peripheral CRU device, and the transfer of one data-bit from a peripheral CRU device to memory, is shown in *Figure 12.* To transfer a data-bit to a peripheral CRU device, the SBP 9900A outputs the corresponding CRU-bit-address on address bus bits A3 through A14 and the respective data-bit on CRUOUT. During the second clock cycle of the operation, the SBP 9900A outputs a pulse, on CRUCLK, indicating to the peripheral CRU device the presence of a data-bit. This process is repeated until transfer of the entire field of data-bits specified by the CRU instruction has been accomplished. To transfer a data-bit from a peripheral CRU device, the SBP 9900A outputs the corresponding CRU-bit-Address on address bus bits A3 through A14 and receives the respective data-bit on CRUIN. No CRUCLK pulses occur during a CRU input operation.

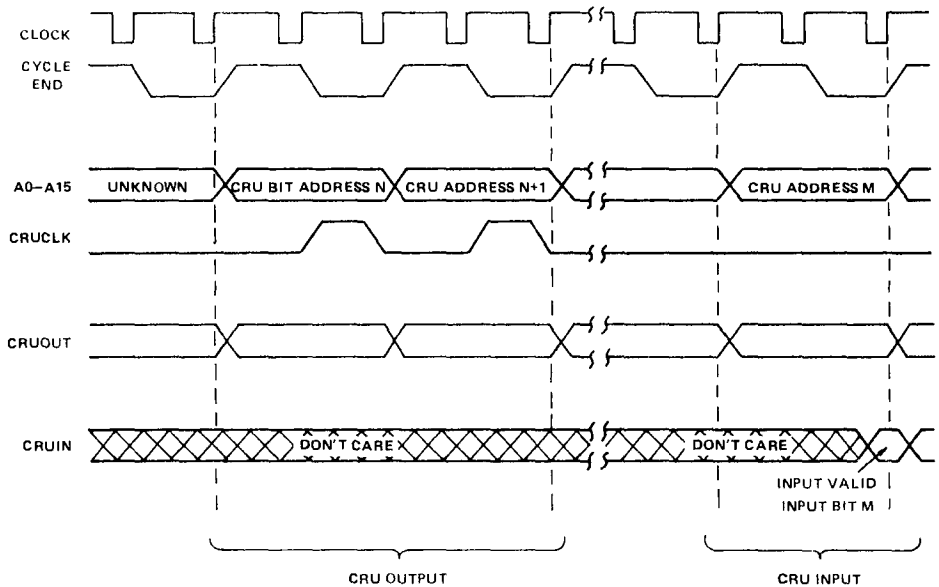


Figure 12. SBP 9900A CRU Interface Timing

MICROINSTRUCTION CYCLE

The SBP 9900 includes circuitry which will indicate the completion of a microinstruction cycle. Designated as the $\overline{\text{CYCEND}}$ function, it provides CPU status that can simplify system design. The $\overline{\text{CYCEND}}$ output will go to a low logic level as a result of the low-to-high transition of each clock pulse which initiates the last clock of a microinstruction.

INSTRUCTION EXECUTION TIMES

SBP 9900A INSTRUCTION EXECUTION TIMES

Instruction execution times for the SBP 9900A are a function of:

1) Clock cycle time, $t_c(\phi)$
2) Addressing mode used where operands have multiple addressing mode capability
3) Number of wait states required per memory access.

Table 3 lists the number of clock cycles and memory accesses required to execute each SBP 9900A instruction. For instructions with multiple addressing modes for either or both operands, the table lists the number of clock cycles and memory accesses with all operands addressed in the workspace-register mode. To determine the additional number of clock cycles and memory accesses required for modified addressing, add the appropriate values from the referenced tables. The total instruction-execution time for an instruction is:

$$T = t_c(\phi)\,(C + W \cdot M)$$

where:

T = total instruction execution time;
$t_c(\phi)$ = clock cycle time;
C = number of clock cycles for instruction execution plus address modification;
W = number of required wait states per memory access for instruction execution plus address modification;
M = number of memory accesses.

Table 3. Instruction Execution Times

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION	
A	14	4	B	B
AB	14	4	B	B
ABS (MSB = 0)	12	2	A	–
(MSB = 1)	14	3	A	–
AI	14	4	–	–
ANDI	14	4	–	–
B	8	2	A	–
BL	12	3	A	–
BLWP	26	6	A	–
C	14	3	A	A
CB	14	3	B	B
CI	14	3	–	–
CKOF	12	1	–	–
CKON	12	1	–	–
CLR	10	3	A	–
COC	14	3	A	–
CZC	14	3	A	–
DEC	10	3	A	–
DECT	10	3	A	–
DIV (ST4 is set)	16	3	A	–
DIV (ST4 is reset)*	92-124	6	A	–
IDLE	12	1	–	–
INC	10	3	A	–
INCT	10	3	A	–
INV	10	3	A	–
Jump (PC is changed)	10	1	–	–
(PC is not changed)	8	1	–	–
LDCR (C = 0)	52	3	A	–
(1≤C≤8)	20+2C	3	B	–
(9≤C≤15)	20+2C	3	A	–
LI	12	3	–	–
LIMI	14	2	–	–
LREX	12	1	–	–
function	26	5	–	–
unction	22	5	–	–
interrupt context switch	22	5	–	–

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION	
LWPI	10	2		
MOV	14	4	A	A
MOVB	14	4	B	B
MPY	52	5	A	–
NEG	12	3	A	–
ORI	14	4	–	–
RSET	12	1	–	–
RTWP	14	4	–	–
S	14	4	A	A
SB	14	4	B	B
SBO	12	2	–	–
SBZ	12	2	–	–
SETO	10	3	A	–
Shift (C≠0)	12+2C	3	–	–
(C=0, Bits 12–15 of WRO=0)	52	4	–	–
(C=0, Bits 12–15 of WRP=N≠0)	20+2N	4	–	–
SOC	14	4	A	A
SOCB	14	4	B	B
STCR (C=0)	60	4	A	–
(1≤C≤7)	42	4	B	–
(C=8)	44	4	B	–
(9≤C≤15)	58	4	A	–
STST	8	2	–	–
STWP	8	2	–	–
SWPB	10	3	A	–
SZC	14	4	A	A
SZCB	14	4	B	B
TB	12	2	–	–
X **	8	2	A	–
XOP	36	8	A	–
XOR	14	4	A	–
Undefined op codes 0000-01FF, 0320-033F, 0C00-0FFF, 0780-07FF	6	1	–	–

*Execution time is dependent upon the partial quotient after each clock cycle during execution.
**Execution time is added to the execution time of the instruction located at the source address minus 4 clock cycles and 1 memory access time.
†The letters A and B refer to the respective tables that follow.

Table A. Address Modification

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	4	1
WR indirect auto-increment (T_S or T_D = 11)	8	2
Symbolic (T_S or T_D = 10, S or D = 0)	8	1
Indexed (T_S or T_D = 10, S or D ≠ 0)	8	2

Table B. Address Modification

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	4	1
WR indirect auto-increment (T_S or T_D = 11)	6	2
Symbolic (T_S or T_D = 10, S or D = 0)	8	1
Indexed (T_S or T_D = 10, S or D ≠ 0)	8	2

As an example, the instruction MOVB is used in a system with $t_c(\phi) = 0.333$ μs and no wait states are required to access memory. Both operands are addressed in the workspace register mode:

$T = t_c(\phi)\ (C + W \cdot M) = 0.333\ (14 + 0 \cdot 4)\ \mu s = 4.662\ \mu s.$

If two wait states per memory access were required, the execution time is:

$T = 0.333\ (14 + 2 \cdot 4)\ \mu s = 7.326 = \mu s.$

If the source operand was addressed in the symbolic mode and two wait states were required:

$T = t_c(\phi)\ (C + W \cdot M)$
$C = 14 + 8 = 22$
$M = 4 + 1 = 5$
$T = 0.333\ (22 + 2 \cdot 5)\ \mu s = 10.656\ \mu s.$

INTERFACING

The input/output (I/O) accommodations have been designed for TTL compatibility. Direct interfacing, supportable by the entire families of catalog devices, is shown in *Figure 13.*

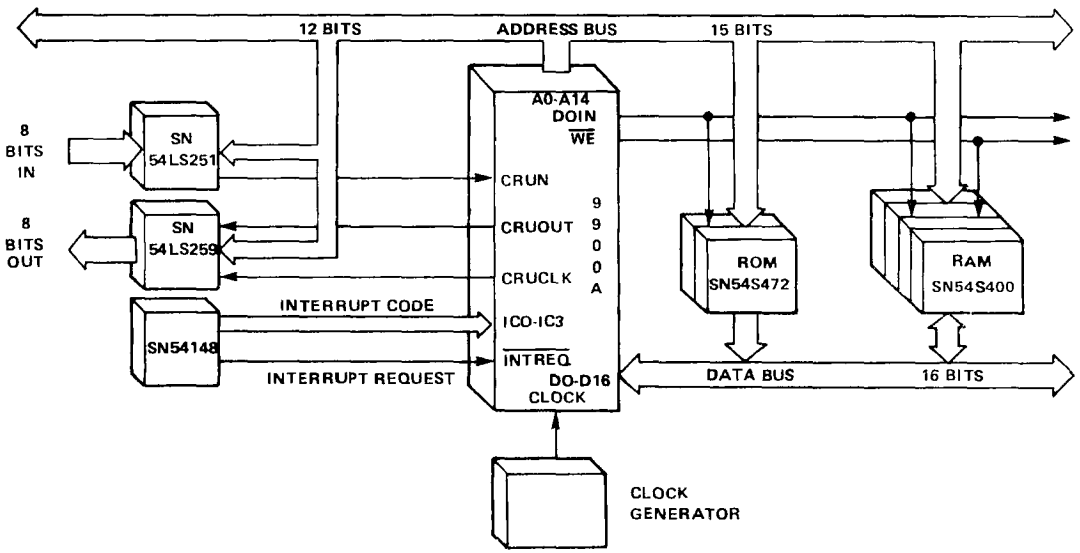


Figure 13. Typical SBP 9900A System

Input Circuit

The input circuit used on the SBP 9900A is basically an RTL configuration which has been modified for TTL compatibility as shown in *Figure 14A*. An input-clamping diode is incorporated to limit negative excursions (ringing) when the SBP 9900A is on the receiving end of a transmission line; an input switching threshold of nominally + 1.5 volts has been specified for improved noise immunity. This threshold is achieved via two resistors which function as a voltage divider to increase the one V_{BE} threshold of the I^2L input transistor to + 1.5 volts. Since this input circuit is independent of injector current, input threshold compatibility is maintained over the entire speed/power performance range.

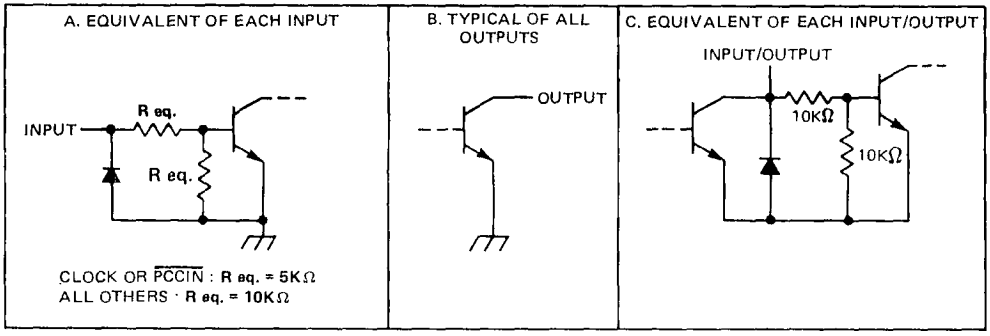


Figure 14. Schematics of Equivalent Inputs, Outputs, Inputs/Outputs

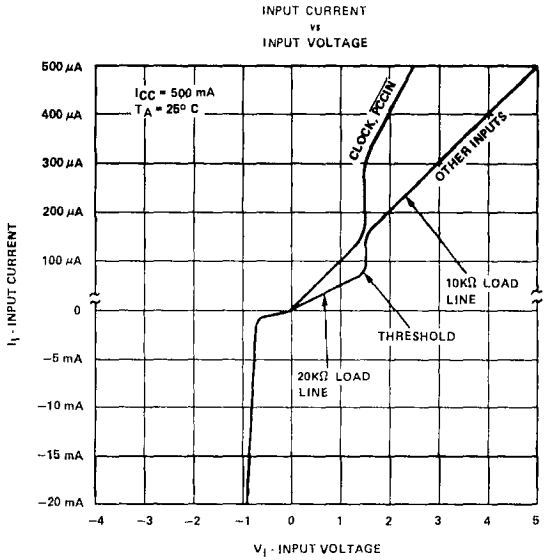


Figure 15. Typical Input Characteristics

The input circuit characteristics for input current versus input voltage are shown in *Figure 15.* The 10K and 20K ohm load lines and threshold knee at +1.5 volts provide a high-impedance characteristic to reduce input loading and improve the low-logic level input noise immunity over some standard TTL inputs. Full compatibility is maintained with virtually all 5 volt logic families even when the SBP 9900A is powered down (injector current reduced).

Sourcing Inputs

The inputs may be sourced directly by most 5 volt logic families. Five volt functions which feature internal pull-up resistors at their outputs require no external interface components; five volt functions which feature open-collector outputs generally require external pull-up resistors.

Terminating Unused Inputs

Inputs which are selected to be hardwired to a logic-level low may be connected directly to ground. Inputs which are selected to be hardwired to a logic-level high must be tied, via a current limiting (pull-up) resistor, to a logic-level-high low-impedance voltage source such as V_{CC}. A single transient protecting resistor may be utilized common to (N) inputs.

Output Circuit

The output circuit selected for the SBP 9900A is an injected open-collector transistor shown in *Figure 14B.* Since this transistor is injected, output sourcing capability is directly related to injector current. In other words, the number of loads which may be sourced by an SBP 9900A output is directly reduced as injector current is reduced.

The output circuit characteristic for logic-level low output voltage (V_{OL}) versus logic-level low output current (I_{OL}) is shown in *Figure 16.* At rated injector current, the SBP 9900A output circuit offers a low-level output voltage of typically 220 mV.

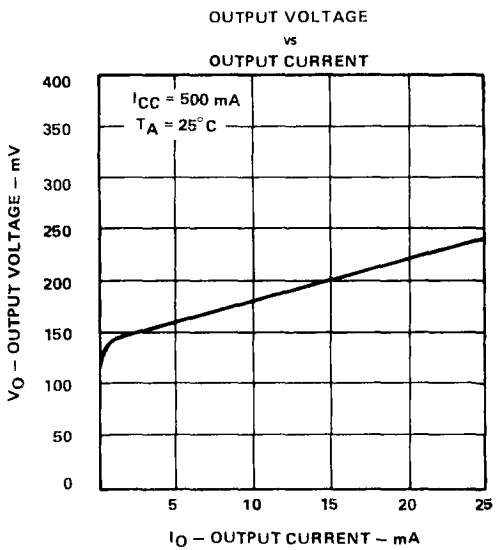


Figure 16. Typical Output Characteristics

The output circuit characteristics for 1) logic-level high output voltage (V_{OH}) and current (I_{OH}), 2) rise times, and 3) next stage input noise immunity, are a function of the load circuit being sourced. The load circuit may be either:

A) the direct input, if no source current is required, of a five-volt logic family function,

or, for greater noise immunity and improved rise times,

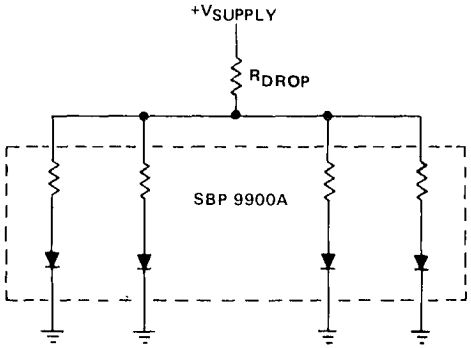
B) the direct input of a five-volt logic family function in conjunction with a discrete pull-up resistor.

When a discrete pull-up resistor (R_L) is utilized, the fanout requirements placed on a particular SBP 9900A output restrict both the maximum and minimum value of R_L.

POWER SOURCE

I^2L is a current-injected logic. When placed across a curve tracer, the processor will resemble a silicon switching diode. Any voltage or current source capable of supplying the desired current at the injector mode voltage required will suffice. A dry-cell battery, a 5-volt TTL power supply, a programmable current supply (for power-up/power-down operation) — literally whatever power source is convenient can be used for most cases. For example, if a 5-volt TTL power supply is to be used, a series dropping resistor would be connected between the 5-volt supply and the injector pins of the I^2L device, as illustrated in *Figure 17*, to select the desired operating current.

An alternate solution utilizes the Texas Instruments TL497 switching-regulator as illustrated in *Figure 18*.



GENERAL FORMULA (OHM'S LAW)

$$R_{DROP} = \frac{V_{SUPPLY} - V_{CC}}{I_{CC}}$$

EXAMPLE FOR V_{SUPPLY} = 5V, AND I_{CC} = 500 mA:

$$R_{DROP} = \frac{5 - 1.05}{0.5} = \frac{3.95}{0.5} = 7.9 \text{ OHMS}$$

Figure 17. Injector Current Calculations

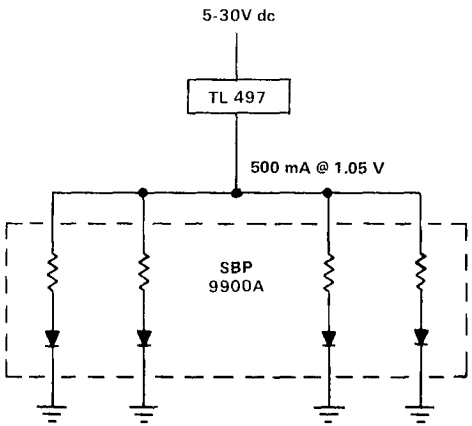


Figure 18. Switching-Regulator Injector Source

Operating from a constant current power source, the SBP 9900A may be powered-up/powered-down with complete maintenance of data integrity to execute instructions over a speed/power range spanning several orders of user-selectable injector-supply-current range as illustrated in *Figure 19*.

Figures 20 and *21* show the typical injector node voltages which occur across the temperature and injector current ranges.

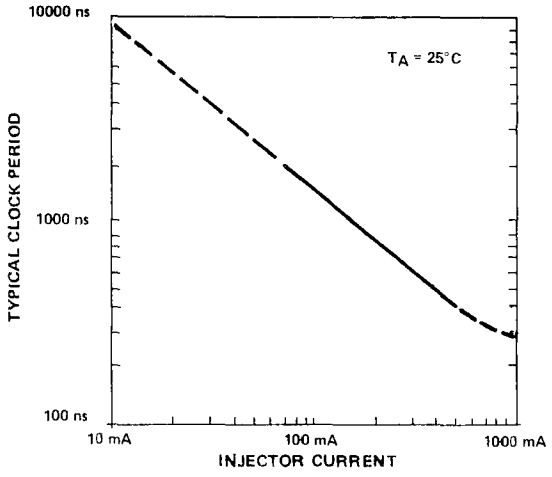


Figure 19. SBP 9900A Clock Period vs. Injector Current

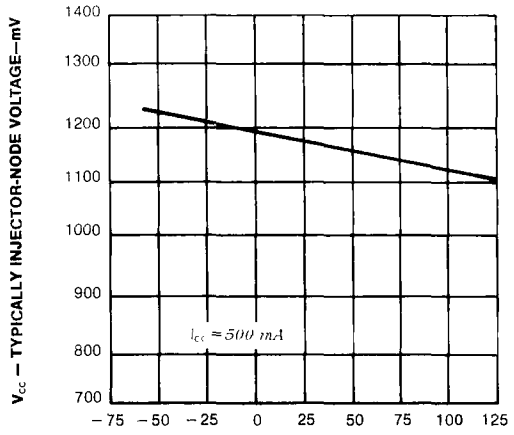


***Figure 20.** SBP 9900A Injector Node Voltage vs. Free-Air Temperature*

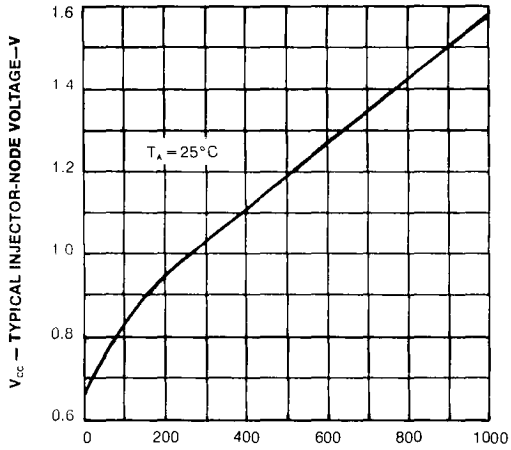


Figure 21. SBP 9900A Injector Node Voltage vs. Injector Current

ELECTRICAL SPECIFICATIONS

SBP 9900A Recommended Operating Conditions, Unless Otherwise Noted $I_{CC} = 500$ mA

		MIN	NOM	MAX	UNIT
Supply current, I_{CC}		450	500	550	mA
High-level output voltage, V_{OH}				5.5	V
Low-level output current, I_{OL}				20	mA
Clock frequency, f_{clock}		0		2	MHz
Width of clock pulse, t_w	High (67%) (V_{IH} = 2.5 V max)	330			ns
	Low (33%)	170			
Clock rise time, t_r			10		ns
Clock fall time, t_f			10		ns
Setup time, t_{su} (see Figure 24)	$\overline{HOLD}$	210↑			ns
	READY	140↑			
	D0 – D15	85↑			
		65↑			
		25↑			
	- IC3	25↑			
Hold time, t_h (see Figure 24)	D	25↑			ns
	READY	65↑			
	D0 – D15	65↑			
		55↑			
		90↑			
	IC0 – IC3	90↑			
Operating free-air temperature, T_A	SBP 9900 M/N	−55		125	°C
	SBP 9900E	−40		85	

↑Rising edge of clock pulse is reference.

SBP 9900A ELECTRICAL CHARACTERISTICS (OVER RECOMMENDED OPERATING FREE-AIR TEMPERATURE RANGE, UNLESS OTHERWISE NOTED)

PARAMETER			TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage				2			V
V_{IL}	Low-level input voltage						0.8	V
V_{IK}	Input clamp voltage		I_{CC} = MIN,	I_I = −12 mA			−1.5	V
I_{OH}	High-level output current		I_{CC} = 500 mA, V_{IL} = 0.8 V,	V_{IH} = 2 V, V_{OH} = 5.5 V			400	μA
V_{OL}	Low-level output voltage		I_{CC} = 500 mA, V_{IL} = 0.8 V,	V_{IH} = 2 V, I_{OL} = 20 mA			0.4	V
I_I	Input current	Clock	I_{CC} = 500 mA,	V_I = 2.4 V		480	600	μA
		All other inputs				240	300	

†For conditions shown as MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at I_{CC} = 500 mA, T_A = 25°C.

SBP 9900A SWITCHING CHARACTERISTICS, (I_{CC} = NOM, T_A = RECOMMENDED OPERATING FREE-AIR TEMPERATURE RANGE UNLESS OTHERWISE NOTED) SEE *Figures 22* AND *23*.

PARAMETER	FROM	TO	TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
f_{max}	MAXIMUM CLOCK FREQUENCY		C_L = 150 pF, R_L = 280 Ω	2			MHz
t_{PLH} or t_{PHL}	CLOCK	ADDRESS BUS (A0 – A14)			170	[illegible]	ns
t_{PLH} or t_{PHL}	CLOCK	DATA BUS (D0 – D15)			170	265	ns
t_{PLH} or t_{PHL}	CLOCK	WRITE ENABLE ($\overline{WE}$)			220	295	ns
t_{PLH} or t_{PHL}	CLOCK	CYCLE END ($\overline{CYCEND}$)			170	[illegible]	ns
t_{PLH} or t_{PHL}	CLOCK	DATA BUS IN (DBIN)			190	[illegible]	ns
t_{PLH} or t_{PHL}	CLOCK	MEMORY ENABLE ($\overline{MEMEN}$)			155	205	ns
t_{PLH} or t_{PHL}	CLOCK	CRU CLOCK (CRUCLK)			187	280	ns
t_{PLH} or t_{PHL}	CLOCK	CRU DATA OUT (CRUOUT)			210	265	ns
t_{PLH} or t_{PHL}	CLOCK	HOLD ACKNOWLEDGE (HLDA)			320	410	ns
t_{PLH} or t_{PHL}	CLOCK	WAIT			155	210	ns
t_{PLH} or t_{PHL}	CLOCK	INSTRUCTION ACQUISITION (IAQ)			155	210	ns

‡All typical values are at 25°C.

8

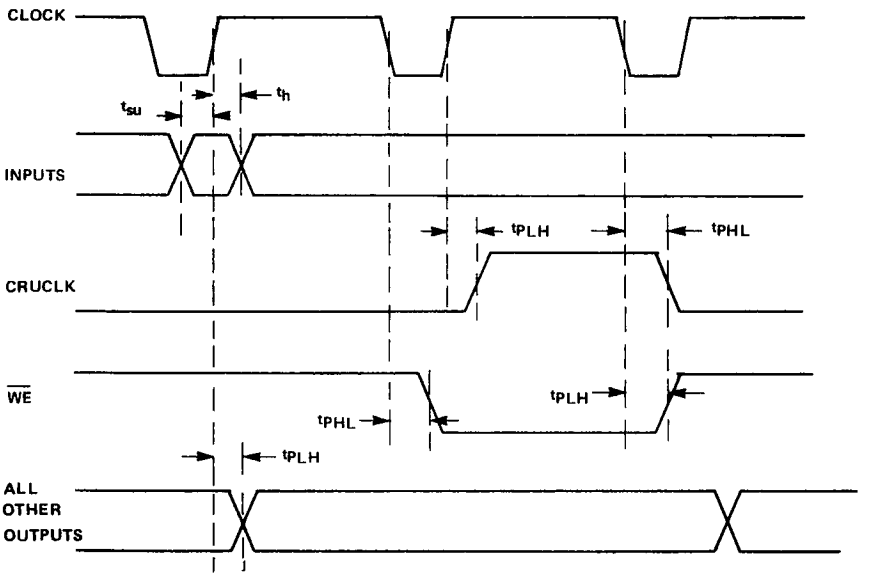


Figure 22. Switching Times — Voltage Waveforms

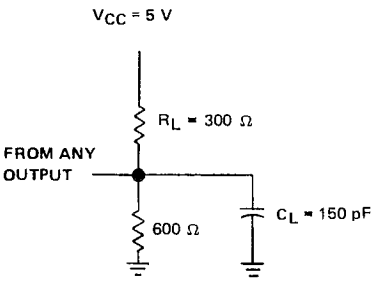


Figure 23. Switching Times Load Circuits

Clock Frequency vs. Temperature

Stability of the operational frequency over the full temperature range of −55°C to 125°C is illustrated in *Figure 24*. The effects of temperature on clock frequency are nil above 0°C. Below 0°C the effects are typically less than −8% with respect to the typical performance.

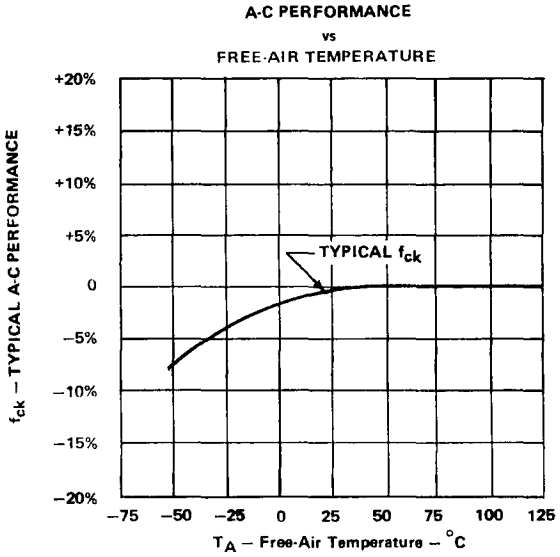


Figure 24. SBP 9900A A-C Performance vs. Temperature

TMS 9980A/
TMS 9981

1. INTRODUCTION

1.1 DESCRIPTION

The TMS 9980A/TMS 9981 is a software-compatible member of TI's 9900 family of microprocessors. Designed to minimize the system cost for smaller systems, the TMS 9980A/TMS 9981 is a single-chip 16-bit central processing unit (CPU) which has an 8-bit data bus, on-chip clock, and is packaged in a 40-pin package (see Figure 1). The instruction set of the TMS 9980A/TMS 9981 includes the capabilities offered by full minicomputers and is exactly the same as the 9900's. The unique memory-to-memory architecture features multiple register files, resident in memory, which allow faster response to interrupts and increased programming flexibility. The separate bus structure simplifies the system design effort. Texas Instruments provides a compatible set of MOS and TTL memory and logic function circuits to be used with a TMS 9980A/TMS 9981 system.

1.2 KEY FEATURES

- 16-Bit Instruction Word
- Full Minicomputer Instruction Set Capability Including Multiply and Divide
- Up to 16,384 Bytes of Memory
- 8-Bit Memory Data Bus
- Advanced Memory-to-Memory Architecture
- Separate Memory, I/O, and Interrupt-Bus Structures
- 16 General Registers
- 4 Prioritized Interrupts
- Programmed and DMA I/O Capability
- On-Chip 4-Phase Clock Generator
- 40-Pin Package
- N-Channel Silicon-Gate Technology

1.3 TMS 9980A/TMS 9981 DIFFERENCES

The TMS 9980A and the TMS 9981 although very similar, have several differences which user should be aware.

1. The TMS 9980A requires a V_{BB} supply (pin 21) while the TMS 9981 has an internal charge pump to generate V_{BB} from V_{CC} and V_{DD}.
2. The TMS 9981 has an optional on-chip crystal oscillator in addition to the external clock mode of the TMS 9980A.
3. The pin-outs are not compatible for D0-D7, INT0-INT2, and $\overline{\phi}3$.

8◀

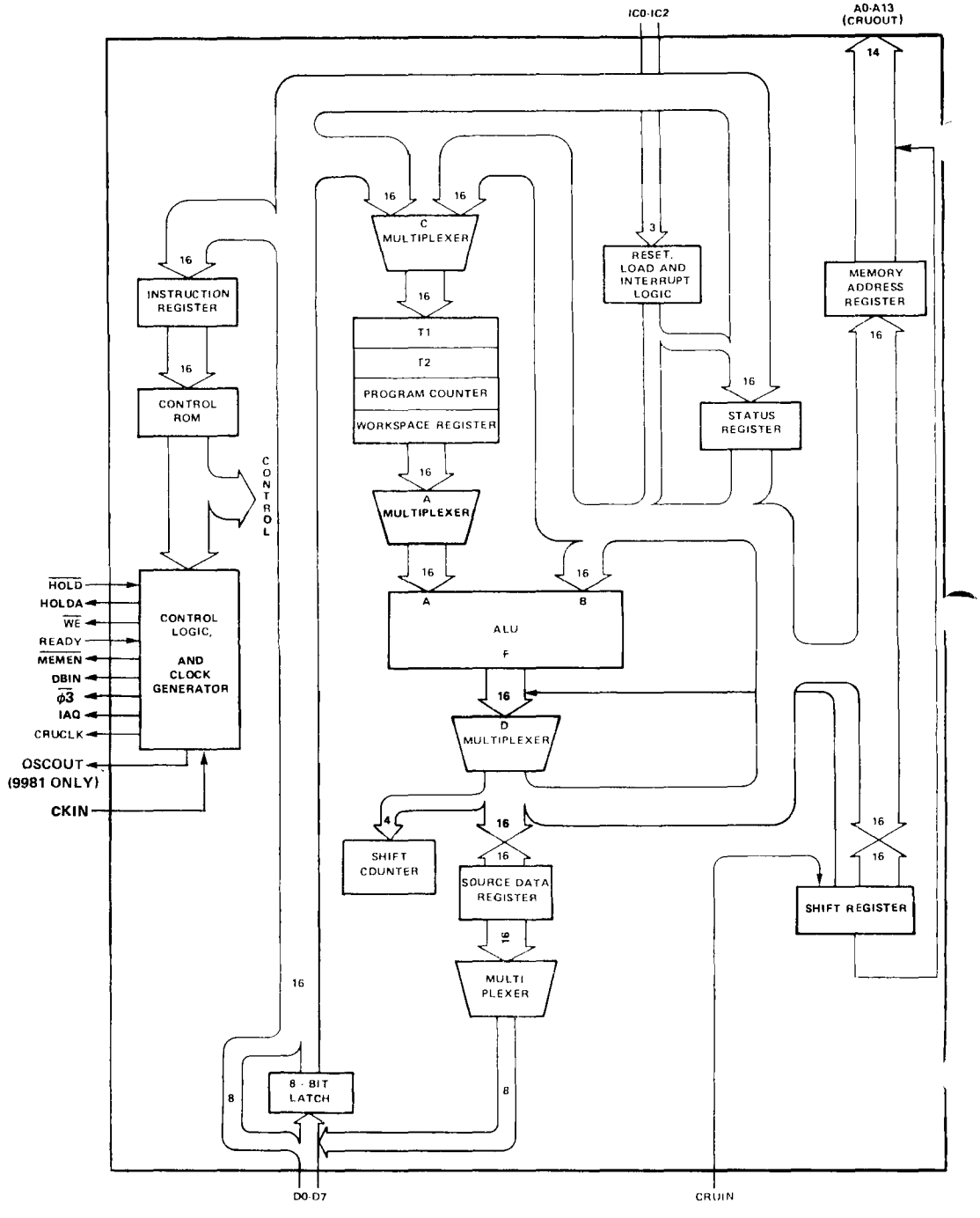
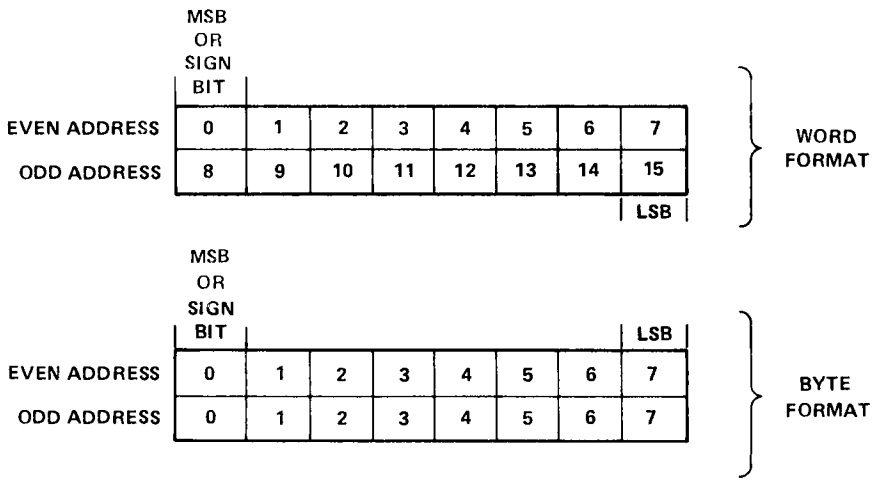


FIGURE 1 – ARCHITECTURE

2. ARCHITECTURE

The memory for the TMS 9980A/TMS 9981 is addressable in 8-bit bytes. A word is defined as 16 bits or 2 consecutive bytes in memory. The words are restricted to be on even address boundaries, i.e., the most-significant half (8 bits) resides at even address and the least-significant half resides at the subsequent odd address. A byte can reside at even or odd address. The word and byte formats are shown below.



2.1 REGISTERS AND MEMORY

The TMS 9980A/TMS 9981 employs an advanced memory-to-memory architecture. Blocks of memory designated as workspace replace internal hardware registers with program-data registers. The TMS 9980A/TMS 9981 memory map is shown in Figure 2. The first two words (4 bytes) are used for RESET trap vector. Addresses 0004_{16} through 0013_{16} are used for interrupt vectors. Addresses 0040 through 007F are used for the extended operation (XOP) instruction trap vectors. The last four bytes at address $3FFC_{16}$ to 3FFF are used for trap vector for the LOAD function.

The remaining memory is available for programs, data, and workspace registers. If desired, any of the special areas may also be used as general memory.

Three internal registers are accessible to the user. The program counter (PC) contains the address of the instruction following the current instruction being executed. This address is referenced by the processor to fetch the next instruction from memory and is then automatically incremented. The status register (ST) contains the present state of the processor and will be further defined in Section 3.4. The workspace pointer (WP) contains the address of the first word in the currently active set of workspace registers.

8◀

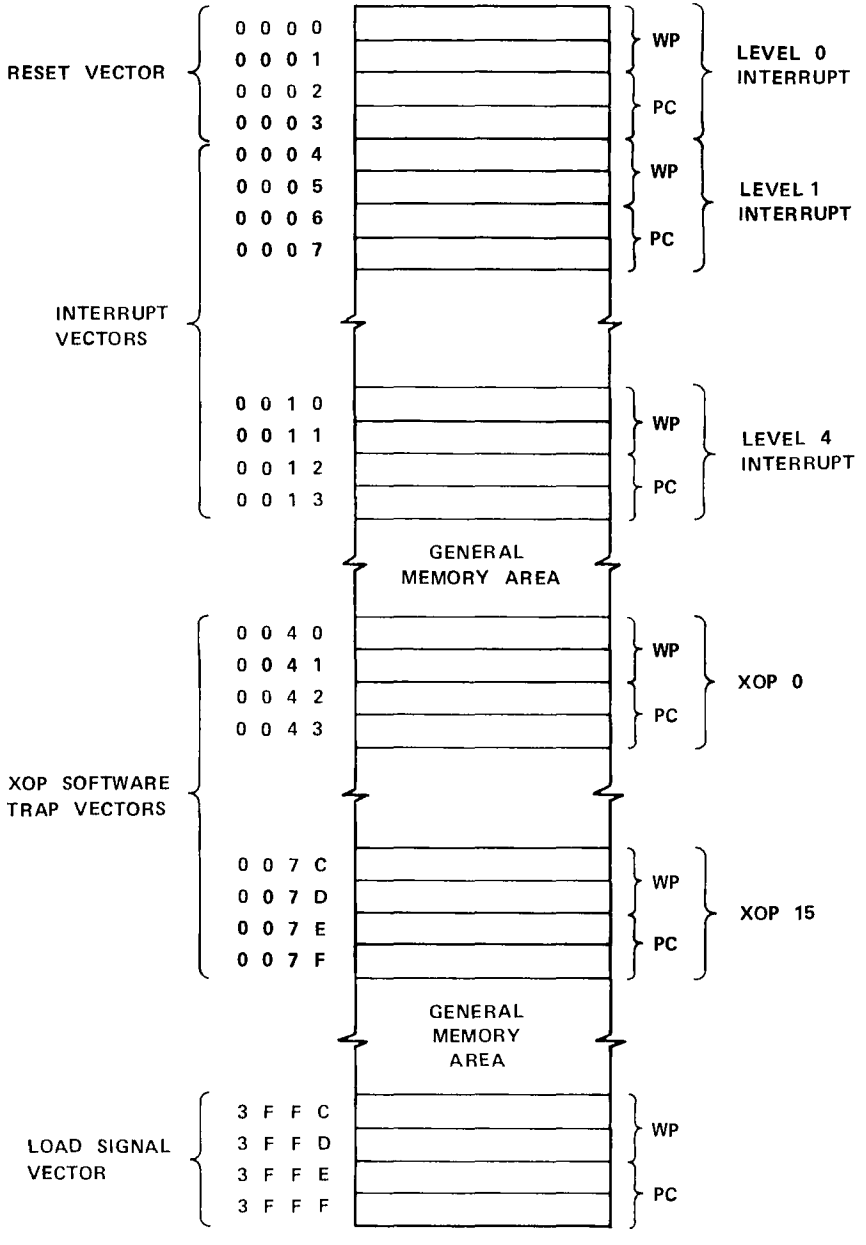
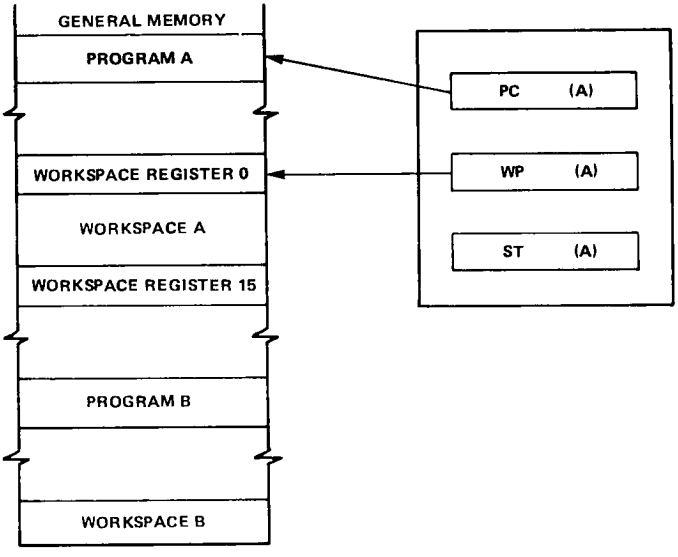


FIGURE 2 – MEMORY MAP

A workspace-register file occupies 16 contiguous memory words in the general memory area. Each workspace register may hold data or addresses and function as operand registers, accumulators, address registers, or index registers. During instruction execution, the processor addresses any register in the workspace by adding the register number to the contents of the workspace pointer and initiating a memory request for the word. The relationship between the workspace pointer and its corresponding workspace is shown below.



The workspace concept is particularly valuable during operations that require a context switch, which is a change from one program environment to another (as in the case of an interrupt or call to a subroutine). Such an operation, using a conventional multi-register arrangement, requires that at least part of the contents of the register file be stored and reloaded. A memory cycle is required to store or fetch each word. By exchanging the program counter, status register, and workspace pointer, the TMS 9980A/TMS 9981 accomplishes a complete context switch with only six store cycles and six fetch cycles. After the switch the workspace pointer contains the starting address of a new 16-word workspace in memory for use in the new routine. A corresponding time saving occurs when the original context is restored Instructions in the TMS 9980A/TMS 9981 that result in a context switch include:

1. Branch and Load Workspace Pointer (BLWP)
2. Return from Subroutine (RTWP)
3. Extended Operation (XOP)

Device interrupts, $\overline{\text{RESET}}$, and $\overline{\text{LOAD}}$ also cause a context switch by forcing the processor to trap to a service subroutine.

2.2 INTERRUPTS

The architecture of the 9900 family allows vectoring of 16 interrupts. These interrupts are assigned levels from 0 to 15. The interrupt at level 0 has the highest priority and the interrupt at level 15 has the lowest priority. The TMS 9900 implements all 16 interrupt levels. The TMS 9980A/TMS 9981 implements only 5 levels (level 0 and levels 1 through 4). Level 0 is reserved for $\overline{\text{RESET}}$ function.

Levels 1 through 4 may be used for external devices. The external levels may also be shared by several device interrupts, depending upon system requirements. The TMS 9980A/TMS 9981 continuously compares the interrupt code (IC0 through IC2) with the interrupt mask contained in status-register bits 12 through 15. When the level of the pending interrupt is less than or equal to the enabling mask level (higher or equal priority interrupt), the processor recognizes the interrupt and initiates a context switch following completion of the currently executing instruction. The processor fetches the new context WP and PC from the interrupt vector locations. Then, the previous context WP, PC, and ST are stored in workspace registers 13, 14, and 15, respectively, of the new workspace. The TMS 9980A/TMS 9981 then forces the interrupt mask to a value that is one less than the level of the interrupt being serviced. This allows only interrupts of higher priority to interrupt a service routine. The processor also inhibits interrupts until the first instruction of the service routine has been executed to allow modification of interrupt mask if needed (to mask out certain interrupts). All interrupt requests should remain active until recognized by the processor in the device-service routine. The individual service routines must reset the interrupt requests before the routine is complete. The interrupt code (IC0-IC2) may change asynchronously within the constraints specified in Section 2.10.4.

If a higher priority interrupt occurs, a second context switch occurs to service the higher-priority interrupt. When that routine is complete, a return instruction (RTWP) restores the first service routine parameters to the processor to complete processing of the lower-priority interrupt. All interrupt subroutines should terminate with the return instruction to restore original program parameters. The interrupt-vector locations, device assignment, enabling mask value and the interrupt code are shown in Table 1.

TABLE 1
INTERRUPT LEVEL DATA

INTERRUPT CODE (IC0–IC2)	FUNCTION	VECTOR LOCATION (MEMORY ADDRESS IN HEX)	DEVICE ASSIGNMENT	INTERRUPT MASK VALUES TO ENABLE (ST12 THROUGH ST15)
1 1 0	Level 4	0 0 1 0	External Device	4 Through F
1 0 1	Level 3	0 0 0 C	External Device	3 Through F
1 0 0	Level 2	0 0 0 8	External Device	2 Through F
0 1 1	Level 1	0 0 0 4	External Device	1 Through F
0 0 1	Reset	0 0 0 0	Reset Stimulus	Don't Care
0 1 0	Load	3 F F C	Load Stimulus	Don't Care
0 0 0	Reset	0 0 0 0	Reset Stimulus	Don't Care
1 1 1	No-Op	-----	-----	-----

▶8 Note that $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ functions are also encoded on the interrupt code input lines. Figure 3 illustrates some of the possible configurations. To realize $\overline{\text{RESET}}$ and one interrupt no external component is needed. If $\overline{\text{LOAD}}$ is also needed, a three input AND gate is wired as shown. If the system requires more than one interrupt, a single SN74148 (TIM 9907) is required.

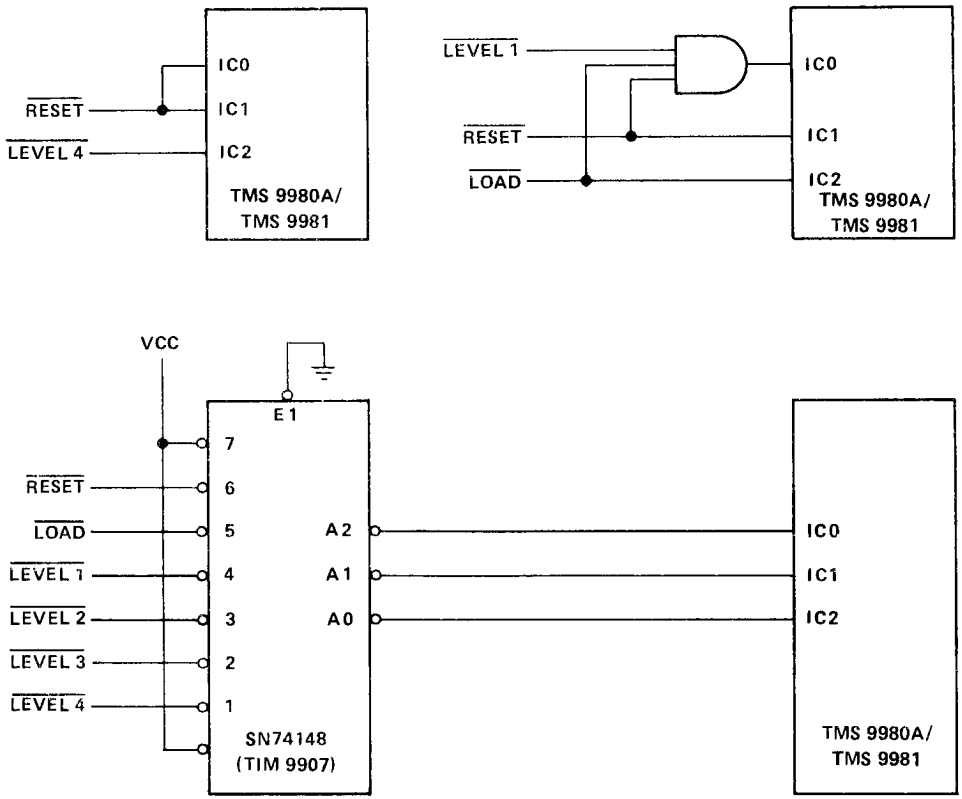


FIGURE 3 – TMS 9980A/TMS 9981 INTERRUPT INTERFACE

2.3 INPUT/OUTPUT

The TMS 9980A/TMS 9981 utilizes a versatile direct command-driven I/O interface designated as the communications-register unit (CRU). The CRU provides up to 2,048 directly addressable output bits. Both input and output bits can be addressed individually or in fields of from 1 to 16 bits. The TMS 9980A/TMS 9981 employs CRUIN, CRUCLK, and A13 (for CRUOUT) and 11 bits (A2-A12) of the address bus to interface with the CRU system. The processor instructions that drive the CRU interface can set, reset, or test any bit in the CRU array or move between memory and CRU data fields.

2.4 SINGLE-BIT CRU OPERATIONS

The TMS 9980A/TMS 9981 performs three single-bit CRU functions: test bit (TB), set bit to one (SBO), and set bit to zero (SBZ). To identify the bit to be operated upon, the TMS 9980A/TMS 9981 develops a CRU-bit address and places it on the address bus, A2 to A12.

For the two output operations (SBO and SBZ), the processor also generates a CRUCLK pulse, indicating an output operation to the CRU device and places bit 7 of the instruction word on the A13 line to accomplish the specified

operation (bit 7 is a one for SBO and a zero for SBZ). A test-bit instruction transfers the addressed CRU bit from the CRUIN input line to bit 2 of the status register (EQUAL).

The TMS 99B0A/TMS 9981 develops a hardware base address for the single-bit operations from the software base address contained in workspace register 12 and the signed displacement count contained in bits 8 through 15 of the instruction. The displacement allows two's complement addressing from base minus 128 bits through base plus 127 bits. The hardware base address (bits 4 through 14 of WR12) is added to the signed displacement specified in the instruction and the result is loaded into the address bus. *Figure 4* illustrates the development of a single-bit CRU address.

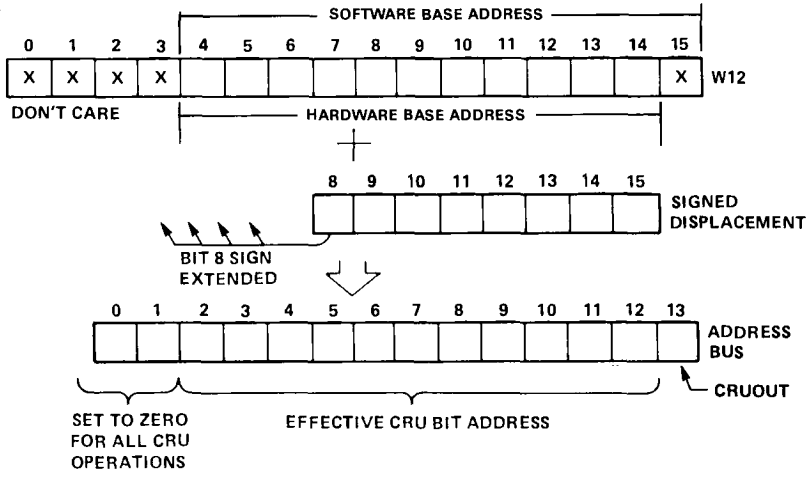


FIGURE 4 – TMS 9980A/TMS 9981 SINGLE-BIT CRU ADDRESS DEVELOPMENT

2.5 MULTIPLE-BIT CRU OPERATIONS

The TMS 9980A/TMS 9981 performs two multiple-bit CRU operations: store communications register (STCR) and load communications register (LDCR). Both operations perform a data transfer from the CRU-to-memory or from memory-to-CRU as illustrated in Figure 5. Although the figure illustrates a full 16-bit transfer operation, any number of bits from 1 through 16 may be involved. The LDCR instruction fetches a word from memory and right-shifts it to serially transfer it to CRU output bits. If the LDCR involves eight or fewer bits, those bits come from the right-justified field within the addressed byte of the memory word. If the LDCR involves nine or more bits, those bits come from the right-justified field within the whole memory word. When transferred to the CRU interface, each successive bit receives an address that is sequentially greater than the address for the previous bit. This addressing mechanism results in an order reversal of the bits; that is, bit 15 of the memory word (or bit 7) becomes the lowest addressed bit in the CRU and bit 0 becomes the highest addressed bit in the CRU field.

▶8

An STCR instruction transfers data from the CRU to memory. If the operation involves a byte or less transfer, the transferred data will be stored right-justified in the memory byte with leading bits set to zero. If the operation involves from nine to 16 bits, the transferred data is stored right-justified in the memory word with leading bits set to zero.

When the input from the CRU device is complete, the first bit from the CRU is the least-significant-bit position in the memory word or byte.

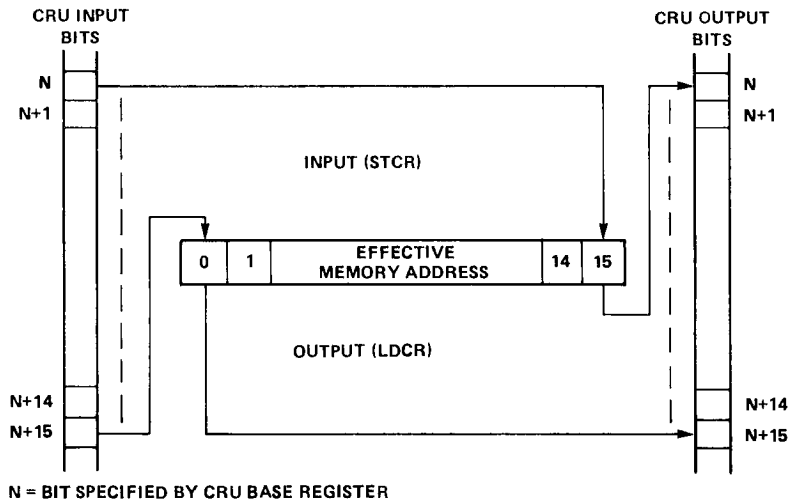


FIGURE 5 – TMS 9980A/TMS 9981 LDCR/STCR DATA TRANSFERS

Figure 6 illustrates how to implement a 16-bit input and a 16-bit output register in the CRU interface. CRU addresses are decoded as needed to implement up to 128 such 16-bit interface registers. In system application, however, only the exact number of interface bits needed to interface specific peripheral devices are implemented. It is not necessary to have a 16-bit interface register to interface an 8-bit device.

2.6 EXTERNAL INSTRUCTIONS

The TMS 9980A/TMS 9981 has five external instructions that allow user-defined external functions to be initiated under program control. These instructions are CKON, CKOF, RSET, IDLE, and LREX. These mnemonics, except for IDLE, relate to functions implemented in the 990 minicomputer and do not restrict use of the instructions to initiate various user-defined functions. IDLE also causes the TMS 9980A/TMS 9981 to enter the idle state and remain until an interrupt, RESET, or LOAD occurs. When any of these five instructions are executed by the TMS 9980A/TMS 9981, a unique 3-bit code appears on the address bus, bits A13, A0, and A1, along with a CRUCLK pulse. When the TMS 9980A/TMS 9981 is in an idle state, the 3-bit code and CRUCLK pulses occur repeatedly until the idle state is terminated. The codes are:

EXTERNAL INSTRUCTION	A13	A0	A1
LREX	H	H	H
CKOF	H	H	L
CKON	H	L	H
RSET	L	H	H
IDLE	L	H	L
CRU INSTRUCTIONS	H/L	L	L

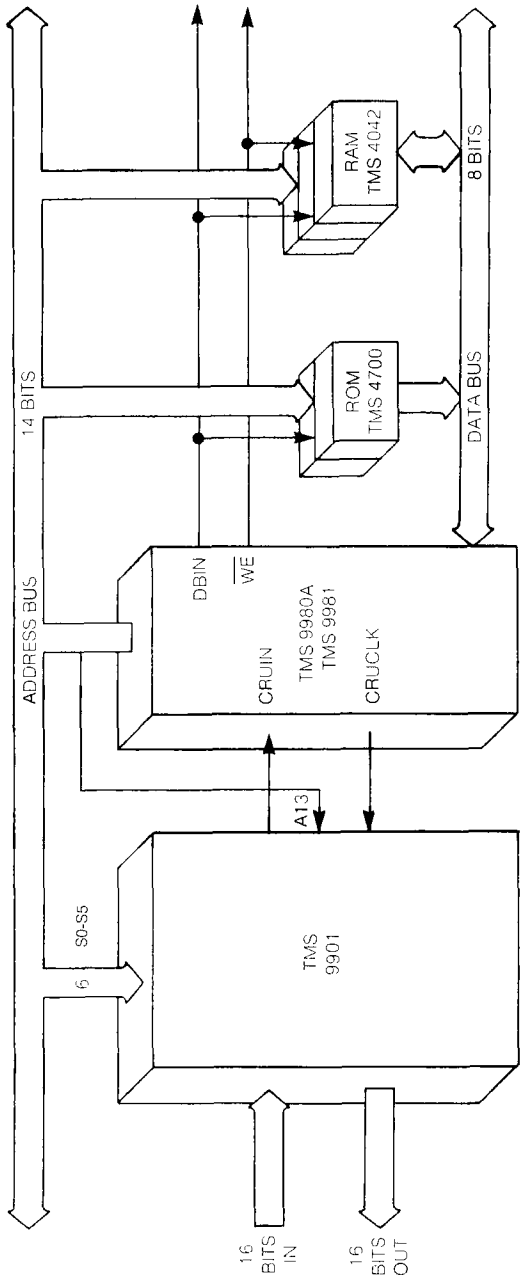


FIGURE 6 – TMS 9980A/9981 16-BIT INPUT/OUTPUT INTERFACE

Note that during external instructions bits (A2-A12) of the address bus may have any of the possible binary patterns. Since these bits (A2-A12) are used as CRU addresses, CRUCLK to the CRU must be gated with a decode of 0 on A0 and A1 to avoid erroneous strobe to CRU bits during external instruction execution.

Figure 7 illustrates typical external decode logic to implement these instructions. Note CRUCLK to the CRU is inhibited during external instructions.

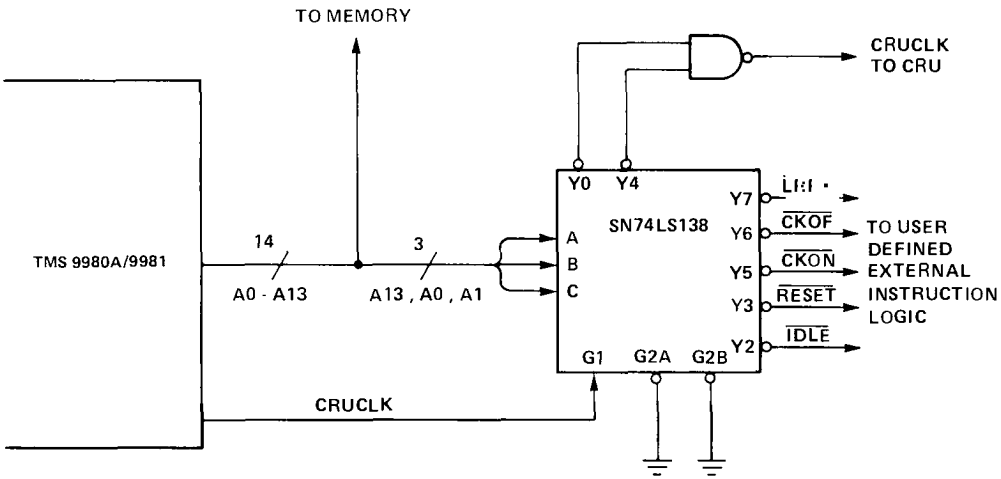


FIGURE 7 – EXTERNAL INSTRUCTION DECODE LOGIC

2.7 NON-MASKABLE INTERRUPTS

2.7.1 LOAD Function

The LOAD stimulus is an unmaskable interrupt that allows cold-start ROM loaders and front panels to be implemented for the TMS 9980A/TMS 9981. When the TMS 9980A/TMS 9981 decodes LOAD on IC0-IC2 lines, it initiates an interrupt sequence immediately following the instruction being executed. Memory location 3FFC is used to obtain the vector (WP and PC). The old PC, WP, and ST are loaded into the new workspace and the interrupt mask is set to 0000. Then the program execution resumes using the new PC and WP. Recognition of LOAD by the processor will also terminate the idle condition. External stimulus for LOAD must be held active (on IC0-IC2) for one instruction period by using IAQ signal

2.7.2 RESET

When the TMS 9980A/TMS 9981 recognizes a RESET on IC0-IC2, it resets and inhibits $\overline{WE}$ and CRUCLK. Upon removal of the RESET code, the TMS 9980A/TMS 9981 initiates a level-zero interrupt sequence that acquires WP and PC from location 0000 and 0002, sets all status register bits to zero and starts execution. Recognition of RESET by the processor will also terminate an idle state. External stimulus for RESET must be held active for a minimum of three clock cycles.

8

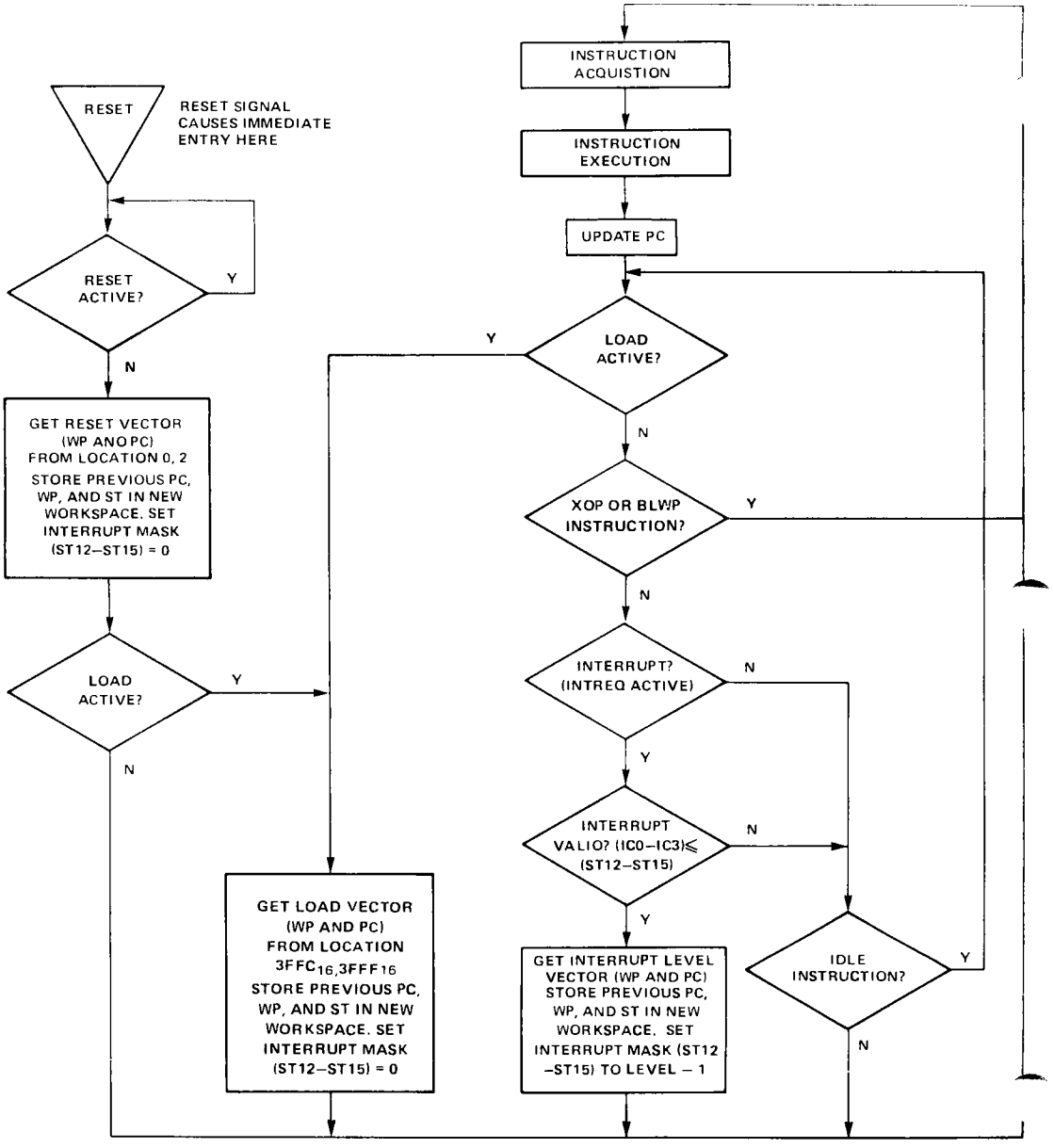


FIGURE 8 – TMS 9980A/TMS 9981 CPU FLOW CHART

2.8 TMS 9980A PIN DESCRIPTION

Table 2 defines the TMS 9980A pin assignments and describes the function of each pin.

TABLE 2
TMS 9980A PIN ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
			ADDRESS BUS
A0 (MSB)	17	OUT	A0 through A13 comprise the address bus. This 3-state bus provides the memory-address vector to the external-memory system when MEMEN is active and I/O-bit addresses and external-inst addresses to the I/O system when $\overline{M}$ ** . s inactive. The address bus assumes the high-impedance state when HOLDA is active.
A1	16	OUT	
A2	15	OUT	
A3	14	OUT	
A4	13	OUT	
A5	12	OUT	
A6	11	OUT	
A7	10	OUT	
A8	9	OUT	
A9	8	OUT	
A10	7	OUT	
A11	6	OUT	
A12	5	OUT	
A13/CRUOUT	4	OUT	**CRUOUT** Serial I/O data appears on A13 when an LDCR, SBZ and SBO instruction is executed. This data should be sampled by the I/O interface logic when CRUCLK goes active (high). One bit of the external instruction code appears on A13 during external instruction execution.
			DATA BUS
D0 (MSB)	26	I/O	D0 through D7 comprise the bidirectional 3-state data bus. This bus transfers memory data to (when writing) and from (when reading) the external-memory system when $\overline{MEMEN}$ is active. The data bus assumes the high-impedance state when HOLDA is active.
D1	27	I/O	
D2	28	I/O	
D3	29	I/O	
D4	30	I/O	
D5	31	I/O	
D6	32	I/O	
D7 (LSB)	33	I/O	
			POWER SUPPLIES
V_{BB}	21		Supply voltage (—5V NOM)
V_{CC}	20		Supply voltage (5 V NOM)
V_{DD}	36		Supply voltage (12 V NOM)
V_{SS}	35		Ground reference
			CLOCKS
CKIN	34	IN	Clock In. A TTL compatible input used to generate the internal 4-phase clock. CKIN frequency is 4 times the desired system frequency.
$\overline{\phi 3}$	22	OUT	Clock phase 3 ($\phi 3$) inverted; used as a timing reference.
			BUS CONTROL
DBIN	18	OUT	Data bus in. When active (high), DBIN indicates that the TMS 9980A has disabled its output buffers to allow the memory to place memory-read data on the data bus during MEMEN. DBIN remains low in all other cases except when HOLDA is active at which time it is in the high-impedance state.

TMS 9980A PIN ASSIGNMENTS

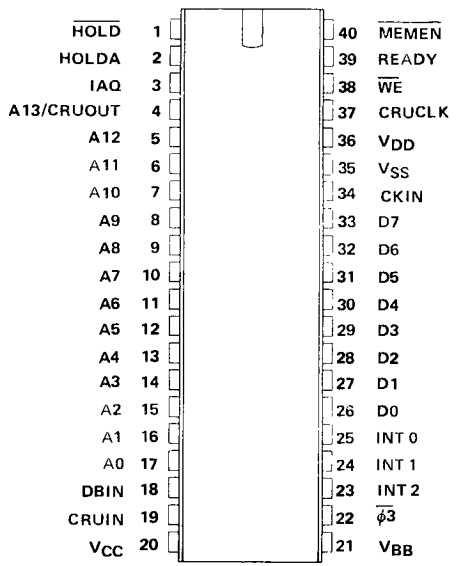


TABLE 2 (CONTINUED)

[illegible]TURE	PIN	I/O	DESCRIPTION
[illegible]	40	OUT	Memory enable. V [illegible] (low), $\overline{\text{MEMEN}}$ indicates that the address bus contains a memory address. When HOLDA is active [illegible] is in the high impedance state.
WE	38	OUT	Write enable. When active (low), $\overline{\text{WE}}$ indicates that memory-write data is available from the TMS 9980 to b written into memory. When HOLDA is active, $\overline{\text{WE}}$ is in the high-impedance state.
CRUCLK	37	OUT	CRU clock. When active (high), CRUCLK indicates that external interface logic should sample the output data on CRUOUT or should decode external instructions on A0, A1, A13.
CRUIN	19	IN	CRU data in. CRUIN, normally driven by 3-state or open-collector devices, receives input data from external interface logic. When the processor executes a STCR or TB instruction, it samples CRUIN for the level of the CRU input bit specified by the address bus (A2 through A12).
INT2	23	IN	Interrupt code. Refer to Section 2.2 for detailed description.
INT1	24	IN	
INT0	25	IN	
			MEMORY CONTROL
$\overline{\text{HOLD}}$	1	IN	Hold. When active (low), $\overline{\text{HOLD}}$ indicates to the processor that an external controller (e.g , DMA device) desires to utilize the address and data buses to transfer data to or from memory. The TMS 9980A enters the hold state following a hold signal when it has completed its present memory cycle.* The processor then places the address and data buses in the high-impedance state (along with $\overline{\text{WE}}$, $\overline{\text{MEMEN}}$, and DBIN) and responds with a hold-acknowledge signal (HOLDA). When $\overline{\text{HOLD}}$ is removed, the processor returns to normal operation.
HOLDA	2	OUT	Hold acknowledge. When active (high), HOLDA indicates the processor is in the hold state and the address and data buses and memory control outputs ($\overline{\text{WE}}$, [illegible] $\overline{\text{EN}}$, and DBIN) are in the high-impedance state.
READY	39	IN	Ready. When active (high), READY indicates that memory will be ready to read or write during the ne> clock cycle. When not-ready is indicated during a memory operation, the TMS 9980A enters a wait state and suspends internal operation until the memory systems indicated ready.
			TIMING AND CONTROL
IAQ	3	OUT	Instruction acquisition. IAQ is active (high) during any memory cycle when the TMS 9980A is acquiring an instruction. IAQ can be used to detect illegal op codes. It may also be used to synchronize LOAD stimulus.

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

2.9 TMS 9981 PIN DESCRIPTION

Table 3 defines the TMS 9981 pin assignments and describes the function of each pin.

TABLE 3
TMS 9981 PIN ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
			ADDRESS BUS
A0(MSB)	17	OUT	A0 through A13 comprise the address bus. This 3-state bus provides the memory-address vector to the external-memory system when MEMEN is active and I/O-bit addresses and external-instruction addresses to the I/O system when $\overline{\text{MEMEN}}$ is inactive. The address bus assumes the high-impedance state when HOLDA is active.
A1	16	OUT	
A2	15	OUT	
A3	14	OUT	
A4	13	OUT	
A5	12	OUT	
A6	11	OUT	
A7	10	OUT	
A8	9	OUT	
A9	8	OUT	
A10	7	OUT	
A11	6	OUT	
A12	5	OUT	
A13/CRUOUT	4	OUT	**CRUOUT** Serial I/O data appears on A13 when an LDCR, SBZ and SBO instruction is executed. This data should be sampled by the I/O interface logic when CRUCLK goes active (high). One bit of the external instruction code appears on A13 during external instruction execution.
			DATA BUS
D0 (MSB)	25	I/O	D0 through D7 comprise the bidirectional 3-state data bus. This bus transfers memory data to (when writing) and from (when reading) the external-memory system when $\overline{\text{MEMEN}}$ is active. The data bus assumes the high-impedance state when HOLDA is active.
D1	26	I/O	
D2	27	I/O	
D3	28	I/O	
D4	29	I/O	
D5	30	I/O	
D6	31	I/O	
D7 (LSB)	32	I/O	
			PDWER SUPPLIES
V_{CC}	20		Supply voltage (5 V NOM)
V_{DD}	36		Supply voltage (12 V NOM)
V_{SS}	35		Ground reference
			CLOCKS
CKIN	34	IN	Clock In and Oscillator Out. These pins may be used in either of two modes to generate the internal 4-phase clock. In mode 1 a crystal of 4 times the desired system frequency is connected between CKIN and OSCOUT (see Figure 13). In mode 2 OSCOUT is left floating and CKIN is driven by a TTL compatible source whose frequency is 4 times the desired system frequency.
OSCOUT	33	OUT	
$\overline{\phi 3}$	21	OUT	Clock phase 3 ($\phi 3$) inverted; used as a timing reference.
			BUS CONTROL
DBIN	18	OUT	Data bus in. When active (high), DBIN indicates that the TMS 9981 has disabled its output buffers to allow the memory to place memory-read data on the data bus during $\overline{\text{MEMEN}}$. DBIN remains low in all other cases except when HOLDA is active at which time it is in the high-impedance state.

TMS 9981 PIN ASSIGNMENTS

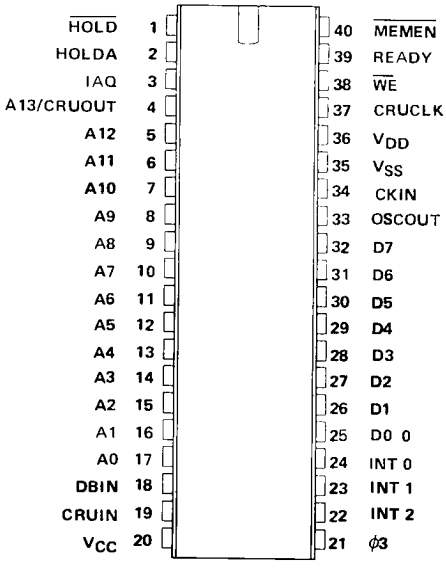


TABLE 3 (CONTINUED)

SIGNATURE	PIN	I/O	DESCRIPTION
$\overline{\text{MEMEN}}$	40	OUT	Memory enable. When active (low), $\overline{\text{MEMEN}}$ indicates that the address bus contains a memory address. When HOLDA is active [illegible] is in the high-impedance state.
$\overline{\text{WE}}$	38	OUT	Write enable. When active (low), $\overline{\text{WE}}$ indicates that memory-write data is available from the TMS 9981 to be written into memory. When HOLDA is active, $\overline{\text{WE}}$ is in the high-impedance state.
CRUCLK	37	OUT	CRU clock. When active (high), CRUCLK indicates that external interface logic should sample the output data on CRUOUT or should decode external instructions on A0, A1, A13.
CRUIN	19	IN	CRU data in. CRUIN, normally driven by 3-state or open-collector devices, receives input data from external interface logic. When the processor executes a STCR or TB instruction, it samples CRUIN for the level of the CRU input bit specified by the address bus (A2 through A12).
INT2 INT1 INT0	22 23 24	IN IN IN	Interrupt code. Refer to Section 2.2 for detailed description.
$\overline{\text{HOLD}}$	1	IN	**MEMORY CONTROL** Hold. When active (low), HOLD indicates to the processor that an external controller (e.g., DMA device) desires to utilize the address and data buses to transfer data to or from memory. The TMS 9981 enters the hold state following a hold signal when it has completed its present memory cycle.* The processor then places the address and data buses in the high-impedance state (along with $\overline{\text{WE}}$, $\overline{\text{MEMEN}}$, and DBIN) and responds with a hold-acknowledge signal (HOLDA). When $\overline{\text{HOLD}}$ is removed, the processor returns to normal operation.
HOLDA	2	OUT	Hold acknowledge. When active (high), HOLDA indicates that the processor is in the hold state and the address and data buses and memory control outputs ($\overline{\text{WE}}$, $\overline{\text{MEMEN}}$, and DBIN) are in the high-impedance state.
READY	39	IN	Ready. When active (high), READY indicates that memory will be ready to read or write during the next clock cycle. When not-ready is indicated during a memory operation, the TMS 9981 enters a wait state and suspends internal operation until the memory systems indicated ready.
IAQ	3	OUT	**TIMING AND CONTROL** Instruction acquisition. IAQ is active (high) during any memory cycle when the TMS 9981 is acquiring an instruction. IAQ can be used to detect illegal op codes. It may also be used to synchronize LOAD stimulus.

*If the cycle following the present memory cycle is also a memory cycle it, too, is completed before TMS 9981 enters hold state

2.10 TIMING

2.10.1 Memory

Basic memory read and write cycles are shown in Figures 9a and 9b. Figure 9a shows a read and a write cycle with no wait states while Figure 9b shows a read and a write cycle for a memory requiring one wait state.

8

$\overline{\text{MEMEN}}$ goes active (low) during each memory cycle. At the same time that $\overline{\text{MEMEN}}$ is active, the memory address appears on the address bits A0 through A13. Since the TMS 9980A/TMS 9981 has an 8-bit data bus, every memory operation consists of two consecutive memory cycles. Address bit A13 is 0 for the first of the two cycles and goes to 1 for the second. If the cycle is a memory-read cycle, DBIN will go active (high) at the same time $\overline{\text{MEMEN}}$ and A0 through A13 become valid. The memory-write ($\overline{\text{WE}}$) signal remains inactive during a read cycle.

The READY signal allows extended memory cycle as shown in Figure 9b.

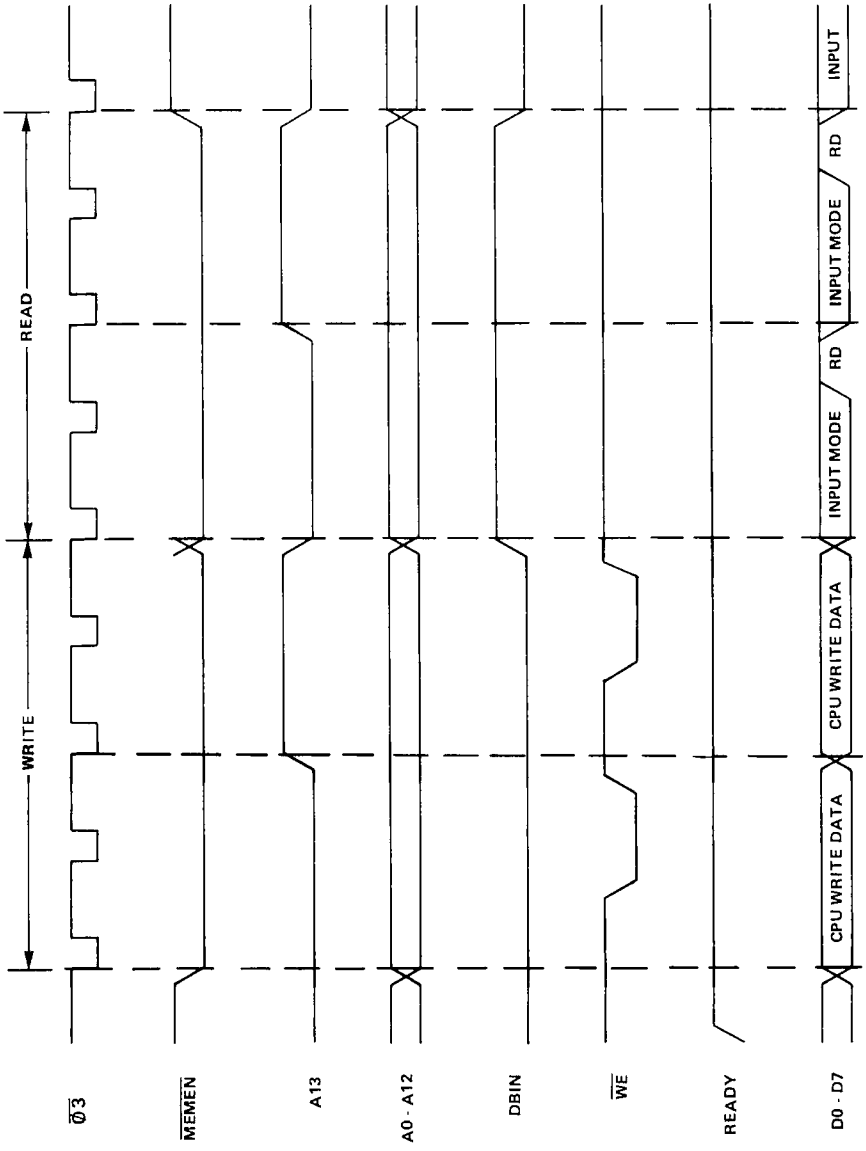


FIGURE 9a – TMS 9980A/TMS 9981 MEMORY BUS TIMING (NO WAIT STATES)

8

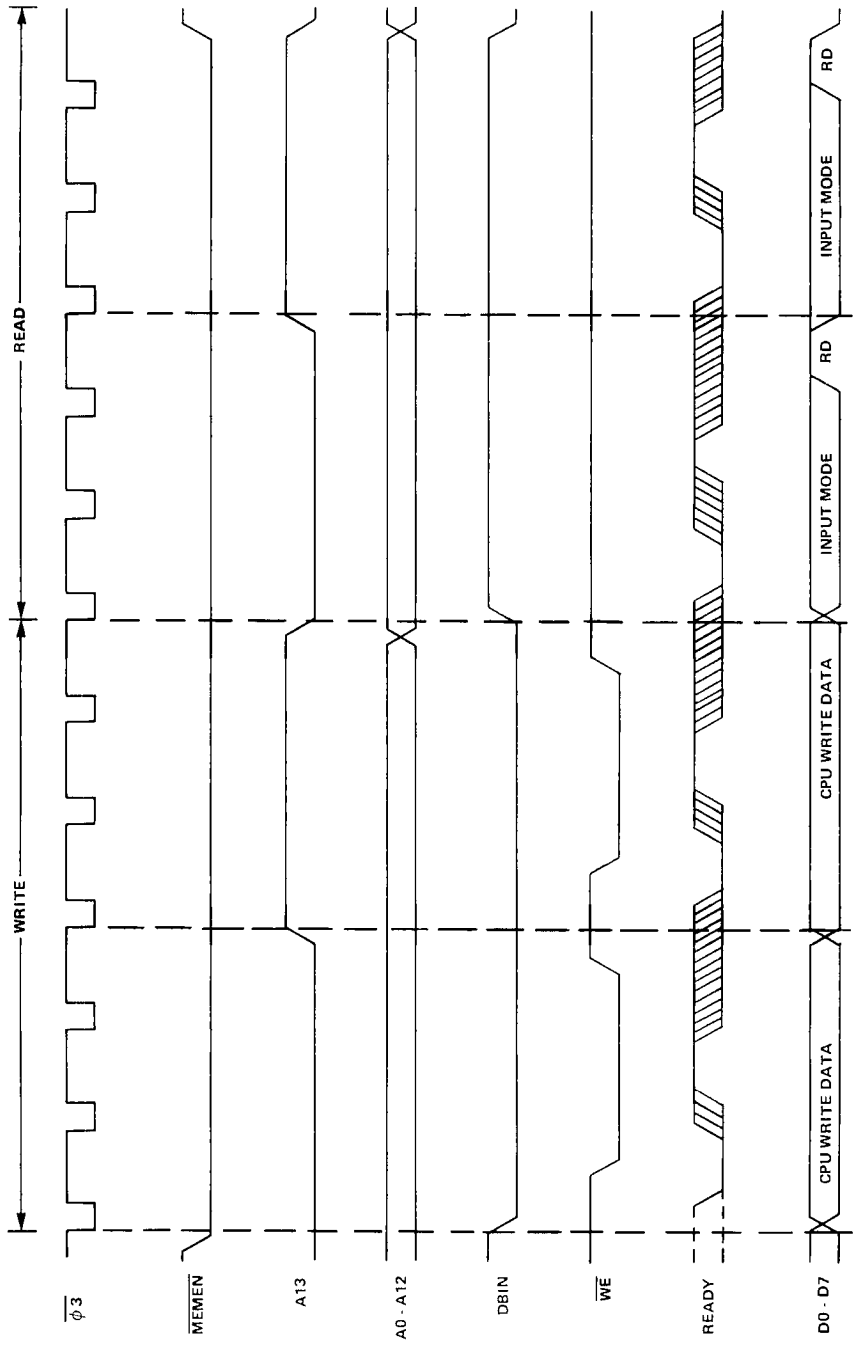


FIGURE 9b – TMS 9980A/TMS 9981 MEMORY BUS TIMING (ONE WAIT STATE)

▶8

At the end of the read cycle, $\overline{\text{MEMEN}}$ and DBIN go inactive (high and low respectively). The address bus also changes at this time, however, the data bus remains in the input mode for one clock cycle after the read cycle.

A write cycle is similar to read cycle except that $\overline{\text{WE}}$ goes active (low) as shown and valid write data appears on the data bus at the same time the address appears.

2.10.2 HOLD

Other interfaces may utilize the TMS 9980A/TMS 9981 memory bus by using the hold operation (illustrated in Figure 10) of the TMS 9980A/TMS 9981. When $\overline{\text{HOLD}}$ is active (low), the TMS 9980A/TMS 9981 enters the hold state at the next available non-memory cycle clock period. When the TMS 9980A/TMS 9981 has entered the hold state HOLDA goes active (high), A0 through A13, D0 through D7, DBIN, $\overline{\text{MEMEN}}$, and $\overline{\text{WE}}$ go into high-impedance state to allow other devices to use the memory buses. When $\overline{\text{HOLD}}$ goes inactive, TMS 9980A/TMS 9981 resumes processing as shown. Considering that there can be a maximum of 6 consecutive memory operations, the maximum delay between $\overline{\text{HOLD}}$ going active to HOLDA going active (high) could be $t_{c(\phi)}$ (for set up) + (12 + 6 W) $t_{c(\phi)}$ (delay for HOLDA), where W is the number of wait states per memory cycle and $t_{c(\phi)}$ is the clock cycle time. If hold occurs during a CRU operation, the TMS 9980A/TMS 9981 uses an extra clock cycle (after the removal of the $\overline{\text{HOLD}}$ signal) to reassert the CRU address providing the normal setup times for the CRU bit transfer that was interrupted.

2.10.3 CRU

CRU interface timing is shown in Figure 11. The timing for transferring two bits out and one bit in is shown. These transfers would occur during the execution of a CRU instruction. The other cycles of the instruction execution are not illustrated. To output a CRU bit, the CRU-bit address is placed on the address bus A2 through A12 and the actual bit data on A13. During the second clock cycle a CRU pulse is supplied by CRUCLK. This process is repeated until the number of bits specified by the instruction are completed.

The CRU input operation is similar in that the bit address appears on A2 through A12. During the subsequent cycle, the TMS 9980A/TMS 9981 accepts the bit input data as shown. No CRUCLK pulses occur during a CRU input operation.

2.10.4 Interrupt Code (IC0-IC2)

The TMS 9980A/TMS 9981 uses 4 phase clock (ϕ1, ϕ2, ϕ3, and ϕ4) for timing and control of the internal operations. IC0-IC2 are sampled during ϕ4 and then during ϕ2.

If these two successive samples are equal, the code is accepted and latched for internal use on the subsequent ϕ1. In systems with simple interrupt structures this allows the interrupt code to change asynchronously without the TMS 9980A/TMS 9981 accepting erroneous codes. Figure 3 shows systems with a single level of external interrupt implemented that would require no external timing. When implementing multiple external interrupts, as in the bottom diagram of Figure 3, external synchronization of interrupt requests is required. See Figure 12 for a timing diagram. In systems with more than one external interrupt, the interrupts should be synchronized with the $\overline{\phi 3}$ output of the TMS 9980A/TMS 9981 to avoid code transitions on successive sample cycles. This synchronization ensures that the TMS 9980A/TMS 9981 will service only the proper active interrupt level.

8◀

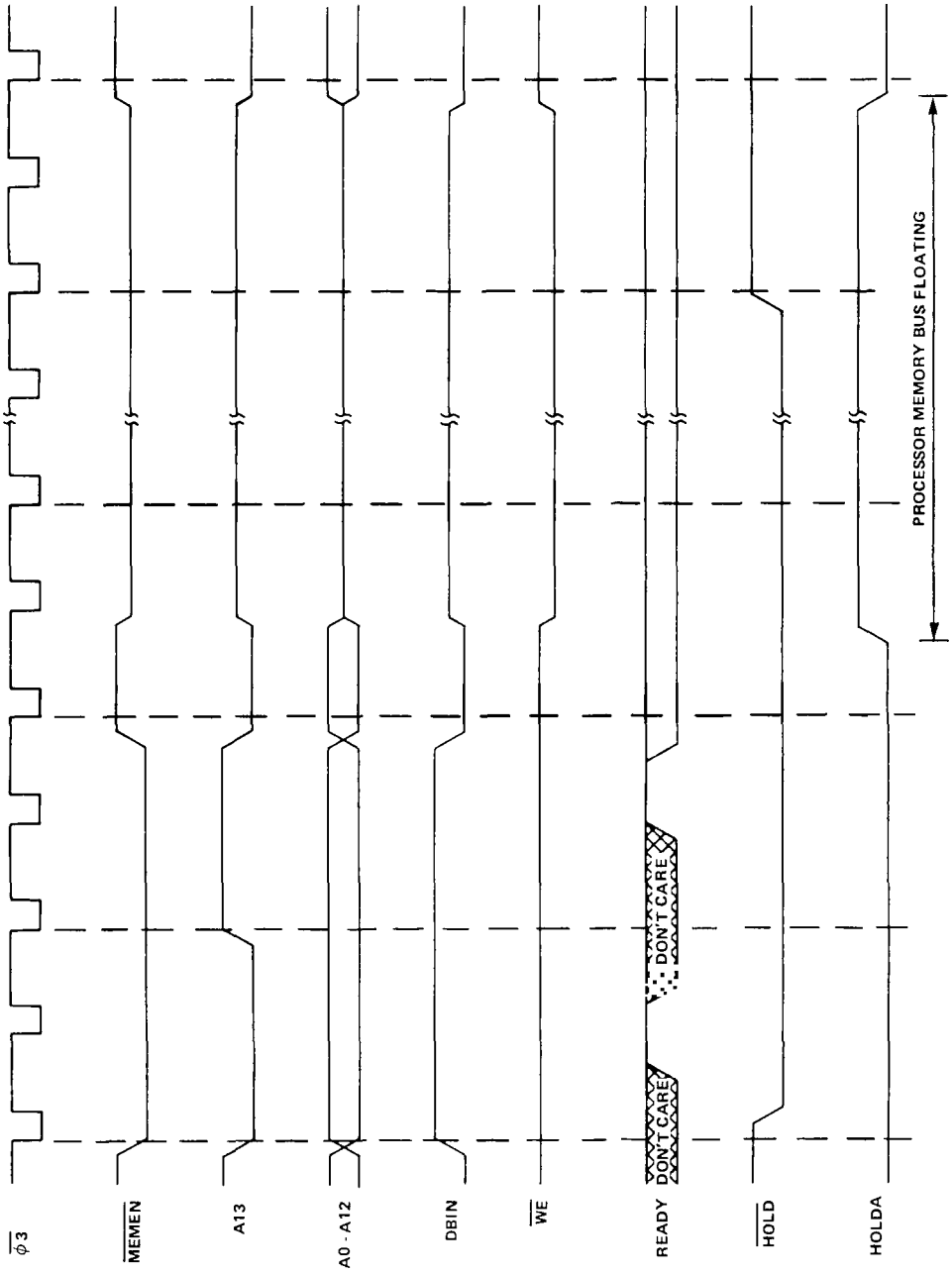


FIGURE 10 – TMS 9980A/TMS 9981 HOLD TIMING

▶8

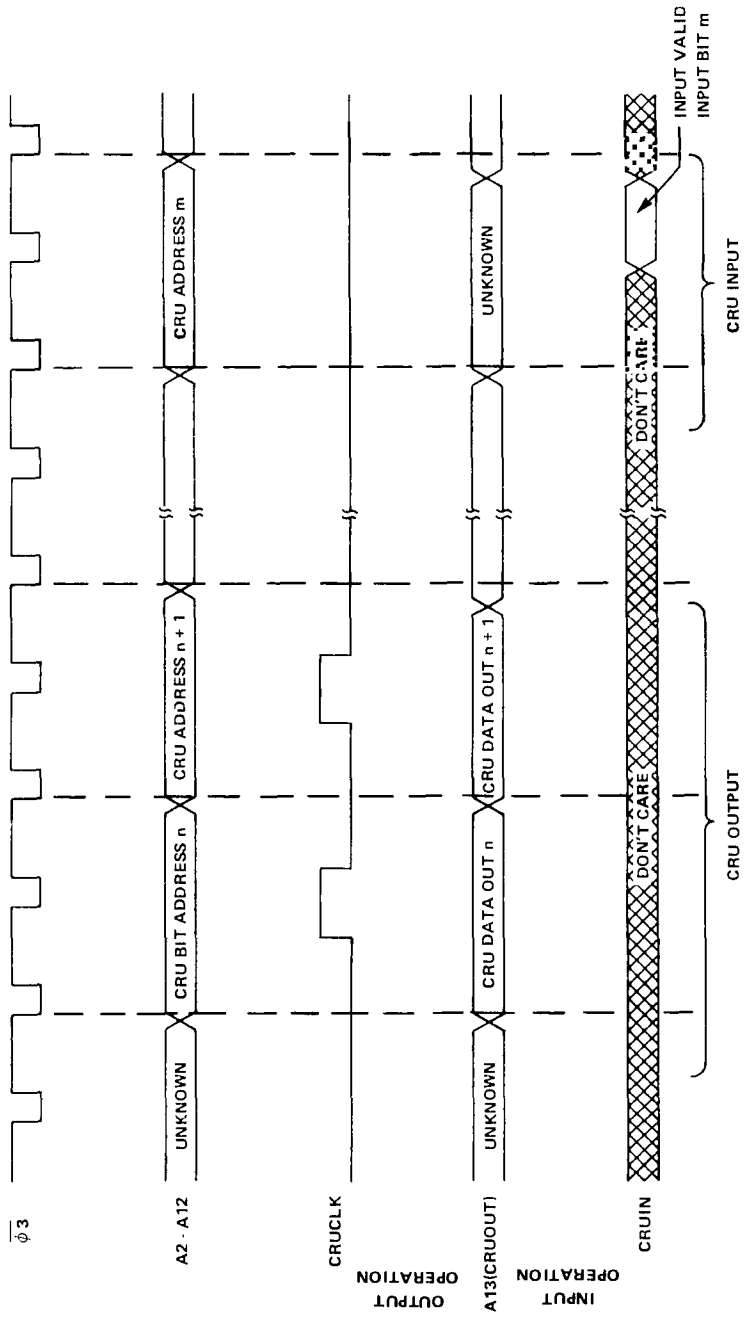


FIGURE 11 – TMS 9980A/TMS 9981 CRU INTERFACE TIMING

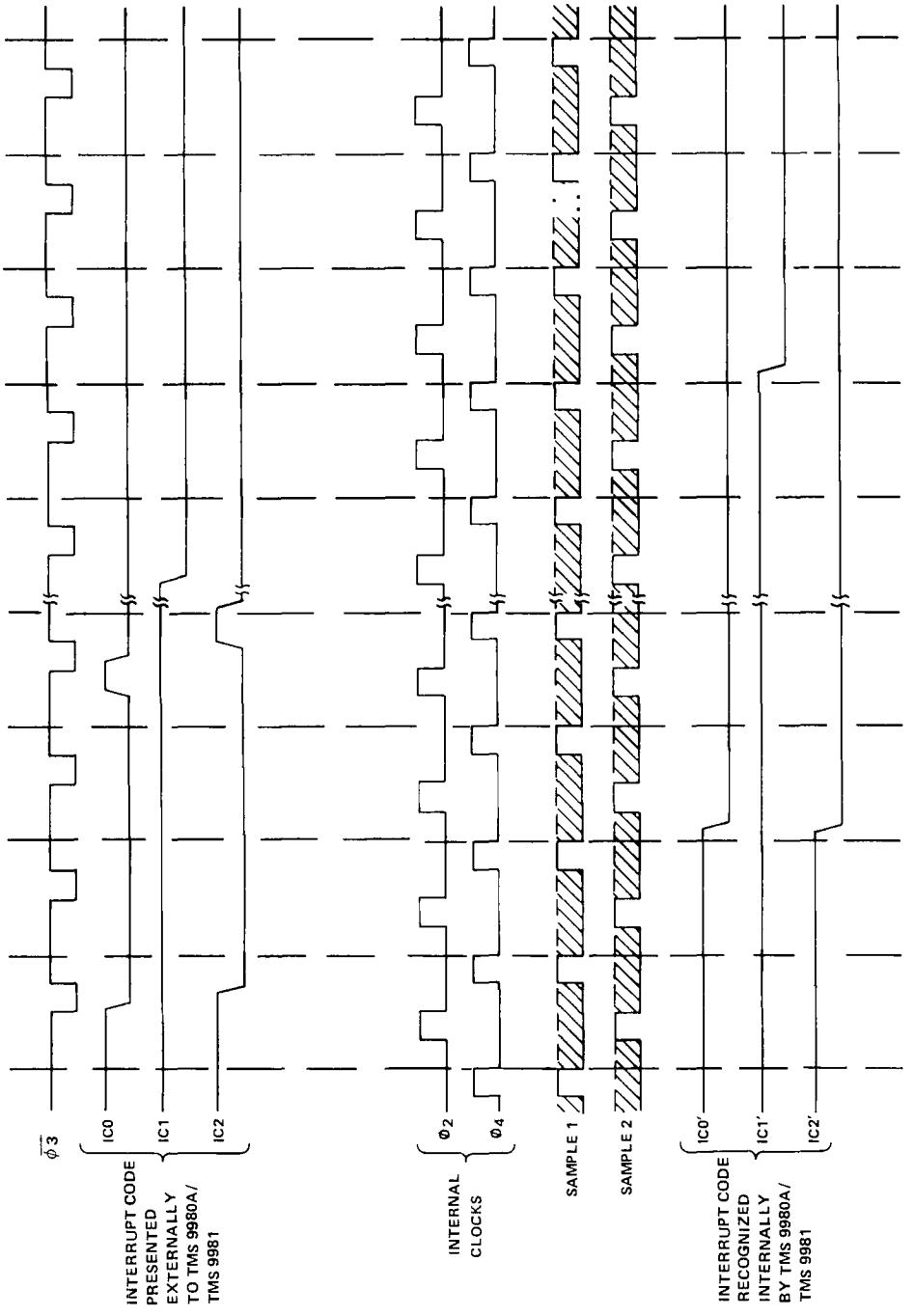


FIGURE 12 – INTERRUPT CODE TIMING

▶8

3.6 TMS 9980A/TMS 9981 INSTRUCTION EXECUTION TIMES

Instruction execution times for the TMS 9980A/TMS 9981 are a function of:

1) Clock cycle time, $t_{c(\phi)}$

2) Addressing mode used where operands have multiple addressing mode capability

3) Number of wait states required per memory access.

Table 4 lists the number of clock cycles and memory accesses required to execute each TMS 9980A/TMS 9981 instruction. For instructions with multiple addressing modes for either or both operands, Table 4 lists the number of clock cycles and memory accesses with all operands addressed in the workspace-register mode. To determine the additional number of clock cycles and memory accesses required for modified addressing, add the appropriate values from the referenced tables. The total instruction-execution time for an instruction is:

$$T = t_{c(\phi)} (C+W \cdot M)$$

where:

T = total instruction time;

$t_{c(\phi)}$ = clock cycle time;

C = number of clock cycles for instruction execution plus address modification;

W = number of required wait states per memory access for instruction execution plus address modification;

M = number of memory accesses.

As an example, the instruction MOVB is used in a system with $t_{c(\phi)} = 0.400\mu s$ and no wait states are required to access memory. Both operands are addressed in the workspace register mode:

$$T = t_{c(\phi)} (C+W \cdot M) = 0.400\ (22+0 \cdot 8) = 8.8\ \mu s.$$

If two wait states per memory access were required, the execution time is:

$$T = 0.400\ (22 + 2 \cdot 8)\ \mu s = 15.2\ \mu s.$$

If the source operand was addressed in the symbolic mode and two wait states were required:

$$T = t_{c(\phi)} (C+W \cdot M)$$

$$C = 22 + 10 = 32$$

$$M = 8 + 2 = 10$$

$$T = 0.400\ (32 + 2 \cdot 10) = 20.8\ \mu s.$$

8◀

TABLE 4
INSTRUCTION EXECUTION TIMES

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION*** SOURCE	DESTINATION
A	22	8	A	A
AB	22	8	B	B
ABS (MSB = 0)	16	4	A	–
(MSB = 1)	20	6	A	–
AI	22	8	–	–
ANDI	22	8	–	–
B	12	4	A	–
BL	18	6	A	–
BLWP	38	12	A	–
C	20	6	A	A
CB	20	6	B	B
CI	20	6	–	–
CKOF	14	2	–	–
CKON	14	2	–	–
CLR	16	6	A	–
COC	20	6	A	–
CZC	20	6	A	–
DEC	16	6	A	–
DECT	16	6	A	–
DIV (ST4 is set)	22	6	A	–
DIV (ST4 is reset)*	104-136	12	A	–
IDLE	14	2	–	–
INC	16	6	A	–
INCT	16	6	A	–
INV	16	6	A	–
Jump (PC is changed)	12	2	–	–
(PC is not changed)	10	2	–	–
LDCR (C = 0)	58	6	A	–
(1<C<8)	26+2C	6	B	–
(9<C<15)	26+2C	6	A	–
LI	18	6	–	–
LIMI	22	6	–	–
LREX	14	2	–	–
LWPI	14	4	–	–
MDV	22	8	A	A
MOVB	22	8	B	B
MPY	62	10	A	–
NEG	18	6	A	–
ORI	22	8	–	–
RSET	14	2	–	–
RTWP	22	8	–	–
S	22	8	A	A
SB	22	8	B	B
SBO	16	4	–	–
SBZ	16	4	–	–
SETO	16	6	A	–
Shift (C ≠ 0)	18+2C	6	–	–
(C ≠0, Bits 12–15 of WRO ≠ 0)	60	B	–	–
(C = 0, Bits 12–15 of WRP = N ≠ 0)	28+2N	8	–	–
SOC	22	8	A	A
SOCB	22	8	B	B
STCR (C = 0)	68	8	A	–
(1≤C≤7)	50	8	B	–
(C = 8)	52	8	B	–
(9≤C≤15)	66	8	A	–

*Execution time is dependent upon the partial quotient after each clock cycle during execution.

***The letters A and B refer to the respective tables that follow.

TABLE 4 (CONTINUED)

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION*** SOURCE	DESTINATION
STST	12	4	–	–
STWP	12	4	–	–
SWPB	16	6	A	–
SZC	22	8	A	A
SZC8	22	8	B	B
TB	16	4	–	–
X**	12	4	A	–
XOP	52	16	A	–
XOR	22	8	A	–
RESET function	36	10	–	–
LOAD function	32	10	–	–
Interrupt context switch	32	10	–	–
Undefined op codes. 0000-01FF, 0320 033F, 0C00-0FFF, 0780-07FF	8	2	–	–

**Execution time is added to the execution time of the instruction located at the source address.
***The letters A and B refer to the respective tables that follow.

ADDRESS MODIFICATION – TABLE A

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	6	2
WR indirect auto-increment (T_S or T_D = 11)	12	4
Symbolic (T_S or T_D = 10, S or D = 0)	10	2
Indexed (T_S or T_D = 10, S or D ≠ 0)	12	4

ADDRESS MODIFICATION – TABLE B

ADDRESSING MODE	CLOCK CYCLES C	MEMORY ACCESSES M
WR (T_S or T_D = 00)	0	0
WR indirect (T_S or T_D = 01)	6	2
WR indirect auto-increment (T_S or T_D = 11)	10	4
Symbolic (T_S or T_D = 10, S or D = 0)	10	2
Indexed (T_S or T_D = 10, S or D ≠ 0)	12	4

8◀

4. TMS 9980A/TMS 9981 ELECTRICAL SPECIFICATIONS

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

Supply voltage, V_{CC} (see Note 1)	−0.3 to 15 V
Supply voltage, V_{DD} (see Note 1)	−0.3 to 15 V
Supply voltage, V_{BB} (see Note 1) (9980A only)	−5.25 to 0 V
All input voltages (see Note 1)	−0.3 to 15 V
Output voltage (see Note 1)	−2 V to 7 V
Continuous power dissipation	1.4 W
Operating free-air temperature range	0°C to 70°C
Storage temperature range	−55°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.

NOTE 1: Under absolute maximum ratings voltage values are with respect to V_{SS}.

4.2 RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{BB} (9980A only)	−5.25	−5	−4.75	V
Supply voltage, V_{CC}	4.75	5	5.25	V
Supply voltage, V_{DD}	11.4	12	12.6	V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH}	2.2	2.4	V_{CC}+1	V
Low-level input voltage, V_{IL}	−1	0.4	0.8	V
Operating free-air temperature, T_A	0	20	70	°C

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

PARAMETER			TEST CONDITIONS	MIN	TYP*	MAX	UNIT
I_I	Input current	[illegible]	$V_I = V_{SS}$ to V_{CC}			±75	μA
		[illegible] during HOLDA	$V_I = V_{SS}$ to V_{CC}			±75	
		Any other inputs	$V_I = V_{SS}$ to V_{CC}			±10	
V_{OH}	High-level output voltage		I_O = −0.4 mA	2.4			V
V_{OL}	Low-level output voltage		I_O = 2 mA			0.5	V
			I_O = 3.2 mA			0.65	
I_{BB}	Supply current from V_{BB} (9980A Only)					1	mA
I_{CC}	Supply current from V_{CC}		0°C		50	60	mA
			70°C		40	50	
I_{DD}	Supply current from V_{DD}		0°C		70	80	mA
			70°C		65	75	
C_I	Input capacitance (any inputs except data bus)		f = 1 MHz, unmeasured pins at V_{SS}		15		pF
C_{DB}	Data bus capacitance		f = 1 MHz, unmeasured pins at V_{SS}		25		pF
C_O	Output capacitance (any output except data bus)		f = 1 MHz, unmeasured pins at V_{SS}		15		pF

*All typical values are at T_A = 25°C and nominal voltages

▶8

4.4 CLOCK CHARACTERISTICS

The TMS 9980A and TMS 9981 have an internal 4-phase clock generator/driver. This is driven by an external TTL compatible signal to control the phase generation. In addition, the TMS 9981 provides an output (OSCOUT) that in conjunction with CKIN forms an on-chip crystal oscillator. This oscillator requires an external crystal and two capacitors as shown in Figure 13. The external signal or crystal must be 4 times the desired system frequency.

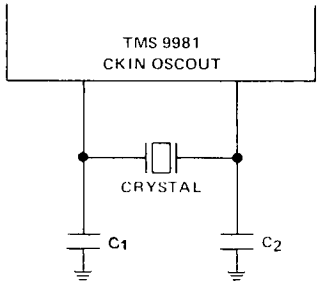


FIGURE 13 – CRYSTAL OSCILATOR CIRCUIT

4.4.1 Internal Crystal Oscillator (9981 Only)

The internal crystal oscillator is used as shown in Figure 13. The crystal should be a fundamental series resonant type C_1 and C_2 represent the total capacitance on these pins including strays and parasitics.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Crystal frequency	0°C-70°C	6		10	MHZ
C_1, C_2	0°C-70°C	10	15	25	pf

4.4.2 External Clock

The external clock on the TMS 9980A and optional on the TMS 9981, uses the CKIN pin. In this mode the OSCOUT pin of the TMS 9981 must be left floating. The external clock source must conform to the following specifications.

PARAMETER		MIN	TYP	MAX	UNIT
f_{ext}	External source frequency*	6		10	MHz
V_H	External source high level	2.2			V
V_L	External source low level			0.8	V
T_r/T_f	External source rise/fall time		10		ns
T_{WH}	External source high level pulse width	40			ns
T_{WL}	External source low level pulse width	40			ns

*This allows a system speed of 1.5 MHz to 2 MHz.

8

4.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

The timing of all the inputs and outputs are controlled by the internal 4 phase clock; thus all timings are based on the width of one phase of the internal clock. This is $1/f_{(CKIN)}$ (whether driven or from a crystal). This is also $\frac{1}{4}f_{system}$. In the following table this phase time is denoted t_w.

All external signals are with reference to $\phi 3$ (see Figure 14).

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_r(\phi 3)$	Rise time of $\phi 3$	$t_w = 1/f(CKIN) = \frac{1}{4}f_{system}$ C_L = 200pf	3	5	10	ns
$t_f(\phi 3)$	Fall time of $\phi 3$		5	7.5	15	ns
$t_w(\phi 3)$	Pulse width of $\phi 3$		t_w-15	t_w-10	t_w+10	ns
t_{su}	Data or control setup time*		t_w-30			ns
t_h	Data hold time*		$2t_w+10$			ns
$t_{PHL}(\overline{WE})$	Propagation delay time WE high to low		t_w-10	t_w	t_w+20	ns
t_{PLH} ∴	Propagation delay time WE low to high		t_w	t_w+10	t_w+30	ns
t_{PHL}(CRUCLK)	Propagation delay time, CRUCLK high to low		−20	−10	+10	ns
t_{PLH}(CRUCLK)	Propagation delay time, CRUCLK low to high		$2t_w-10$	$2t_w$	$2t_w+20$	ns
t_{OV}	Delay time from output valid to $\phi 3$ low		t_w-50	t_w-30		ns
t_{OX}	Delay time from output invalid to $\phi 3$ low			t_w-20	t_w	ns

*All inputs except IC0-IC2 must be synchronized to meet these requirements. IC0-IC2 may change asynchronously. See section 2.10.4.

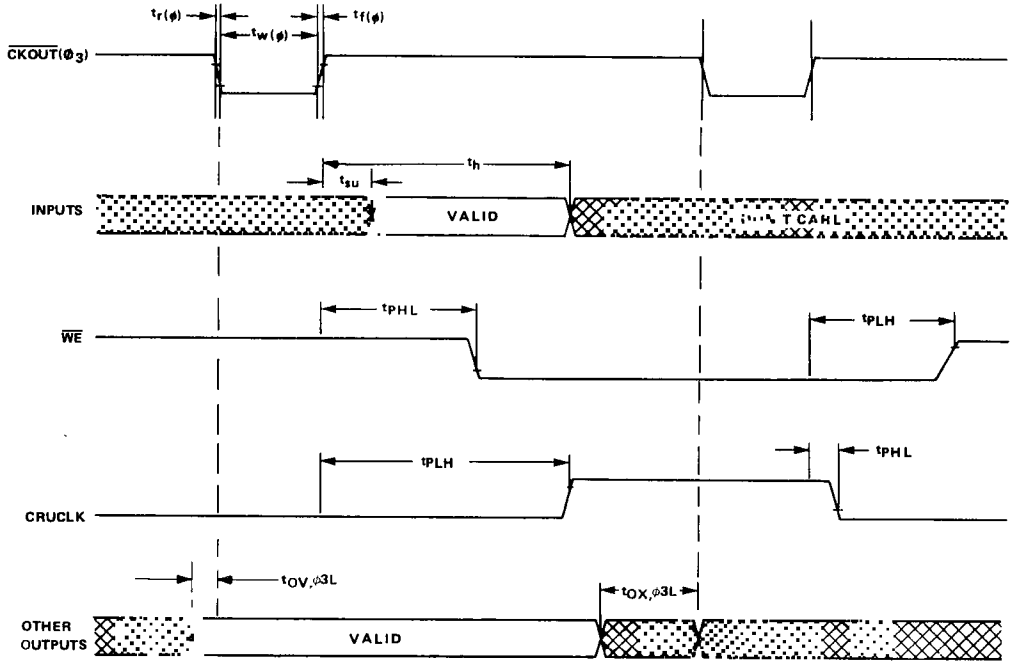


FIGURE 14 – EXTERNAL SIGNAL TIMING DIAGRAM

INTRODUCTION

DESCRIPTION

The TMS 9940 is a single-chip, 16-bit microcomputer containing a CPU, memory (RAM and EPROM/ROM), and extensive I/O. Except for four instructions that do not apply to the TMS 9940 microcomputer configuration, the TMS 9940 instruction set matches that of the TMS 9900 and includes capabilities offered by minicomputers. In addition, the TMS 9940 instruction set includes two instructions that facilitate manipulation of binary coded decimal (BCD) data, and a single-word load-interrupt-mask (LIIM) instruction.

The unique memory-to-memory architecture features multiple register files, resident in the RAM, which allow faster response to interrupts and increased programming flexibility. The memory consists of 128 bytes of RAM and 2048 bytes of EPROM/ROM. The TMS 9940 implements four levels of interrupts, including an internal decrementer which can be programmed as a timer or an event counter. All members of the TMS 9900 family of peripheral circuits are compatible with the TMS 9940. The TMS 9940 is fully supported by software and hardware development systems. The TMS 9940 is fully supported by factory applications engineers and technical answering services.

KEY FEATURES

- 16-bit instruction word;
- Minicomputer instruction set including multiply and divide;
- 2048 bytes of EPROM (TMS 9940E)/ROM (TMS 9940M) on chip;
- 128 bytes of RAM on chip;
- 16 general purpose registers;
- 4 prioritized interrupts;
- On-chip timer/event counter;
- 32 bits general purpose I/O Ports;
- 256 bits I/O expansion;
- Easy test function ;
- Multiprocessor system interface;
- Power down capability for low stand-by power;
- Five speed ranges for maximum performance;
- N-channel silicon gate MOS, 5 volt power supply;
- An EPROM device, the TMS 9940E, is contained in a 40-pin, 600-mil, dual-in-line ceramic package with quartz lid;
- A mask ROM device, the TMS 9940M, is contained in a 40-pin, 600-mil, dual-in-line plastic or ceramic package.

PARTS IDENTIFICATION

EPROM DEVICE	MASK-ROM DEVICE	OSCILLATOR FREQUENCY MHz (NOM)
TMS 9940E	TMS 9940M	5
TMS 9940E-40	TMS 9940M-40	4
TMS 9940E-30	TMS 9940M-30	3
TMS 9940E-20	TMS 9940M-20	2
TMS 9940E-10	TMS 9940M-10	1

NOTE: An additional MPXXXX number is used to identify custom ROM codes for TMS 9940M devices.

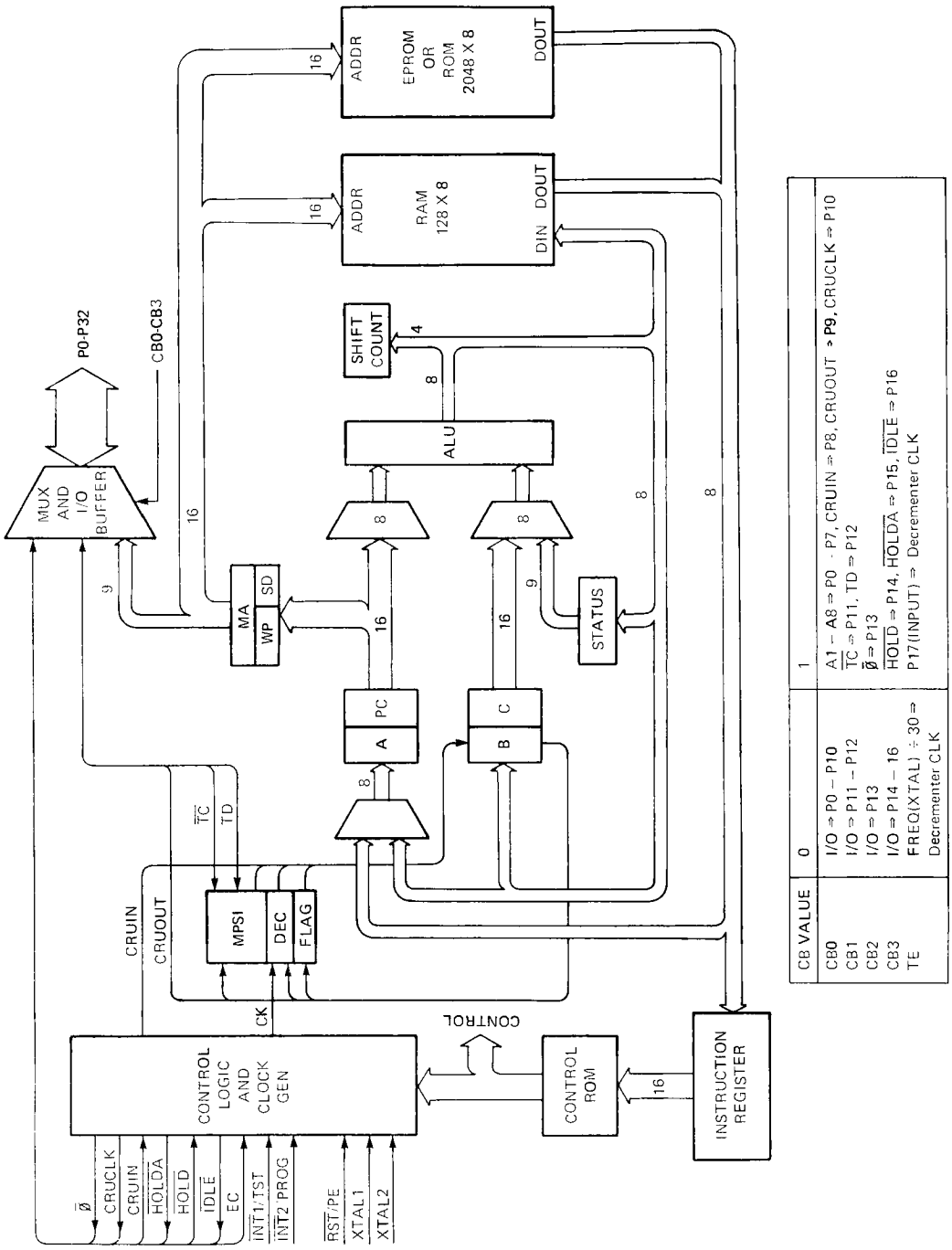
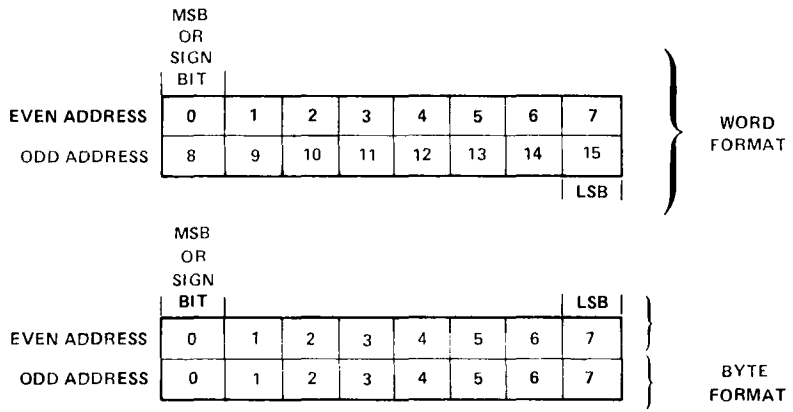


Figure 1. TMS 9940 Architecture.

ARCHITECTURE

Memory for the TMS 9940 is organized in 8-bit bytes. The processors are nevertheless 16-bit processors requiring two memory accesses for each 16-bit word. A word is defined as 16 bits or two consecutive bytes in memory. The words are restricted to be on even address boundaries, i.e., the most significant half (8 bits) resides at even address and the least significant half resides at the subsequent odd address. A byte can reside at even or odd addresses. The word and byte formats are shown below.



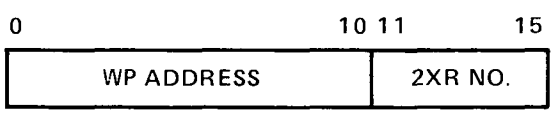
Registers and Memory

The TMS 9940 employs an advanced memory-to-memory architecture where blocks of memory designated as workspaces replace dedicated hardware registers with program-data registers. The TMS 9940 memory map is shown in *Figure 2.* The 2k x 8 EPROM/ROM is assigned memory addresses 0000_{16} through $07FF_{16}$, and the 128 x 8 RAM is assigned memory addresses 8300_{16} through $837F_{16}$.

The first eight words in the EPROM/ROM (addresses 0000_{16} through $000F_{16}$) are used for the interrupt vectors, and 24 words (addresses 0050_{16} through $007F_{16}$) are used for the extended operation (XOP) instruction trap vectors. The remaining memory is available for programs, data, and workspace registers. If desired, any of the special areas may also be used as general EPROM/ROM memory.

Three machine registers are accessible to the user. The 15-bit program counter (PC) contains the address of the instruction following the current instruction being executed. This address is referenced by the processor to fetch the next instruction from memory and is then automatically incremented. The 16-bit status register (ST) contains the present state of the processor. The 11-bit workspace register (WP) points to the first word in the currently active set of workspace registers.

The workspace-register files are nonoverlapping and contain 16 contiguous memory words. Each workspace register may hold data or an address, and function as an operand register, accumulator, address register, or index register. During instruction execution, the processor addresses any register in the workspace by concatenating the 11-bit WP value (bits 0 to 10) with two times the specified register number (bits 11 to 15) as shown below. WP addresses in RAM will be one of four values: 8300_{16}, 8320_{16}, 8340_{16}, and 8360_{16}.



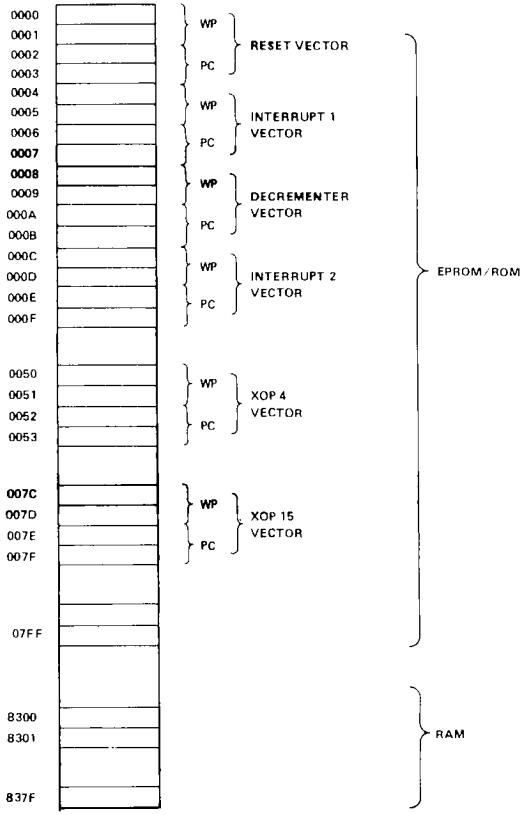
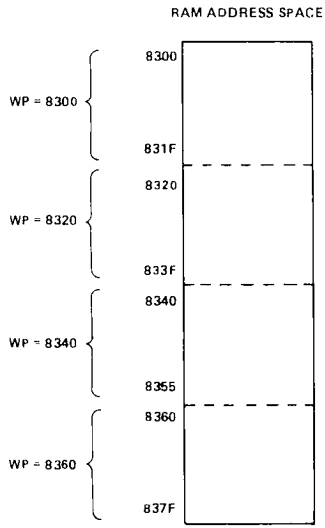


Figure 2. TMS 9940 Memory Map

Up to four nonoverlapping workspaces can be defined in the RAM. The relationship between the workspace pointer value and its corresponding workspace are shown below:



The workspace concept is particularly valuable during operations that require a context switch, which is a change from one program environment to another (as in the case of an interrupt or call to a subroutine). Such an operation, using a conventional multi-register arrangement, requires that at least part of the contents of the register file be stored and reloaded. The TMS 9940, however, can accomplish a complete context switch simply by exchanging the values in the PC, ST, and WP. Instructions in the TMS 9940 that result in a context switch include:

1. Branch and Load Workspace Pointer (BLWP);
2. Return from Subroutine (RTWP);
3. Extended Operation (XOP).

$\overline{\text{RESET}}$, the decrementer interrupt, and the external device interrupts ($\overline{\text{INT}}1$ and $\overline{\text{INT}}2$) also cause a context switch by forcing the processor to trap to a service subroutine.

INTERRUPTS

The TMS9940 implements four hardware interrupt levels. The highest priority interrupt level (level 0) is reserved for the $\overline{\text{RESET}}$ function followed by a user defined external interrupt $\overline{\text{INT}}1$ (level 1), the decrementer (level 2), and the second user defined external interrupt $\overline{\text{INT}}2$ (level 3). The $\overline{\text{RESET}}$ function will be accepted whenever it goes active (e.g., in the middle of an instruction), whereas all other levels are accepted at the end of the presently executing instruction.

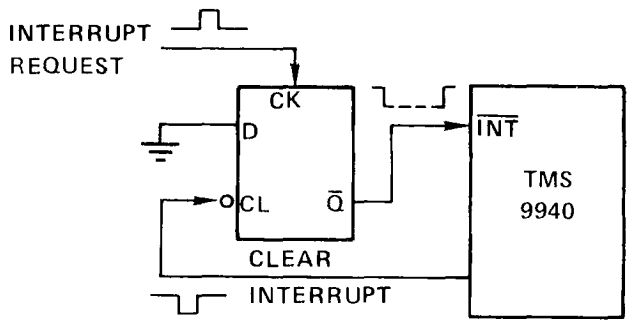
The TMS9940 external interrupt interface consists of three discrete input lines ($\overline{\text{RESET}}$, $\overline{\text{INT}}1$, $\overline{\text{INT}}2$). The input levels are standard TTL levels and the signals require no external synchronization.

The TMS9940 continuously compares the value of the highest priority active interrupt level with the interrupt mask contained in status register bits 14 and 15. When the level of the pending interrupt is less than or equal to the enabling mask value (higher or equal priority interrupt), the processor recognizes the interrupt and initiates a context switch. The processor fetches the new context WP and PC from the interrupt vector locations and stores the previous context WP into R13, PC into R14, and ST into R15, of the new workspace. The interrupt mask is loaded with a value that is one less than the interrupt level being serviced (NOTE: $\overline{\text{RESET}}$ forces the mask value to zero) so that only higher priority interrupts will be recognized during the service routine. The processor also inhibits interrupts until the first instruction of the service routine has been executed to preserve program linkage should a higher priority interrupt occur.

$\overline{\text{RESET}}$ must be held active for a minimum of five clock cycles to guarantee recognition. When $\overline{\text{RESET}}$ is removed the status register and configuration word are set to zero and a level zero interrupt is initiated.

The decrementer interrupt is discussed in detail in a later section. If the decrementer is programmed as an external event counter with a start value of 1, P17/EC will function as a positive edge-triggered interrupt input.

External device interrupt requests are priority level sensitive and, if masked out, must remain active until recognized by the processor executing in the device service routine. The individual service routines must reset the interrupt mask and request before the service routine is complete. A typical schematic to latch in an interrupt requests is shown below:



If a higher priority interrupt becomes active during a service routine, a second context switch occurs to service the higher priority interrupt. When that routine is complete, a return instruction (RTWP) restores the first service routine parameters to the processor to complete processing of the lower priority interrupt. All interrupt service routines should end with the return instruction to restore original program parameters. The interrupt-vector locations, device assignments, and enabling-mask values are shown on *Table 1.*

Table 1. Interrupt Level Data

Interrupt Level		Vector Location (Memory Address In Hex)	Device Assignment	Interrupt Mask Value To Enable Respective Interrupts ST14 & ST15
(Highest Priority)	0	0000	Reset	0 through 3*
	1	0004	External Device	1 through 3
	2	0008	Decrementer	2 and 3
(Lowest Priority)	3	000C	External Device	3 only

*Level 0 cannot be disabled

As shown in *Table 3,* the positive-logic value of the interrupt pins may be read by CRU instructions even though the interrupt may be masked.

INPUT/OUTPUT

The TMS9940 has a communications register unit (CRU) drive I/O interface. The I/O features implemented on a single 32-bit channel (P0 to P31) are:

- General Purpose I/O: The 32-bits (P0 to P31) of individually controlled I/O data.
- I/O Expansion Via CRU: 256 user configured external I/O bits (P0 to P10).
- Multiprocessor System Interface (P11 and P12): A register for passing commands/data between processors.
- External event counter (P17).
- Power Down

The system engineer's flexibility in the TMS9940 applications is extended greatly by software I/O structuring. The key element is only four bits, called configuration bits (CB), contained in the configuration control register. This register holds the multiplexer control that selects optional modes for 17 of the 32 I/O terminals. The configuration bits controlling these modes are as follows:

Configuration Bit	*Function Controllers*
CB 0	CRU I/O Expansion
CB 1	Multiprocessor System Interface
CB 2	External synchronization (Clock output)
CB 3	Power down and hold Logic

The decrementer used as an event counter has its input available from P17 continuously and does not need a configuration bit to control it.

Generally, a dedicated control system will need only one configuration set up; however, the flexibility allows for multiple configurations dynamically changing for more I/O capacity.

The TMS9940 allows the user to configure part of the I/O pins as special functions for system applications. The configurable pins are shown in *Table 2.*

Table 2. Configuration Bit Effects

MODE	PIN NAME	PIN NO.	CONFIGURATION BIT (CB) NO.	CRU BIT (HEX)	EFFECT OF CB BIT 0	EFFECT OF CB BIT 1	I/O CB=1
CRU Expansion	P0/A1 (MSB)	23	0	183	P0	A1	OUT
	P1/A2	24	0	183	P1	A2	OUT
	P2/A3	25	0	183	P2	A3	OUT
	P3/A4	26	0	183	P3	A4	OUT
	P4/A5	27	0	183	P4	A5	OUT
	P5/A6	28	0	183	P5	A6	OUT
	P6/A7	29	0	183	P6	A7	OUT
	P7/A8	30	0	183	P7	A8	OUT
	P8/CRUIN	18	0	183	P8	CRUIN	IN
	P9/CRUOUT	17	0	183	P9	CRUOUT	OUT
	P10/CRUCLK	16	0	183	P10	CRUCLK	OUT
MPSI	P11/$\overline{TC}$	14	1	184	P11	$\overline{TC}$	I/O
	P12/TD	11	1	184	P12	TD	I/O
SYNC	P13/$\overline{\phi}$	15	2	185	P13	$\overline{\phi}$	OUT
Power Down & Hold	P14/$\overline{HLD}$	10	3	186	P14	$\overline{HLD}$	IN
	P15/$\overline{HLDA}$	9	3	186	P15	$\overline{HLDA}$	OUT
	P16/$\overline{IDLE}$	8	3	186	P16	$\overline{IDLE}$	OUT

Note: P17 is continuously available if the decrementer is used as an event counter.

That is, CB0 controls the I/O Expansion Channel, CB1 controls the MPSI, CB2 allows a clock output, and CB3 configures $\overline{HLD}$, $\overline{HLDA}$, and $\overline{IDLE}$ for power down. Application of $\overline{RESET}$ forces the configuration bits to zero, the all I/O line condition. The configuration can then be changed by outputting the desired bit value to the designated CRU address. (See *Table 3.*)

Communications Register Unit (CRU)

The CRU is a bit-oriented I/O interface through which both input and output bits can be directly addressed individually, or in fields of from 1 to 16 bits. The processor instructions that drive the CRU interface can set, reset, or test any bit in the CRU array or move between memory and CRU data fields. The CRU bit address assignments for all I/O and dedicated functions are shown in *Table 3.*

CRU instructions that manipulate external data are the only instructions which send CRUCLK pulses out of terminal 16.

Table 3. CRU Bit Address Assignments

CRU Bit Address	CRU Read Data	CRU Write Data
000	I/O Expansion	I/O Expansion
↓	↓	↓
0FF	I/O Expansion	I/O Expansion
180	INT1	—
181	Decrementer Interrupt	Clear Decrementer Interrupt
182	INT2	—
183	—	Configuration Bit 0
184	—	Configuration Bit 1
185	—	Configuration Bit 2
186	—	Configuration Bit 3
190	Decrementer (LSB)	Decrementer (LSB)
↓	↓	↓
19D	Decrementer (MSB)	Decrementer (MSB)
19E	—	T/C (See Decrementer) 1 = Timer, 0 = Counter
1A0	MPSI (LSB)	MPSI (LSB)
↓	↓	↓
1AF	MPSI (MSB)	MPSI (MSB)
1B0	FLAG 0	FLAG 0
↓	↓	↓
1BF	FLAG F	FLAG F
1C0	—	P0 Direction (1 = OUT, 0 = IN)
↓		↓
1DF	—	P31 Direction (1 = OUT, 0 = IN)
1E0	P0 DATA	P0 DATA
↓	↓	↓
1FF	P31 DATA	P31 DATA

Note: CRU addresses not listed above are not usable.

Single-Bit CRU Operations

The TMS9940 performs three single-bit CRU functions: test bit (TB), set bit to one (SBO), and set bit to zero (SBZ). To identify the bit to be operated upon, the TMS9940 develops a CRU-bit address and places it on the address bus.

For the two output operations (SBO and SBZ) the processor also generates a CRUCLK pulse, indicating an output operation to the CRU device, and places bit 7 of the instruction word on the CRUOUT line to accomplish the specified operation (bit 7 is a one for SBO and a zero for SBZ). A test-bit instruction transfers the addressed CRU bit from the CRUIN input line to bit 2 of the status register (EQUAL).

The TMS 9940 develops a hardware base address for the single-bit operations from the software base address contained in workspace register 12 and the signed displacement count contained in bits 8 through 15 of the instruction. The displacement allows two's complement addressing from base minus 128 bits through base plus 127 bits. The hardware base address (bits 6 through 14 of WR12) is added to the signed displacement specified in the instruction, and the result is loaded onto the address bus. *Figure 3* illustrates the development of a single-bit CRU address for the SBO, SBZ, and TB instruction.

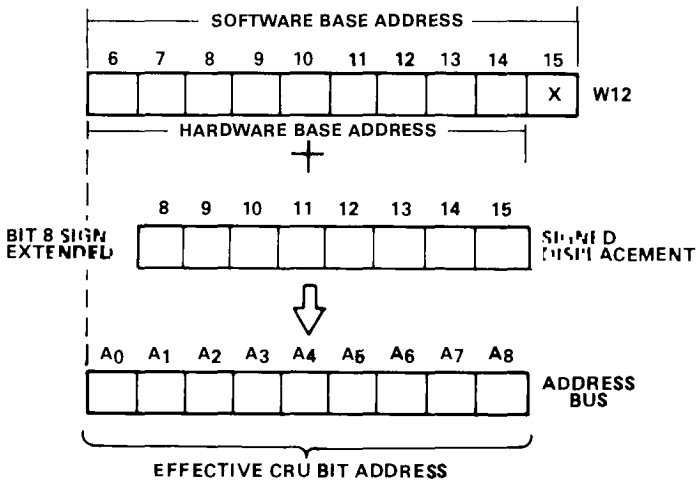


***Figure** 3. TMS9940 Single-Bit CRU Address Development*

Multiple-Bit CRU Operations

The TMS9940 performs two multiple-bit CRU operations: store communications register (STCR) and load communications register (LDCR). Both operations perform a data transfer from the CRU-to-memory or from memory-to-CRU as illustrated in *Figure 4.* Although the figure illustrates a full 16-bit transfer operation, any number of bits from 1 through 16 may be involved. The LDCR instruction fetches a word from memory and right-shifts it to transfer it serially to CRU output bits. If the LDCR involves eight or fewer bits, those bits come from the right-justified field within the addressed byte of the memory word. If the LDCR involves nine or more bits, those bits come from the right-justified field within the whole memory word. When transferred to the CRU interface, each successive bit receives an address that is sequentially greater than the address for the previous bit. This addressing mechanism results in an order reversal of the bits; this is, bit 15 of the memory word (or bit 7) becomes the lowest addressed bit in the CRU, and bit 0 becomes the highest bit in the CRU field.

An STCR instruction transfers data from the CRU to memory. If the operation involves a byte or less transfer, the transferred data will be stored right-justified in the memory byte with leading bits set to zero. if the operation involves from 9 to 16 bits, the transferred data is stored right-justified in the memory word with leading bits set to zero.

When the input from the CRU device is complete, the first bit from the CRU is in the least-significant-bit position in the memory word or byte.

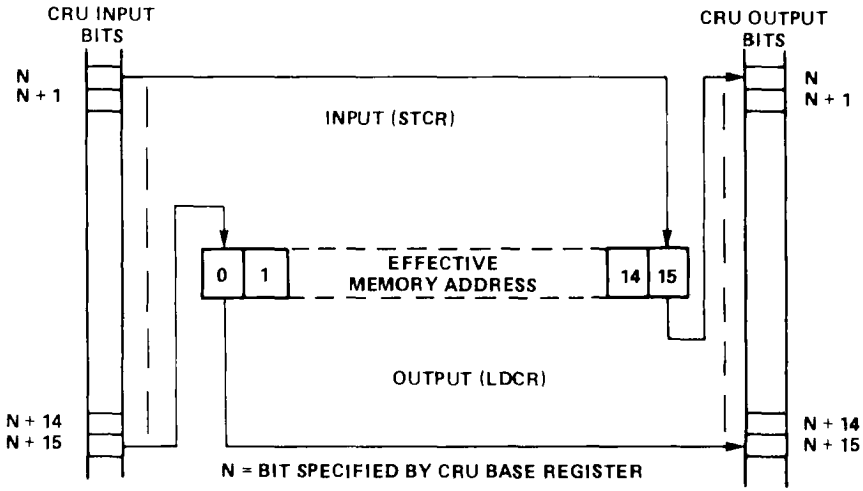
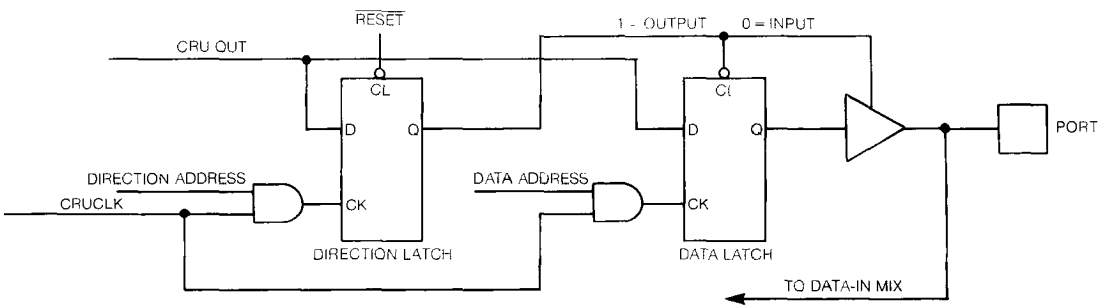


Figure 4. TMS9940 LDCR/STCR Data Transfers

General Purpose I/O

The TMS9940 contains 32 I/O pins which can be used as individually controlled I/O lines with each line independently programmed as an input or output. $\overline{\text{RESET}}$ forces all I/O lines to the input mode until programmed, by an I/O command. Once programmed, the line will stay in the designated state until it is reprogrammed or $\overline{\text{RESET}}$ again becomes active. Reading a line will input data present on the pin without affecting its direction. The lines can be accessed individually by the single bit CRU instructions (SBO, SBZ, TB) or in groups of 1 to 16 by the multiple bit CRU instructions (STCR, LDCR). The I/O data and direction bits are accessed through dedicated bit addresses as shown in *Table 3.*

When an I/O port is programmed to be an input, the previous output data is reset to zero. Thus the data direction and configuration bits should be set first, then the desired output data is set by the appropriate CRU instruction. The equivalent logic for the output control of ports 17 to 31 is shown below.



I/O Expansion

The TMS9940 allows direct I/O expansion for up to 256 bits by use of a standard 9900 family CRU interface. I/O lines P0-P10 can be configured as an 8-bit address bus (A1-A8), CRUIN, CRUOUT, and CRUCLK to interface to any CRU based peripheral. (See the configuration section for details.)

Figure 5 illustrates how to implement a system containing a TMS9901 programmable system interface, a TMS9902 asynchronous communications controller, and a TMS9903 synchronous communications controller.

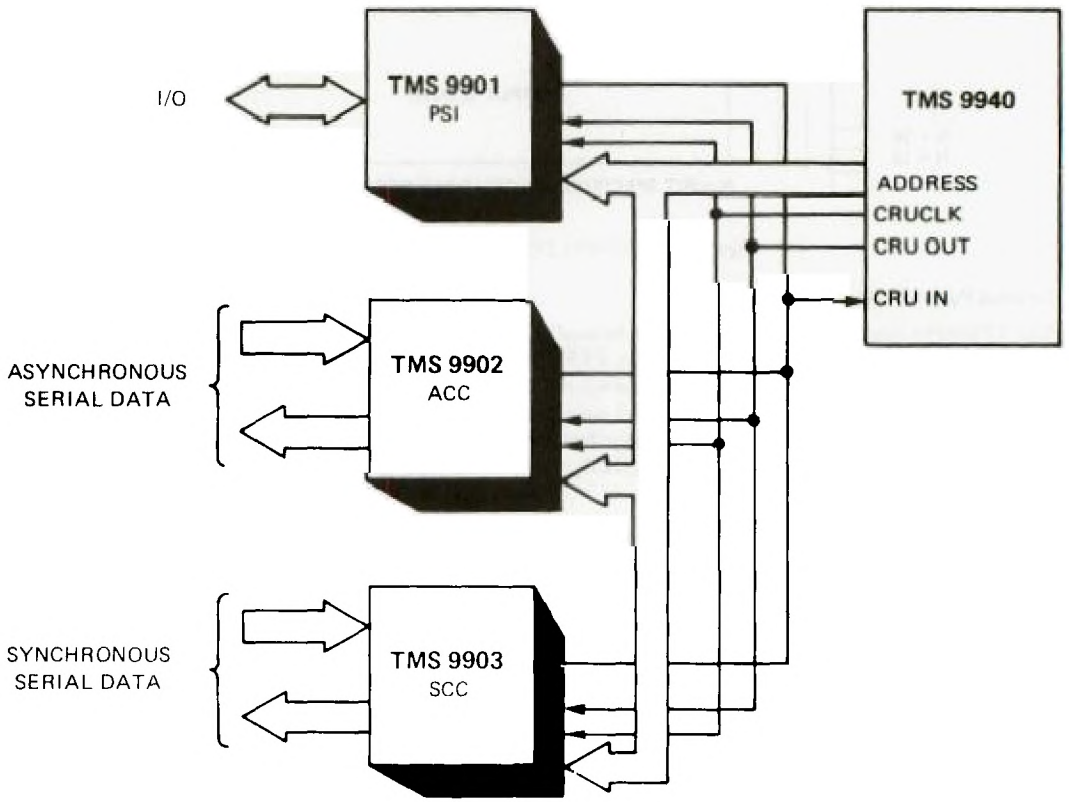
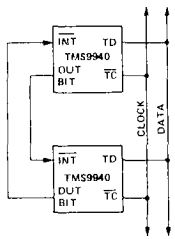


Figure 5. TMS 9940 Input/Output Expansion Interface

Multiprocessor System Interface (MPSI)

The MPSI is a two-wire interface for transferring data in a multiple processor system. Since the TMS9940 can execute instructions out of its RAM, the MPSI allows the capability of efficiently downloading instruction sequences which can then be executed. Thus, multiple processor systems can reconfigure themselves in system applications. The MPSI can also be used to transfer data to be operated on, such as in a master-slave situation with the master distributing tasks to the slaves.



For multiple TMS9940 systems the MPSI is connected as shown. Additional CPU's can be connected simply by "wire ORing" to the MPSI signals.

A block diagram of the internal MPSI logic is shown in *Figure 6.*

The protocol of the system is such that all devices are "receivers" except when actually transmitting data (the "sender" mode). The TD input signal feeds a 16-bit shift register that is clocked by the TC input to allow 16 bits of data to be shifted into the shift register completely transparent to the rest of the CPU operation. After the data has been sent, the "sender" interrupts the "receiver" (through a normal interrupt input) so that the "receiver" can execute an STCR instruction to input its MPSI data from its dedicated MPSI CRU bit addresses (see *Table 3*). As needed, the "receiver" can then interrupt the "sender" to acknowledge receipt and/or request new data.

To become a "sender" the TMS9940 executes an LDCR instruction to the dedicated MPSI CRU addresses. Automatically, the TD signal switches to the output mode to send data, and the TC signal sends out the CRUCLK strobe. After completion of the instruction, TD and TC again revert to the input mode to switch the device back to "receiver" status.

The MPSI is compatible with the standard 9900 family CRU interface. An example illustrating the TMS9940 and TMS9900 communicating through the MPSI is shown in *Figure 7.*

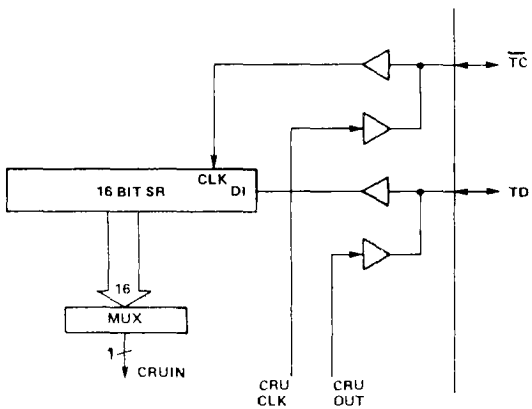
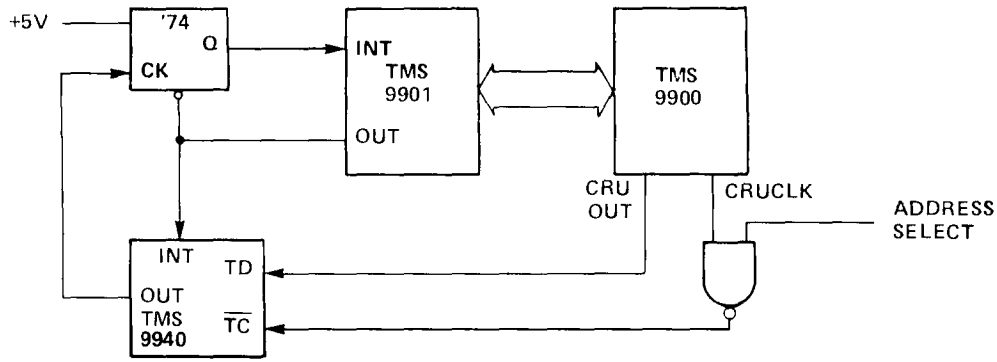
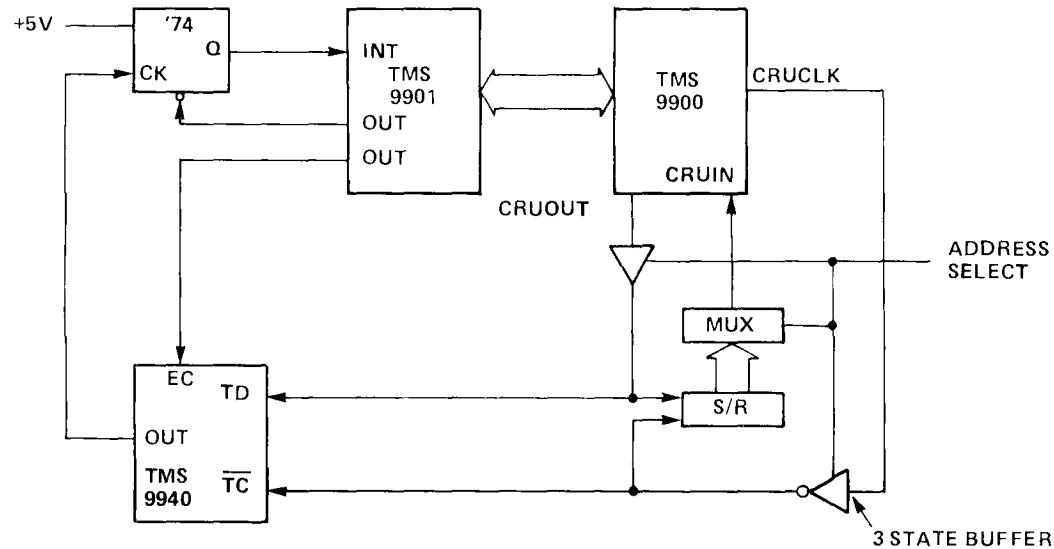


Figure 6. MPSI Block Diagram



(A) ONE-WAY COMMUNICATION; TMS 9900 DOWNLOADS TO TMS 9940

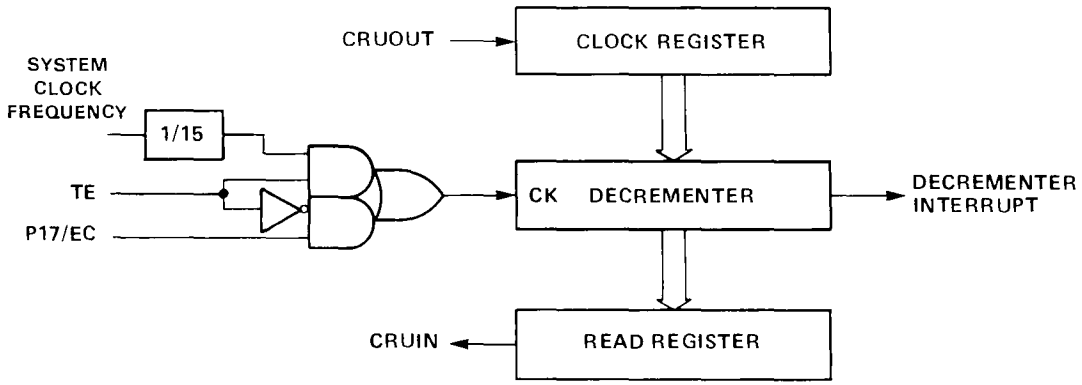


(B) TWO-WAY COMMUNICATION

Figure 7. TMS 9900 and TMS9940 Communication through MPSI

Decrementer (Timer/Event Counter)

The TMS9940 contains a 14-bit decrementing register which can function as a programmable real-time clock, an event timer, or an external event counter. A block diagram of the timer/counter is shown below.



When $\overline{\text{RESET}}$ is active, a zero value is forced into the clock register to disable the decrementer. Writing a non-zero value into the clock register through the dedicated CRU bit addresses (bits 190_{16} to $19D_{16}$ as shown in *Table 3)* enables the decrementer to start at the programmed value, count down to zero at a rate equal to system oscillator x 1/30, issue an interrupt, and restart at the programmed value. The interrupt is then automatically cleared by the interrupt context switch.

The decrementer is programmed to function as a timer or event counter by a dedicated Timer Enable CRU bit, TE (see *Table 3).* Writing a one (1) into TE will program the decrementer as a timer, and a zero will program the decrementer as an event counter.

When programmed as a timer, the decrementer can function as an interval timer simply by loading the proper start value in CRU bits 190_{16} to $19D_{16}$. The decrementer will then issue interrupts at the chosen interval.

The decrementer can also be used as an event timer when programmed as a timer by reading the timer values through the dedicated CRU bit addresses at the start and stop points of the event of interest and comparing the two values. The difference will be a direct measurement of the elapsed time.

When programmed as an event counter, the decrementer functions as above except that pin P17/EC is the clock input instead of the system clock. A positive edge transition on P17/EC will decrement the count. When the count reaches zero, the decrementer is reloaded with the programmed start value and an interrupt is issued. Note that P17/EC can function as a positive edge-triggered interrupt by loading a start value of one.

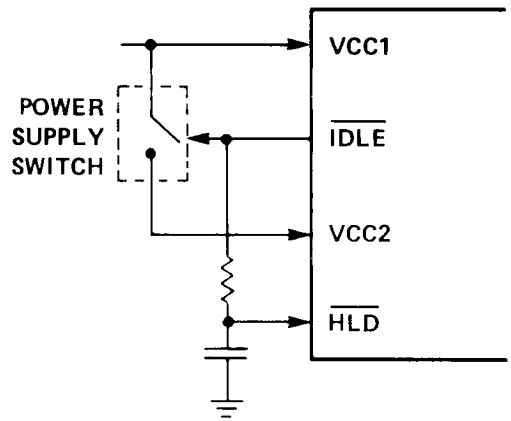
8◀

Flag Register

The TMS9940 incorporates a 16-bit flag register internally. Each of the bits is under program control and can be SET, RESET, and TESTED. The bits are accessed through dedicated CRU bit addresses and utilize the CRU instructions (LDCR, STCR, TB, SBO, SBZ) or control. The CRU bit addresses assignments for the flag register are shown in *Table 3.*

Power Down

Applications which have low duty cycles (for example, those which are human interactive) and/or require low power dissipation, can make use of the power down capability to lower average power. The TMS9940 is powered by two separate power supplies: (1) V_{CC1}, which powers the RAM and interrupt logic, and (2) V_{CC2}, which powers the rest of the circuitry. A diagram showing the way to connect power to use the power down feature is shown below.



In the above circuit when the IDLE instruction is executed, a low value will be output on $\overline{\text{IDLE}}$ to open the power supply switch. Inputting an interrupt into the CPU will force the processor out of IDLE and drive $\overline{\text{IDLE}}$ HIGH, which will close the switch and power up the rest of the circuitry. The HLD input is a Schmitt Trigger input which will keep the CPU stopped until V_{CC2} has settled. The particular values of R and C chosen are system dependent.

To use the power down feature configuration bit (CB) 3 must be set to a 1 to enable (low) $\overline{\text{HLD}}$, $\overline{\text{HLDA}}$, and $\overline{\text{IDLE}}$. Execution of the IDLE instruction will disable the decrementer interrupt (level 2) and $\overline{\text{INT2}}$ (level 3) and the processor can be powered down with the circuit shown in the figure. External lows to either $\overline{\text{RST}}$ or $\overline{\text{INT1}}$ can be used to force the CPU out of the IDLE state; however, if the processor was powered down, $\overline{\text{INT1}}$ must be used to maintain RAM data integrity as the RAM write latches temporarily float during $\overline{\text{RESET}}$. If decrementer CB is equal to a zero when IDLE is executed, the decrementer interrupt and $\overline{\text{INT2}}$ are not disabled and the CPU cannot be powered down.

Hold and Hold Acknowledge

Multiple processor operation may require temporary suspension of operation of one microcomputer (e.g., where both microcomputers access common devices through the general purpose I/O lines). This could entail a "master/slave" situation. One microcomputer (master) can place the other (slave) on hold by activating the slaves' $\overline{\text{HLD}}$ line. When in the hold state, the slave issues $\overline{\text{HLDA}}$. When the master deactivates the slaves' $\overline{\text{HLD}}$ line, the slave leaves the hold state. Configuration bit 3 must be a one at the slave so that its $\overline{\text{HLD}}$ pin will be active.

Synchronization Mode ($\overline{\phi}$)

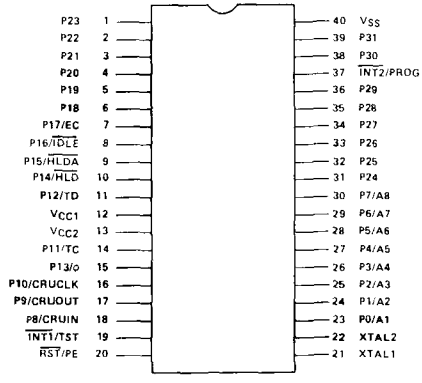
A clock output for use with external hardware is available on terminal 15, P13/$\overline{\phi}$. When configured in the sync mode (see *Table 2*), P13/$\overline{\phi}$ sends out the internal clock that is half of the oscillator frequency.

TMS9940 Terminal Assignments

Table 4 defines the TMS9940 pin assignments and describes the function of each pin.

Table 4. *TMS9940 Pin Assignments and Functions*

SIGNATURE	PIN	I/O	DESCRIPTION
XTAL1	21	IN	Crystal input pin for control of internal oscillator
XTAL2	22	IN	Crystal input pin for control of internal oscillator. Also input pin for external oscillator
V_{CC1}	12		Supply voltage (+5 V) The internal RAM and interrupt logic are powered by this supply
V_{CC2}	13		Supply voltage (+5V). All logic except the RAM and interrupt logic are powered by this supply
V_{SS}	40		Ground reference.
$\overline{RST}$/PE	20	IN	$\overline{RESET}$/Program Enable. When active low (Schmitt Trigger Input, V_{IL}) the $\overline{RESET}$ sequence is initiated. $\overline{RESET}$ must be held active for a minimum of five clock cycles. When active high (V_{IP}) the EPROM programming function is enabled. (See EROM Programming Section for detailed description)
$\overline{INT1}$/TST	19	IN	$\overline{Interrupt\ 1}$/TEST When active low (V_{IL}) external device interrupt 1 is active When active high (V_{IP}) the device is switched into the test mode (see TEST FUNCTION Section for detailed description).
$\overline{INT2}$/PROG	37	IN	$\overline{Interrupt\ 2}$/PROGRAM PULSE. When active low (V_{IL}) and $\overline{RST}$/PE is not active high, external device interrupt 2 is active. When $\overline{RST}$/PE is active high (V_{IP}), $\overline{INT2}$/PROG becomes the programming pulse input for EPROM programming. (See Programming Section for description.)
P0/A1	23	I/O	General Purpose I/O lines. P0-P7 can also be configured as the address bus (A1 is MSB) of the I/O expansion channel (see Configuration Section for details).
P1/A2	24		
P2/A3	25		
P3/A4	26		
P4/A5	27		
P5/A6	28		
P6/A7	29		
P7/A8	30		



8◀

Table 4. TMS 9940 Pin Assignments and Functions (Continued)

SIGNATURE	PIN	I/O	DESCRIPTION
P8/CRUIN	18	I/O	General Purpose I/O Line. P8 can also be configured as the CRUIN data input signal for the I/O expansion channel (see I/O Section for configuration details).
P9/CRUOUT	17	I/O	General Purpose I/O Line. P9 can also be configured as the CRUOUT data output signal for the I/O expansion channel (see I/O Section for configuration details).
P10/CRUCLK	16	I/O	General Purpose I/O Line. P10 can also be configured as the CRUCLK data strobe output signal for the I/O expansion channel (see I/O Section for configuration details).
P11/$\overline{TC}$	14	I/O	General Purpose I/O Line. P11 can also be configured as the transfer clock for the Multiprocessor System Interface (see I/O Section for configuration details).
P12/TD	11	I/O	General Purpose I/D Line. P12 can also be configured as the transfer data signal for the Multiprocessor System Interface (see I/O Section for configuration details).
P13/$\overline{\phi}$	15	I/O	General Purpose I/O Line. P13 can also be configured as a clock output signal (see I/O Section for configuration details).
P14/$\overline{HLD}$	10	I/O	General Purpose I/O Line. P14 can also be configured as the $\overline{HOLD}$ (active low) Schmitt Trigger input to force the processor to stop until $\overline{HOLD}$ returns to the inactive state. (See I/O Section for configuration details.)
P15/$\overline{HLDA}$	9	I/O	General Purpose I/O Line. P15 can also be configured as the Hold Acknowledge output (active low). When the processor enters the HOLD state, $\overline{HOLDA}$ becomes active (See I/O Section for configuration details.)
P16/$\overline{IDLE}$	8	I/O	General Purpose I/O. P16 can also be configured as the $\overline{IDLE}$ output signal (active low) for power down. (5ee I/O Section for configuration details.)
P17/EC	7	I/O	General Purpose I/O Line. P17 can also be programmed as the event counter input. The decrementer will decrement on each positive transition of EC. (See Decrementer Section for programming details.)
P18	6	I/O	General Purpose I/O Line
P19	5	I/O	General Purpose I/O Line
P20	4	I/O	General Purpose I/O Line
P21	3	I/O	General Purpose I/O Line
P22	2	I/O	General Purpose I/O Line
P23	1	I/O	General Purpose I/O Line
P24	31	I/O	General Purpose I/O Line

Table 4. TMS 9940 Pin Assignments and Functions (Concluded)

SIGNATURE	PIN	I/O	DESCRIPTION
P25	32	I/O	General Purpose I/O Line
P26	33	I/O	General Purpose I/O Line
P27	34	I/O	General Purpose I/O Line
P28	35	I/O	General Purpose I/O Line
P29	36	I/O	General Purpose I/O Line
P30	38	I/O	General Purpose I/O Line
P31	39	I/O	General Purpose I/O Line

TMS9940 INSTRUCTION SET

Definition

Each instruction of the TMS9940 set performs one of the following operations:

- Arithmetic, logical, comparison, or manipulation operations on data;
- Loading or storage of machine registers (program counter, workspace pointer, or status);
- Data transfer between memory and external devices via the CRU;
- Control functions.

This instruction set is identical to that of the TMS9900 with the following exceptions:

- Instructions added (as dedicated XOP's 0 to 3);
  —DCA —LIIM —DCS
- Instructions deleted
  —RSET —CKOF
  —CKON —LREX

A complete listing of the instructions and addressing modes is found in a later section.

TMS9940 Instruction Execution Times

Instruction execution times for the TMS9940 are a function of:

1. Clock cycle time, $t_c(\phi) \equiv 2 \cdot t_{cy}$ where $t_{cy} \equiv 1/\text{Oscillator Frequency } (f_{osc})$
2. Addressing mode used where operands have multiple addressing mode capability.

Table 5 lists the number of clock cycles required to execute each TMS9940 instruction. For instructions with multiple addressing modes for either or both operands, the table lists the number of clock cycles with all operands addressed in the workspace register mode. To determine the additional number of clock cycles required for modified addressing, add the appropriate values from Table A. The total instruction execution time for an instruction is

$T = t_c(\phi) \cdot C$

where,

T = total instruction time
$t_c(\phi)$ = clock cycle time
C = number of clock cycles required for instruction execution plus address modification.

8◀

Table 5. TMS 9940 Instruction Execution Times

INSTRUCTION	CLOCK CYCLES	ADDRESS MODIFICATION
A	10	See Table A
AB	7	See Table A
ABS (MSB = 0)	12	See Table A
(MSB = 1)	12	See Table A
AI	12	
ANDI	12	
B	8	See Table A
BL	10	See Table A
BLWP	20	See Table A
C	10	See Table A
CB	7	See Table A
CI	12	
CLR	8	See Table A
COC	10	See Table A
CZC	10	See Table A
DCA	7	See Table A
DCS	7	See Table A
DEC	8	See Table A
DECT	8	See Table A
DIV (ST 4 is SET)	14	See Table A
DIV (ST 4 is RESET)*	128	See Table A
IDLE	10	
INC	8	See Table A
INCT	8	See Table A
INV	8	See Table A
JUMP	6	
LDCR (C = 0)	42	See Table A
(1≤C≤8)	8 + 2C	See Table A
(9≤C≤15)	10 + 2C	See Table A
LI	12	
LIIM	10	

▶8

Table 5. TMS 9940 Instruction Execution Times (Continued)

INSTRUCTION	CLOCK CYCLES	ADDRESS MODIFICATION
LIMI	14	
LWPI	12	
MOV	8	See Table A
MOVB	6	See Table A
MPY	82	See Table A
NEG	10	See Table A
ORI	12	
RTWP	14	See Table A
S	10	See Table A
SB	7	See Table A
SBO	10	
SBZ	10	
SETO	8	See Table A
SHIFT (C = 0)	12 + 2N	
(C = 0, BITS 12-15 of WR0 = 0)	46	
(C = 0, BITS 12-15 of WR0 = N ≠ 0)	14 + 2N	
SOC	10	See Table A
SOCB	7	See Table A
STRC (C = 0)	46	See Table A
(1 ≤ C ≤ 8)	29	See Table A
(9 ≤ C ≤ 15)	46	See Table
STST	8	
STWP	8	
SWPB	8	See Table A
SZC	10	See Table A
SZCB	7	See Table A
TB	10	
X**	6	See Table A
XOP	26	See Table A
XOR	10	See Table A
Reset Function	16	
Interrupt Context Switch	16	

*Execution time is dependent on the partial quotient after each clock cycle during execution.
**Execution time is added to the execution time of the instruction located at the source address.

8◀

TABLE A
ADDRESS MODIFICATION

ADDRESSING MODE	CLOCK CYCLES (C)
WR (T_s or T_d = 00)	0
WR indirect (T_s or T_d = 01)	2
WR indirect auto increment (T_s or T_d = 11)	4
Symbolic (T_s or T_d =10, S or D = 0)	6
Indexed (T_s or T_d =10, S or D ≠ 0)	8

TMS9940E EPROM PROGRAMMING

Erasure

Before programming, the TMS9940E is erased by exposing the chip through the transparent lid to high intensity ultraviolet light (wavelength: 2537 angstroms). The recommended exposure is 10 watt-seconds per square centimeter. This can be obtained by, for instance, 20 to 30 minutes exposure of a filterless Model S52 shortwave UV lamp about 2.5 centimeters above the EPROM. After exposure all bits are in the "0" state.

Programming

The TMS9940E should be initialized by RESET before the programming sequence begins. The EPROM consists of 16K bits of program memory organized as 2K bytes (8 bits) located at (starting) address 0000_{16}. Data is transferred into the CPU for programming through P24(MSB)–P31 (LSB). Taking the PE signal active high (V_{IP}) initializes the internal address pointer of 0000_{16} and inputs the first byte of data (see *Figure 8*). After a minimum delay of 40 clock cycles, PROG can be applied (V_{IP}, 50 ms) and the data present on P24-P31 updated to the next byte. Tha falling edge of PROG inputs the new byte of data to the next location and after a minimum delay of 25 clock cycles the PROG pulse can be applied again. This sequence is continued until the entire 2K bytes have been programmed. Note that the memory is programmed in sequence starting at 0000_{16}, and the input data must be valid at the rising edge of PE or falling edge of PROG.

Programming/Test Function Electrical Characteristics

▶8

	PARAMETER	MIN	NOM	MAX	UNIT
t_r	TST, PE, PROG input rise time		100		ns
t_f	TST, PE, PROG input fall time		100		ns
t_{su}	Input data setup time to rising edge of PE, TST or to falling edge of PROG		0		ns
t_h	Input data hold time past rising edge of PE, TST		80 $t_{c(\phi)}$		ns
t_h(P-da)	Input data hold time past falling edge of PROG		50 $t_{c(\phi)}$		ns
t_h(P-PE,T)	PE, TST input hold time past falling edge of PROG		0		ns
t_{su}(P-PE,T)	PROG input setup time to rising edge of PE, TST		0		ns
t_h(T-PL)	PROG input pulse low past rising edge of TST, PE		80 $t_{c(\phi)}$		ns
t_w(PL)	PROG input pulse width low		50 $t_{c(\phi)}$		ns
t_w(PHP)	PROG input pulse width high in the programing mode		50		ms
t_w(PHT)	PROG input pulse width high in the test mode		4 $t_{c(\phi)}$		ns

NOTE: Timing diagrams in Figure 8.

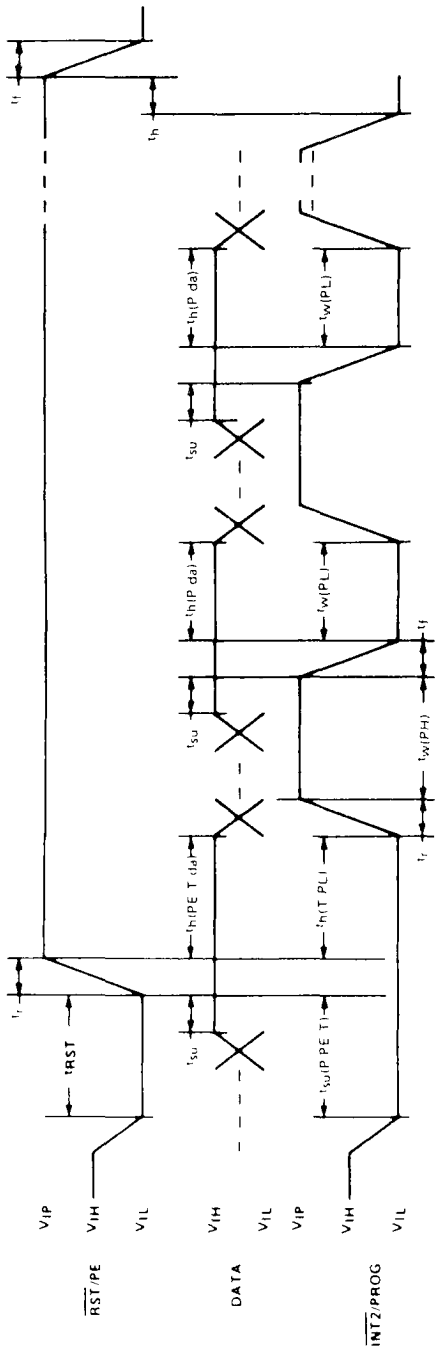


Figure 8. EPROM Programming Timing Diagram

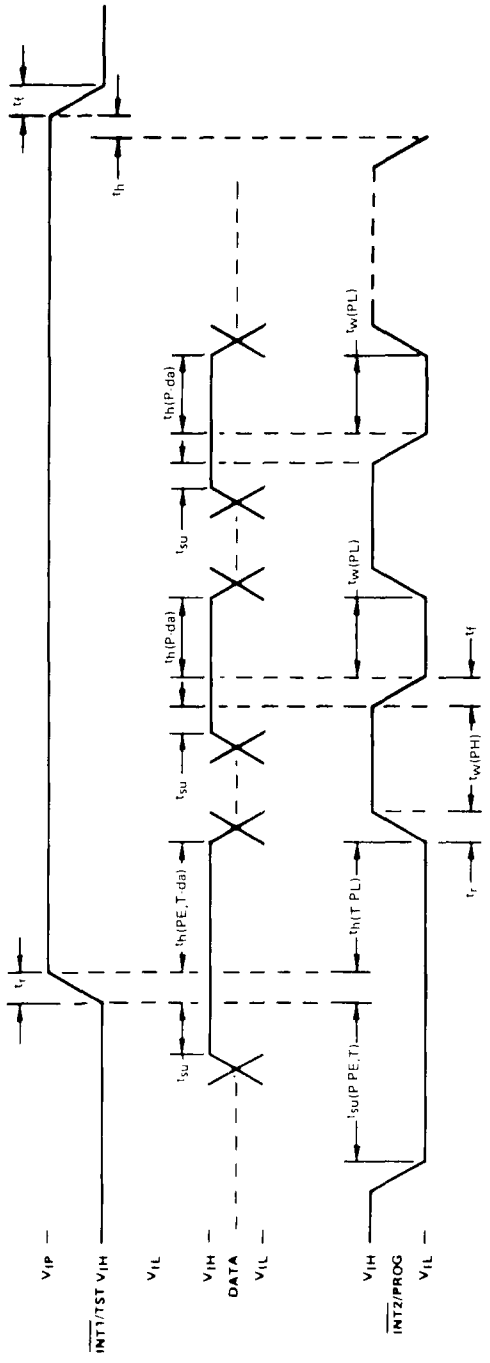


Figure 9. Test Function Timing Diagram

▶8

TEST FUNCTION

This test function allows loading a program into the RAM area of the TMS9940 through pins P24 through P31. This program can then be executed, and the results of this execution used to verify operation of the TMS9940. The program could include error messages as well as a successful completion message sent to a peripheral device accessed through the CRU.

The processor should be initialized by $\overline{\text{RESET}}$ before any test sequence begins. Data is directly loaded in sequence into the RAM through P24 (MSB)—P31 (LSB). Taking the TEST signal active high (V_{IP}) initializes the internal address pointer to 8300_{16} (starting address of RAM) and inputs the first byte of data (see *Figure 9*). After a minimum delay of 40 clock cycles PROG can be applied (V_{IH}, 4 clock cycles minimum) and the data present on P24—P31 updated to the next byte. The falling edge of PROG inputs the new byte of data to the next location and, after a minimum delay of 25 clock cycles, PROG can be applied again. This sequence is continued until the desired data has been loaded into the RAM. Taking TEST inactive will then jump the processor to the address specified by the last 16 bits loaded. Note that the RAM is loaded in sequence starting at 8300_{16}, and the input data must be valid at the rising edge of TST or on the falling edge of PROG.

TMS9940 ELECTRICAL SPECIFICATIONS

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

Supply Voltage, V_{CC1}† . . . . . −0.3 to 20 V
Supply Voltage, V_{CC2} . . . . . −0.3 to 20 V
Programming Voltage, PE . . . . . −0.3 to 35 V
All Input Voltages . . . . . −0.3 to 20 V
Output Voltage . . . . . −2 to 7 V
Continuous Power Dissipation . . . . . 1.5 watt
Operating Free-Air Temperature Range . . . . . 0°C to 70°C
Storage Temperature Range . . . . . −55°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.

†All voltage values are with respect to V_{SS}.

Recommended Operating Conditions

PARAMETER	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC1}		5		V
Supply voltage, V_{CC2}		5		V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH}	2.0			V
Low-level input voltage, V_{IL}			0.8	V
Program/test input voltage, V_{IP}		26		V
Operating free-air temperature, T_A	0		+70	°C

ELECTRICAL CHARACTERISTICS

PARAMETER	TEST CONDITIONS	MIN	NOM	MAX	UNITS
I_I input current, all inputs	$V_I = V_{SS}$ to V_{CC}		±10		μA
V_{OH}, high-level output voltage, all outputs	$I_O = -0.4$ mA		2.4		V
V_{OL}, low-level output voltage, all outputs	$I_O = 2$ mA		0.4		V
I_{CC1}, supply current from V_{CC1}			10		mA
I_{CC2}, supply current from V_{CC2}			150		mA
C_I, input capacitance, all inputs	f = 1 MHz unmeasured pins at V_{SS}		15		pF
C_O, output capacitance, all outputs	f = 1 MHz unmeasured pins at V_{SS}		15		pF

CLOCK CHARACTERISTICS

The TMS 9940 has an internal oscillator and a two-phase clock generator controlled by an external or crystal. The user may also disable the oscillator and directly inject a frequency source into the XTAL2 input. The crystal frequency and the external frequency source must be double the desired system CLOCK frequency.

Internal Oscillator

The internal oscillator is enabled by connecting a crystal across XTAL 1 and XTAL 2. The system CLOCK frequency $1/t_{c(\phi)}$, is one-half the crystal oscillator frequency, f_{osc}.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
f_{osc}, TMS 9940E, TMS 9940M		.5	5.0	5.12	MHz
f_{osc}, TMS 9940E-40, TMS 9940M-40		.5	4.0	4.10	MHz
f_{osc}, TMS 9940E-30, TMS 9940M-30	$0°C \leq T \leq +70°C$	.5	3.0	3.07	MHz
f_{osc}, TMS 9940E-20, TMS 9940M-20		.5	2.0	2.05	MHz
f_{osc}, TMS 9940E-10, TMS 9940M-10		.5	1.0	1.02	MHz

Note: $t_{cy} \equiv 1/f_{osc}$

$t_{c(\phi)} \equiv 2 \cdot t_{cy}$

▶8

TMS 9940 DESIGN SUPPORT

Program Support Tools

Table 6 defines four of the major program development methods available for the TMS 9940 microcomputer. Each program development product supports assembly language programming, options 1 and 2 support hardware emulation for target system evaluation.

Table 6. TMS 9940 Program Support

OPTION	SYSTEM	HOST COMPUTER	ASSEMBLER	EMULATION	SUPPLIER
1	TM 990/40DS Low Cost Standalone Development System	16 - Bit TI Microcomputer TM 990/40DS User Supplied Terminal (TTY, TI 733, or TI 745)	Yes, Line - By - Line (Non Symbolic)	Yes, Non - Real Time	Semiconductor Group - TI -
2	AMPL System (Advanced Microprocessor Prototyping Laboratory)	16 - Bit TI Minicomputer, FS 990/4 (Includes Terminal)	Yes, TXMIRA, Full Assembly	Yes, Real - Time	Digital Systems Group - TI -
3	TMSWIOIT Transportable FORTRAN Source, Cross Support Software Package	User Supplied 16 - Bit Minicomputer or larger (eg 32 Bit Mainframe) User Supplied I/O	Yes, Full Assembly	No	Semiconductor Group - TI -
4	Timeshare	Timeshare 32 - Bit Mainframe User Supplied I/O	Yes, Macro Assembly And Full Assembly	No	G.E N.C.S S TYMSHARE

Options 3 and 4 support compatible computer simulation based on the TMS 9900 microprocessor; thus, functional instruction simulation without TMS 9400 timing or I/O data is possible.

To determine the most cost-effective tool, the second column relates to the computer equipment required. Timeshare has a repeating cost to consider, whereas, the remainder are one-time investments.

To assemble short program modules via option 1, the Line-By-Line Assembler is the fastest method available. It interactively provides mnemonic-to-object assembly, excluding symbolic addressing references. Standalone program debug is then performed through the terminal keyboard.

The bulk of the TMS 9940 instruction set is identical to the TMS 9900 assembly language. Available cross assemblers are:

- PX9 ASM on the CS 990/4 cassette development system
- TXMIRA on the FS 990/4 floppy disk development system
- SYSMAC on the DS 990/10
- TMS 9900 cross-assembler (available on several timesharing networks).

8

Three instructions on the TMS 9940 are not found on TMS 9900 assemblers: DCA, DCS, and LIIM. However, these mnemonics are made acceptable by the 'DXOP' assembler directive. The 'DXOP' assembler directive is available on all of the above mentioned assemblers. The DXOP function is to define a label for a specific XOP value. The directive should appear at the beginning of the source life. The following listing shows how the DCA, DCS, and LIIM mnemonics are defined using the DXOP directive, and a short sample program using the three instructions. Note that the DXOP directives are used prior to using the instructions.

```
SAMPLE          TXMIRA      936227**                    PAGE 0001
9940 SAMPLE PROGRAM

0002                        DXOP    DCA,0       DEFINE XOP 0 AS DCA
0003                        DXOP    DCS,1       DEFINE XOP 1 AS DCS.
0004                        DXOP    LIIM,2      DEFINE XOP 2 AS LIIM
0005                        IDT     'SAMPLE'
0006 0000 A081  START       AB      R1, R2      ADD REGISTERS 1 AND 2 TOGETHER
0007 0002 2C02              DCA     2           CORRECT THE RESULT FOR BCD.
0008 0004 1002              JMP     QUIT        GO TO CLEAN UP.
0009 0006 6081  STARTS      SB      R1, R2      SUBTRACT REGISTER 1 FROM 2.
0010 0008 2C42              DC8     R2          CORRECT THE RESULT FOR BCD.
0011 000A 2C82  QUIT        LIIM    R2          LOAD NEW INTERRUPT MASK.
0012 000C 0380              RTWP    GO HOME
0013                        END     START

0000 ERRORS
```

Factory Programming — TMS 9940M

Produced from any of the program support tools, a TI standard TMS 9900 family object format is accepted for factory programming. The absolute object form with a custom MPXXXX number in the program identifier field are acceptable. The object file can be sent and subsequently verified through a timeshare transmission or TI 733 — compatible digital cassettes (punched cards, paper tape, FS 990 floppy discs are also accepted) when developed on the TM 990/40DS or non-TI designed support tools, a user can send a master TMS 9940E device containing the code to be produced in volume.

User Programming — TMS 9940E

The TM 990/40DS low-cost development system can program, verify, and download TMS 9940E devices. A TI designed test program or user defined programs (modifying the TIBUGII resident monitor) also provide functional testing on the development system. Refer to the Test Function section for a detailed description.

A programmer module is an accessory to FS 990 minicomputers. The program, verify and download functions work together with the sophisticated AMPL package in FS 990/4 systems.

Custom Applications

Through a staff of experienced application programmers and microprocessor specialists, Texas Instruments will, upon request, assist customers in evaluating applications, in training designers to program the TMS 9940, and in simulating programs. TI will also contract to write programs to customer's specifications.

INTRODUCTION

DESCRIPTION

The TMS 9985 is a software compatible member of TI's 9900 family of microprocessors and microcomputers and contains a 16-bit CPU, 256 bytes of RAM, on chip timer/event counter, external 16-bit address bus and 8-bit data bus, and is packaged in a 40-pin package. The instruction set of the TMS 9985 includes the capabilities offered by full minicomputers and is exactly the same as the TMS 9940 microcomputer's. The unique memory-to-memory architecture features multiple register files, resident in memory, which allows faster response to interrupts and increased programming flexibility. The separate bus structure (see *Figure 8-46*) simplifies the system design effort. All members of the TMS 9900 family of peripheral circuits are compatible with the TMS 9985. The TMS 9985 is fully supported by software and hardware development systems.

KEY FEATURES DIFFERENT FROM THE TMS 9900

- 5-MHz Speed
- 8-Bit Memory Data Bus
- 5 Prioritized Interrupts
- 40-Pin Package
- On Chip Timer/Event Counter
- 256 Bits of RAM on Chip
- Separate Memory, I/O and Interrupt Bus Structures
- On Chip Programmable Flags (16)
- Multiprocessor System Interface
- Single 5-Volt Supply
- Speed Selected Versions

DIFFERENCES BETWEEN THE TMS 9985 AND THE TMS 9940

The TMS 9985 is so similar to the TMS 9940 that only the differences are described here.

Key Features Different from the TMS 9940

- 5-MHz Speed
- Up to 65,536 Bytes of Memory
- 256 Bytes of RAM On Chip
- 8-Bit Memory Data Bus
- Separate Memory, I/O and Interrupt Bus Structures
- 5 Prioritized Interrupts

ARCHITECTURE

Registers and Memory

See the TMS 9940.

INTERRUPTS

The TMS 9985 implements five hardware interrupt level. Interrupt level data is shown in *Table 1.*

***Table 1.** Interrupt Level Data*

▶8

INTERRUPT LEVEL		VECTOR LOCATION (MEMORY ADDRESS IN HEX)	DEVICE ASSIGNMENT	INT. MASK VALUE TO ENABLE RESPECTIVE INTERNAL STATUS REGISTERS 14 AND 15
(Highest Priority)	0	0000	RESET	0 THROUGH 3*
	1	FFFC	LOAD	0 THROUGH 3*
	2	0004	EXTERNAL DEV	1 THROUGH 3
	3	0008	DECREMENTER	2 AND 3
	4	000C	EXTERNAL DEV	3 ONLY

* $\overline{\text{RESET}}$ AND $\overline{\text{LOAD}}$ CAN NOT BE DISABLED

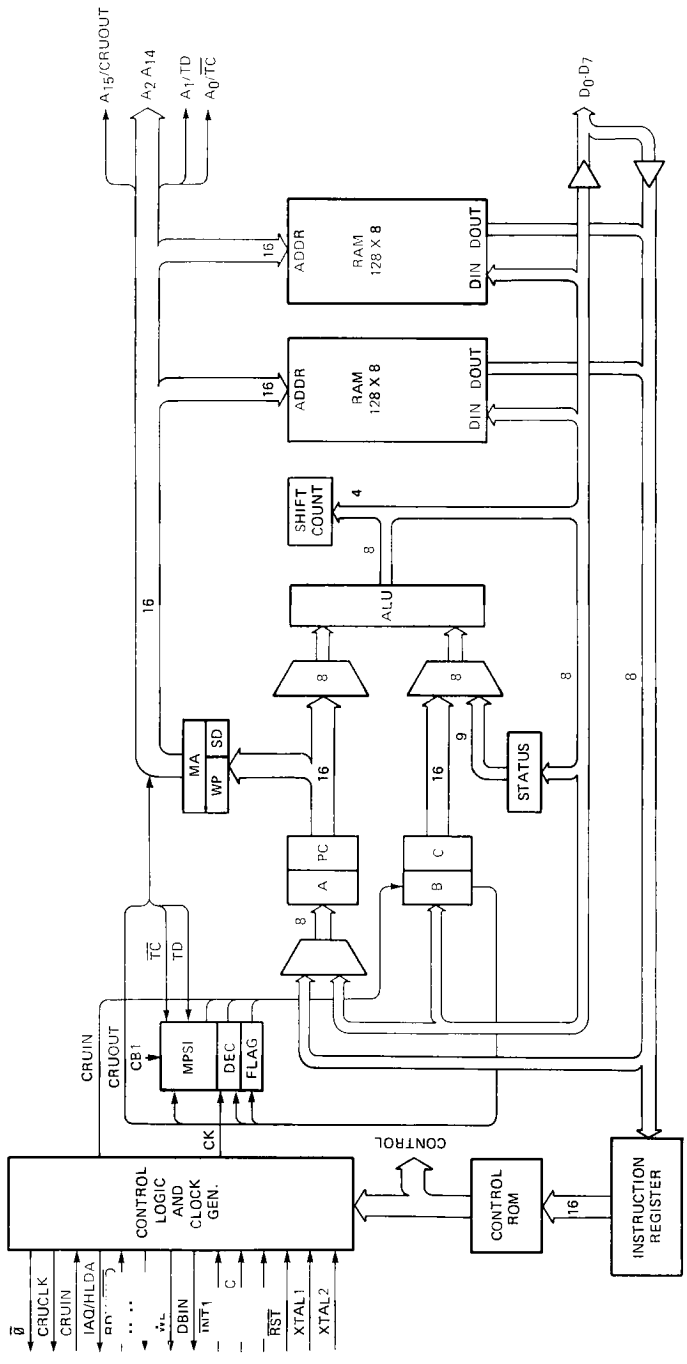


Figure 1. TMS 9985 Architecture

Input/Output

The TMS 9985 supports four types of I/O channels:

1. Communications Register Unit (CRU)
2. Memory Mapped (MM)
3. Direct Memory Access (DMA)
4. The Multiprocessor System Interface (MPSI)

The CRU and MPSI are much the same as those in the TMS 9940. See the TMS 9940 for a discussion of the decrementer and flag register.

Memory Mapped I/O Channel

Memory Mapped I/O is a byte oriented I/O interface through which input or output bytes can be directly addressed. The interface is defined to exist in memory address space and is accessed as if it were a memory location. All processor instructions that access memory can be used to drive the Memory Mapped interface, and thus, arithmetic and logical operations can be performed directly on MM I/O. *Figure 1* illustrates how to implement a 1 byte input and 1 byte output MM register.

Direct Memory Access (DMA)

Direct Memory Access (DMA) is a block oriented I/O interface through which blocks of data can be moved into and out of the system memory under external control. The external controller applies $\overline{\text{HOLD}}$ to initiate a DMA request. $\overline{\text{HOLD}}$ is sampled during nonmemory cycles and, when detected, forces the TMS 9985 to enter a hold state. The processor places the address and data buses into the high impedance state and responds with a hold acknowledge signal (HOLDA). When $\overline{\text{HOLD}}$ is removed the TMS 9985 will then return to normal operation. *Figure 2* shows how to implement a DMA system using the TMS 9911 DMA controller. The maximum latency time between a $\overline{\text{HOLD}}$ request and a HOLDA response is equal to 37 clock cycles. The DMA channel cannot be used to move data into or out of the internal RAM.

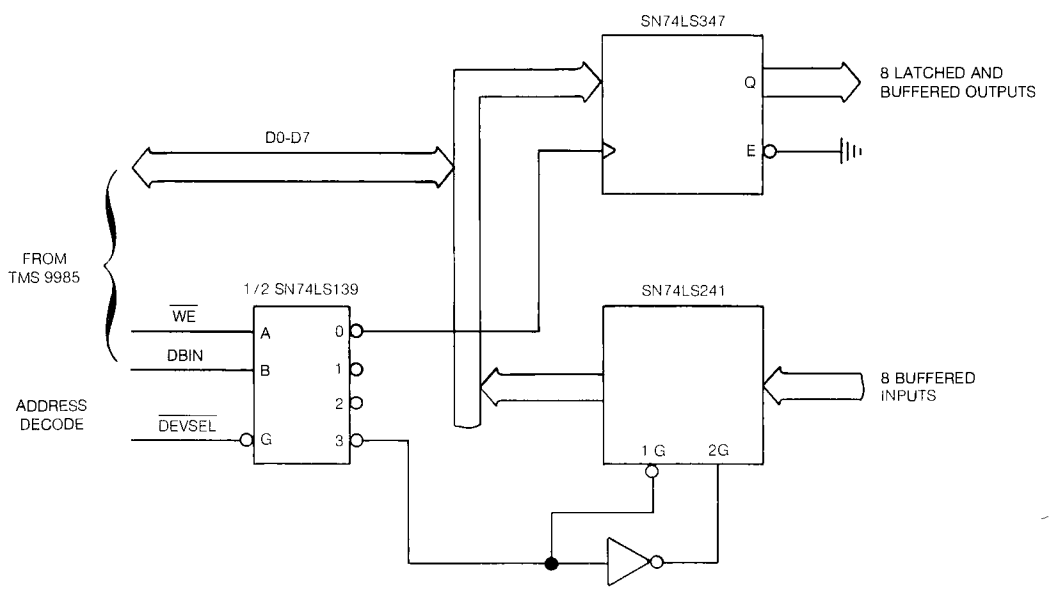


Figure 2. 8-Bit Memory mapped I/O Interface

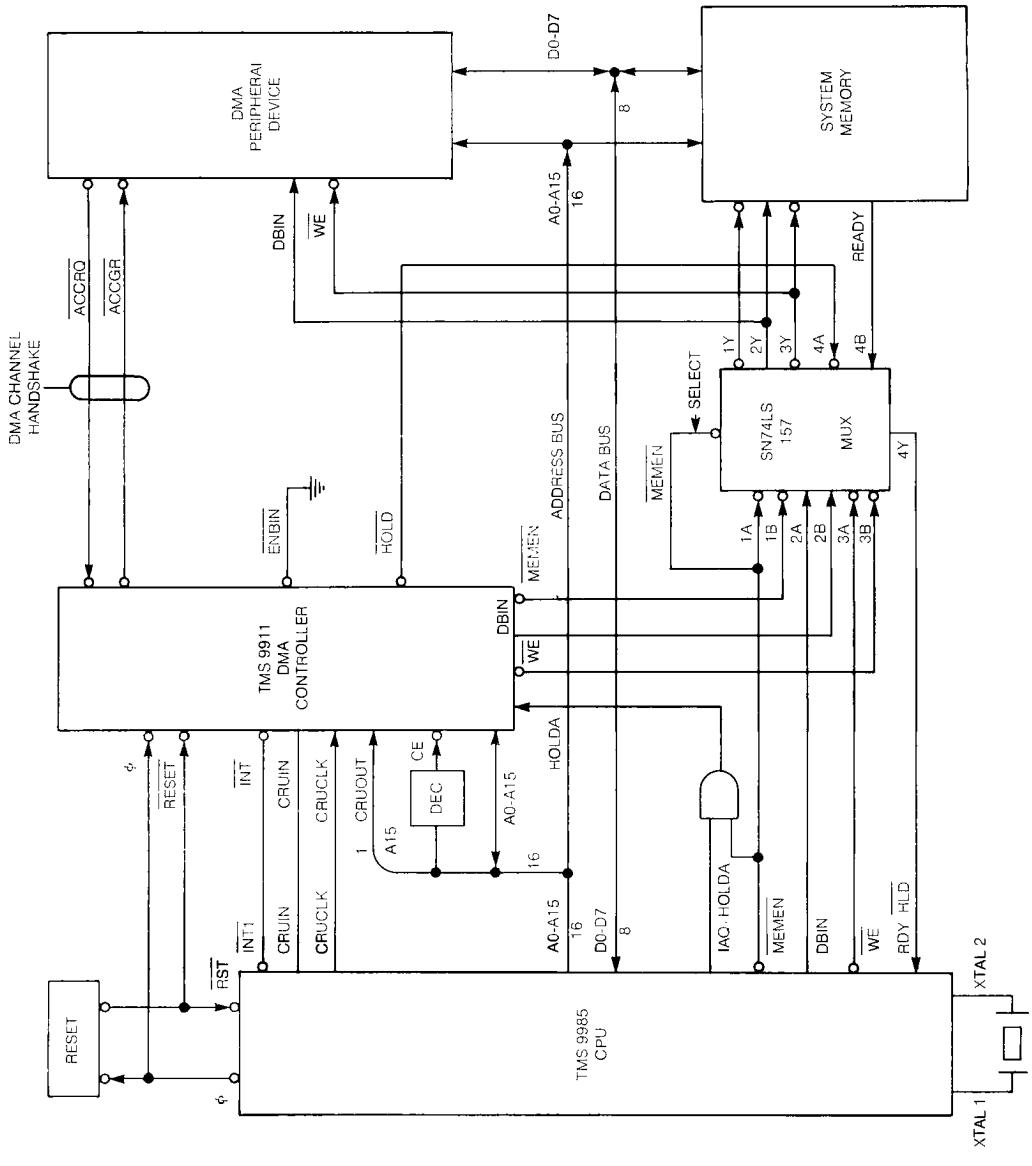


Figure 3. TMS 9985 DMA Interface Using the TMS 9911 DMA Controller

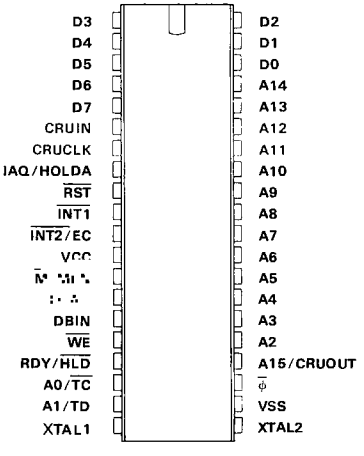
System Configuration

The TMS 9985 allows the user to configure part of the pins as special functions for system applications. The configurable pins are shown below.

Pin Name	*Configuration Bit*	*Configuration*	
		Output	Input
A0/$\overline{TC}$	1	A0	$\overline{TC}$
A1/TD	1	A1	TD

TMS 9985 Pin Description

SIGNATURE	I/O	DESCRIPTION
XTAL 1	IN	Crystal input pin for control of internal oscillator.
XTAL 2	IN	Crystal input pin for control of internal oscillator. Also input pin for external oscillator.
VCC		Supply voltage (+5 volts)
VSS		Ground reference
$\overline{\phi}$	OUT	Clock output signal. The frequency of $\overline{\phi}$ is ½ of the oscillator input frequency.
$\overline{RST}$	IN	$\overline{RESET}$. When active (LOW), (Schmitt Trigger input) the RESET sequence is initiated. RESET must be held active for a minimum of five clock cycles.
$\overline{INT1}$	IN	$\overline{INTERRUPT}$ 1. When active (LOW), external device interrupt 1 is active.
$\overline{INT2}$/EC	IN	$\overline{INTERRUPT}$ 2/EVENT COUNTER. When active (LOW), external device interrupt 2 is active. When the decrementer is programmed as an event counter (T/$\overline{C}$=0), a positive transition on $\overline{INT2}$/EC will decrement the count.
$\overline{LOAD}$	IN	$\overline{LOAD}$. when active (LOW), $\overline{LOAD}$ causes the TMS 9985 to execute a non maskable interrupt with memory address $FFFC_{16}$ containing the trap vector (WP and PC). The load sequence begins after the instruction being executed is completed. $\overline{LOAD}$ will also terminate an idle state. If $\overline{LOAD}$ is active during the time $\overline{RESET}$ is released, then the LOAD trap will occur after the $\overline{RESET}$ function is completed. LOAD should remain active for one instruction period. IAQ can be used to determine instruction boundaries.
A0/$\overline{TC}$	I/O	ADDRESS BIT 0/$\overline{TRANSFER\ CLOCK}$ when configured as address, A0 is the MSB of the 16 bit memory address bus and the 15 bit CRU address bus. When configured for the MPSI, $\overline{TC}$ is the transfer clock Line (See I/O Section for configuration details).
A1/TD	I/O	ADDRESS BIT 1/TRANSFER DATA. When configured as address, A1 is the 2nd most significant bit of the 16 bit memory address bus and the 15 bit CRU address bus. When configured for the MPSI, TD is the transfer data line (See I/O Section for configuration details).



SIGNATURE	I/O	DESCRIPTION
A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14	OUT	ADDRESS BUS. A2 is the 3rd most significant bit of the 16 bit memory address bus and the 15 bit CRU address bus. A14 is the 2nd least significant bit of the 16 bit memory address bus and the LSB of the 15 bit CRU address bus. The address bus assumes the high impedance state when holda is active.
A15/ CRUOUT	OUT	ADDRESS BIT 15/CRU OUTPUT DATA. A15/CRUOUT is the LSB of the 16 bit memory address bus and the output data line for CRU output instructions. A15/CRUOUT assumes the high impedance state when HOLDA is active.
D0 D1 D2 D3 D4 D5 D6 D7	I/O	DATA BUS. D0 is the MSB of the 8 bit data bus, and D7 is the LSB. This bus transfers data to (when writing) and from (when reading) the external memory system when $\overline{\text{MEMEN}}$ is active. The data bus assumes the high impedance state when HOLDA is active.
DBIN	OUT	DATA BUS IN. When active (HIGH), DBIN indicates that the TMS 9985 has disabled its output buffers to allow the external memory to place memory-read data on the data bus during $\overline{\text{MEMEN}}$. DBIN remains low in all other cases.
$\overline{\text{MEMEN}}$	OUT	MEMORY ENABLE. When active (LOW), $\overline{\text{MEMEN}}$ indicates that the address bus contains an external memory address, the READY/$\overline{\text{HOLD}}$ input is sampling READY, and the IAQ/HOLDA output is outputting IAQ.
$\overline{\text{WE}}$	OUT	WRITE ENABLE. When active (LOW), $\overline{\text{WE}}$ indicates that memory write data is available from the TMS 9985 to be written into external memory during $\overline{\text{MEMEN}}$. $\overline{\text{WE}}$ remains high in all other cases.
CRUIN	IN	CRU DATA IN. CRUIN is the input data line for CRU input instructions and is sampled during the TB instruction.
CRUCLK	OUT	CRU CLOCK. When active (HIGH), CRUCLK indicates that the external interface logic should sample the output data on CRUOUT.
RDY/$\overline{\text{HLD}}$	IN	READY/$\overline{\text{HOLD}}$. When $\overline{\text{MEMEN}}$ is active the RDY/$\overline{\text{HLD}}$ input sample READY; when $\overline{\text{MEMEN}}$ is inactive, $\overline{\text{HOLD}}$ is sampled. When active (HIGH), READY indicates that external memory will be ready to read or write with no additional wait states When not-ready is indicated during an external memory operation the TMS 9985 enters a wait state and suspends internal operation until the memory systems indicate ready. When active (LOW), $\overline{\text{HOLD}}$ indicates to the processor that an external controller desires to utilize the address and data buses to transfer data to or from external memory. Following a hold signal, the TMS 9985 enters a hold state at the next memory cycle The processor places the address and data buses in the high impedance state and responds with a hold acknowledge signal (HOLDA). When $\overline{\text{HOLD}}$ is removed, the TMS 9985 returns to normal operation.
IAQ/HOLDA	OUT	INSTRUCTION ACQUISITION/HOLD ACKNOWLEDGE. When $\overline{\text{MEMEN}}$ is active the IAQ/HLDA line outputs IAQ; when $\overline{\text{MEMEN}}$ is inactive HOLDA is output. IAQ is active (HIGH), during any memory cycle when the TMS 9985 is acquiring an instruction HOLDA is active (HIGH) when the processor is in the hold state and the address and data buses are in the high impedance state.

TIMING

Memory

Basic memory read and write cycles are shown in *Figures 4* and *5*. *Figure 4* shows read and write cycles with no wait states while *Figure 5* shows read and write cycles for a memory requiring one or two wait states.

$\overline{\text{MEMEN}}$ goes active (LOW) during each memory. At the same time that $\overline{\text{MEMEN}}$ is active, the memory address appears on the address bus (A0 through A15). If the cycle is a memory read cycle, DBIN will be active (HIGH) and the data present on the data bus (D0 through D7) will be input into the processor. If the cycle is a memory write cycle, $\overline{\text{WE}}$ will go active (LOW) and data will be input by the CPU onto the data bus. At the end of the cycle $\overline{\text{MEMEN}}$ and DBIN or $\overline{\text{WE}}$ will go inactive.

Hold

Other interfaces may utilize the TMS 9985 memory bus by using the hold operation (illustrated in *Figure 6*). The external $\overline{\text{HOLD}}$ input is sampled during nonmemory cycles and when active (LOW), forces the TMS 9985 to enter the hold state. The processor places the address and data buses into the high impedance state to allow other devices to use the memory buses, and outputs the hold acknowledge signal (HOLDA, active HIGH). When $\overline{\text{HOLD}}$ goes inactive, the TMS 9985 resumes processing as shown. The maximum latency time between a $\overline{\text{HOLD}}$ request and a HOLDA response is equal to 37 clock cycles.

CRU

CRU interface timing is shown in *Figure 7*. The timing for transferring two bits out and one bit in is shown. These transfers would occur during the execution of a CRU instruction. The other cycles of the instruction execution are not illustrated. To output a CRU bit, the CRU-bit address is placed on the address bus A0 through A14 and the actual bit data on A15/CRUOUT. During the second clock cycle a CRU pulse is supplied by CRUCLK. This process is repeated until the number of bits specified by the instruction are completed.

The CRU input operation is similar in that the bit address appears on A0 through A14.

During the subsequent cycle, the TMS 9985 accepts the bit input data as shown. No CRUCLK pulses occur during a CRU input operation.

8

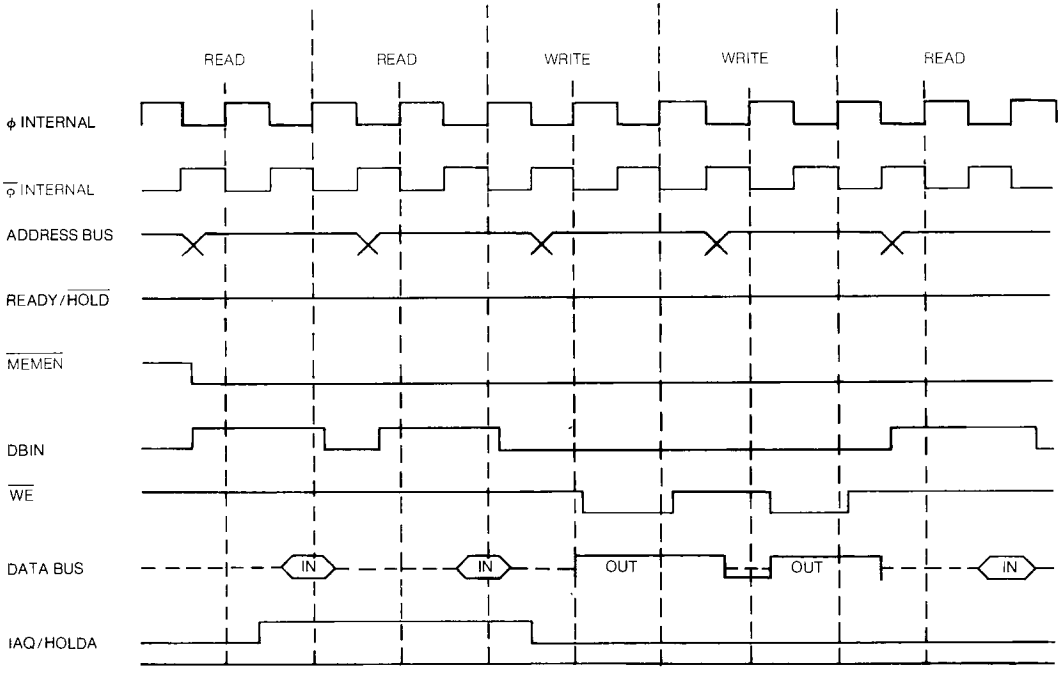


Figure 4. TMS 9985 Memory Bus Timing (no wait states)

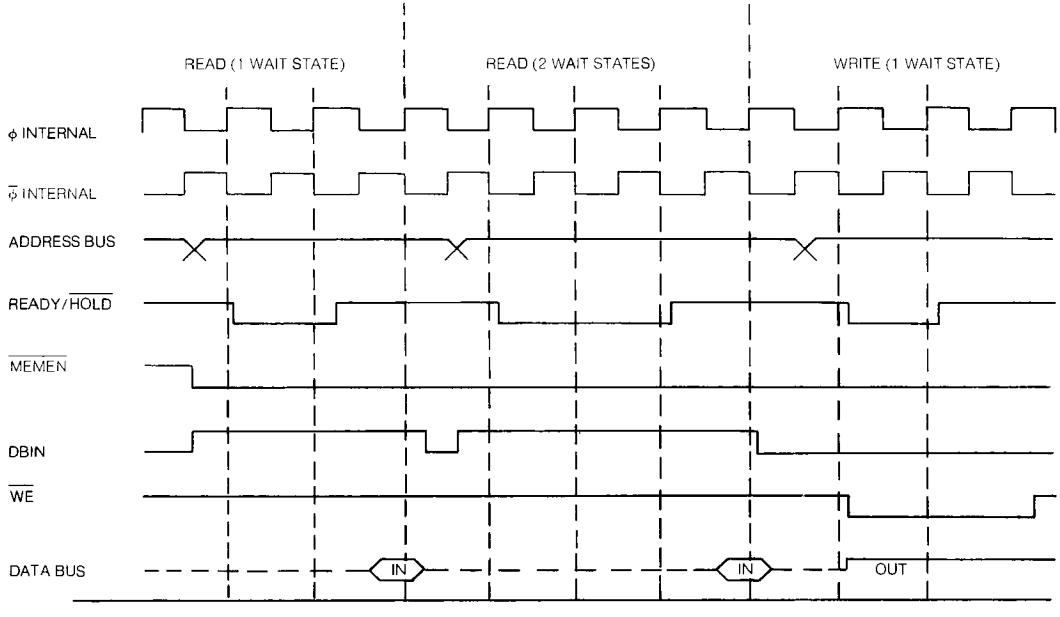


Figure 5. TMS 9985 Memory Bus Timing (with wait states)

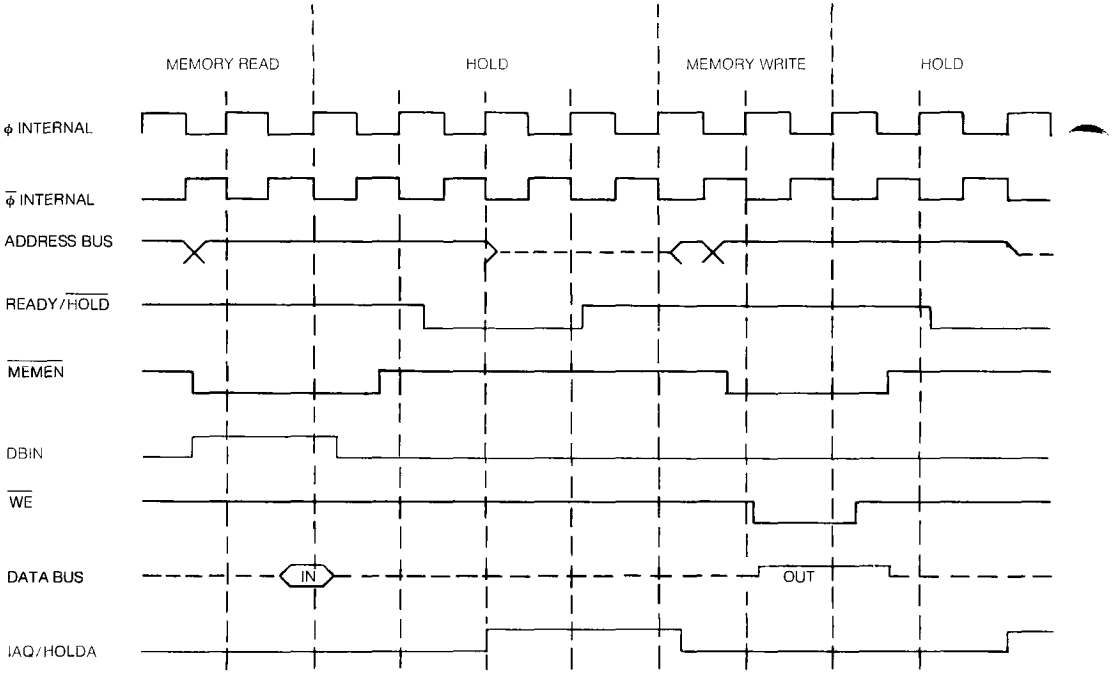


Figure 6. TMS 9985 Hold Timing

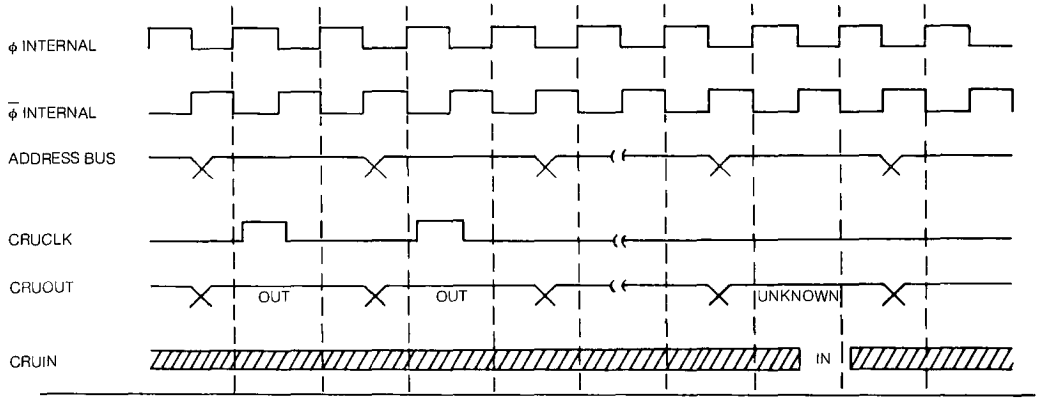


Figure 7. TMS 9985 CRU Interface Timing

9900 Instruction Set

9900 INSTRUCTION SET

Definition

Each 9900 instruction performs one of the following operations:

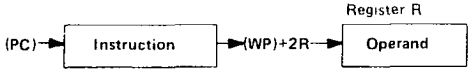
- Arithmetic, logical, comparison, or manipulation operations on data
- Loading or storage of internal registers (program counter, workspace pointer, or status)
- Data transfer between memory and external devices via the CRU
- Control functions.

Addressing Modes

The 9900 instructions contain a variety of available modes for addressing random-memory data (e.g., program parameters and flags), or formatted memory data (character strings, data lists, etc.). The following figures graphically describe the derivation of the effective address for each addressing mode. The applicability of addressing modes to particular instructions is described in the Instructions Section along with the description of the operations performed by the instruction. The symbols following the names of the addressing modes [R, *R, *R+, @ LABEL, or @ TABLE (R)] are the general forms used by 9900 assemblers to select the addressing mode for register R.

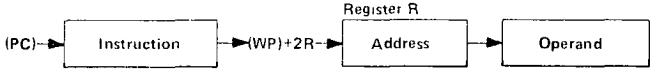
Workspace Register Addressing R

Workspace Register R contains the operand.



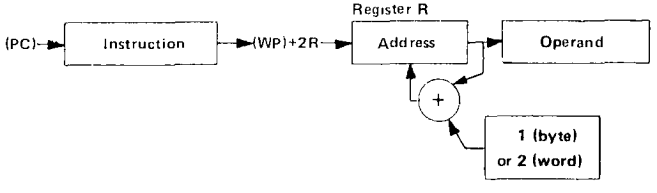
Workspace Register Indirect Addressing *R

Workspace Register R contains the address of the operand.



Workspace Register Indirect Auto Increment Addressing *R+

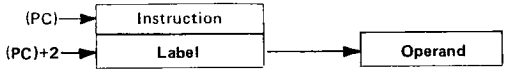
Workspace Register R contains the address of the operand. After acquiring the operand, the contents of workspace register R are incremented.



▶8

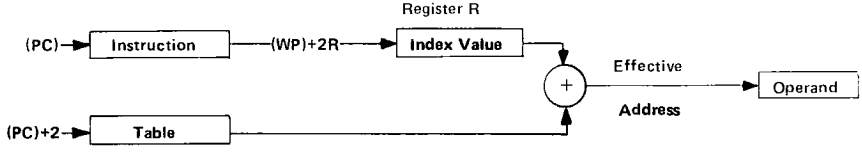
Symbolic (Direct) Addressing @LABEL

The word following the instruction contains the address of the operand.



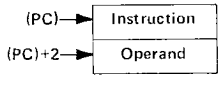
Indexed Addressing @ TABLE (R)

The word following the instruction contains the base address. Workspace register R conatins the index value. The sum of the base address and the index value results in the effective address of the operand.



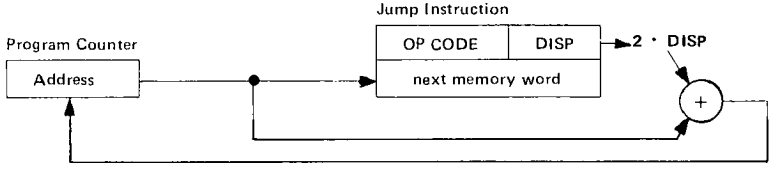
Immediate Addressing

The word following the instruction contains the operand.



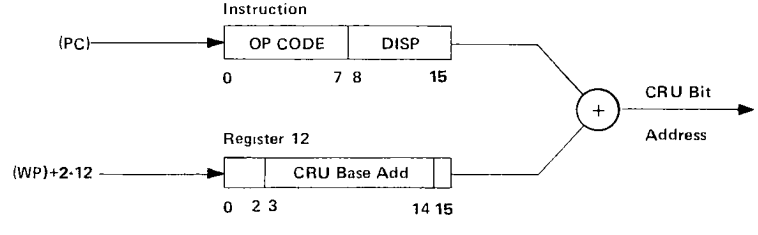
Program Counter Relative Addressing

The 8-bit signed displacement in the right byte (bits 8 through 15) of the instruction is multiplied by 2 and added to the updated contents of the program counter. The result is placed in the PC.



CRU Relative Addressing

The 8-bit signed displacement in the right byte of the instruction is added to the CRU base address (bits 3 through 14 of the workspace register 12). The result is the CRU address of the selected CRU bit.



Terms and Definitions

The following terms are used in describing the instructions of the 9900:

TERM	DEFINITION
B	Byte indicator (1=byte, 0 = word)
C	Bit count
D	Destination address register
DA	Destination address
IOP	Immediate operand
LSB(n)	Least significant (right most) bit of (n)
MSB(n)	Most significant (left most) bit of (n)
N	Don't care
PC	Program counter
Result	Result of operation performed by instruction
S	Source address register
SA	Source address
ST	Status register
STn	Bit n of status register
T_D	Destination address modifier
T_S	Source address modifier
W	Workspace register
WRn	Workspace register n
(n)	Contents of n
a→b	a is transferred to b
|n|	Absolute value of n
+	Arithmetic addition
–	Arithmetic subtraction
AND	Logical AND
OR	Logical OR
⊕	Logical exclusive OR
$\overline{n}$	Logical complement of n

Status Register

The status register contains the interrupt mask level and information pertaining to the instruction operation.

0	1	2	3	4	5	6	7 8 9 10 11	12 13 14 15
ST0	ST1	ST2	ST3	ST4	ST5	ST6	not used (=0)	ST12 ST13 ST14 ST15
L>	A>	=	C	O	P	X		Interrupt Mask

8

BIT	NAME	INSTRUCTION	CONDITION TO SET BIT TO 1
ST0	LOGICAL GREATER THAN	C,CB	If MSB(SA) = 1 and MSB(DA) = 0, or if MSB(SA) = MSB(DA) and MSB of [(DA)-(SA)] = 1
		CI	If MSB(W) = 1 and MSB of IOP = 0, or if MSB(W) = MSB of IOP and MSB of [IOP-(W)] = 1
		ABS	If (SA) ≠ 0
		All Others	If result ≠ 0
ST1	ARITHMETIC GREATER THAN	C,CB	If MSB(SA) = 0 and MSB(DA) = 1, or if MSB(SA) = MSB(DA) and MSB of [(OA)-(SA)] = 1
		CI	If MSB(W) = 0 and MSB of IOP = 1, or if MSB(W) = MSB of IOP and MSB of [IOP-(W)] = 1
		ABS	If MSB(SA) = 0 and (SA) ≠ 0
		All Others	If MSB of result = 0 and result ≠ 0

BIT	NAME	INSTRUCTION	CONDITION TO SET BIT TO 1
ST2	EQUAL	C, CB	If (SA) = (DA)
		CI	If (W) = IOP
		COC	If (SA) and $\overline{(DA)}$ = 0
		CZC	If (SA) and (DA) = 0
		TB	If CRUIN = 1
		ABS	If (SA) = 0
		All others	If result = 0
ST3	CARRY	A, AB, ABS, AI, DEC, DECT, INC, INCT, NEG, S, SB	If CARRY OUT = 1
		SLA, SRA, SRC, SRL	If last bit shifted out = 1
ST4	OVERFLOW	A, AB	If MSB(SA) = MSB(DA) and MSB of result ≠ MSB(DA)
		AI	If MSB(W) = MSB of IOP and MSB of result ≠ MSB(W)
		S, SB	If MSB(SA) ≠ MSB(DA) and MSB of result ≠ MSB(DA)
		DEC, DECT	If MSB(SA) = 1 and MSB of result = 0
		INC, INCT	If MSB(SA) = 0 and MSB of result = 1
		SLA	If MSB changes during shift
		DIV	If MSB(SA) = 0 and MSB(DA) = 1, or if MSB(SA) = MSB(DA) and MSB of [(DA)-(SA)] = 0
		ABS, NEG	If (SA) = 8000_{16}
ST5	PARITY	CB, MOVB	If (SA) has odd number of 1's
		LDCR, STCR	If 1 ≤ C ≤ 8 and (SA) has odd number of 1's
		AB, SB, SOCB, SZCB	If result has odd number of 1's
ST6	XOP	XOP	If XOP instruction is executed
ST12–ST15	INTERRUPT MASK	LIMI	If corresponding bit of IOP is 1
		RTWP	If corresponding bit of WR15 is 1

The TMS 9940 has a slightly different arrangement of its status register. Note that the first six status bits are the same as for the TMS 9900.

0	1	2	3	4	5	6	7	8 9 10 11 12 13	14 15
ST0 L>	ST1 A>	ST2 =	ST3 C	ST4 O	ST5 P	not used (= 0)	ST7 DC	not used (=0)	ST14 ST15 INTERRUPT MASK

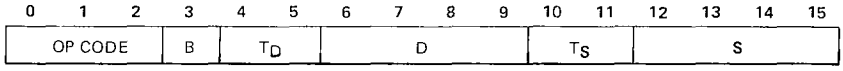
ST7	DIGIT CARRY	A,ABS,AI,DEC, DECT,INC,INCT NEG,S	If carry out of least significatn BCD Digit of most significant byte = 1
		AB,DCA,DCS,SB	If carry out of least significant BCD Digit = 1
ST14-ST15	INTERRUPT MASK	LIIM	If corresponding bit of S is 1
		LIMI	If corresponding bit of IOP is 1
		RTWP	If corresponding bit of WR 13 is 1

8

INSTRUCTIONS

DUAL OPERAND INSTRUCTIONS WITH MULTIPLE ADDRESSING MODES FOR SOURCE AND DESTINATION OPERAND

General format:



If B = 1 the operands are bytes and the operand addresses are byte addresses. If B = 0 the operands are words and the operand addresses are word addresses.

The addressing mode for each operand is determined by the T field of that operand.

T_S OR T_D	S OR D	ADDRESSING MODE	NOTES
00	0, 1, . . . 15	Workspace register	1
01	0, 1, . . . 15	Workspace register indirect	
10	0	Symbolic	4
10	1, 2, . . . 15	Indexed	2,4
11	0, 1, . . . 15	Workspace register indirect auto-increment	3

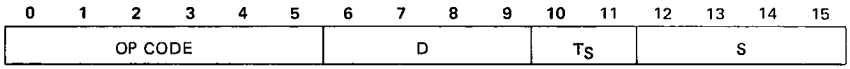
Notes: 1. When a workspace register is the operand of a byte instruction (bit 3 = 1), the left byte (bits 0 through 7) is the operand and the right byte (bits 8 through 15) is unchanged.
2. Workspace register 0 may not be used for indexing.
3. The workspace register is incremented by 1 for byte instructions (bit 3 = 1) and is incremented by 2 for word instructions (bit 3 = 0).
4. When $T_S = T_D = 10$, two words are required in addition to the instruction word. The first word is the source operand base address and the second word is the destination operand base address.

MNEMONIC	OP CODE 0 1 2	B 3	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
A	1 0 1	0	Add	Yes	0-4	(SA)+(DA) → (DA)
AB	1 0 1	1	Add bytes	Yes	0-5	(SA)+(DA) → (DA)
C	1 0 0	0	Compare	No	0-2	Compare (SA) to (DA) and set appropriate status bits
CB	1 0 0	1	Compare bytes	No	0-2,5	Compare (SA) to (DA) and set appropriate status bits
S	0 1 1	0	Subtract	Yes	0-4	(DA) − (SA) → (DA)
SB	0 1 1	1	Subtract bytes	Yes	0-5	(DA) − (SA) → (DA)
SOC	1 1 1	0	Set ones corresponding	Yes	0-2	(DA) OR (SA) → (DA)
SOCB	1 1 1	1	Set ones corresponding bytes	Yes	0-2,5	(DA) OR (SA) → (DA)
SZC	0 1 0	0	Set zeroes corresponding	Yes	0-2	(DA) AND $(\overline{SA})$ → (DA)
SZCB	0 1 0	1	Set zeroes corresponding bytes	Yes	0-2,5	(DA) AND $(\overline{SA})$ → (DA)
MOV	1 1 0	0	Move	Yes	0-2	(SA) → (DA)
MOVB	1 1 0	1	Move bytes	Yes	0-2,5	(SA) → (DA)

▶8

DUAL OPERAND INSTRUCTIONS WITH MULTIPLE ADDRESSING MODES FOR THE SOURCE OPERAND AND WORKSPACE REGISTER ADDRESSING FOR THE DESTINATION

General format:



The addressing mode for the source operand is determined by the T_S field.

T_S	S	ADDRESSING MODE	NOTES
00	0, 1, . . . 15	Workspace register	
01	0, 1, . . . 15	Workspace register indirect	
10	0	Symbolic	
10	1, 2, . . . 15	Indexed	1
11	0, 1, . . . 15	Workspace register indirect auto increment	2

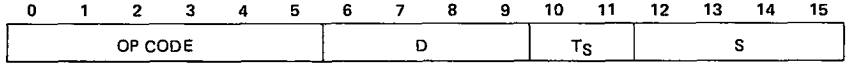
Notes: 1. Workspace register 0 may not be used for indexing.
2. The workspace register is incremented by 2.

MNEMONIC	OP CODE 0 1 2 3 4 5	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
COC	0 0 1 0 0 0	Compare ones corresponding	No	2	Test (D) to determine if 1's are in each bit position where 1's are in (SA). If so, set ST2.
CZC	0 0 1 0 0 1	Compare zeros corresponding	No	2	Test (D) to determine if 0's are in each bit position where 1's are in (SA). If so, set ST2.
XOR	0 0 1 0 1 0	Exclusive OR	Yes	0-2	(D) ⊕ (SA) → (D)
MPY	0 0 1 1 1 0	Multiply	No		Multiply unsigned (D) by unsigned (SA) and place unsigned 32-bit product in D (most significant) and D+1 (least significant). If WR15 is D, the next word in memory after WR15 will be used for the least significant half of the product.
DIV	0 0 1 1 1 1	Divide	No	4	If unsigned (SA) is less than or equal to unsigned (D), perform no operation and set ST4. Otherwise, divide unsigned (D) and (D+1) by unsigned (SA). Quotient → (D), remainder → (D+1). If D = 15, the next word in memory after WR 15 will be used for the remainder.

8

EXTENDED OPERATION (XOP) INSTRUCTION

General format:



The T_S and S fields provide multiple mode addressing capability for the source operand. When the XOP is executed, ST6 is set and the following transfers occur:

$(40_{16}+4D) \rightarrow (WP)$
$(42_{16}+4D) \rightarrow (PC)$
$SA \rightarrow$ (new WR11)
(old WP) $\rightarrow$ (new WR 13)
(old PC) $\rightarrow$ (new WR 14)
(old ST) $\rightarrow$ (new WR 15)

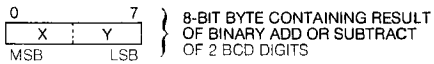
The TMS 9900 does not test interrupt requests ($\overline{INTREQ}$) upon completion of the XOP instruction. The TMS 9980A/TMS 9981 tests for reset and load but does not test for interrupt requests ($\overline{INTREQ}$) upon completion of the XOP instruction.

The TMS 9940 has the same general format for extended operations as the TMS 9900 with the differences described below.

MNEMONIC	D FIELD 6789	MEANING	RESULT COMPARED TO ZERO?	STATUS BITS AFFECTED	DESCRIPTION
DCA	0000	Decimal Correct Addition	Yes	0-3,5,7	The byte specified by SA is corrected to form 2 BCD digits as shown in Table 4
DCS	0001	Decimal Correct Subtraction	Yes	0-3,5,7	The byte specified by SA is corrected to form 2 BCD digits as shown in Table 4
LIIM	001X	Load Interrupt Mask	No	14,15	Ts must equal 0. S, Bits 14 and 15 → ST 14 and ST 15.
XOP	01XX 10XX 11XX	General XOP	No	--	$(40_{16}+4D) \rightarrow (WP)$ $(42_{16}+4D) \rightarrow (PC)$. SA → (New WR11); (Old WP) → (New WR 13), (Old PC) → (New WR 14), (Old ST) → (New WR 15); Following execution of an XOP instruction, the TMS 9940 inhibits interrupt levels 1,2, and 3 until one more instruction is executed.

▶8

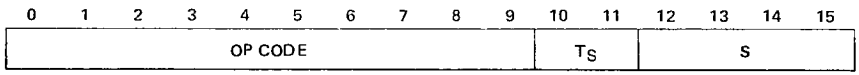
RESULT OF DCA AND DCS INSTRUCTIONS



BYTE BEFORE EXECUTION				BYTE AFTER DCA				BYTE AFTER DCS			
C	X	DC	Y	C	X	DC	Y	C	X	DC	Y
0	X<10	0	Y<10	0	X	0	Y	–	–	–	–
0	X<10	1	Y<10	0	X	0	Y+6	–	–	–	–
0	X<9	0	Y≥10	0	X+1	1	Y+6	–	–	–	–
1	X<10	0	Y<10	1	X+6	0	Y	–	–	–	–
1	X<10	1	Y<10	1	X+6	0	Y16	–	–	–	–
1	X<10	0	Y≥10	1	X+7	1	Y+6	–	–	–	–
0	X≥10	0	Y<10	1	X+6	0	Y	–	–	–	–
0	Z≥10	1	Y<10	1	X+6	0	Y+6	–	–	–	–
0	X≥9	0	Y≥10	1	X+7	1	Y+6	–	–	–	–
0	X	0	Y	–	–	–	–	0	X+10	1	Y+10
0	X	1	Y	–	–	–	–	0	X+10	0	Y
1	X	0	Y	–	–	–	–	1	X	1	Y+10
1	X	1	Y	–	–	–	–	1	X	0	Y

SINGLE OPERAND INSTRUCTIONS

General format:



The T_S and S fields provide multiple mode addressing capability for the source operand.

8◀

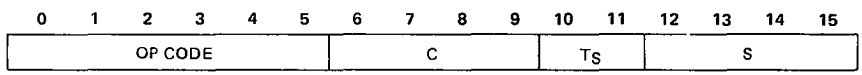
MNEMONIC	OP CODE 0 1 2 3 4 5 6 7 8 9	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
B	0 0 0 0 0 1 0 0 0 1	Branch	No	–	SA → (PC)
BL	0 0 0 0 0 1 1 0 1 0	Branch and link	No	–	(PC) → (WR11); SA → (PC)
BLWP	0 0 0 0 0 1 0 0 0 0	Branch and load workspace pointer	No	–	(SA) → (WP); (SA+2) → (PC); (old WP) → (new WR13); (old PC) → (new WR14); (old ST) → (new WR15); the interrupt input ($\overline{\text{INTREQ}}$) is not tested upon completion of the BLWP instruction. The TMS 9980A/TMS 9981 tests for reset and load but does not test for interrupt requests ($\overline{\text{INTREQ}}$) upon completion of the XOP instruction.
CLR	0 0 0 0 0 1 0 0 1 1	Clear operand	No	–	0 → (SA)
SETO	0 0 0 0 0 1 1 1 0 0	Set to ones	No	–	$FFFF_{16}$ → (SA)
INV	0 0 0 0 0 1 0 1 0 1	Invert	Yes	0-2	$\overline{(SA)}$ → (SA)
NEG	0 0 0 0 0 1 0 1 0 0	Negate	Yes	0-4	–(SA) → (SA)
ABS	0 0 0 0 0 1 1 1 0 1	Absolute value*	No	0-4	|(SA)| → (SA)
SWPB	0 0 0 0 0 1 1 0 1 1	Swap bytes	No	–	(SA), bits 0 thru 7 → (SA), bits 8 thru 15; (SA), bits 8 thru 15 → (SA), bits 0 thru 7.
INC	0 0 0 0 0 1 0 1 1 0	Increment	Yes	0-4	(SA) + 1 → (SA)
INCT	0 0 0 0 0 1 0 1 1 1	Increment by two	Yes	0-4	(SA) + 2 → (SA)
DEC	0 0 0 0 0 1 1 0 0 0	Decrement	Yes	0-4	(SA) – 1 → (SA)
DECT	0 0 0 0 0 1 1 0 0 1	Decrement by two	Yes	0-4	(SA) – 2 → (SA)
X†	0 0 0 0 0 1 0 0 1 0	Execute	No	–	Execute the instruction at SA.

*Operand is compared to zero for status bit.

†If additional memory words for the execute instruction are required to define the operands of the instruction located at SA, these words will be accessed from PC and the PC will be updated accordingly. The instruction acquisition signal (IAQ) will not be true when the 9900 accesses the instruction at SA. Status bits are affected in the normal manner for the instruction executed.

CRU Multiple-Bit Instructions

General format:



▶8

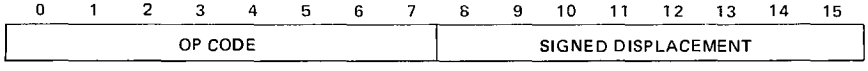
The C field specifies the number of bits to be transferred. If C = 0, 16 bits will be transferred. The CRU base register (WR 12, bits 3 through 14) defines the starting CRU bit address. The bits are transferred serially and the CRU address is incremented with each bit transfer, although the contents of WR 12 is not affected. T_S and S provide multiple mode addressing capability for the source operand. If 8 or fewer bits are transferred (C = 1 through 8), the source address is a byte address. If 9 or more bits are transferred (C = 0, 9 through 15), the source address is a word address. If the source is addressed in the workspace register indirect auto increment mode, the workspace register is incremented by 1 if C = 1 through 8, and is incremented by 2 otherwise.

MNEMONIC	OP CODE 0 1 2 3 4 5	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
LDCR	0 0 1 1 0 0	Load communcation register	Yes	0-2,5†	Beginning with LSB of (SA), transfer the specified number of bits from (SA) to the CRU.
STCR	0 0 1 1 0 1	Store communcation register	Yes	0-2,5†	Beginning with LSB of (SA), transfer the specified number of bits from the CRU to (SA). Load unfilled bit positions with 0.

†ST5 is affected only if $1 \leq C \leq 8$.

CRU Single-Bit Instructions

General format:

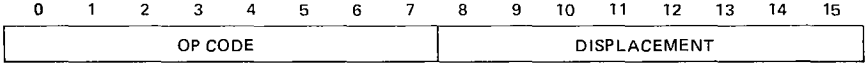


CRU relative addressing is used to address the selected CRU bit.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7	MEANING	STATUS BITS AFFECTED	DESCRIPTION
SBO	0 0 0 1 1 1 0 1	Set bit to one	–	Set the selected CRU output bit to 1.
SBZ	0 0 0 1 1 1 1 0	Set bit to zero	–	Set the selected CRU output bit to 0.
TB	0 0 0 1 1 1 1 1	Test bit	2	If the selected CRU input bit = 1, set ST2.

Jump Instructions

General format:



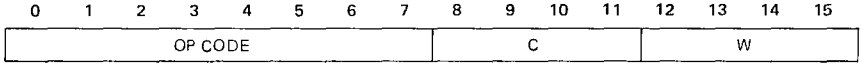
Jump instructions cause the PC to be loaded with the value selected by PC relative addressing if the bits of ST are at specified values. Otherwise, no operation occurs and the next instruction is executed since PC points to the next instruction. The displacement field is a word count to be added to PC. Thus, the jump instruction has a range of −128 to 127 words from memory-word address following the jump instruction. No ST bits are affected by jump instruction.

8◀

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7	MEANING	ST CONDITION TO LOAD PC
JEQ	0 0 0 1 0 0 1 1	Jump equal	ST2 = 1
JGT	0 0 0 1 0 1 0 1	Jump greater than	ST1 = 1
JH	0 0 0 1 1 0 1 1	Jump high	ST0 = 1 and ST2 = 0
JHE	0 0 0 1 0 1 0 0	Jump high or equal	ST0 = 1 or ST2 = 1
JL	0 0 0 1 1 0 1 0	Jump low	ST0 = 0 and ST2 = 0
JLE	0 0 0 1 0 0 1 0	Jump low or equal	ST0 = 0 or ST2 = 1
JLT	0 0 0 1 0 0 0 1	Jump less than	ST1 = 0 and ST2 = 0
JMP	0 0 0 1 0 0 0 0	Jump unconditional	unconditional
JNC	0 0 0 1 0 1 1 1	Jump no carry	ST3 = 0
JNE	0 0 0 1 0 1 1 0	Jump not equal	ST2 = 0
JNO	0 0 0 1 1 0 0 1	Jump no overflow	ST4 = 0
JOC	0 0 0 1 1 0 0 0	Jump on carry	ST3 = 1
JOP	0 0 0 1 1 1 0 0	Jump odd parity	ST5 = 1

Shift Instructions

General format:

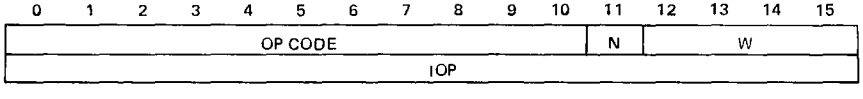


If C = 0, bits 12 through 14 of WR0 contain the shift count. If C = 0 and bits 12 through 15 of WR0 = 0, the shift count is 16.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
SLA	0 0 0 0 1 0 1 0	Shift left arithmetic	Yes	0-4	Shift (W) left. Fill vacated bit positions with 0.
SRA	0 0 0 0 1 0 0 0	Shift right arithmetic	Yes	0-3	Shift (W) right. Fill vacated bit positions with original MSB of (W).
SRC	0 0 0 0 1 0 1 1	Shift right circular	Yes	0-3	Shift (W) right. Shift previous LSB into MSB.
SRL	0 0 0 0 1 0 0 1	Shift right logical	Yes	0-3	Shift (W) right. Fill vacated bit positions with 0's.

Immediate Register Instructions

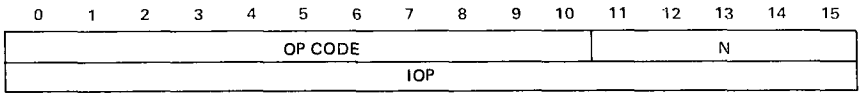
▸8 General format:



MNEMONIC	OP CODE 0 1 2 3 4 5 6 7 8 9 10	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
AI	0 0 0 0 0 0 1 0 0 0 1	Add immediate	Yes	0-4	(W) + IOP → (W)
ANDI	0 0 0 0 0 0 1 0 0 1 0	AND immediate	Yes	0-2	(W) AND IOP → (W)
CI	0 0 0 0 0 0 1 0 1 0 0	Compare immediate	Yes	0-2	Compare (W) to IOP and set appropriate status bits
LI	0 0 0 0 0 0 1 0 0 0 0	Load immediate	Yes	0-2	IOP → (W)
ORI	0 0 0 0 0 0 1 0 0 1 1	OR immediate	Yes	0-2	(W) OR IOP → (W)

Internal Register Load Immediate Instructions

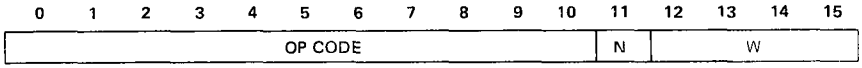
General format:



MNEMONIC	OP CODE 0 1 2 3 4 5 6 7 8 9 10	MEANING	DESCRIPTION
LWPI	0 0 0 0 0 0 1 0 1 1 1	Load workspace pointer immediate	IOP → (WP), no ST bits affected
LIMI	0 0 0 0 0 0 1 1 0 0 0	Load interrupt mask	IOP, bits 12 thru 15 → ST12 thru ST15

Internal Register Store Instructions

General format:

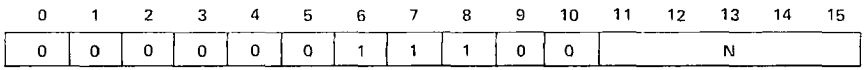


No ST bits are affected.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7 8 9 10	MEANING	DESCRIPTION
STST	0 0 0 0 0 0 1 0 1 1 0	Store status register	(ST) → (W)
STWP	0 0 0 0 0 0 1 0 1 0 1	Store workspace pointer	(WP) → (W)

Return Workspace Pointer (RTWP) Instruction

General format:

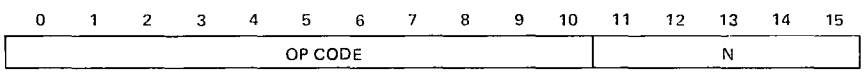


The RTWP instruction causes the following transfers to occur:

(WR 15)→(ST)
(WR 14)→(PC)
(WR 13)→(WP)

External Instructions

General format:

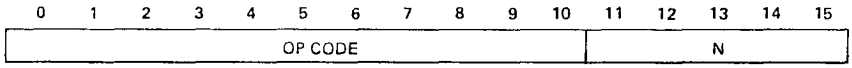


External instructions cause the three most-significant address lines (A0 through A2) to be set to the below-described levels and the CRUCLK line to be pulsed, allowing external control functions to be initiated.

MNEMONIC	OP CODE 0 1 2 3 4 5 6 7 8 9 10	MEANING	STATUS BITS AFFECTED	DESCRIPTION	ADDRESS BUS A0	A1	A2
IDLE	0 0 0 0 0 0 1 1 0 1 0	Idle		Suspend TMS 9900 instruction execution until an interrupt, $\overline{\text{LOAD}}$, or $\overline{\text{RESET}}$ occurs	L	H	L
RSET	0 0 0 0 0 0 1 1 0 1 1	Reset	12–15	0 → ST12 thru ST15	L	H	H
CKOF	0 0 0 0 0 0 1 1 1 1 0	User defined		— — —	H	H	L
CKON	0 0 0 0 0 0 1 1 1 0 1	User defined		— — —	H	L	H
LREX	0 0 0 0 0 0 1 1 1 1 1	User defined		— — —	H	H	H

IDLE INSTRUCTION — TMS 9940

General format:



The IDLE instruction stops the TMS 9940 until an interrupt or $\overline{\text{RESET}}$ occurs. See the *Power Down* section for use of the IDLE instruction.

▸8

Peripheral and Interface Circuits

8

1. INTRODUCTION

1.1 DESCRIPTION

The TMS 9901 Programmable Systems Interface (PSI) is a multifunctional component designed to provide low cost interrupt and I/O ports and an interval timer for TMS 9900-family microprocessor systems. The TMS 9901 is fabricated using N-channel silicon-gate MOS technology. The TMS 9901 is TTL-compatible on all inputs and outputs, including the power supply (+5 V) and single-phase clock.

1.2 KEY FEATURES

- Low Cost
- 9900-Family Peripheral
- Performs Interrupt and I/O Interface functions:
  - — Six Dedicated Interrupt Lines
  - — Seven Dedicated I/O Lines
  - — Nine Programmable Lines as I/O or Interrupt
  - — Up to 15 Interrupt Lines
  - — Up to 22 Input Lines
  - — Up to 16 Output Lines
- Easily Cascaded for Expansion
- Interval or Event Timer
- Single 5 V Power Supply
- All Inputs and Outputs TTL-Compatible
- Standard 40-Pin Plastic or Ceramic Package
- N-Channel Silicon-Gate MOS Technology.

1.3 APPLICATION OVERVIEW

The following example of a typical application may help introduce the user to the TMS 9901 PSI. Figure 1 is a block diagram of a typical application. Each of the ideas presented below is described in more detail in later sections of this manual.

The TMS 9901 PSI interfaces to the CPU through the Communications Register Unit (CRU) and the interrupt control lines as shown in Figure 1. The TMS 9901 occupies 32 bits of CRU input and output space. The five least significant bits of address bus are connected to the S lines of the PSI to address one of the 32 CRU bits of the TMS 9901. The most significant bits of the address bus are decoded on CRU cycles to select the PSI by taking its chip enable ($\overline{CE}$) line active (LOW).

Interrupt inputs to the TMS 9901 PSI are synchronized with $\bar{\phi}$, inverted, and then ANDed with the appropriate mask bit. Once every $\bar{\phi}$ clock time, the prioritizer looks at the 15 interrupt input AND gates and generates the interrupt control code. The interrupt control code and the interrupt request line constitute the interrupt interface to the CPU.

After reset all I/O ports are programmed as inputs. By writing to any I/O port, that port will be programmed as an output port until another reset occurs, either software or hardware. Data at the input pins is buffered on to the TMS 9901. Data to the output ports is latched and then buffered off-chip by the PSI's MOS-to-TTL buffers.

The interval timer on the TMS 9901 is accessed by writing a ONE to select bit zero, (control bit) which puts the PSI CRU interface in the clock mode. Once in the clock mode the 14-bit clock contents can be read or written. Writing to the clock register will reinitialize the clock and cause it to start decrementing. When the clock counts to zero, it will cause an interrupt and reload to its initial value. Reading the clock contents permits the user to see the decrementer contents at that point in time just before entering the clock mode. The clock read register is not updated when the PSI is in the clock mode.

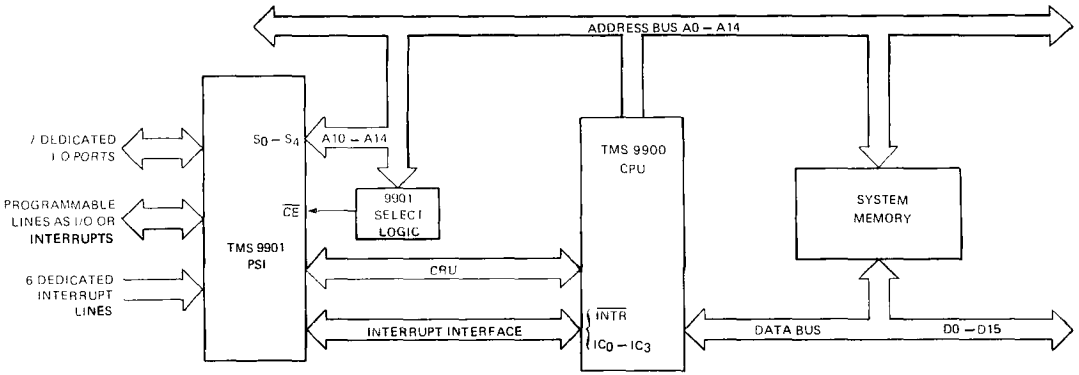


FIGURE 1– TYPICAL TMS 9901 PROGRAMMABLE SYSTEM INTERFACE (PSI) APPLICATION

2. ARCHITECTURE

The architecture of the TMS 9901 Programmable Systems Interface (PSI) is designed to provide the user maximum flexibility when designating system I/O ports and interrupts. The TMS 9901 can be divided into four subsystems: CRU interface, interrupt interface, input/output interface, and interval timer. Figure 2 is a general block diagram of the TMS 9901 internal architecture. Each of the subsystems of the PSI is discussed in detail in subsequent paragraphs.

2.1 CRU Interface

The CPU communicates with the TMS 9901 PSI via the CRU. The TMS 9901 occupies 32 bits in CRU read space and 32 bits in CRU write space. Table 1 shows the mapping for CRU bit addresses to TMS 9901 functions.

The CRU interface consists of five address select lines (S0-S4), chip enable ($\overline{CE}$), and the three CRU lines (CRUIN, CRUOUT, CRUCLK). The select lines (S0-S4) are connected to the five least significant bits of the address bus; for a TMS 9900 system S0-S4 are connected to A10-A14, respectively. Chip enable ($\overline{CE}$) is generated by decoding the most significant bits of the address bus on CRU cycles; for a 9900 based system address bits 0-9 would be decoded. When $\overline{CE}$ goes active (LOW), the five select lines point to the CRU bit being accessed. When $\overline{CE}$ is inactive (HIGH), the PSI's CRU interface is disabled.

NOTE

When $\overline{CE}$ is inactive (HIGH) the 9901 sets its CRUIN pin to high impedance and disables CRUCLK from coming on chip. This means that CRUIN can be used as an OR tied bus. When $\overline{CE}$ is high the 9901 will still see the select lines, but no command action is taken.

In the case of a write operation, the TMS 9901 strobes data off the CRUOUT line with CRUCLK. For a read operation, the data is sent to the CPU on the CRUIN line.

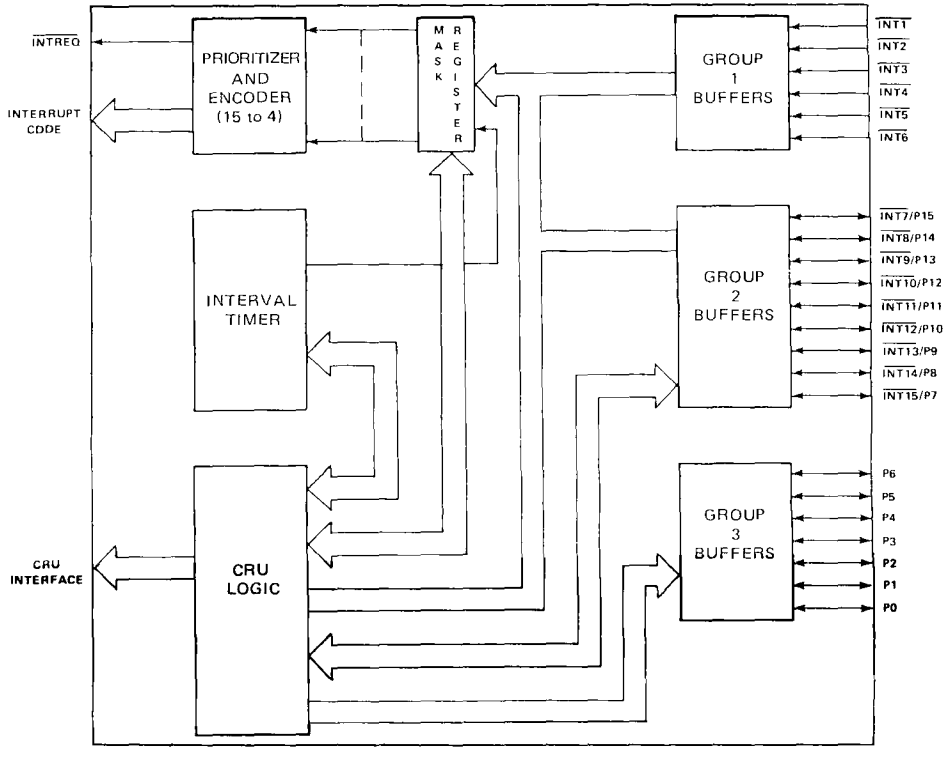


FIGURE 2–TMS 9901 PSI BLOCK DIAGRAM

Several TMS 9901 devices may be cascaded to expand I/O and interrupt handling capability simply by connecting all CRU and address select lines in parallel and providing each device with a unique chip enable signal: the chip enable ($\overline{CE}$) is generated by decoding the high-order address bits (A0-A9) on CRU cycles.

▶8 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.

TABLE 1

SELECT BIT ASSIGNMENTS

SELECT BIT	S_0 S_1 S_2 S_3 S_4	CRU Read Data	CRU Write Data
0	0 0 0 0 0	CONTROL BIT(1)	CONTROL BIT(1)
1	0 0 0 0 1	$\overline{INT}$1/CLK1(2)	Mask 1/CLK1(3)
2	0 0 0 1 0	$\overline{INT}$2/CLK2	Mask 2/CLK2
3	0 0 0 1 1	$\overline{INT}$3/CLK3	Mask 3/CLK3
4	0 0 1 0 0	$\overline{INT}$4/CLK4	Mask 4/CLK4
5	0 0 1 0 1	$\overline{INT}$5/CLK5	Mask 5/CLK5
6	0 0 1 1 0	$\overline{INT}$6/CLK6	Mask 6/CLK6
7	0 0 1 1 1	$\overline{INT}$7/CLK7	Mask 7/CLK7
8	0 1 0 0 0	$\overline{INT}$8/CLK8	Mask 8/CLK8
9	0 1 0 0 1	$\overline{INT}$9/CLK9	Mask 9/CLK9
10	0 1 0 1 0	$\overline{INT}$10/CLK10	Mask 10/CLK10
11	0 1 0 1 1	$\overline{INT}$11/CLK11	Mask 11/CLK11
12	0 1 1 0 0	$\overline{INT}$12/CLK12	Mask 12/CLK12
13	0 1 1 0 1	$\overline{INT}$13/CLK13	Mask 13/CLK13
14	0 1 1 1 0	$\overline{INT}$14/CLK14	Mask 14/CLK14
15	0 1 1 1 1	$\overline{INT}$15/INTREQ(7)	Mask 15/$\overline{RST2}$(4)
16	1 0 0 0 0	PO Input(5)	PO Output(6)
17	1 0 0 0 1	P1 Input	P1 Output
18	1 0 0 1 0	P2 Input	P2 Output
19	1 0 0 1 1	P3 Input	P3 Output
20	1 0 1 0 0	P4 Input	P4 Output
21	1 0 1 0 1	P5 Input	P5 Output
22	1 0 1 1 0	P6 Input	P6 Output
23	1 0 1 1 1	P7 Input	P7 Output
24	1 1 0 0 0	P8 Input	P8 Output
25	1 1 0 0 1	P9 Input	P9 Output
26	1 1 0 1 0	P10 Input	P10 Output
27	1 1 0 1 1	P11 Input	P11 Output
28	1 1 1 0 0	P12 Input	P12 Output
29	1 1 1 0 1	P13 Input	P13 Output
30	1 1 1 1 0	P14 Input	P14 Output
31	1 1 1 1 1	P15 Input	P15 Output

NOTES
(1) 0 = Interrupt Mode 1 = Clock Mode
(2) Data present on $\overline{INT}$ input pin (or clock value) will be read regardless of mask value.
(3) While in the Interrupt Mode (Control Bit = 0) writing a "1" into mask will enable interrupt, a "0" will disable.
(4) Writing a zero to bit 15 while in the clock mode (Control Bit = 1) executes a software reset of the I/O pins.
(5) Data present on the pin will be read. Output data can be read without affecting the data.
(6) Writing data to the port will program the port to the output mode and output the data.
(7) INTREQ is the inverted status of the $\overline{INTREQ}$ pin.

8

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 most significant bit of R12 is not brought out of the CPU. During CRU cycles, the memory control lines (MEMEN, WE, and DBIN) are all inactive; $\overline{\text{MEMEN}}$ being inactive (HIGH) indicates the address is not a memory address and therefore is a CRU address or external instruction code. Also, when 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 $\overline{\text{MEMEN}}$ must be inactive (HIGH) to indicate a CRU cycle.

2.2 Interrupt Interface

A block diagram of the interrupt control section is shown in Figure 3. The interrupt inputs (six dedicated, $\overline{\text{INT1}}$-$\overline{\text{INT6}}$, and nine programmable) are sampled on the falling edge of $\overline{\phi}$ and latched onto the chip for one $\overline{\phi}$ time by the SYNC LATCH, each $\overline{\phi}$ time. The output of the sync latch is inverted (interrupts are LOW active) and ANDed with its respective mask bit (MASK = 1, INTERRUPT ENABLED). On the rising edge of $\overline{\phi}$, the prioritizer and encoder senses the masked interrupts and produces a four-bit encoding of the highest priority interrupt present (see Tables 2 and 3). The four-bit prioritized code and $\overline{\text{INTREQ}}$ are latched off-chip with a sync latch on the falling edge of the next $\overline{\phi}$, which ensures proper synchronization to the processor.

Once an interrupt goes active (LOW), it should stay active until the appropriate interrupt service routine explicitly turns off the interrupt. If an interrupt is allowed to go inactive before the interrupt service routine is entered, an erroneous interrupt code could be sent to the processor. A total of five clock cycles occur between the time the CPU samples the INTREQ line and the time it samples the IC0-IC3 lines. For example, if an interrupt is active and the CPU recognizes that an interrupt is pending, *but before the CPU can sample the interrupt control lines the interrupt goes inactive,* the interrupt control lines will contain an incorrect code.

The interrupt mask bits on the TMS 9901 PSI are individually set or reset under software control. Any unused interrupt line should have its associated mask disabled to avoid false interrupts: To do this, the control bit (CRU bit zero), is first set to a zero for interrupt mode operation. Writing to TMS 9901 CRU bits 1-15 will enable or disable interrupts 1-15, respectively. Writing a one to an interrupt mask will *enable* that interrupt; writing a zero will *disable* that interrupt. Upon application of $\overline{\text{RST1}}$ (power-up reset), all mask bits are reset (LOW), the interrupt code is forced to all zeros, and INTREQ is held HIGH. Reading TMS 9901 CRU bits 1-15 indicates the status of the respective interrupt inputs; thus, the designer can employ the unused (disabled) interrupt input lines as data inputs (true data in).

2.3 Input/Output Interface

A block diagram of the TMS 9901 I/O interface is shown in Figure 4. Up to 16 individually controlled, I/O ports are available (seven dedicated, P0-P6, and nine programmable) and, as discussed above, the unused dedicated interrupt lines also can be used as input lines (true data in). Thus the 9901 can be configured to have more than 16 inputs. $\overline{\text{RST1}}$ (power-up reset) will program all I/O ports to input mode. Writing data to a port will automatically switch that port to the output mode. Once programmed as an output, a port will remain in output mode until $\overline{\text{RST1}}$ or $\overline{\text{RST2}}$ (command bit) is executed. An output port can be read and indicates the present state of the pin. A pin programmed to the output mode *cannot* be used as an input pin: *Applying an input current to an output pin may cause damage to the TMS 9901.* The TMS 9901 outputs are latched and buffered off-chip, and inputs are buffered onto the chip. The output buffers are MOS-to-TTL buffers and can drive two standard TTL loads.

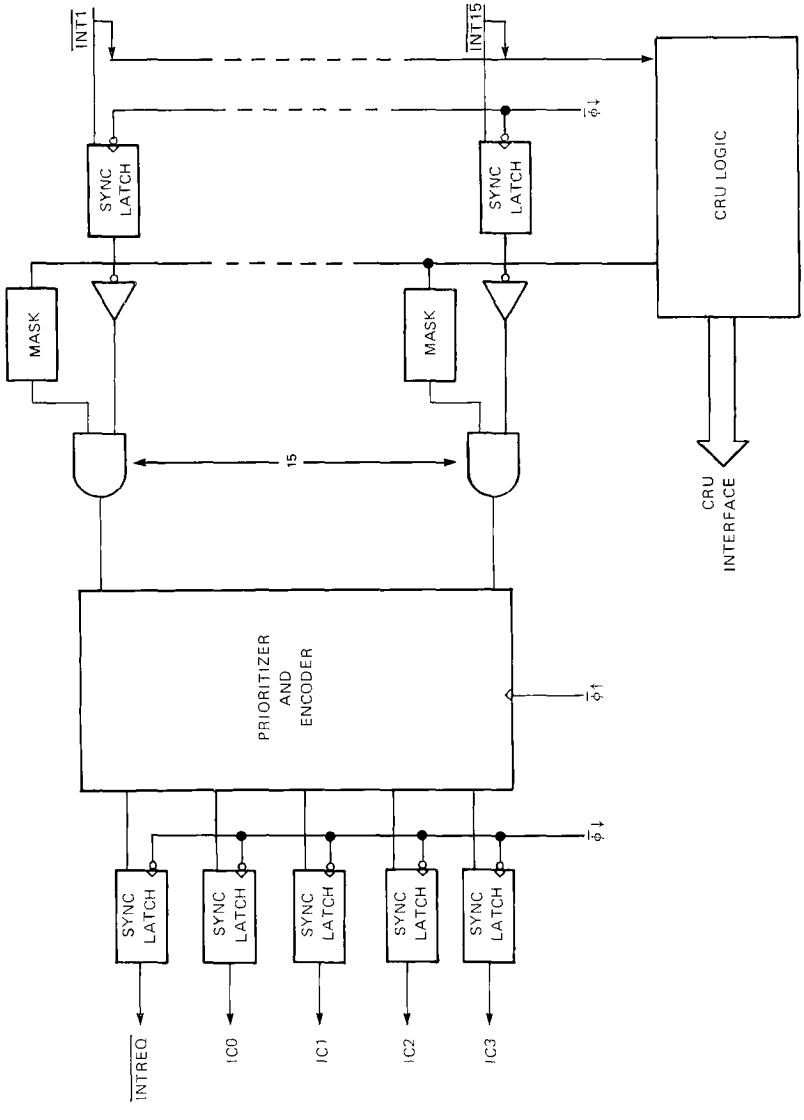


FIGURE 3– TMS 9901 PSI INTERRUPT CONTROL SECTION BLOCK DIAGRAM

8

TABLE 2
INTERRUPT CODE GENERATION

INTERRUPT/STATE	PRIORITY	I_{C0}	I_{C1}	I_{C2}	I_{C3}	$\overline{INTREQ}$
$\overline{RST\ 1}$	–	0	0	0	0	1
$\overline{INT}$ 1	1 (HIGHEST)	0	0	0	1	0
$\overline{INT}$ 2	2	0	0	1	0	0
$\overline{INT}$ 3/CLOCK	3	0	0	1	1	0
$\overline{INT}$ 4	4	0	1	0	0	0
$\overline{INT}$ 5	5	0	1	0	1	0
$\overline{INT}$ 6	6	0	1	1	0	0
$\overline{INT}$ 7	7	0	1	1	1	0
$\overline{INT}$ 8	8	1	0	0	0	0
$\overline{INT}$ 9	9	1	0	0	1	0
$\overline{INT}$ 10	10	1	0	1	0	0
$\overline{INT}$ 11	11	1	0	1	1	0
$\overline{INT}$ 12	12	1	1	0	0	0
$\overline{INT}$ 13	13	1	1	0	1	0
$\overline{INT}$ 14	14	1	1	1	0	0
$\overline{INT}$ 15	15 (LOWEST)	1	1	1	1	0
NO INTERRUPT	–	1	1	1	1	1

TABLE 3
TMS 9980A OR TMS 9981 INTERRUPT LEVEL DATA

INTERRUPT CODE (IC0-IC2)	FUNCTION	VECTOR LOCATION (MEMORY ADDRESS IN HEX)	DEVICE ASSIGNMENT	INTERRUPT MASK VALUES TO ENABLE (ST12 THROUGH ST15)
1 1 0	Level 4	0 0 1 0	External Device	4 Through F
1 0 1	Level 3	0 0 0 C	External Device	3 Through F
1 0 0	Level 2	0 0 0 8	External Device	2 Through F
0 1 1	Level 1	0 0 0 4	External Device	1 Through F
0 0 1	Reset	0 0 0 0	Reset Stimulus	Don't Care
0 1 0	Load	3 F F C	Load Stimulus	Don't Care
0 0 0	Reset	0 0 0 0	Reset Stimulus	Don't Care
1 1 1	No-Op	.....	.....	

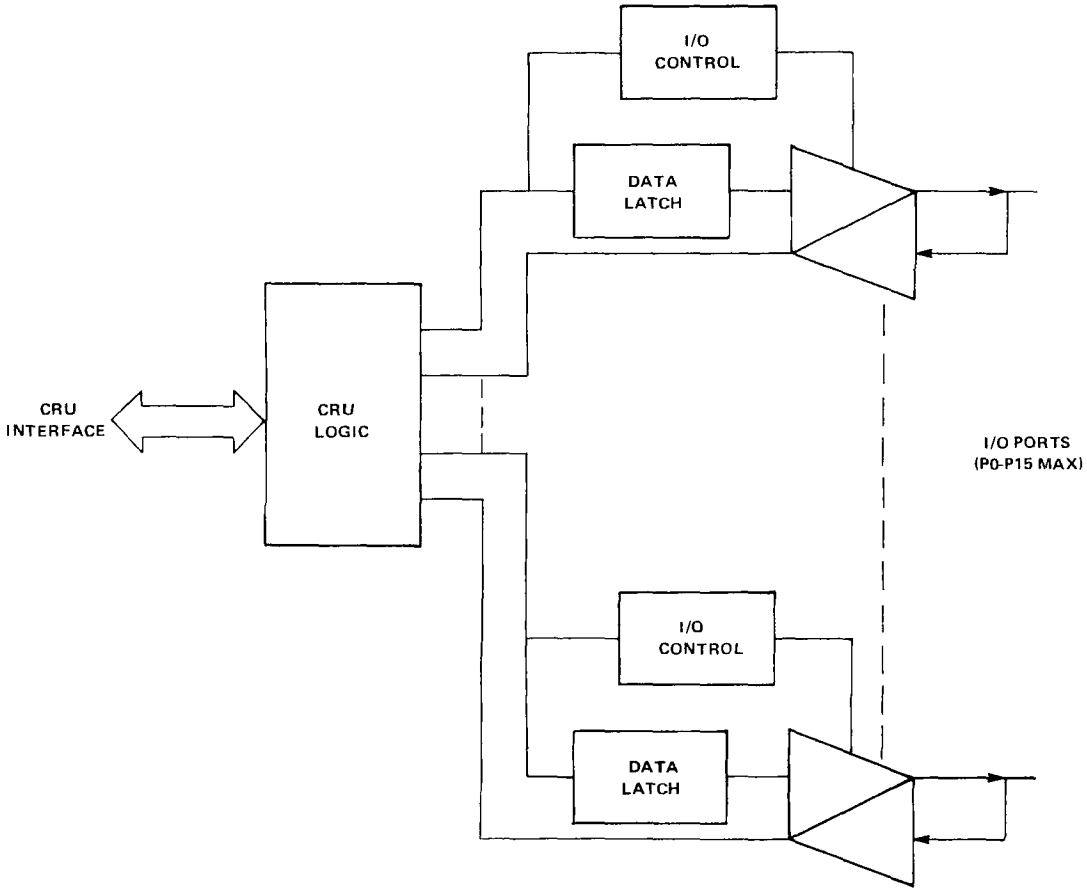


FIGURE 4–TMS 9901 I/O INTERFACE SECTION

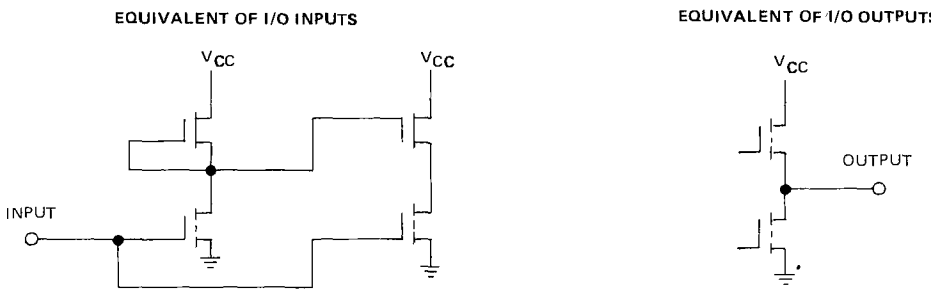


FIGURE 5 – INPUT AND OUTPUT EQUIVALENTS

8◀

2.4 Programmable Ports

A total of nine pins ($\overline{INT}$7/P15-$\overline{INT}$15/P7) on the TMS 9901 are user-programmable as either I/O ports or interrupts. These pins will assume all characteristics of the type pin they are programmed to be (as described in Sections 2.2 and 2.3). Any pin which is not being used for interrupt should have the appropriate interrupt mask disabled (mask = 0) to avoid erroneous interrupts to the CPU. To program one of the pins as an interrupt, its interrupt mask simply is enabled and the line may be used as if it were one of the dedicated interrupt lines. To program a pin as an I/O port, disable the interrupt mask and use that pin as if it were one of the dedicated I/O ports.

2.5 Interval Timer

Figure 6 is a block diagram of the TMS 9901 interval timer section. The clock consists of a 14-bit counter that decrements at a rate of $f(\bar{\phi})/64$ (at 3 MHz this results in a maximum interval of 349 milliseconds with a resolution of 21.3 microseconds). The clock can be used as either an interval timer or an event timer. To access the clock, select bit zero (control bit) must be set to a one. The clock is enabled to ca[illegible] nterrupts by writing a nonzero value to it and is then disabled from interrupting by writing zero to it or by a [illegible]1. The clock starts operating at no more than two $\bar{\phi}$ times after it is loaded. When the clock decrementer is running, it will decrement down to zero and issue a level-3 interrupt. The decrementer, when it becomes zero, will also be reloaded from the clock register and decrementing will start again. (The zero state is counted as any other decrementer state.) The decrementer always runs, but it will not issue interrupts unless enabled; of course, the contents of the unenabled clock read register are meaningless.

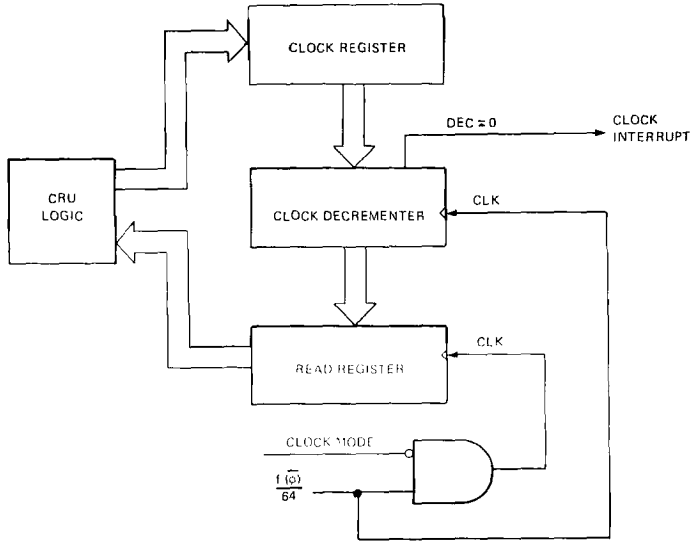


FIGURE 6—TMS 9901 INTERVAL TIMER SECTION

The clock is accessed by writing a one into the control bit (TMS 9901 CRU bit zero) to force CRU bits 1-15 to clock mode. Writing a nonzero value into the clock register then enables the clock and sets its time period. When the clock is enabled, it interrupts on level 3 and external level-3 interrupts are disabled. The mask for level 3 in the PSI must be set to a one so that the processor will see the clock interrupt. When the clock interrupt is active, the clock mask (mask bit 3) must be written into with either a one or zero to clear the interrupt; writing a zero also disables further interrupts.

If a new clock value is required, a new 14-bit clock start value can be programmed by executing a CRU write operation to the clock register. During programming, the decrementer is restarted with the current start value after each start value bit is written. A timer restart is easily implemented by writing a single bit to any of the clock bits. The clock is disabled by $\overline{RST1}$ (power up reset) or by writing a zero value into the clock register; $\overline{RST2}$ does not affect the clock.

The clock read register is updated every time the decrementer decrements when the TMS 9901 is not in clock mode. There are two methods to leave the clock mode : first, a zero is written to the control bit; or second, a TMS 9901 select bit greater than 15 is accessed. Note that when $\overline{CE}$ is inactive(HIGH), the PSI is not disabled from seeing the select lines. As the CPU is addressing memory, A10-A14 could very easily have a value of 15 or greater — A10-A14 are connected to the select lines; therefore, the TMS 9901 interval timer section can "think" it is out of clock mode and update the clock read register. Very simply, this means that a value cannot be locked into the clock read register by writing a one to CRU select bit zero (the control bit). The 9901 must be out of clock mode for at least one timer period to ensure that the contents of the clock read register has been updated. This means that to read the most recent contents of the decrementer, just before reading, the TMS 9901 *must* not be in the clock mode. The only sure way to manipulate clock mode is to use the control bit (select bit zero). When clock mode is reentered to access the clock read register, updating of the read register will cease. This is done so that the contents of the clock read register will not change while it is being accessed.

2.6 Power-Up Considerations

During hardware re· ·· $\overline{RST1}$ must be active (LOW) for a minimum of two clock cycles to force the TMS 9901 into a known state. $\overline{R}$ · T1 will disable all ·· ··upts, disable the clock, program all I/O ports to the input mode, and force IC0-IC3 to all zeros with $\overline{INT}$ held HIGH. The system software must enable the appropriate interrupts, program the clock, and configure the I/O ports as required. After initial power-up the TMS 9901 is accessed only as needed to service the clock, e· ·· (disable) interrupts, or read (write) data to the I/O ports. The I/O ports can be reconfigured by use of the $\overline{R}$ 2 software reset command bit.

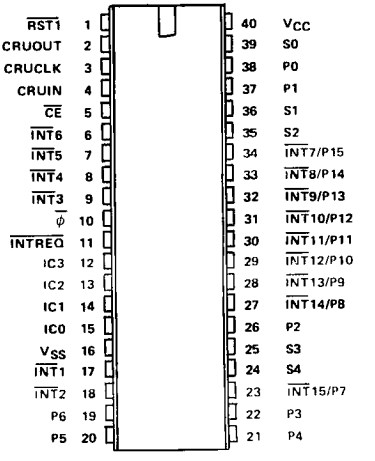
8◀

2.7 Pin Descriptions

Table 4 defines the TMS 9901 pin assignments and describes the function of each pin.

TABLE 4
TMS 9901 PIN ASSIGNMENTS AND FUNCTIONS

SIGNATURE	PIN	I/O	DESCRIPTION
$\overline{INTREQ}$	11	OUT	INTERRUPT Request. When active (low) $\overline{INTREQ}$ indicates that an enabled interrupt has been received. $\overline{INTREQ}$ will stay active until all enabled interrupt inputs are removed.
IC0 (MSB)	15	OUT	Interrupt Code lines. IC0-IC3 output the binary code corresponding to the highest priority enabled interrupt. If no enabled interrupts are active IC0—IC3 = (1,1,1,1).
IC1	14	OUT	
IC2	13	OUT	
IC3 (LSB)	12	OUT	
$\overline{CE}$	5	IN	Chip Enable. When active (low) data may be transferred through the CRU interface to the CPU. $\overline{CE}$ has no effect on the interrupt control section.
S0	39	IN	Address select lines. The data bit being accessed by the CRU interface is specified by the 5-bit code appearing on S0-S4.
S1	36	IN	
S2	35	IN	
S3	25	IN	
S4	24	IN	
CRUIN	4	OUT	CRU data in (to CPU). Data specified by S0-S4 is transmitted to the CPU by CRUIN. When $\overline{CE}$ is not active CRUIN is in a high-impedance state.
CRUOUT	2	IN	CRU data out (from CPU). When $\overline{CE}$ is active, data present on the CRUOUT input will be sampled during CRUCLK and written into the command bit specified by S0-S4.
CRUCLK	3	IN	CRU Clock (from CPU). CRUCLK specifies that valid data is present on the CRUOUT line.
$\overline{RST1}$	1	IN	Power Up Reset. When active (low) $\overline{RST1}$ resets all interrupt masks to "0", resets IC0 – IC3 = (0, 0, 0, 0), $\overline{INTEREQ}$ = 1 disables the clock, and programs all I/O ports to inputs. $\overline{RST1}$ has a Schmitt-triger input to allow implementation with an RC circuit as shown in Figure 7.
V_{CC}	40		Supply Voltage +5 V nominal.
V_{SS}	16		Ground Reference
$\bar{\phi}$	10	IN	System clock ($\bar{\phi}3$ in TMS 9900 system, $\overline{CKOUT}$ in TMS 9980 system).
$\overline{INT1}$	17	IN	Group 1, interrupt inputs. When active (Low) the signal is ANDed with its corresponding mask bit and if enabled sent to the interrupt control section. $\overline{INT1}$ has highest priority.
$\overline{INT2}$	18	IN	
$\overline{INT3}$	9	IN	
$\overline{INT4}$	8	IN	
$\overline{INT5}$	7	IN	
$\overline{INT6}$	6	IN	
$\overline{INT7}$/ P15	34	I/O	Group 2, programmable interrupt (active low) or I/O pins (true logic). Each pin is individually programmable as an interrupt, an input port, or an output port.
$\overline{INT8}$/ P14	33	I/O	
$\overline{INT9}$/ P13	32	I/O	
$\overline{INT10}$/P12	31	I/O	
$\overline{INT11}$/P11	30	I/O	
$\overline{INT12}$/P10	29	I/O	
$\overline{INT13}$/P9	28	I/O	
$\overline{INT14}$/P8	27	I/O	
$\overline{INT15}$/P7	23	I/O	
P0	38	I/O	Group 3, I/O ports (true logic). Each pin is individually programmable as an input port or an output port.
P1	37	I/O	
P2	26	I/O	
P3	22	I/O	
P4	21	I/O	
P5	20	I/O	
P6	19	I/O	



3. APPLICATIONS

3.1 Hardware Interface

Figure 7 illustrates the use of a TMS 9901 PSI in a TMS 9900 system. The TIM 9904 clock generator/driver syncs the $\overline{\text{RESET}}$ for both the TMS 9901 and the CPU. The RC circuit on the TIM 9904 provides the power-up and pushbutton $\overline{\text{RESET}}$ input to the clock chip. Address lines A0-A9 are decoded on CRU cycles to select the TMS 9901. Address lines A10-A14 are sent directly to PSI select lines S0-S4, respectively, to select which TMS 9901 CRU bit is to be accessed.

Figure 8 illustrates the use of a TMS 9901 with a TMS 9985 CPU. No TIM 9904 is needed with the TMS 9985, so the reset circuitry is connected directly to the system reset line. The clock ($\overline{\phi}$) then comes from the TMS 9985. All other circuitry is identical to the TMS 9900 system.

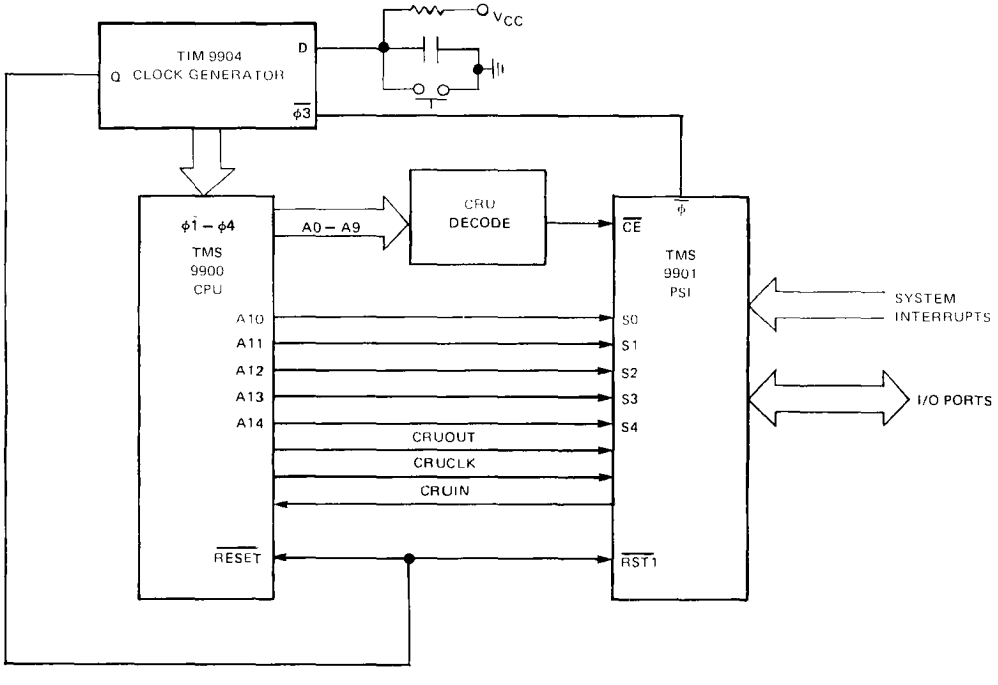


FIGURE 7–TMS 9900/TMS 9901 INTERFACE

8

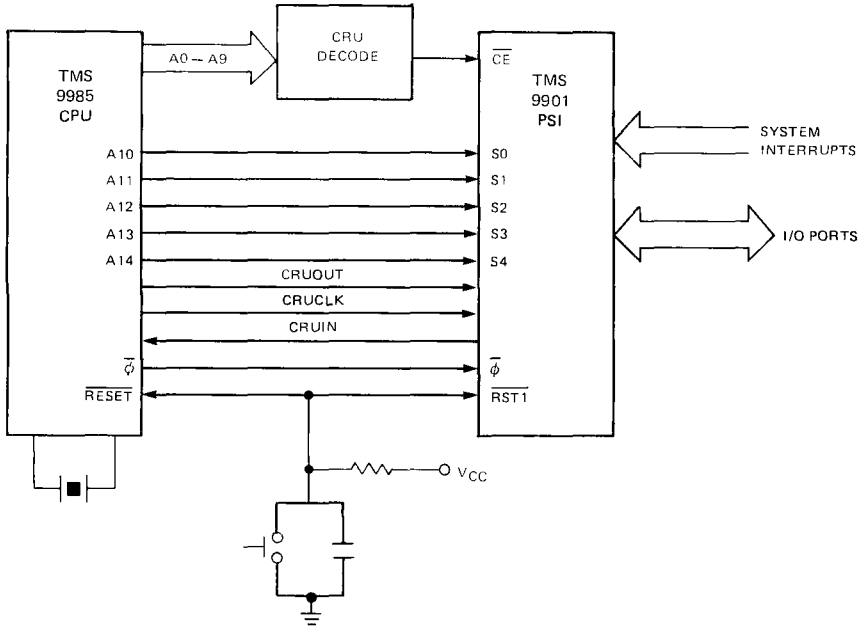


FIGURE 8–TMS 9985/TMS 9901 INTERFACE

3.2 Software Interface

Figure 9 lists the TMS 9900 code needed to control the TMS 9901 PSI. The code initializes the PSI to an eight-bit input port, an eight-bit output port, and enables interrupt levels 1-6. The six dedicated interrupt pins are all used for interrupts; their mask bits are set ON. The nine programmable pins are all used as I/O ports; mask bits 7-15 remain reset. P0-P7 are programmed as an eight-bit output port, and P8-P15 are programmed as an eight-bit input port.

8

Some code is added to read the contents of the clock read-register. The SBZ instruction takes the TMS 9901 out of clock mode long enough for the clock read register to be updated with the most recent decrementer value. When clock mode is reentered, the decrementer will cease updating the clock read-register so that the contents of the register will not be changing during a read operation.

The second section of code is typical code found in a clock interrupt service routine. All interrupts initially are disabled by the routine. These functions are not necessary, but are usually done to ensure system integrity. The interrupt mask should be restored as soon as the sensitive processing is complete. The interrupt is counted in the variable COUNT and is then cleared by writing a one to mask bit 3. If a zero is written to mask bit 3 to clear the interrupt, clock interrupt will be disabled from that point onward, but the clock will continue to run.

ASSUMPTION:

- System uses clock at maximum interval (349 msec @ 3MHz)
- Interrupts 1-6 are used
- Eight bits are used as an output port , P0 –P7
- Eight bits are used as an input port , P8 – P15
- $\overline{\text{RST1}}$ (power-up reset) has been applied
- The most significant byte of R1 contains data to be output.

```
        LI      R12, PSIBAS     Set up CRU base to point to 9901
        LDCR    @CLKSET, 0      16-bit transfer, set clock to max interval

        LDCR    @INTSET, 7      Enter interrupt mode and enable interrupts 1 – 6

        LI      R12, PSIBAS+32  Set CRU base to I/O ports – output
        LDCR    R1, 8           Output byte from R1, program ports 0 – 7 as output

        LI      R12, PSIBAS+48  Set CRU base to I/O ports – input
        STCR    R2, 8           Store a byte from input port into MSBT of R2

        LI      R12, PSIBAS     Set CRU base to 9901
        SBZ     0               Leave clock mode so decremented contents can be latched
        INCT    R12             Set CRU base to clock read register
        SBO     -1              Enter clock mode
        STCR    R3, 14          Read 14-bit clock read register contents into R3

CLKSET  DATA    >FFFF
INTSET  BYTE    >7E

CLKINT  $                       Clock interrupt service routine – level 3
        LIMI    0               Disable interrupts at CPU
        INC     @COUNT          Count the clock interrupt
        LI      R12, PSIBAS     Set CRU base to point to 9901
        SBZ     0               Enter interrupt mode
        SBO     3               Clear clock interrupt
```

8◀

FIGURE 9 – TMS 9900 SAMPLE SOFTWARE TO CONTROL THE TMS 9901

3.3 Interval Timer Application

A TM 990/100M microcomputer board application in which every 10 seconds a specific task must be performed is described below. The TMS 9901 clock is set to interrupt every 333.33 milliseconds. This is accomplished by programming the 14-bit clock register to $3D09_{16}$ ($15{,}625_{10}$). The TM 990/100M microcomputer board system clock runs at 3 MHz, giving a clock resolution of 21.33 microseconds. A decrementer period of 21.33 microseconds multiplied by 15,625 periods until interrupt gives 333.33 milliseconds between interrupts. The interrupt service routine must count 30 interrupts before 10 seconds elapses:

$$f_{(DEC)} = \frac{f(\phi)}{64}, \qquad T_{(DEC)} = \frac{1}{f_{(DEC)}} = \frac{64}{3{,}000{,}000} = 21.3333\ \mu s$$

Figure 10 is a flowchart of the software required to perform the above application, and Figure 11 is a listing of the code. Following the flowchart, the main routine sets up all initial conditions for the 9901 and clock service routine. The interrupt service routine decrements a counter in R2 which was initialized to 30. When the counter in R2 decrements to zero, 10 seconds have elapsed, and the work portion of the service routine is entered. Note carefully that the work portion of the service routine takes longer than 333.33 ms which is the time between clock interrupts from the 9901. Therefore, recursive interrupts are going to occur and some facility must be provided to handle them. Loading a new workspace pointer and transferring the saved WP, PC, and ST (R13-R15) from the interrupt workspace to the new workspace allows one level of recursion.

▶8

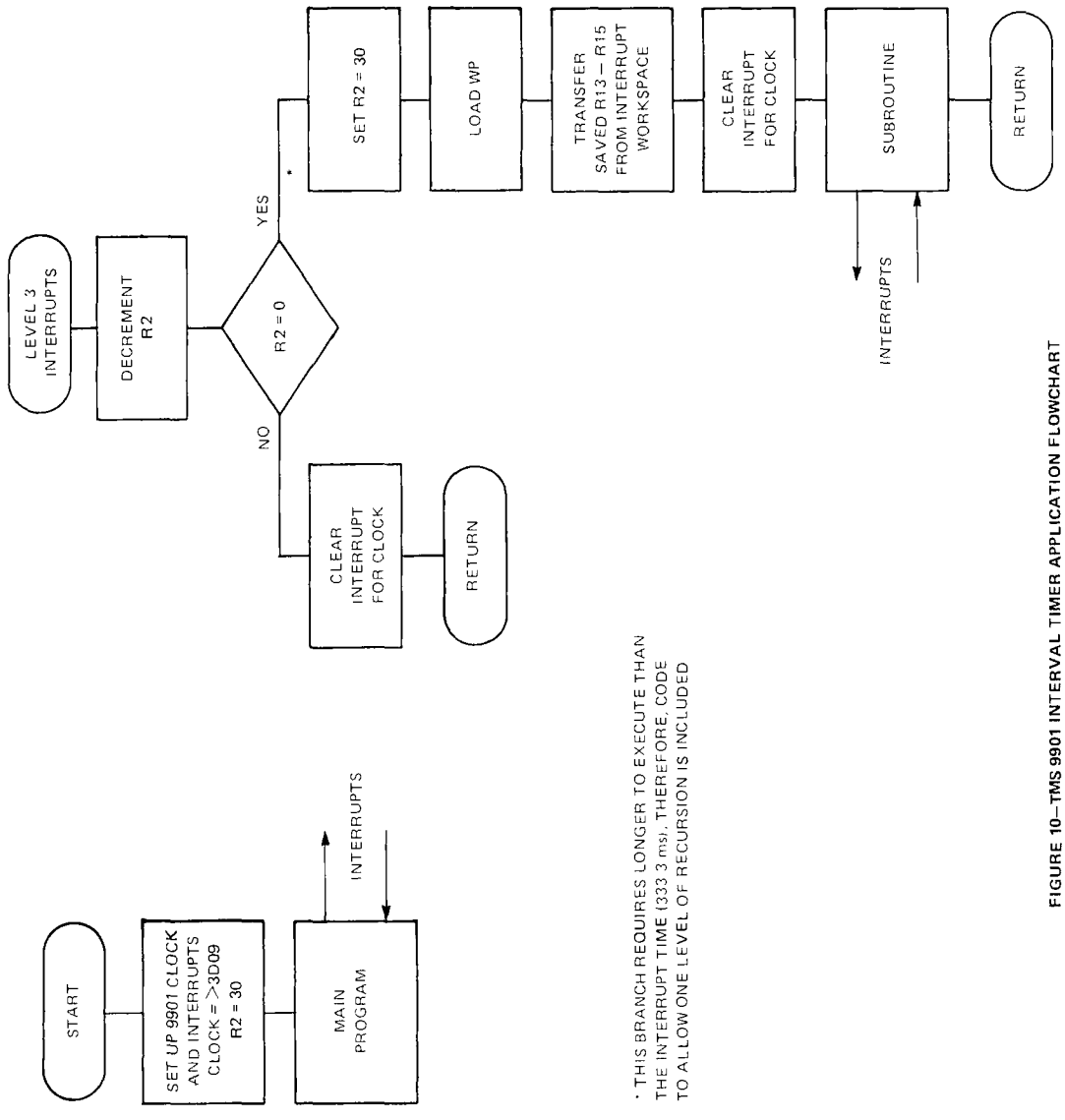


FIGURE 10–TMS 9901 INTERVAL TIMER APPLICATION FLOWCHART

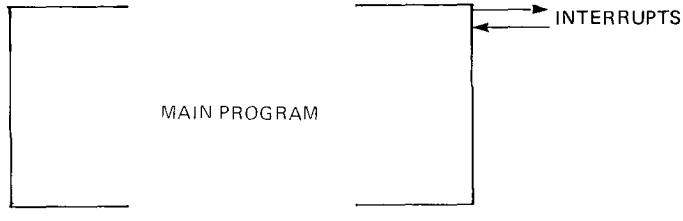
8◀

DEVICE INITIALIZATION

```
FE00 02E0 LWPI >FF20
FE02 FF20
FE04 020C LI R12,>100     9901 CRU BASE ADDRESS
FE06 0100
FE08 02E0 LWPI >FF60      INTERRUPT 3 WORKSPACE
FE0A FF60
FE0C 0201 LI R1,>7A13     DATA FOR 333.33MS CLOCK
FE0E 7A13
FE10 0202 LI R2,30        30 X 333.33MS = 10SEC
FE12 001E
FE14 020C LI R12,>100     9901 CRU BASE ADDRESS
FE16 0100
FE18 3BC1 LDCR R1,15      LOAD 9901 CLOCK
FE1A 1E00 SBZ 0           SET 9901 TO INTERRUPT MODE
FE1C 1D03 SBO 3           UNMASK INTERRUPT 3
```

MAIN PROGRAM

```
FD00 02E0 LWPI  FF00      MAIN PROGRAM WORKSPACE
FD02 FF00
FD04 0300 LIMI 3          ENABLE INT 0-3
FD06 0003
```



▶8

NOTE: This code was assembled using the TM 990/402 line-by-line assembler.

FIGURE 11—INTERVAL TIMER

INTERRUPT 3 SERVICE ROUTINE
(WP = FF68)

```
FD80 0602 DEC R2          COUNT DOWN 30 IN R2
FD82 1302 JEQ >FD88       IF ZERO THEN JUMP
FD84 1D03 SBO 3           CLEAR 9901 CLOCK INTERRUPT
FD86 0380 RTWP            RETURN TO INTERRUPTED ROUTINE
FD88 0202 LI R2,30        RELOAD R2 FOR 10 SEC COUNT DOWN
FD8A 001E
FD8C 0460 B @>FC80        BRANCH TO SUBROUTINE
FD8E FC80
```

ROUTINE TO BE PERFORMED EVERY 10 SECONDS, IT TAKES
LONGER THAN 333.33 MS WHICH IS 9901 CLOCK PERIOD'

```
FC80 02E0 LWPI >FF20      WORKSPACE FOR SUBROUTINE
FC82 FF20
FC84 C360 MOV @>FF82,R13 TRANSFER SAVED WP,PC,ST FROM
FC86 FF82
FC88 C3A0 MOV @>FF84,R14 INT 3 WORKSPACE
FC8A FF84
FC8C C3E0 MOV @>FF86,R15
FC8E FF86
FC90 1D03 SBO 3             CLEAR 9901 CLOCK INTERRUPT
FC92 0300 LIMI 3            ENABLE INT 0-3
FC94 0003
```


```
FC98      RTWP
```

FIGURE 11–(CONCLUDED)

4. TMS 9901 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 ..... 0.85 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}			±100	μA
$I_{CC(av)}$	Average supply current from V_{CC}	$t_{c(\phi)} = 330$ ns, $T_A = 70$°C			150	mA
C_I	Small signal input capacitance, any input	f = 1 MHz			15	pF

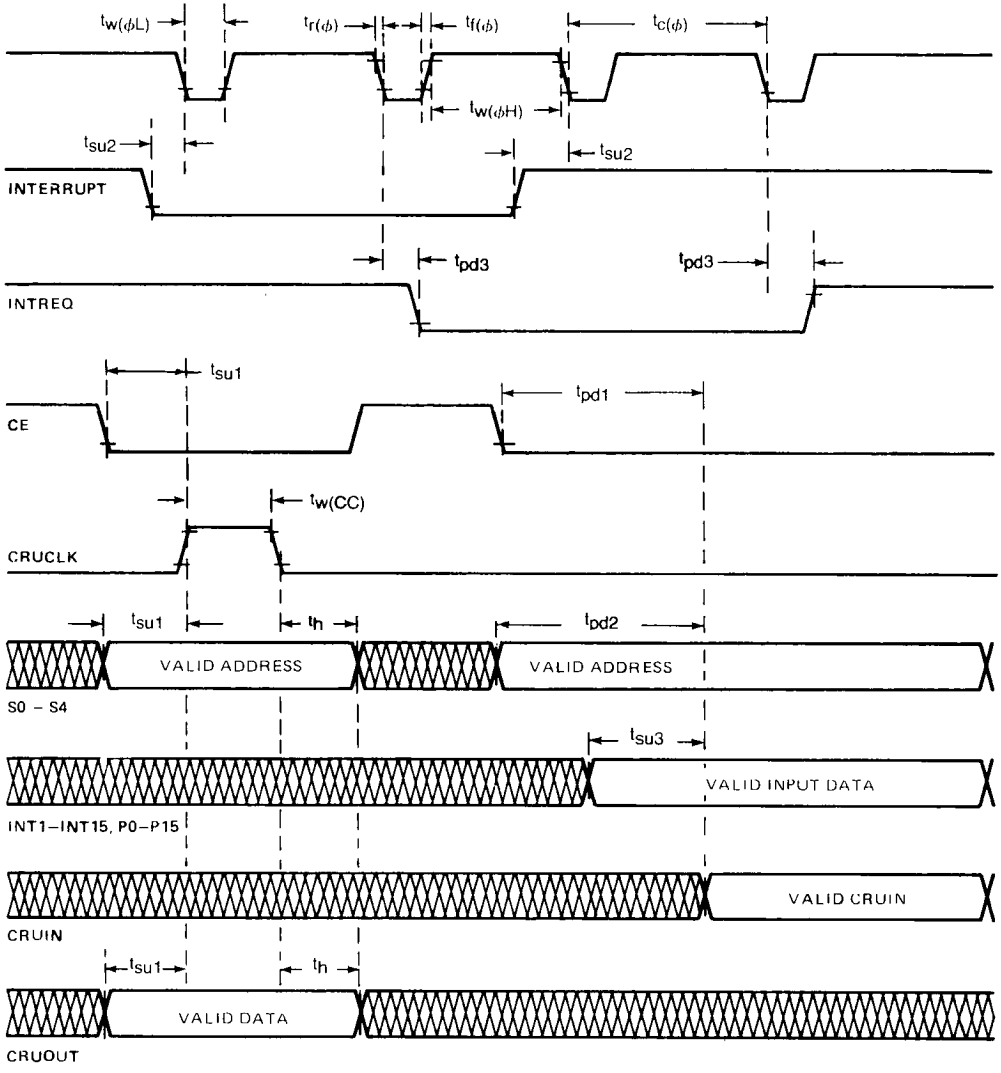
4.4 Timing Requirements Over Full Range of Operating Conditions

PARAMETER		MIN	TYP	MAX	UNIT
$t_{c(\phi)}$	Clock cycle time	300	333	2000	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		300	ns
$t_{w(CC)}$	CRUCLK pulse width	100	185		ns
t_{su1}	Setup time for $\overline{CE}$, S0-S4, or CRUOUT before CRUCLK	100			ns
t_{su2}	Setup time for interrupt before $\overline{\phi}$ low	60			ns
t_{su3}	Setup time for inputs before valid CRUIN	200			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	C_L = 100 pF			300	ns
t_{pd2}	Propagation delay, S0-S4 to valid CRUIN	C_L = 100 pF			[illegible]	ns
t_{pd3}	Propagation delay, $\overline{\phi}$ low to valid $\overline{INT}$ [illegible] IC0-IC3	C_L = 100 pF			[illegible]	ns
t_{pd}	Propagation delay, $\overline{CRUCLK}$ to valid data out (P0-P15)	C_L = 100 pF			300	ns



NOTE 1: ALL TIMING MEASUREMENTS ARE FROM 10% and 90% POINTS.

FIGURE 12–SWITCHING CHARACTERISTICS

5. TMS 9901-40 ELECTRICAL SPECIFICATIONS

5.1 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE AIR TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)*

Supply voltages, V_{CC} . . . . . −0.3 V to 10 V
All input and output voltages . . . . . −0.3 V to 10 V
Continuous power dissipation . . . . . 0.90 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 period may affect device reliability.

5.2 RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	5.25	V
Supply voltage, V_{SS}		0		V
High-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

5.3 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	μA
I_I	Input current (any input)	$V_I = 0$ V to V_{CC}			±100	V
$I_{CC(av)}$	Average supply current from V_{CC}	$t_c(\phi) = 330$ ns, $T_A = 25°C$			150	mA
C_i	Small Signal Input Capacitance, any input	f = 1 MHz			15	mF

5.4 TIMING REQUIREMENTS OVER FULL RANGE OF OPERATING CONDITIONS

PARAMETER		MIN	NOM	MAX	UNIT
$t_c(\phi)$	Clock cycle time	240	250	667	ns
$t_r(\phi)$	Clock rise time	5		40	ns
$t_f(\phi)$	Clock fall time	10		40	ns
$t_w(\phi L)$	Clock pulse width (low level)	40		300	ns
$t_w(\phi H)$	Clock pulse width (high level)	180			ns
$t_w(CC)$	CRUCLK pulse width	80	125		ns
t_{su1}	Setup time for S0-S4, $\overline{CE}$, or CRUOUT before CRUCLK	80	80		ns
t_{su2}	Setup time, interrupt before $\overline{\phi}$ low	50	50		ns
t_{su3}	Setup time for inputs before valid CRUIN	180	180		ns
t_h	Hold time for $\overline{CE}$, S0-S4, or CRUOUT after CRUCLK	50	50		ns

▶8

DESIGN GOAL

This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

5.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PD1}	propagation delay, $\overline{CE}$ to Valid CRUIN	$C_L = 100_P F$		220	220	ns
t_{PD2}	propagation delay, S0-S4 to Valid CRUIN			240	240	ns
t_{PD3}	propagation delay, ϕ low to Valid $\overline{INTREQ}$, IC0-IC3			80	80	ns
t_{PD}	propagation delay, CRUCLK to Valid Data Out (P0-P15)			200	200	ns

8

DESIGN GOAL

This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

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 ($\overline{CE}$), and three CRU lines (CRUIN, CRUOUT, CRUCLK). An additional input to the CPU is the ACC interrupt line ($\overline{INT}$). 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 ($\overline{CE}$) 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 ($\overline{RTS}$), data set ready ($\overline{DSR}$), clear to send ($\overline{CTS}$), serial transmit data (XOUT), and serial receive data (RIN). 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 $\overline{DSR}$ or $\overline{CTS}$ changes level, an interrupt is generated.

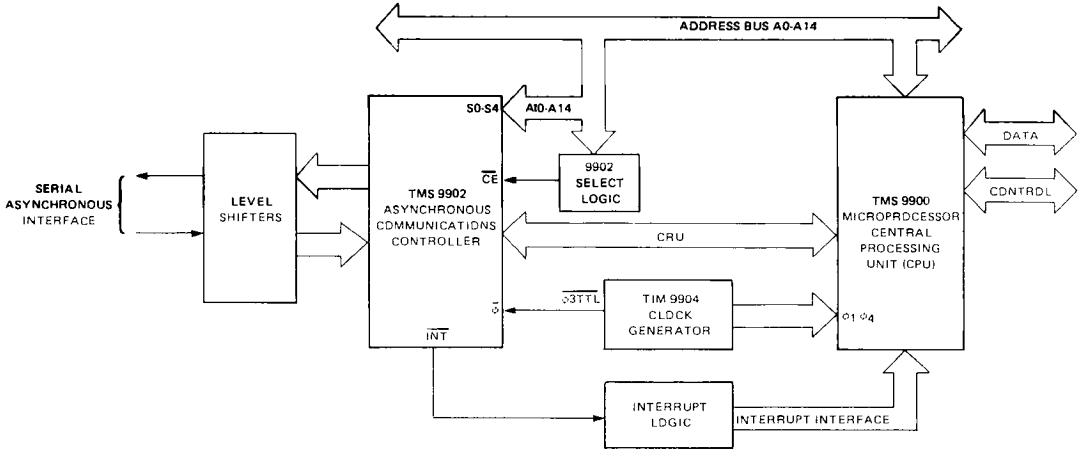


FIGURE 1. TYPICAL APPLICATION, TMS 9902 ASYNCHRONOUS COMMUNICATION CONTROLLER (ACC)

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 9901 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.

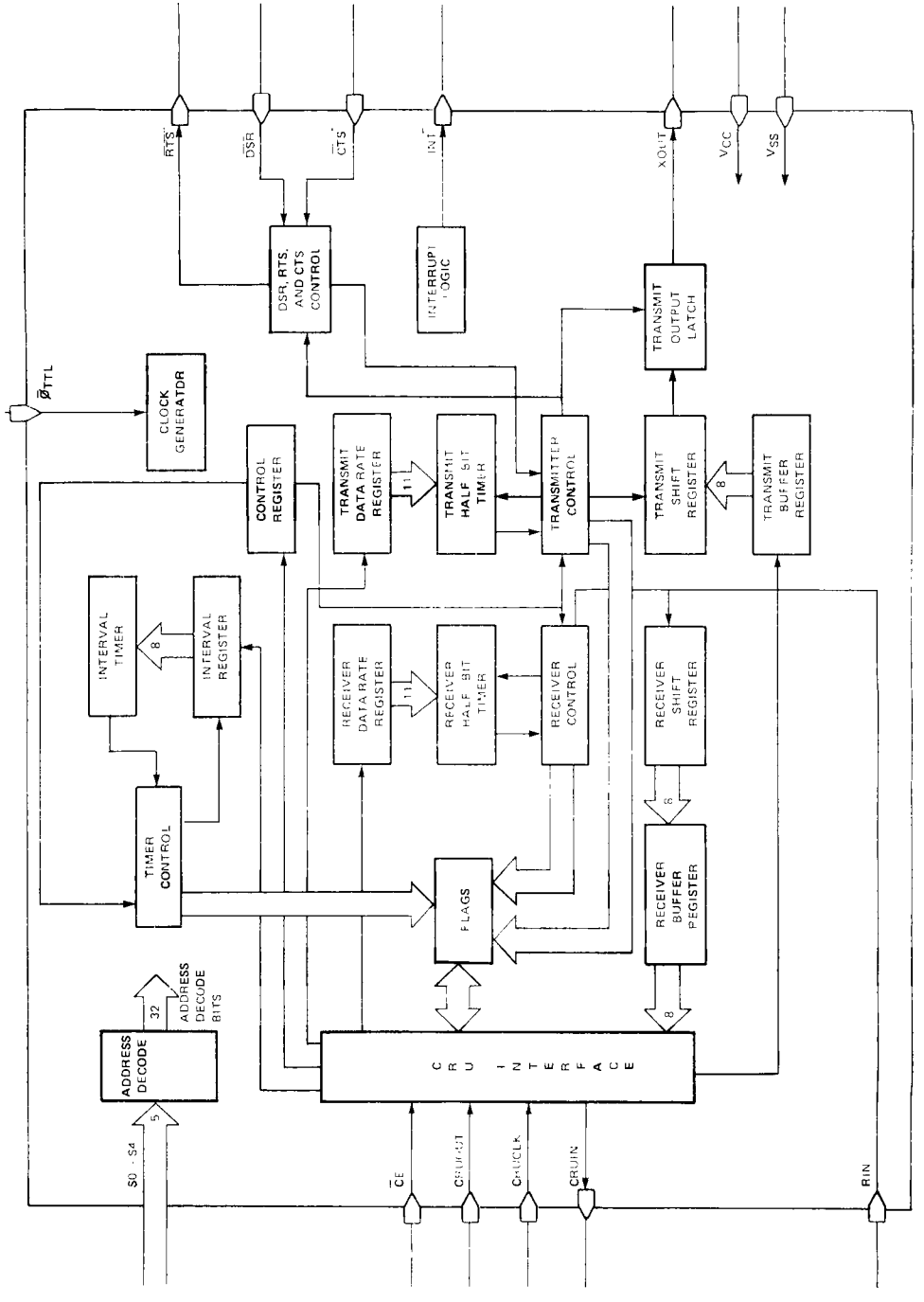


FIGURE 2. TMS 9902 ASYNCHRONOUS COMMUNICATIONS CONTROLLER (ACC) BLOCK DIAGRAM

8

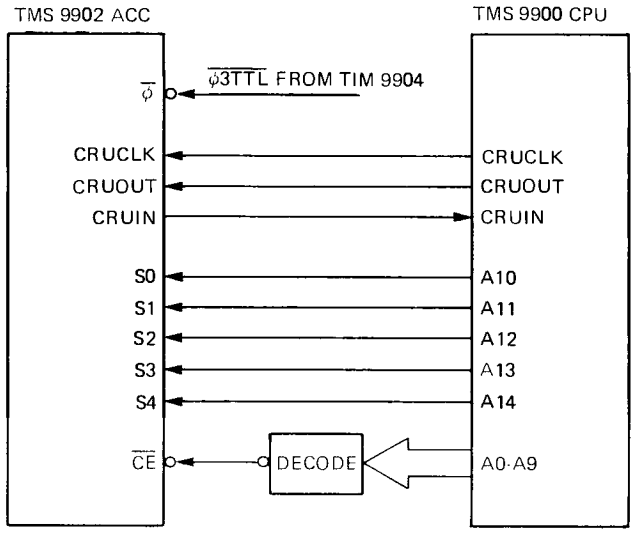


FIGURE 3. TMS 9902 – TMS 9900 CRU INTERFACE

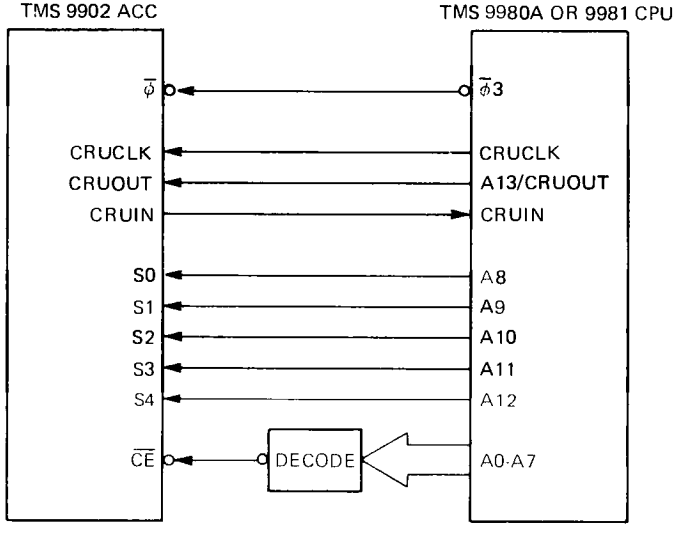


FIGURE 4. TMS 9902 – TMS 9980A OR 9981 CRU INTERFACE

8

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.

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 BIT ADDRESS 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

▶8

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	INT
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 $\overline{INT}$ 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 $\overline{INT}$ 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 $\overline{INT}$ 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 $\overline{INT}$ 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 $\overline{RTS}$ output to be active (LOW). Writing a zero to bit 16 causes $\overline{RTS}$ 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 $\overline{RTS}$ output does not become inactive (HIGH) until *after* character transmission is completed.

Bit 15 (TSTMD) — **Test Mode.** Writing a one to bit 15 causes $\overline{RTS}$ to be internally connected to $\overline{CTS}$, XOUT to be internally connected to RIN, $\overline{DSR}$ 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 LOAO SELECTION

REGISTER LOAD CONTROL FLAG STATUS				REGISTER ENABLED
LDCTRL	LDIR	LRDR	LXOR	
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 r reset to logic zero, disabling loading of the Interval Register. LDIR is als 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.

8

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 tv registers may be loaded simultaneously when data is received and transmitteu 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	Φ 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	RCL0
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

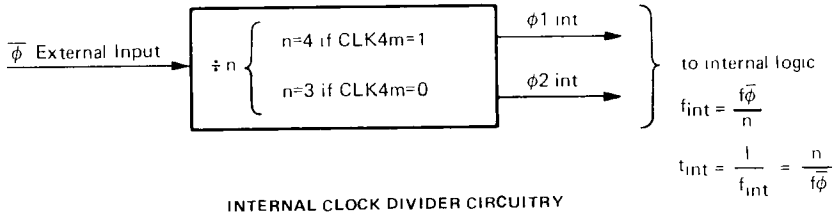
Bits 5 and 4 (PENB and PODD) — **Parity Selection.** The type of parity generated for transmission and detected for reception is selected by bits 5 and 4 of the Control Register as shown below. When parity is enabled (PENB = 1), the parity bit is transmitted and received in addition to the number of 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.



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:

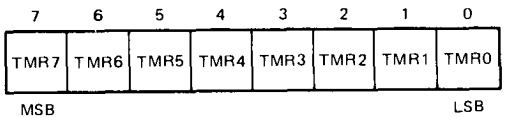
CHARACTER LENGTH SELECTION

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

8

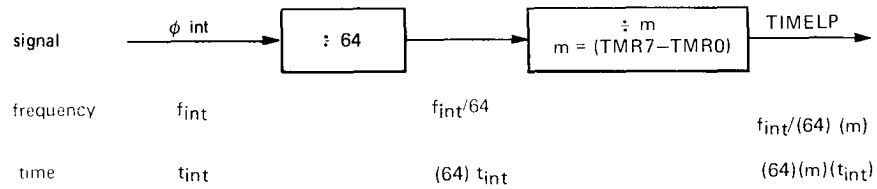
2.1.2.2 Interval Register

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.



INTERVAL REGISTER BIT ADDRESS ASSIGNMENTS

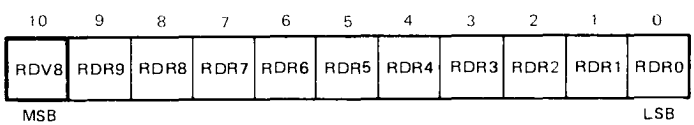
The figure below illustrates the establishment of the interval for the Interval Timer. For example, if the Interval Register is loaded with a value of 80_{16} (128_{10}) the interval at which Timer Interrupts are generated is $t_{itvl} = t_{int} \cdot 64 \cdot M = (1\ \mu s)(64)(128) = 8.192$ ms when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



TIME INTERVAL SELECTION

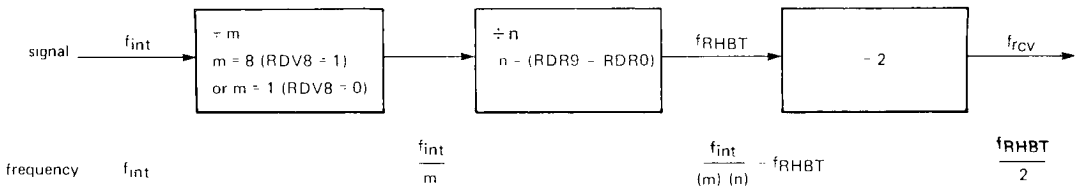
2.1.2.3 Receive Data Rate Register

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.



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 × 10^6 ÷ 1) ÷ 417] ÷ 2 = 1199.0 bits per second.

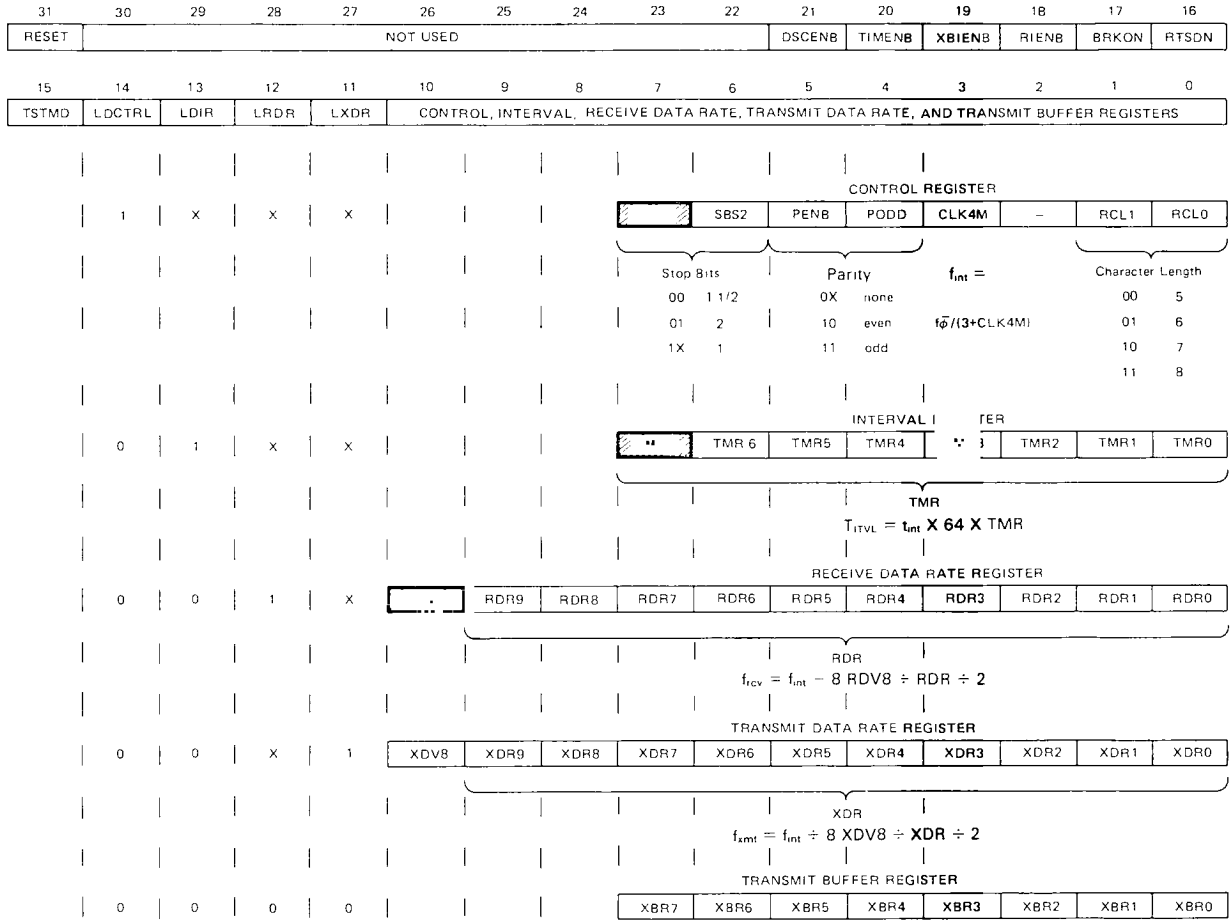
2.1.2.5 Transmit Buffer Register

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	X8R3	XBR2	XBR1	XBR0
MSB							LSB

TRANSMIT BUFFER REGISTER BIT ADDRESS ASSIGNMENTS

TABLE 4. CRU OUTPUT BIT ADDRESS ASSIGNMENTS



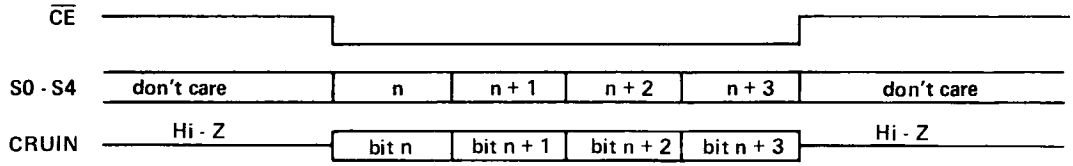
NOTE 1 LOADING OF THE BIT INDICATED BY CAUSES THE LOAD CONTROL FLAG FOR THAT REGISTER TO RESET AUTOMATICALLY

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 BIT ADDRESS 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 ($\overline{INT}$) 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 $\overline{DSR}$ or $\overline{CTS}$ input 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 bit 21 (DSCENB).

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

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

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

8

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 DSR or 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.

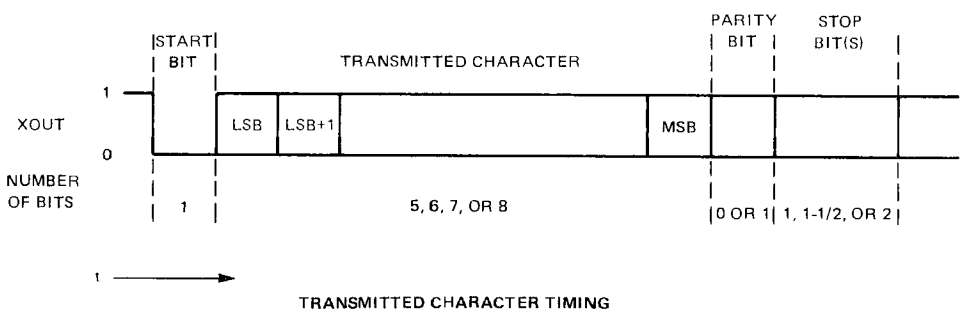
Bit 7-Bit 0
(RBR7-RBR0) — **Receive Buffer Register.** The Receive Buffer Register contains the most recently received character. For character lengths of fewer than eight bits, the character is right-justified, with unused most significant bit(s) all zero(es). The presence of valid data in the Receive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

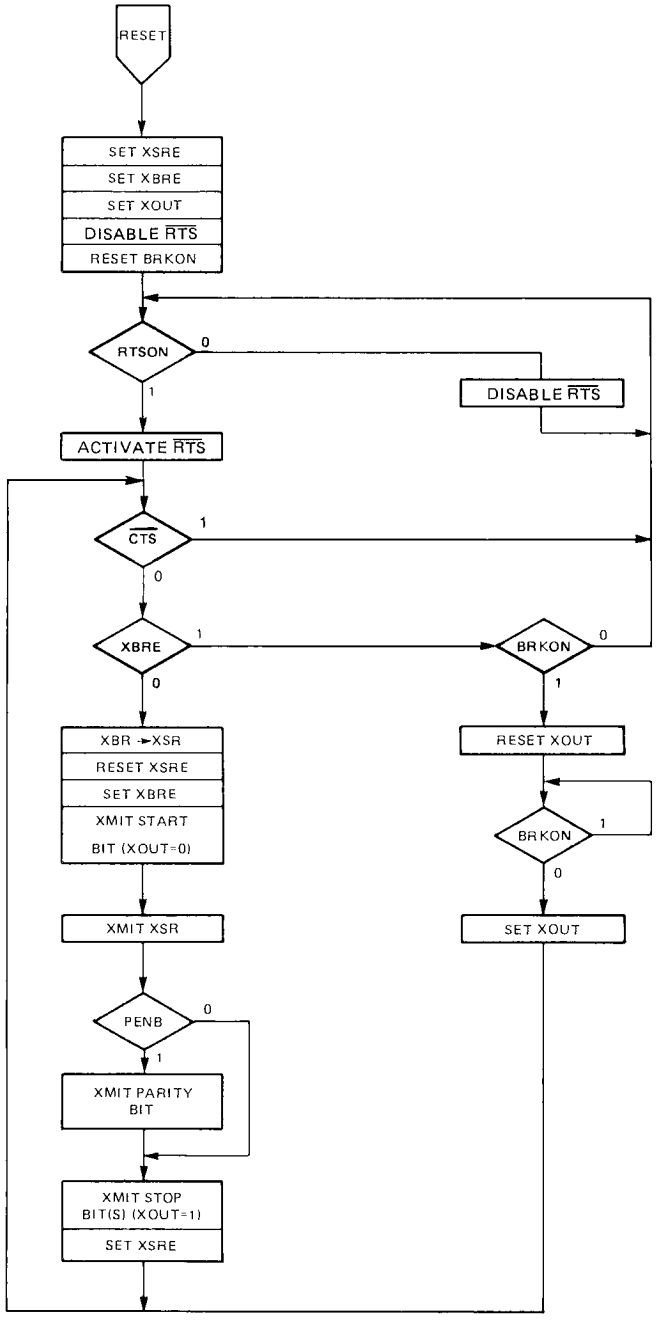


FIGURE 5. TMS 9902 TRANSMITTER OPERATION

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.

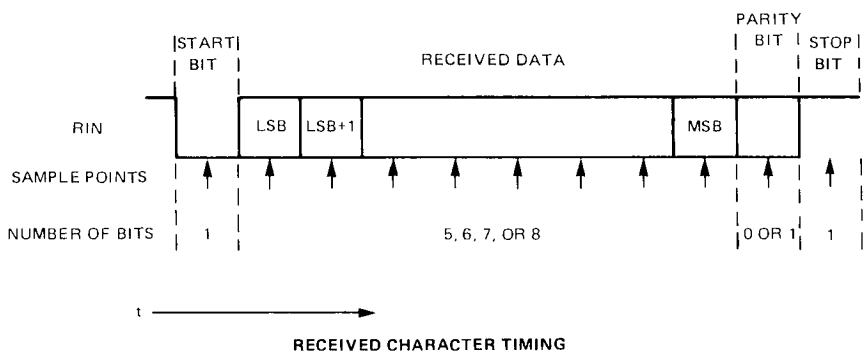
2.3.2 Start Bit Detection

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

8

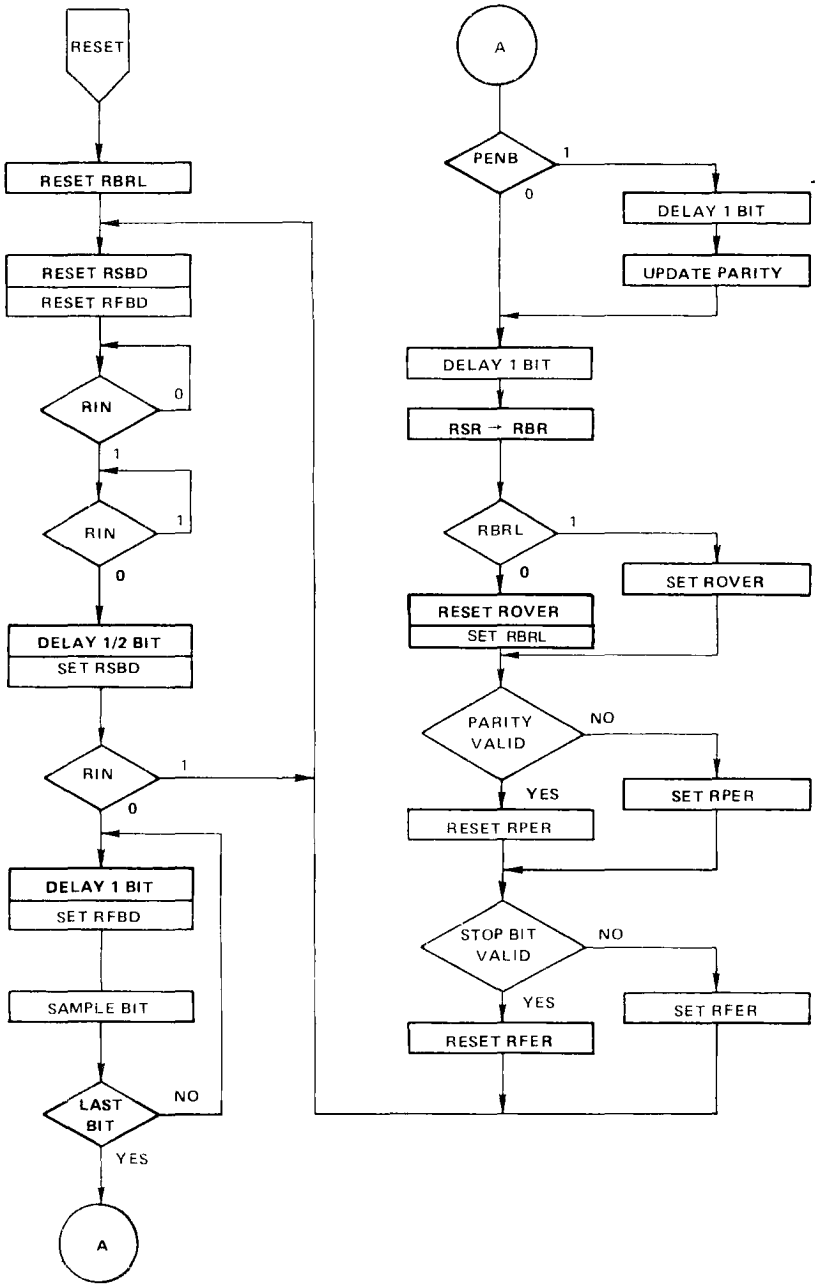


FIGURE 6. TMS 9902 RECEIVER OPERATION

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).

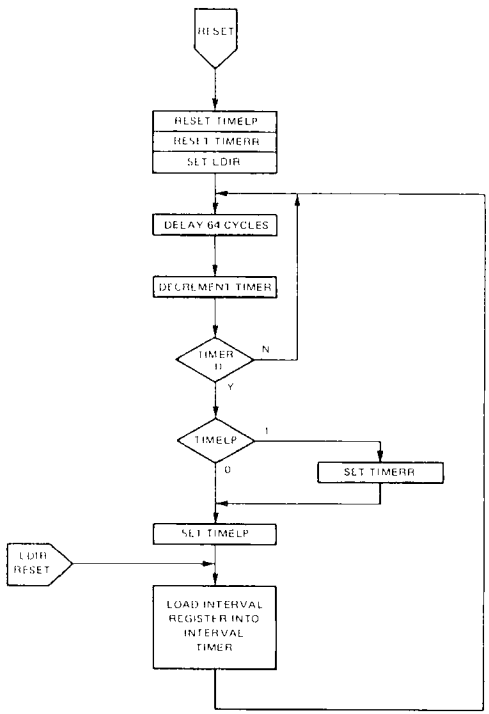


FIGURE 7. TMS 9902 INTERVAL TIMER OPERATION

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).

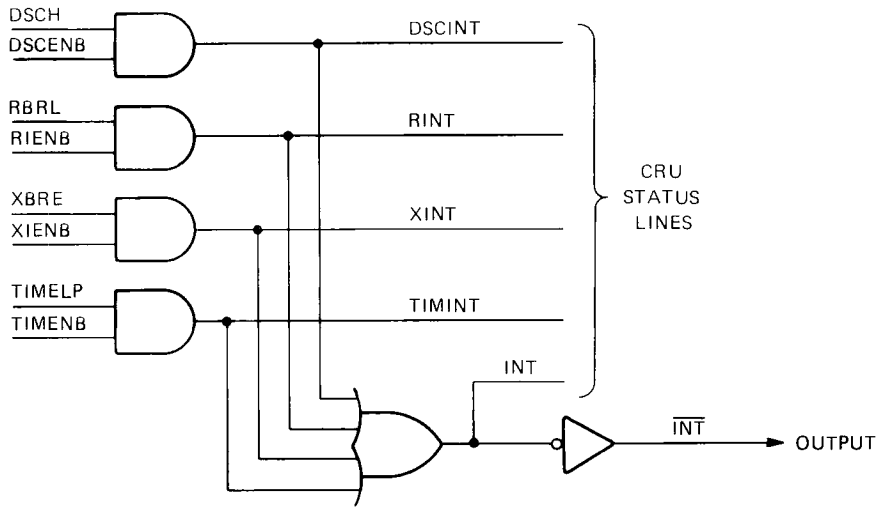


FIGURE 8. $\overline{INT}$ OUTPUT GENERATION

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.

8

TABLE 6
TMS 9902 INTERRUPT CLEARING

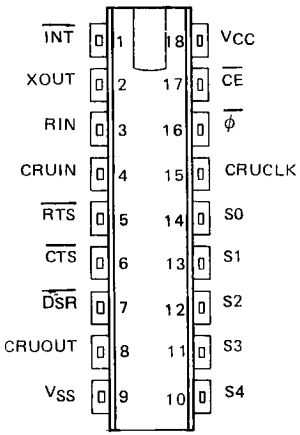
MNEMONIC	CAUSE	ACTION TO RESET
DSCINT	$\overline{CTS}$ or $\overline{DSR}$ change state	Write a bit to DSCENB (bit 21)*
RINT	Recieve Buffer Full	Write a bit to RIENB (bit 18)*
XINT	Transmit Buffer Register Empty	Load Transmit Buffer
TIMINT	Timer Elapsed	Write a bit to TIMENB (bit 20)*

*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)	10	I	Address Select Lines. The data bit being accessed by the CPU interface is specified by the 5-bit code appearing on S0-S4.
S3	11	I	
S2	12	I	
S1	13	I	
S0	14	I	
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 CE is inactive (HIGH), TMS 9902 CRU interface is disabled. CRUIN remains at high-impedance when CE is inactive (HIGH).
V_{CC}	18	I	Supply voltage (+5 V nominal).

TMS 9902
18-PIN PACKAGE



8

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.

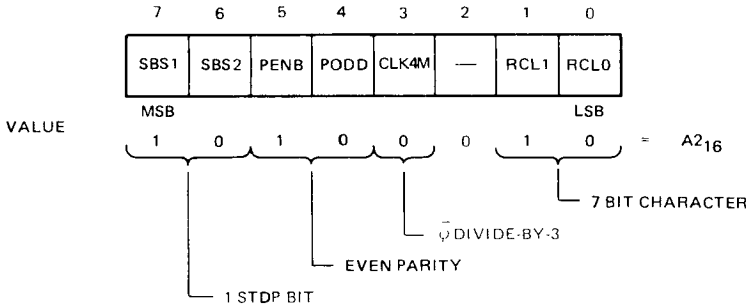
```
        LI      R12,>40     INITIALIZE CRU BASE
        SBO     31          RESET COMMAND
        LDCR    @ CNTRL,8   LOAD CONTROL AND RESET LDCTRL
        LDCR    @ INTVL,8   LOAD INTERVAL AND RESET LDIR
        LDCR    @ RDR,11    LOAD RDR AND RESET LRDR
        LDCR    @ XDR,12    LOAD XDR AND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

▶8

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

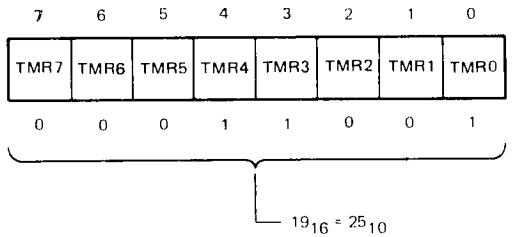
3.1.2 Control Register

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



3.1.3 Interval Register

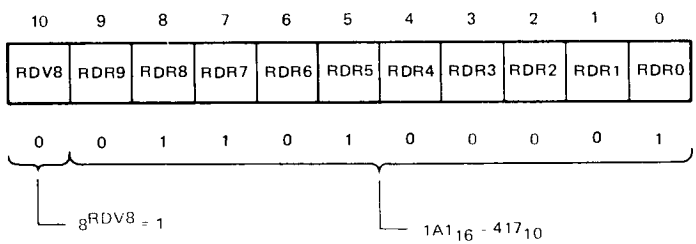
To set up the interval register to generate an interrupt every 1.6 milliseconds, load the value into the interval register to specify the number of 64-microsecond increments in the total interval desired.



25 X 64 MICROSECONDS = 1.6 MILLISECONDS

3.1.4 Receive Data Rate Register

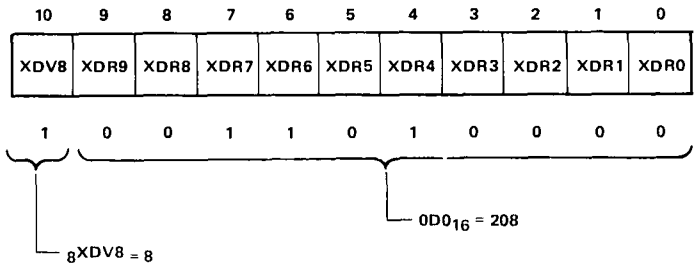
To set the data rate for the receiver to 1200 bits per second, load the value into the Receive Data Rate register as shown below:



$10^6 \div 1 \div 417 \div 2 = 1199.04$ 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:



$1 \times 10^6 \div 8 \div 208 \div 2 = 300.48$ 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 ($\overline{9902SEL}$) signal comes from decode circuitry which looks at A0-A9 on CRU cycles. The interrupt output ($\overline{INT}$) of the TMS 9902 is input to the TMS 9901 for prioritization and encoding. When the 9902 is communicating with a terminal, the $\overline{RTS}$ pin can be connected to the $\overline{CTS}$ pin because the terminal will always be in the clear-to-send ($\overline{CTS}$) 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 defined 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.

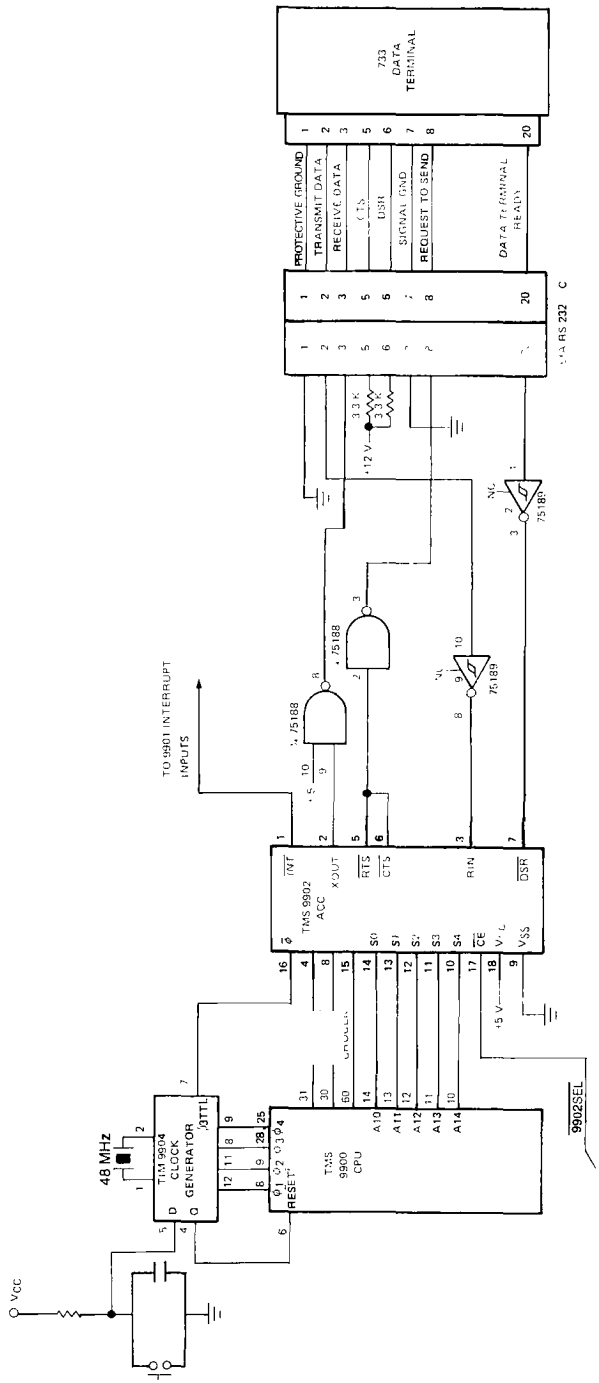


FIGURE 9. INTERFACE TO A 733 DATA TERMINAL

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
```

TABLE 7 (Continued)

```
TIBUG
*** WRITE CHARACTER ***

0290              ***********************************************
0291              * WRITE CHARACTER -- XOP  R,12
0292              *                              ---      NORMAL RETURN
0293              *
0294              * TRANSMIT THE CHARACTER IN THE LEFT BYTE OF
0295              * USER REGISTER R. IF THE CHARACTER IS A
0296              * CARRIAGE RETURN, THE ROUTINE WAITS 200 MSEC FOR
0297              * THE CARRIAGE TO RETURN. IF THE TERMINAL IS
0298              * A 733ASR AS DENOTED IN THE T COMMAND, EACH
0299              * CHARACTER IS PADDED WITH 25 MSEC TO REDUCE
0300              * THE TRANSFER RATE TO 300 BAUD.
0301              ***********************************************
0302 01B6 020A   WENTRY LI   R10,3750
     01B8 0EA6
0303 01BA 020C          LI   CRUBAS,>80    SET CRU BASE REG.
     01BC 0080
0304 01BE 1D10          SBO  16            SET RTSON
0305 01C0 1F16          TB   22            TRANSMIT BUFFER REG. EMPTY?
0306 01C2 16F9          JNE  WENTRY        NO, WAIT UNTIL IT IS
0307 01C4 321B          LDCR *LINK,8       CHARACTER TO UART
0308 01C6 D2DB          MOVB *LINK,LINK
0309 01C8 1E10          SBZ  16            RESET RTSON
0310 01CA 098B          SRL  LINK,8
0311 01CC 028B          CI   LINK,>000D    CARRIAGE RETURN
     01CE 000D
0312 01D0 1608          JNE  ASR733        NO, SKIP
0313 01D2 0A3A          SLA  R10,3
0314 01D4 1F16  WLOOP1  TB   22            WAIT FOR XMISSION TO END
0315 01D6 16FE          JNE  WLOOP1
0316 01D8 1F17          TB   23
0317 01DA 16FC          JNE  WLOOP1
0318 01DC 060A  WLOOP2  DEC  R10           WAIT LOOP
0319 01DE 16FE          JNE  WLOOP2
0320 01E0 0380          RTWP
0321 01E2 C2E0  ASR733  MOV  @DUMPFG,LINK  IN DUMP ROUTINE ?
     01E4 FFF6
0322 01E6 1303          JEQ  WEXIT         YES, IGNORE ASR FLAG
0323 01E8 C2E0          MOV  @ASR,LINK     ASR733 ?
     01EA FFF4
0324 01EC 16F3          JNE  WLOOP1        YES, WAIT 3 NULLS
0325 01EE 0380  WEXIT   RTWP

TIBUG
*** READ CHARACTER ***

0271              ***********************************************
0272              * READ CHARACTER -- XOP  R,13
0273              *                              --       NORMAL RETURN
0274              *
0275              * READ WAITS FOR A CHARACTER TO BE ASSEMBLED IN
0276              * THE UART. THE CHARACTER IS PLACED IN THE LEFT
0277              * BYTE OF USER REGISTER R. THE RIGHT BYTE IS
0278              * ZEROED. ALL ERRORS ARE IGNORED.
0279              ***********************************************
0280              *
0281 01A6 020C  RENTRY LI   CRUBAS,>80    SET CRU BASE REG.
     01A8 0080
0282 01AA 1F15          TB   21            RECEIVE BUFFER REG. FULL?
0283 01AC 16FC          JNE  RENTRY        NO, LOOP
0284 01AE 04DB          CLR  *LINK
0285 01B0 361B          STCR *LINK,8
0286 01B2 1E12          SBZ  18
0287 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 μA	2.4		V_{CC}	V
		I_{OH} = −200 μA	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

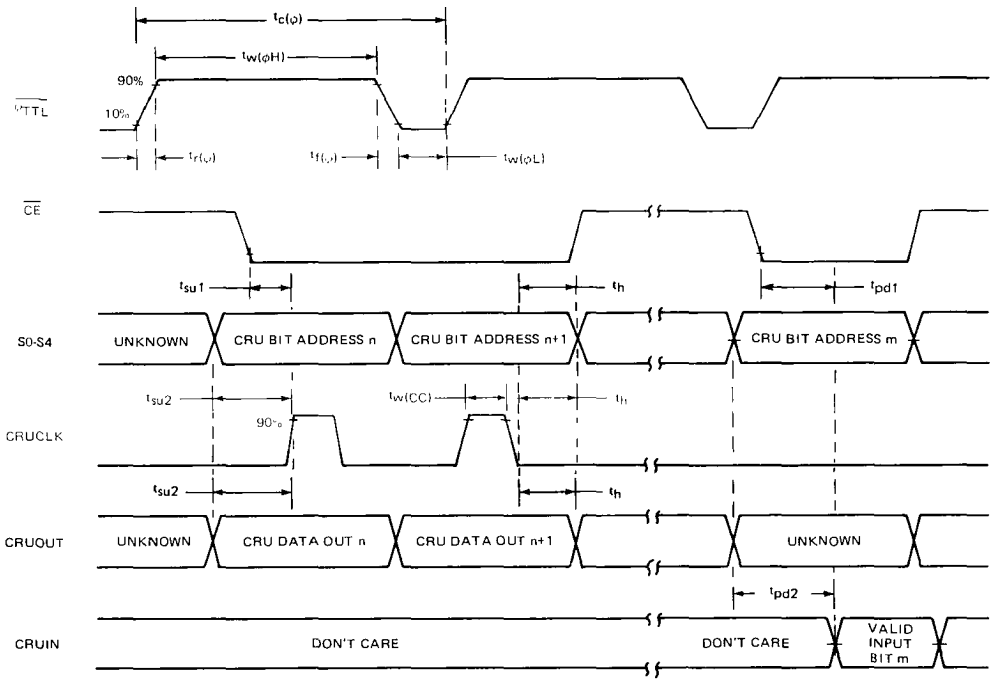
4.4 Timing Requirements Over Full Range of Operating Conditions

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

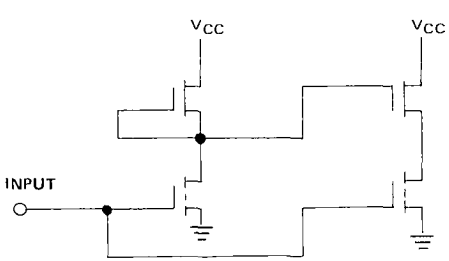
	PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
t_{pd1}	Propagation delay, $\overline{CE}$ to valid CRUIN	CL = 100pF			300	ns
t_{pd2}	Propagation delay, S0-S4 to valid CRUIN	CL = 100pF			320	ns



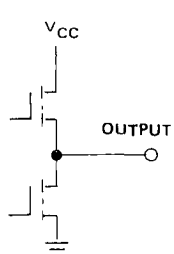
SWITCHING CHARACTERISTICS

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

8◀

5. TMS 9902-40 ELECTRICAL SPECIFICATIONS

5.1 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 . . . . . 0.6 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.

5.2 RECOMMENDED OPERATING CONDITIONS

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.75	5	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

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

	PARAMETERS	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_{OL}	Low-level output voltage	$I_{OL} = 3.2$ mA			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, $T_A = 25°C$			100	mA
C_i	Small Signal Input Capacitance, any input	f = 1 MHz			15	pF

5.4 TIMING REQUIREMENTS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

▸8

	PARAMETER	MIN	TYP	MAX	UNIT
$t_c(\phi)$	Clock cycle time	240	250	667	ns
$t_r(\phi)$	Clock rise time	8		40	ns
$t_f(\phi)$	Clock fall time	10		40	ns
$t_w(\phi H)$	Clock pulse width (high level)	180			ns
$t_w(\phi L)$	Clock pulse width (low level)	40			ns
$t_w(CC)$	CRUCLK pulse width	80			ns
t_{su1}	Setup time for $\overline{CE}$ before CRUCLK	110	110		ns
t_{su2}	Setup time for S0-S4 or CRUOUT before CRUCLK	150	150		ns
t_h	Hold time for CE, S0-S4, or CRUOUT after CRUCLK	50	50		ns

DESIGN GOAL
This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

5.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

PARAMETERS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PD1} Propagation delay, CE to valid CRUIN	$C_L = 100pF$		220	220	ns
t_{PD2} Propagation delay, S0-S4 to valid CRUIN			240	240	ns

8

DESIGN GOAL

This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

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
- 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 +5 V 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 most 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 for 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.

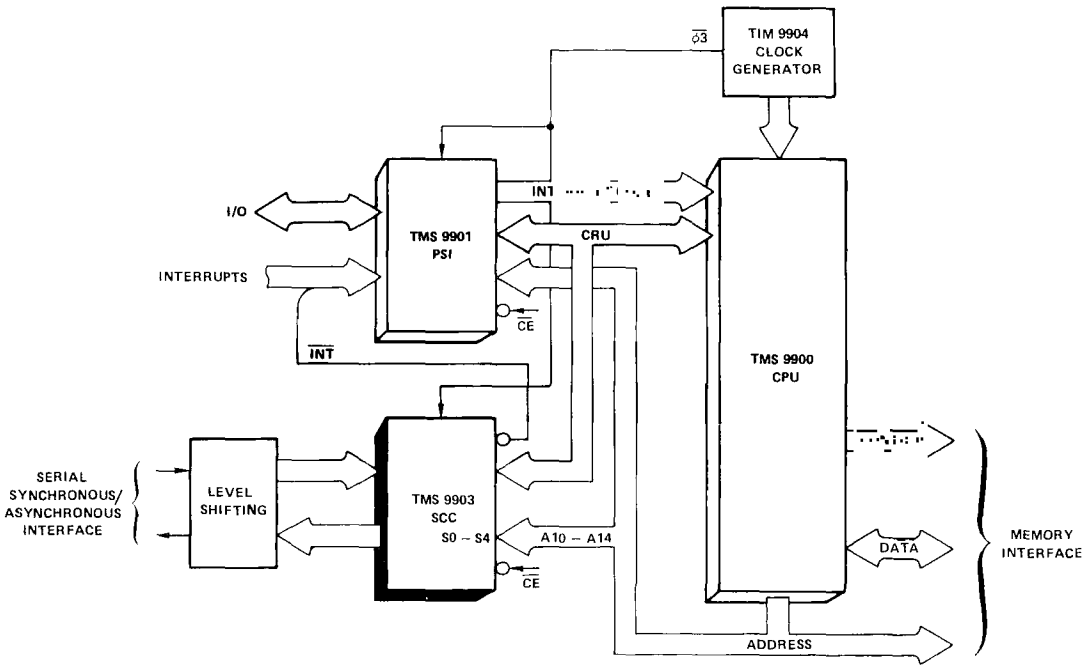


FIGURE 1. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER IN A TMS 9900 SYSTEM

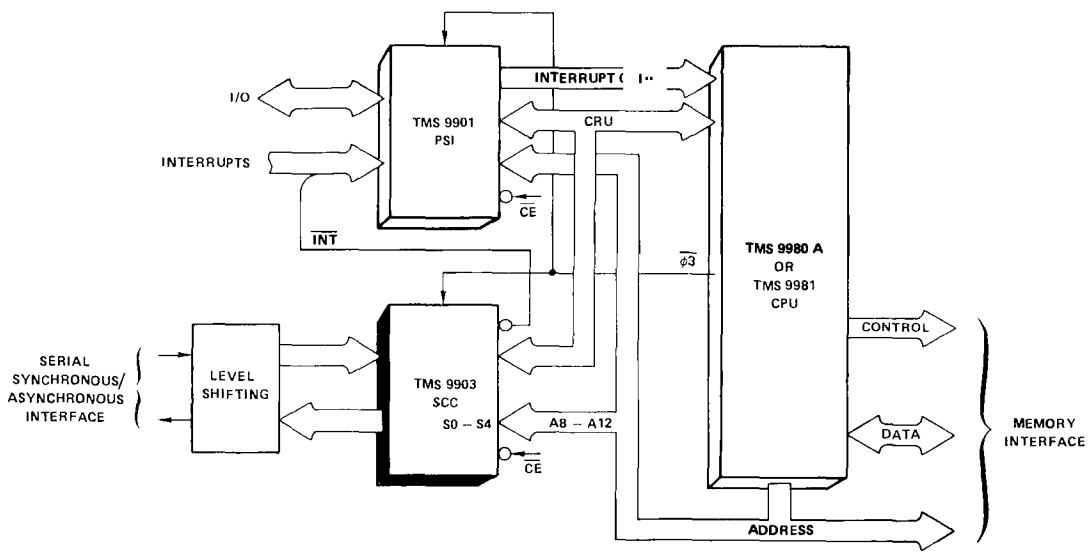


FIGURE 2. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER IN A TMS 9980 A, 9981, SYSTEM

8◀

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

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 ▸8 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.

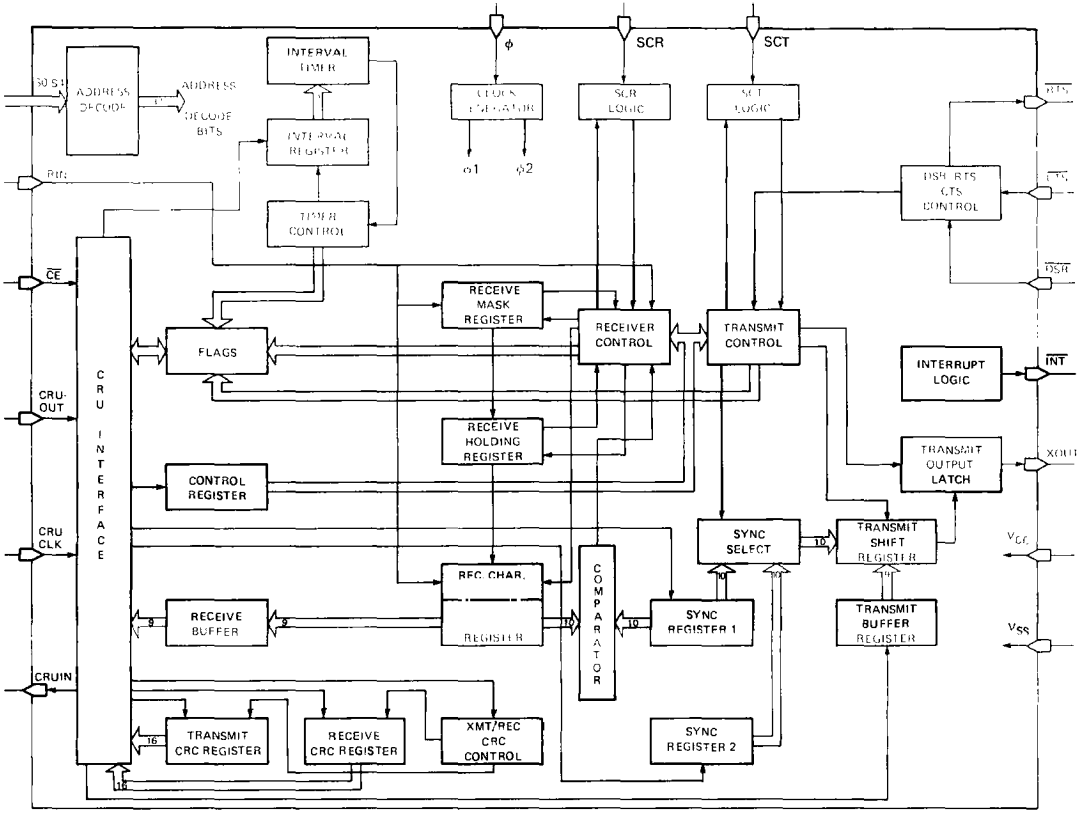


FIGURE 3. TMS 9903 SYNCHRONOUS COMMUNICATION CONTROLLER BLOCK DIAGRAM

8◀

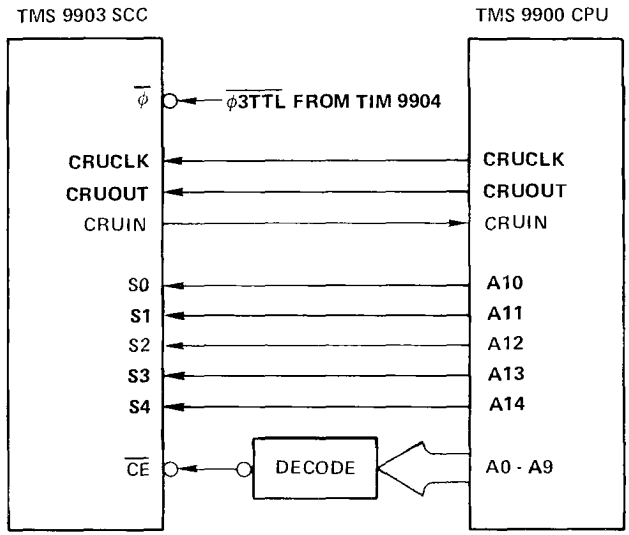


FIGURE 4. TMS 9903 CONTROL SIGNALS (TMS 9900 SYSTEM)

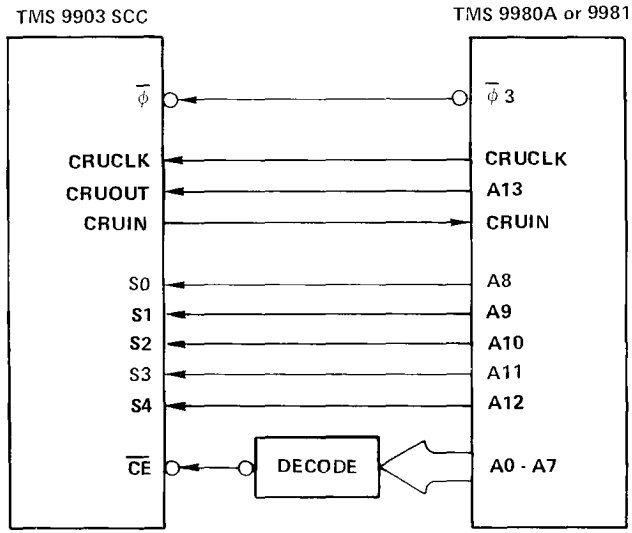


FIGURE 5. TMS 9903 CONTROL SIGNALS (TMS 9980A or 9981 SYSTEM)

TABLE 1. TMS 9903 OUTPUT SELECT BIT ASSIGNMENTS

ADDRESS	NAME	MODE 0	1	2	3	5	6	DESCRIPTION
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 Transmit CRC Register
	CLRCRC(0)	X	X	X	X	X	X	Clear Receive CRC Register
28	—	X		X		X	X	Not used
	XZINH		X					Transmit Zero Insertion Inhibit
	RSYNDL				X			Receive 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					Receive Holding Register Read
	LDSYN1			X	X			Load Sync Character Register 1
25	LXBC	X	X	X	X	X	X	Load Transmit Buffer and Transmit CRC Register
24	LXCRC	X	X	X	X	X	X	Load Transmit CRC Register
23	XPRNT	X	X		X			Transparent
	—			X				Not Used
	BRKON					X	X	Break On
22	XAINB	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 Receive CRC Register
11-0	DATA							Data To Selected Register

8

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 address select (S lines) and SCC functions by operational mode; modes 4 and 7 are not implemented. Each CRU addressable output bit on the TMS 9903 is described in detail following Table 1.

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 disables 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 Transmit CRC Register (XCRC).** Writing a one to bit 29 in all modes clears the XCRC register to all zeros.

(CLRCRC)— **Clear Receive 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)— **Transmit Zero Insertion Inhibit.** Writing a one to bit 28 in mode 1 causes the contents of the transmit 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 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 bit addresses 0–9. Writing a zero to bit 27 in mode 0, 1, 2, 3 resets LDSYN2.

▶8

Modes 5, 6 Not Used

Bit 26

Modes 0, 5, 6 — Not Used.

Mode 1 (RHRRD)— **Receive Holding Register Read.** Writing a one to bit 26 in mode 1 enables reading of the receive–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 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 bit addresses 0–8, and enables reading of XCRC at input bit addresses 0–15. Writing a zero to bit 25 in all modes resets LXBC an 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 bit addresses 0–9, and enables reading XCRC at input 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 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 (CTS, RTS, RTSAUT)
TIMENB	20	TIMELP	TIMINT	Timer Elapsed
XBIENB	19	XBRE	XBINT	Transmit 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)— **Transmit 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 (receive 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 (receive 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 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 SCT. $\overline{SCT}$ 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 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 bit addresses 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 Receive CRC Register. Writing a one to bit 12 in all modes enables the loading of the receive CRC register from output bit addresses 0–9, and enables reading the RCRC (receive CRC register) on input bit addresses 0–15. Writing a zero to bit 12 in all modes resets LRCRC.

2.1.2 Control and Data Registers

Loading of 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 receive 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 transmit buffer register before the transmit 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 receive 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	CLS1	CLS0	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 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 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 SCT frequency to 32 times the transmit data rate, and the SCR frequency to 32 times the receive data rate. SCR is resynched 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

8

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.

MODE BIT 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.

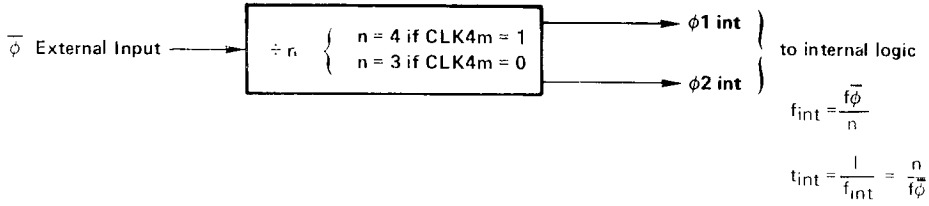
TRANSMIT/RECEIVE CONFIGURATION BIT SELECT

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

▶8

Bit 3
All modes (CKL4M)—

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)

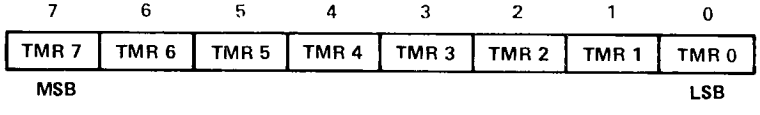
Received Character Length Select. The number of data bits in each received character is determined by bits 2, 1, and 0 of the control register, as shown below.

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

2.1.2.2 Interval Register

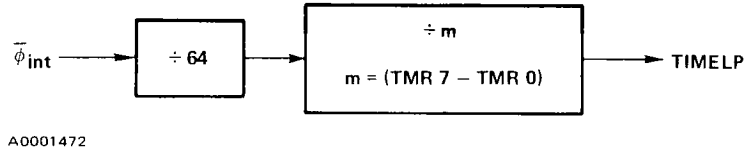
The interval register is enabled for loading whenever LDIR = 1. The interval register is used to select the rate at which timer interrupts are generated by the SCC interval timer. The figure below shows the bit-address assignments for the interval register when enabled for loading.



INTERNAL-REGISTER BIT ADDRESSES

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

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



TIME INTERVAL SELECTION

2.1.2.3 Receive CRC Register

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

NOTE

Single bits of the CRC registers cannot be accessed. As individual bits are sent to or read from the CRC registers, they are shifted in or out, respectively, of the register from CRC bit 16.

2.1.2.4 Transmit CRC Register

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

Operation of the transmit CRC register is analogous to that of the receive CRC register shown in Figure 6

▶8 *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 bit addresses 0–9 is loaded into sync character register 1. The LDSYN1 flag is reset by a command from the CPU.

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 bit addresses 0–9 is loaded into sync character register 2. The LDSYN2 flag is reset by a command from the CPU.

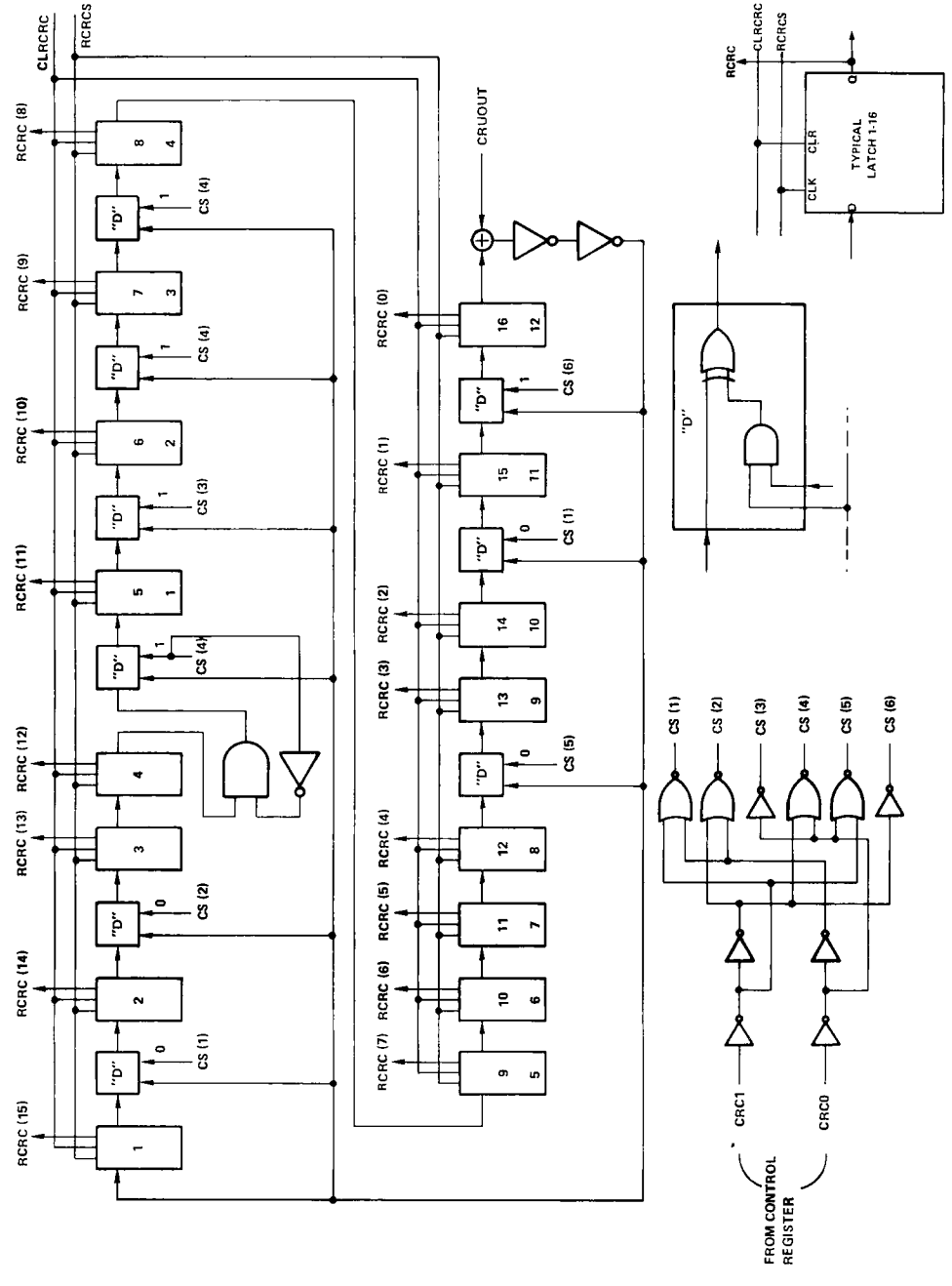


FIGURE 6. RECEIVE CRC REGISTER DIAGRAM

2.1.2.7 Transmit Buffer Register

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

8	7	6	5	4	3	2	1	0
XBR 8	XBR 7	XBR 6	XBR 5	XBR 4	XBR 3	XBR 2	XBR 1	XBR 0
MSB								LSB

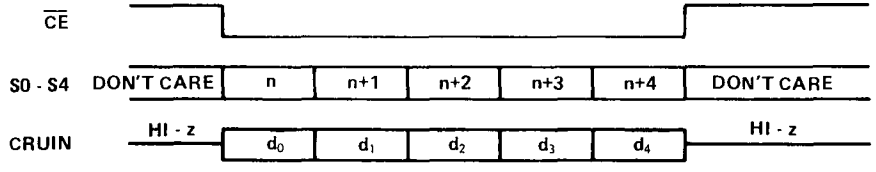
TRANSMIT BUFFER REGISTER BIT ADDRESSES

TABLE 4. CRU OUTPUT SELECT BIT 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)	S1OD(9)	S2OD(9)		RCRC(9)	XCRC(9)			
8	MDSL(2)			IROD(7)			XCRC(8)	XBR(8)	XBR(8)
7	MDSL(1)								
6	MDSL(0)								
5	CSL(1)								
4	CSL(0)								
3	CLK4M								
2	RSCL(2)								
1	RSCL(1)								
0	RSCL(0)	S1OD(0)	S2OD(0)	IROD(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 bit addresses using the five address 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 address assignments of the SCC. Following Table 5 is a detailed discussion of each bit. Addresses 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	Transmit Buffer Register Empty
21	RBRL	X	X	X	X	X	X	Receive 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					Transmit Abort Interrupt
	—			X	X	X	X	Not Used (Always = 0)
17	XBINT	X	X	X	X	X	X	Transmit 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					Receive Abort
	—	X		X	X			Not Used (Always = 0)
	RSBD					X	X	Receive Start Bit Detect
13	RHRL		X					Receive Holding Register Loaded
	—	X		X	X			Not Used (Always = 0)
	RFBD					X	X	Receive Full Bit Detect
12	RHROV		X					Receive Holding Register Overrun
	—	X		X	X			Not Used (Always = 0)
	RFER					X	X	Receive Framing Error
11	ROVER	X	X	X	X	X	X	Receive Overrun
10	RPER	X		X	X	X	X	Receive Parity Error
	RZER		X					Receive Zero Error
9	RCVERR	X		X	X	X	X	Receive Error
	RFLDT		X					Receive Flag Detect
8-0	RBR	X	X	X	X	X	X	Receive Buffer Register (Received Data)

8

Bit 31
All modes (INT)—

Interrupt. All modes INT = DSCINT + TIMINT + RBINT + XAINT + XBINT. The interrupt output control line (INT) is active when this status signal is a logic one.

Bit 30
All modes (FLAG)—

Register Load Control Flag Set. In all modes FLAG = LDCTRL + LDSYN + 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 $\overline{\text{DSR}}$ or $\overline{\text{CTS}}$ inputs, or RTSAUT 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 output bit 21 (DSCENB).

Bit 28
All modes (CTS)—

Clear To Send. The CTS signal indicates the inverted status of the $\overline{\text{CTS}}$ device input in all modes.

Bit 27
All modes (DSR)—

Data Set Ready. The DSR signal indicates the inverted status of the $\overline{\text{DSR}}$ 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 output 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 bit address 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 bit address 22 (XAIENB)

▶8

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 bit 25 (LXBC).

Bit 21
All modes (RBRL)— **Receive Buffer Register Loaded.** The RBRL signal is set in all modes when a complete character has been transferred from the receive shift register (RSR) to the RBR. RBRL is reset by an output to output 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 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)— **Transmit 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)— **Transmit 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 transmit-buffer.

Bit 16
All modes (RINT)— **Receiver Interrupt.** In all modes RINT = [RBRL (input bit 21) OR RHRL (input bit 13) OR RABRT (input bit 14)] AND RIENB (output bit 18). RINT indicates the presence of an enabled interrupt caused by a loaded receive buffer or a loaded receive 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 bit address 18 (RIENB).

Modes 5, 6 (RSBD)— **Receive 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)— **Receive 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 bit address 26, RHRRD (receive holding register read).

Modes 5, 6 (RFBD)— **Receive 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)— **Receive 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 bit address 26 (RHRRD).

Modes 5, 6 (RFER)— **Receive 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 (receive 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 bit address 18 (RIENB).

Bit 10

Modes 0, 2, 3, 5, 6 (RPER)— **Receive 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)— **Receive 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 Hold Register Read).

Bit 9

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

Mode 1 (RFLDT)— **Receive 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 bi address 18 (RIENB). RFLDT is also set when RABRT is set.

Bit 8 - Bit 0

All modes (RBR8-RBR0)— **Receive Buffer Register.** The receive-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	R				
	X	I				
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 interval clock cycle each time a bit is to be sent. If DRCK32 is reset, or $\overline{CTS}$ is inactive (HIGH), SCTX is generated on every one-to-zero transition of 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

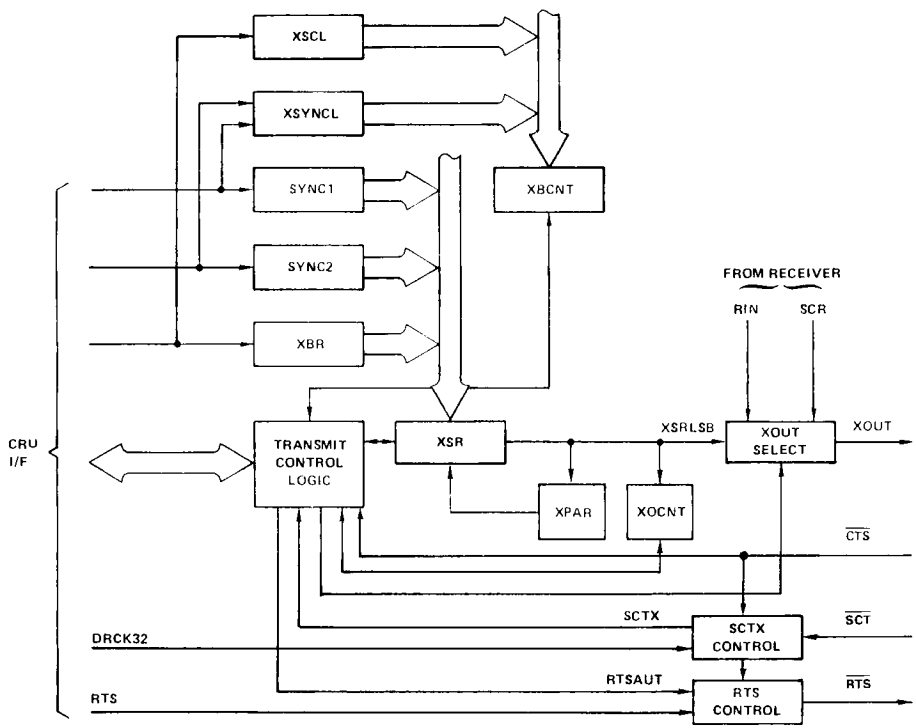


FIGURE 7. TMS 9903 SCC TRANSMITTER BLOCK DIAGRAM

transition of $\overline{SCT}$, the $\overline{RTS}$ signal is updated by the internal, automatic request-to-send signal, (RTSAUT) unless output bit address 17 (RTS) is addressed. If RTS is addressed the $\overline{RTS}$ signal is controlled by the level of output bit 17 until either the RESET or CLRXMT (clear transmitter) command is issued.

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)

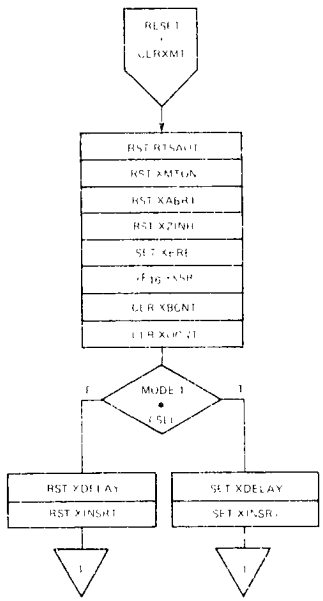


FIGURE 8. TRANSMITTER INITIALIZATION

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 synchronization. The received character is assembled in the receive shift register (RSR) according to the length specified in control register bits 2, 1,0 — receive 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 receive 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 receive buffer register loaded flag RBRL is set. If RBRL was set already, the receiver overrun flag ROVER is set. Incorrect 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.

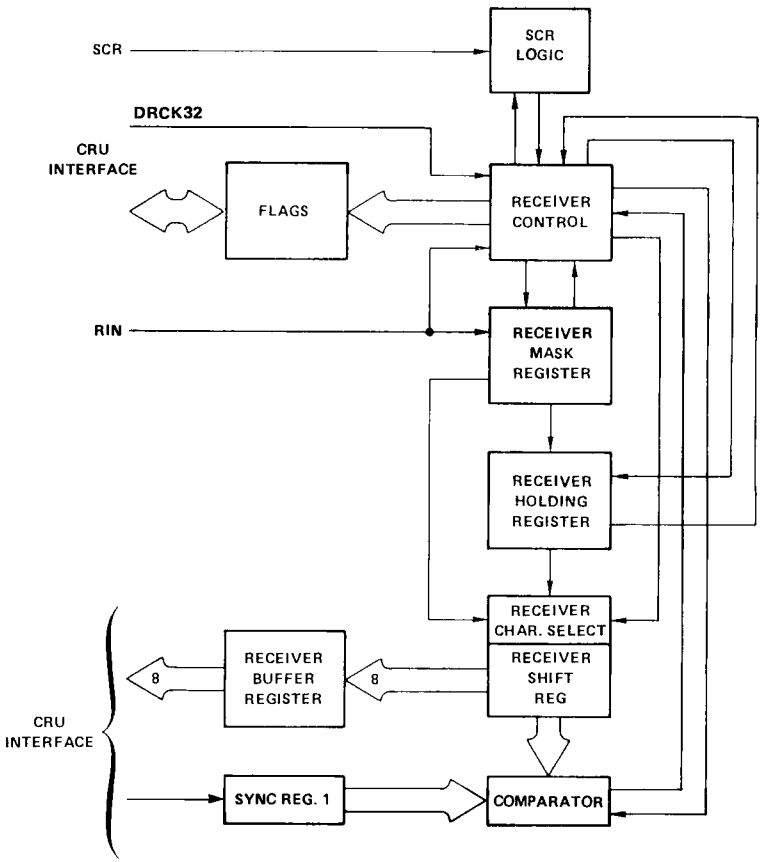


FIGURE 9. TMS 9903 RECEIVER BLOCK DIAGRAM

8

2.3.2 Receiver Initialization

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

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

2.4 TRANSMITTER AND RECEIVER OPERATION

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

2.4.1 Mode 0 Operations

2.4.1.1 Transmitter Operation

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

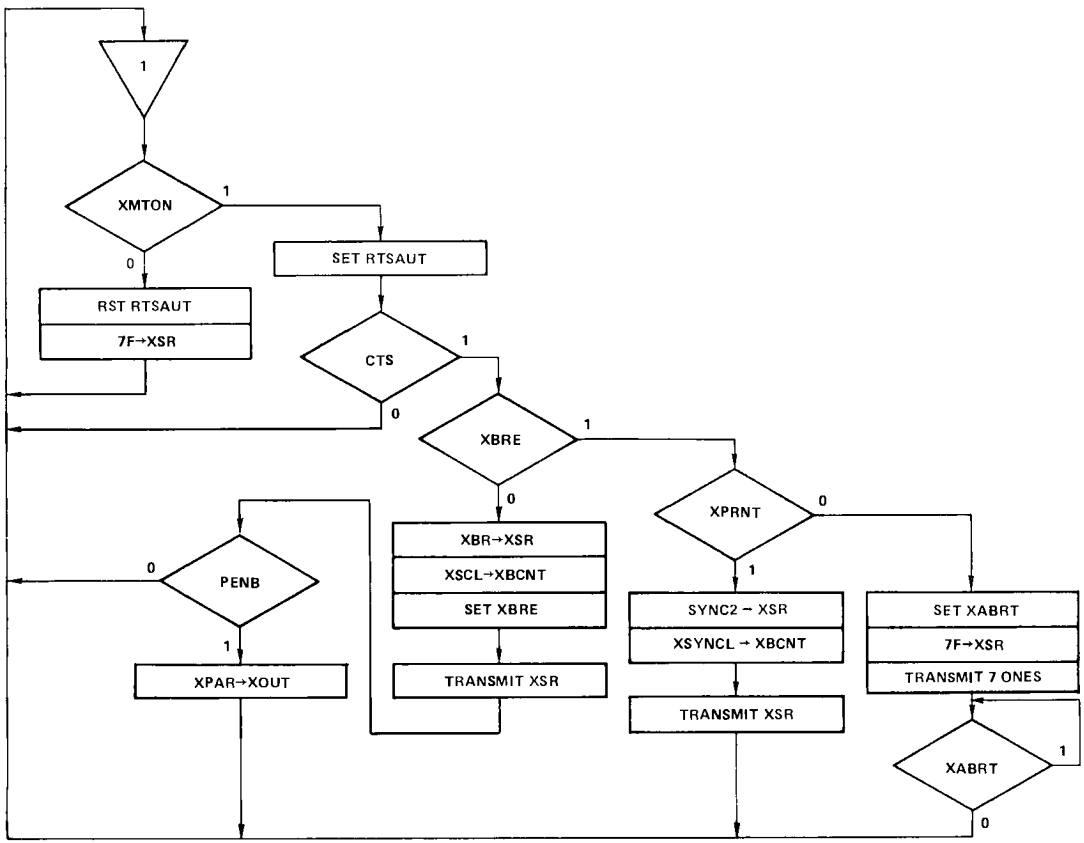


FIGURE 10. MODE 0 TRANSMITTER OPERATION

2.4.1.2 Mode 0 Receiver Operation

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

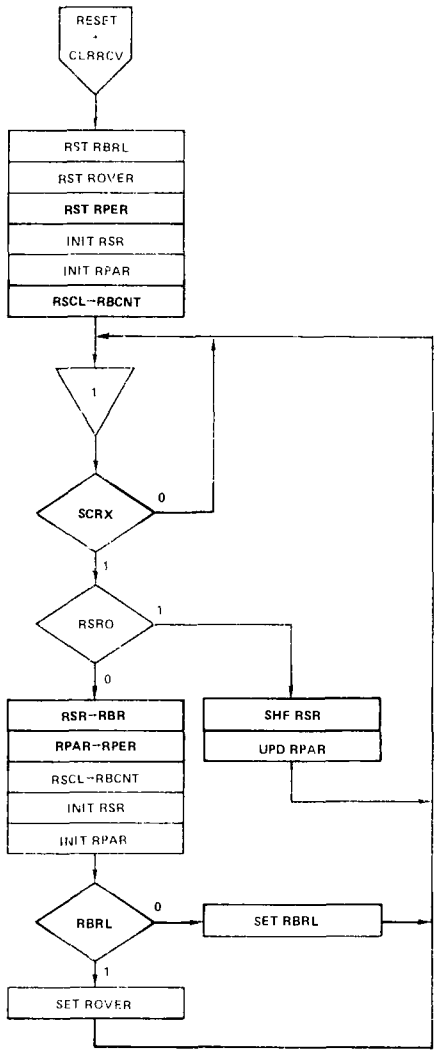


FIGURE 11. MODE 0 RECEIVER OPERATION

2.4.2 Mode 1 Operation

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).

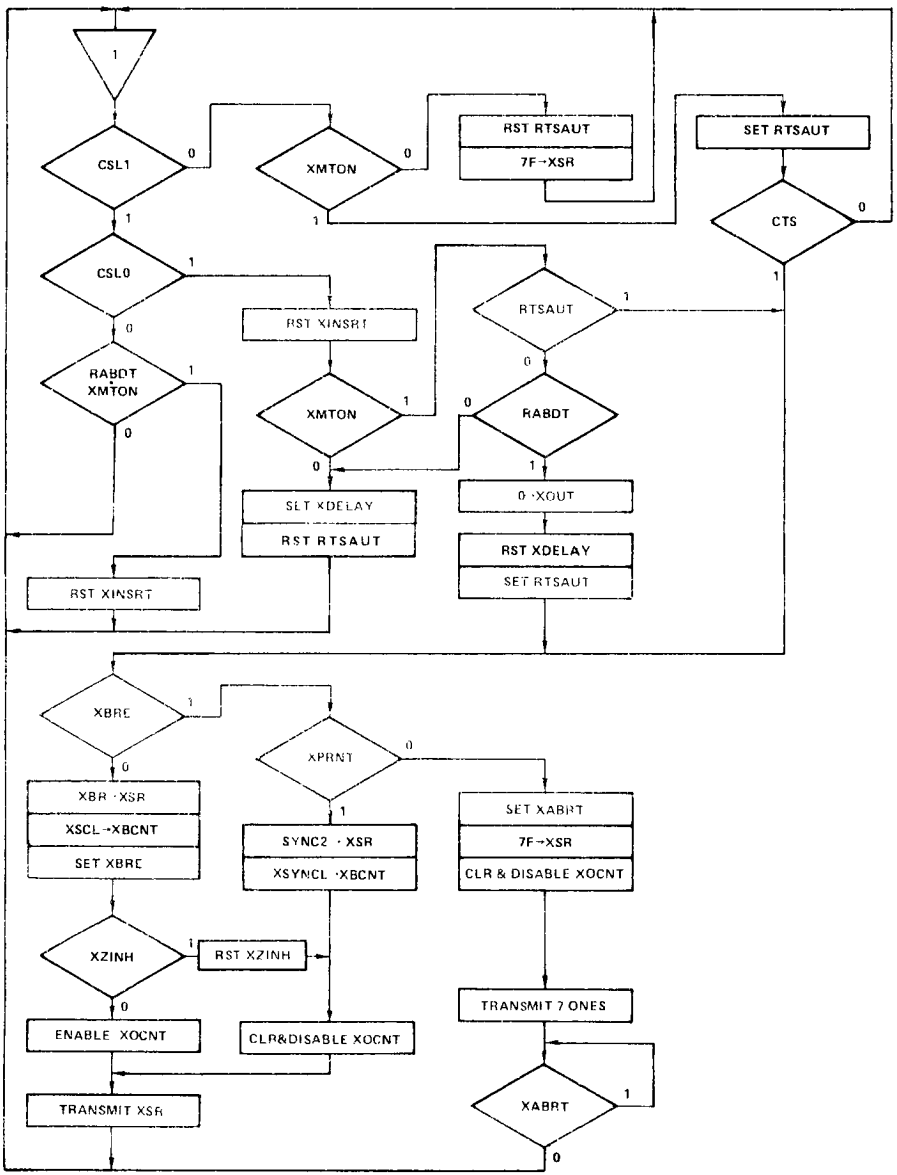


FIGURE 12. MODE 1 TRANSMITTER OPERATION

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, XOUT = RIN. When XINSRT is reset by detection of an EOP, RIN is delayed a single bit-time before being transmitted on XOUT.

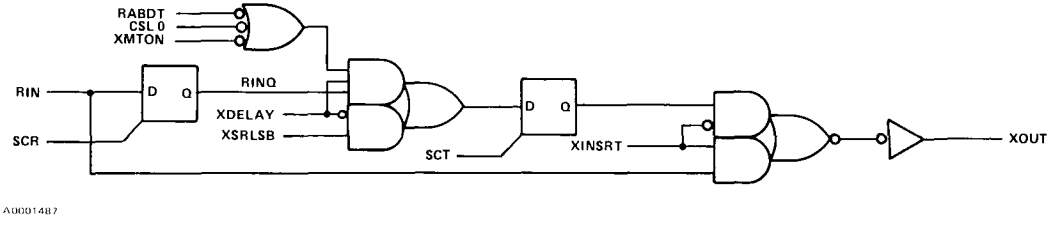


FIGURE 13. XOUT SELECT LOGIC

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.

▶8

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.

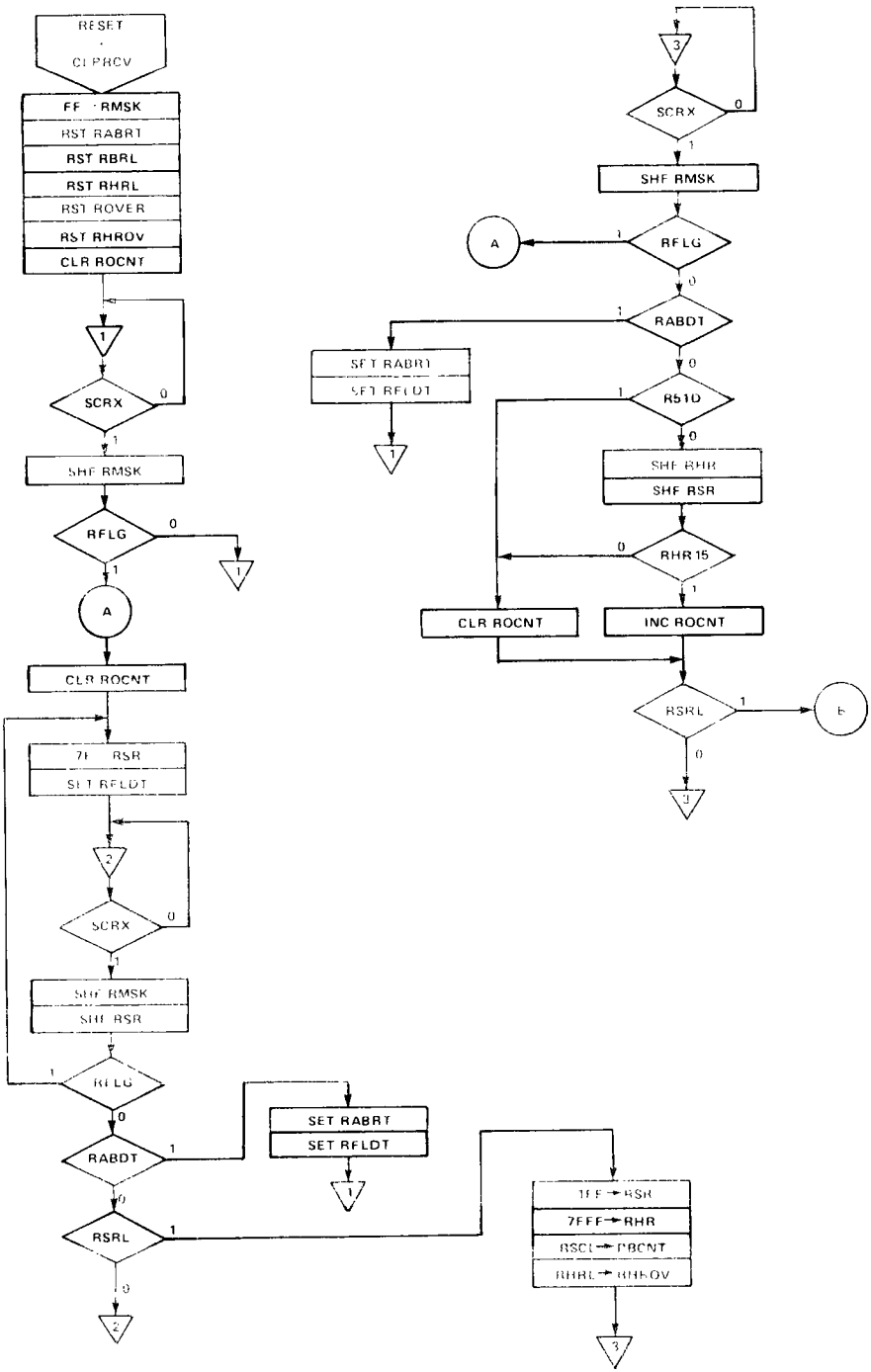


FIGURE 14. MODE 1 RECEIVER OPERATION (PAGE 1 OF 2)

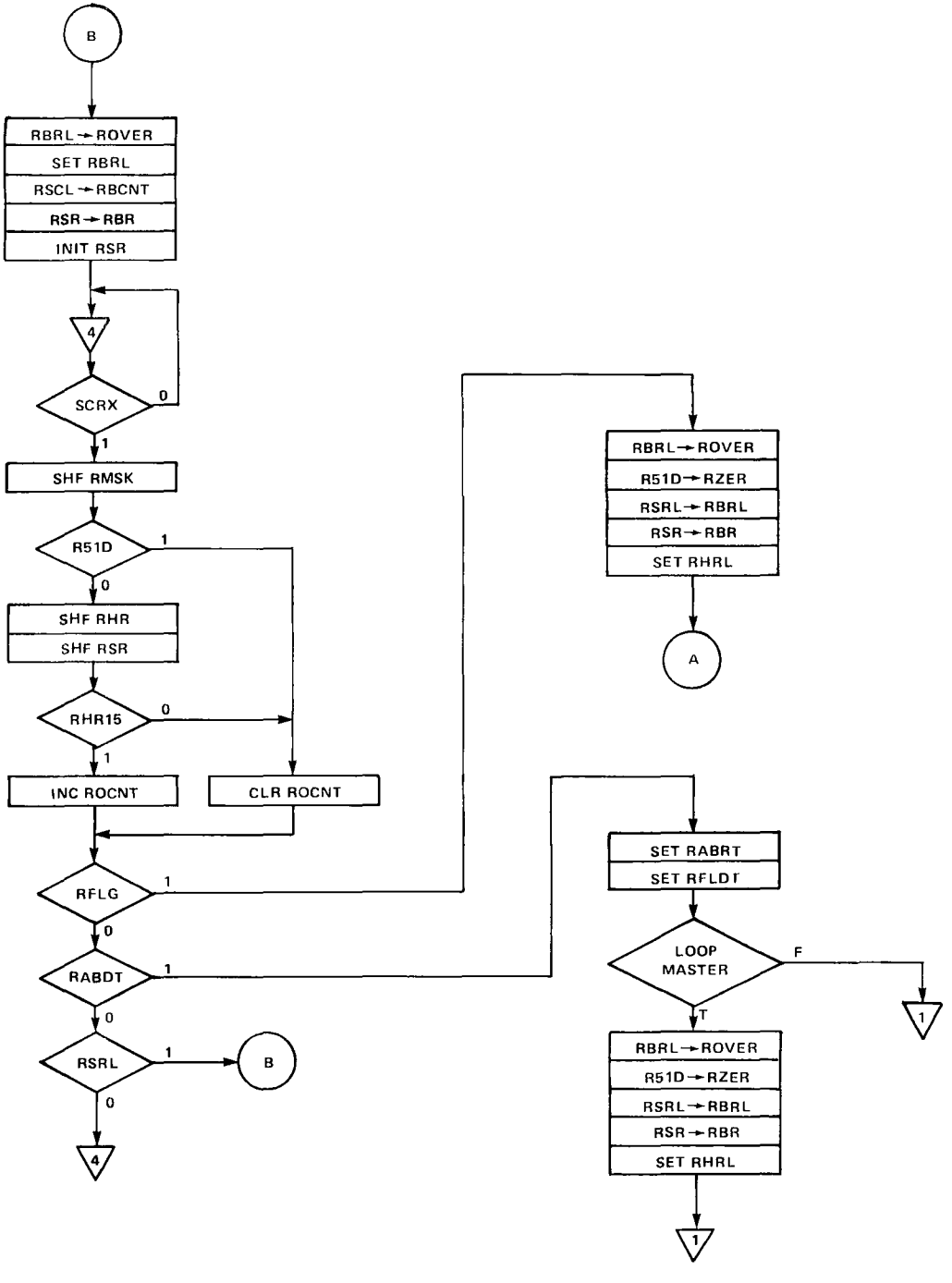


FIGURE 14. MODE 1 RECEIVER OPERATION (PAGE 2 OF 2)

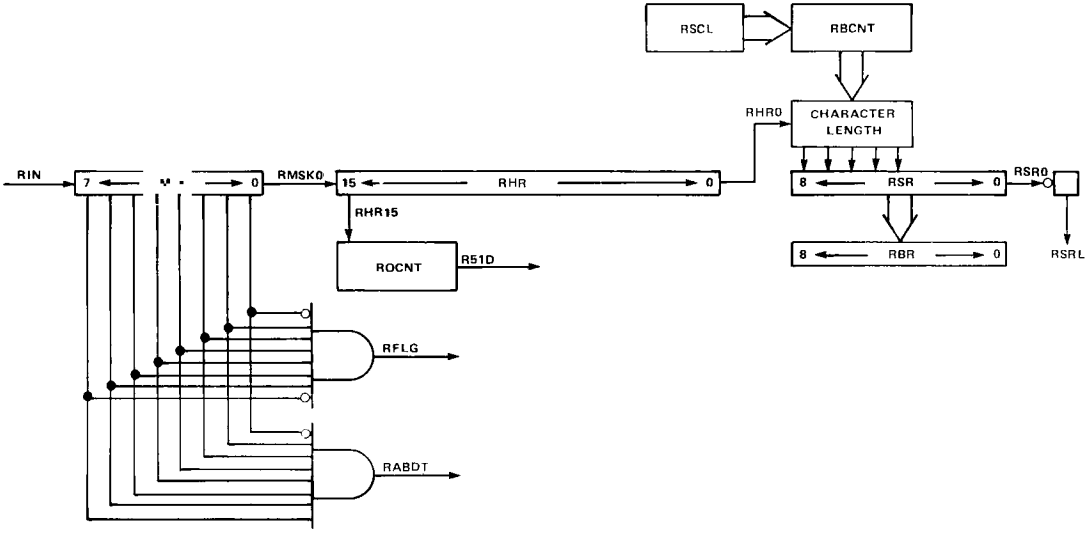


FIGURE 15. MODE 1 RECEIVER CIRCUITRY

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 (receive character length select field of the control register) are loaded into RBCNT (receive 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 receive buffer register (RBR) and the receive buffer register loaded flag (RBRL) is set.

8

2.4.2.2.4 Character Assembly. Each time RSRL = 1, RSCL is transferred to RBCNT, RSR to RBR, RBRL is set; and RSR is initialized to all ones except for the MSB of the selected character length. That is, for a seven-bit character, RSR is loaded with 0001111111_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, 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 receive 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 receive 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.

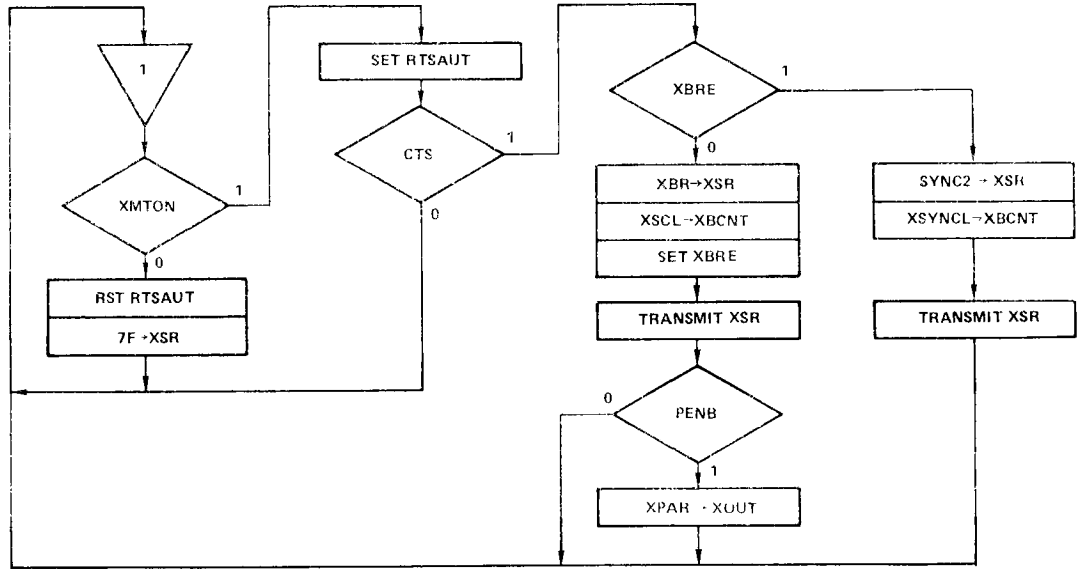


FIGURE 16. MODE 2 TRANSMITTER OPERATION

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.

8

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 SCT. Mode 5 is asynchronous operation with one start and two stop bits, while mode 6 is asynchronous operation with one start and one stop bit.

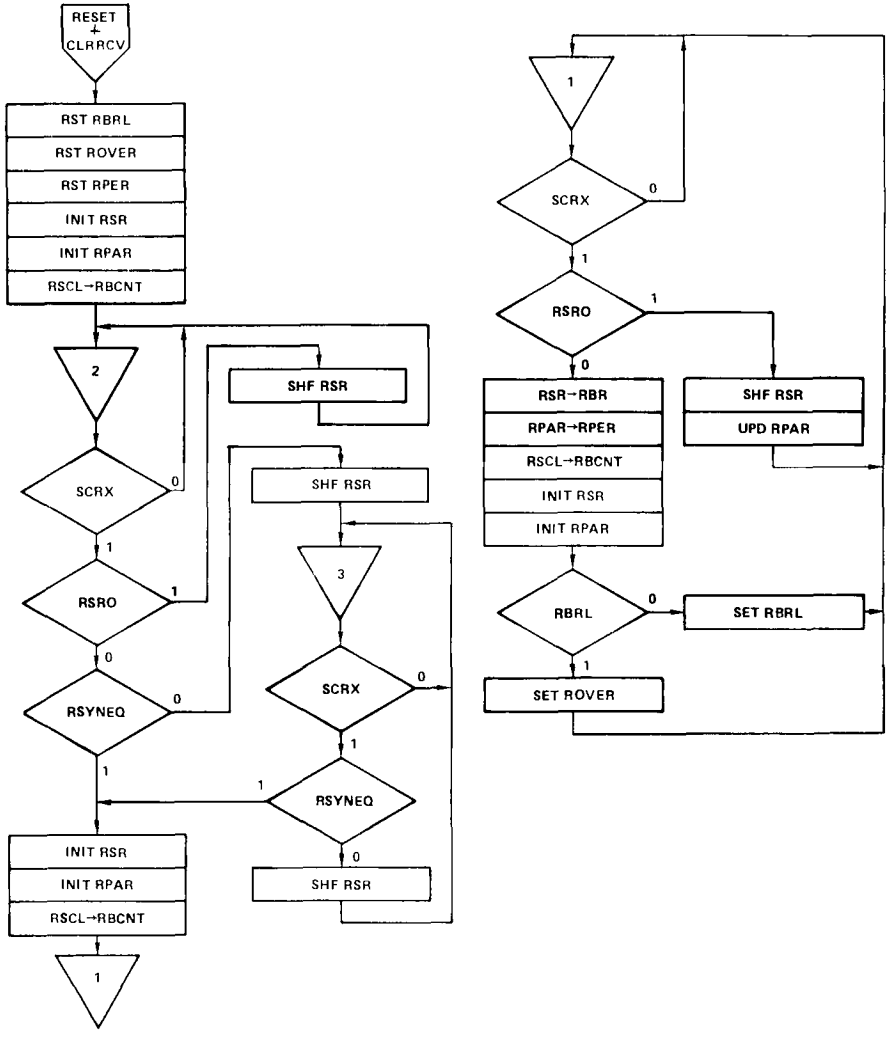


FIGURE 17. MODE 2 RECEIVER OPERATION

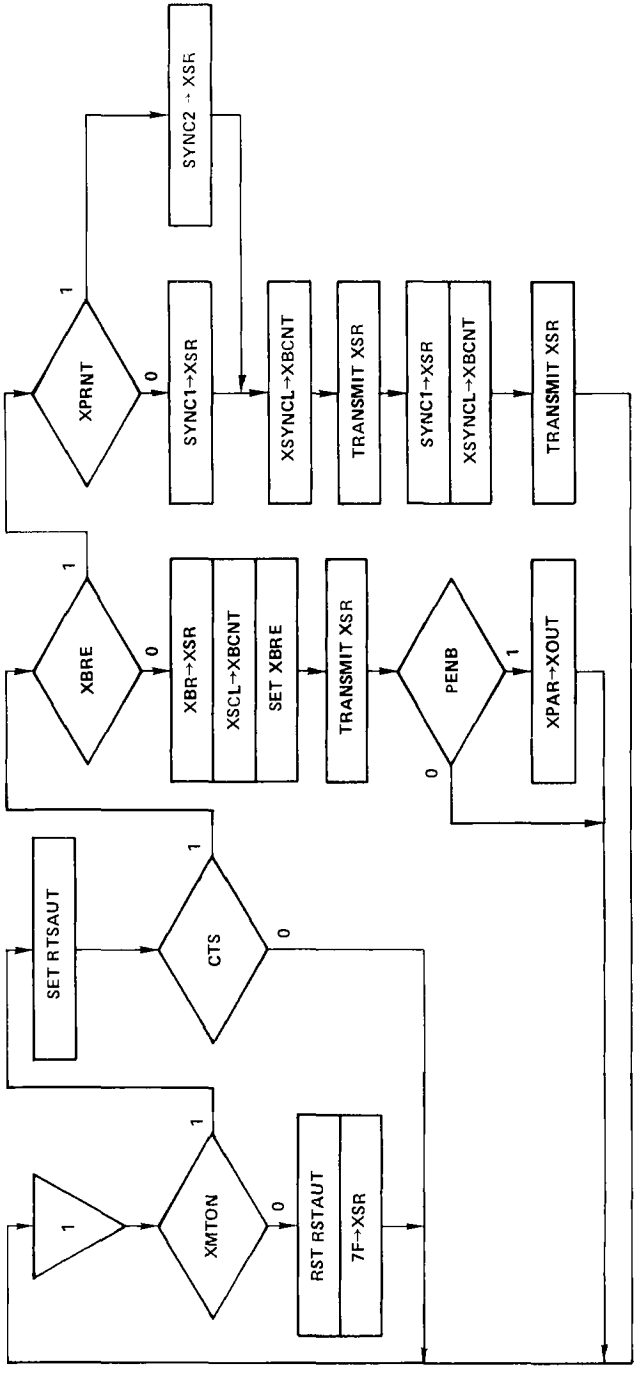


FIGURE 18. MODE 3 TRANSMITTER OPERATION

8◀

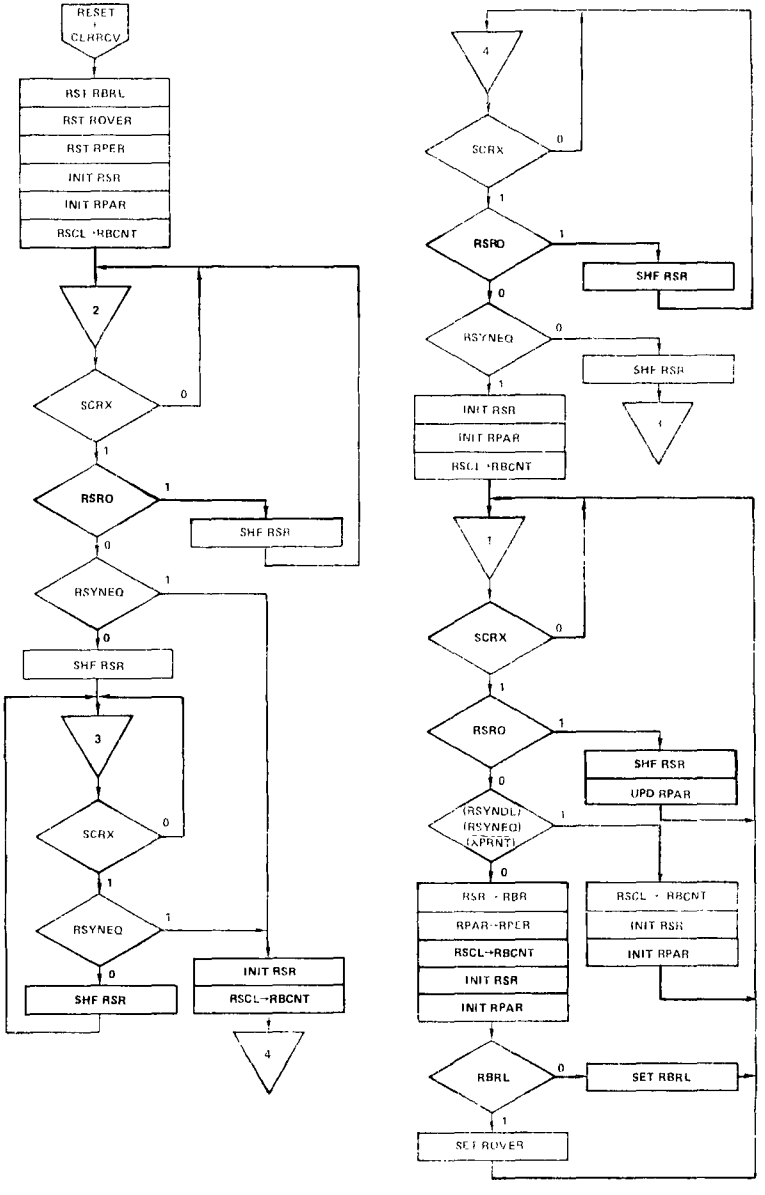


FIGURE 19. MODE 3 RECEIVER OPERATION

▶8

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 RTS and XOUT are set, placing the transmitter in the inactive state. When XMTON is set by the CPU, the 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.

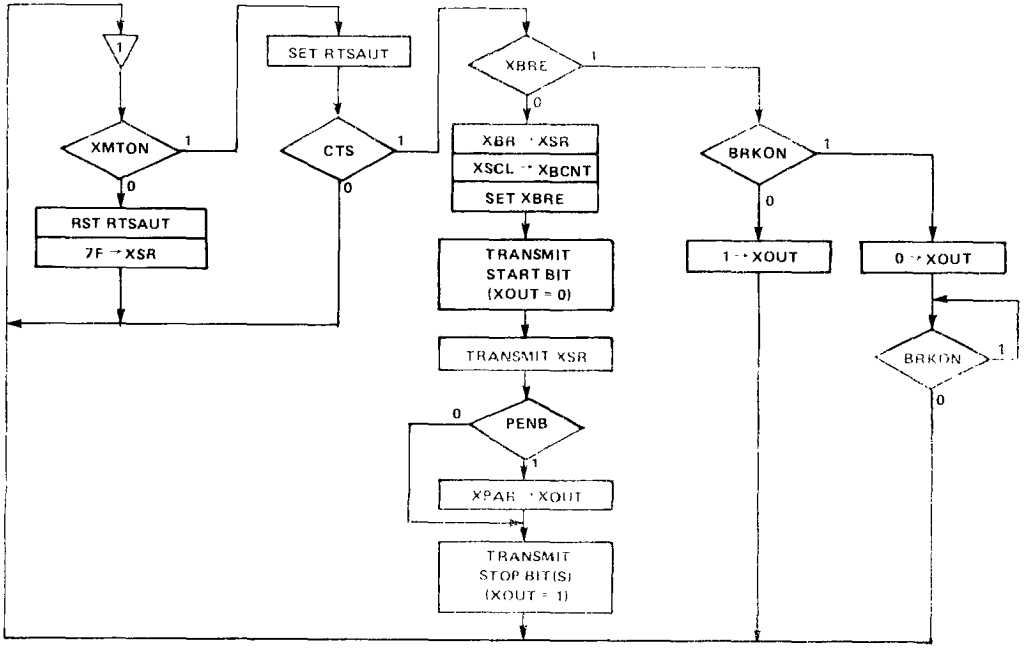


FIGURE 20. MODE 5 OR 6 TRANSMITTER OPERATION

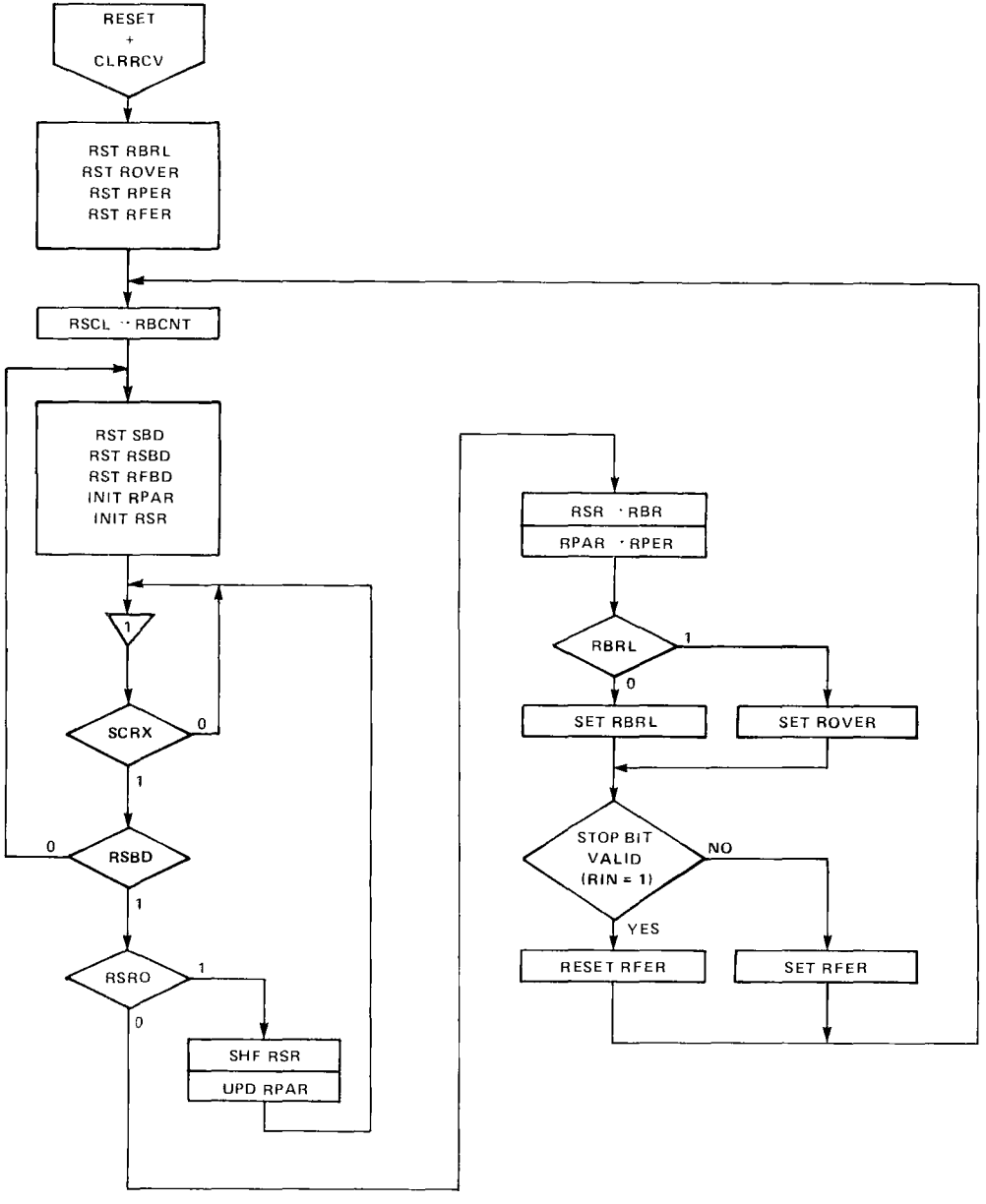


FIGURE 21. MODE 5 OR 6 RECEIVER OPERATION

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 (receive 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 verfying 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 receive 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, and the interval timer is enabled. 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 (the zero state is counted the same as other counter states). A flowchart for interval timer operation is illustrated in Figure 23.

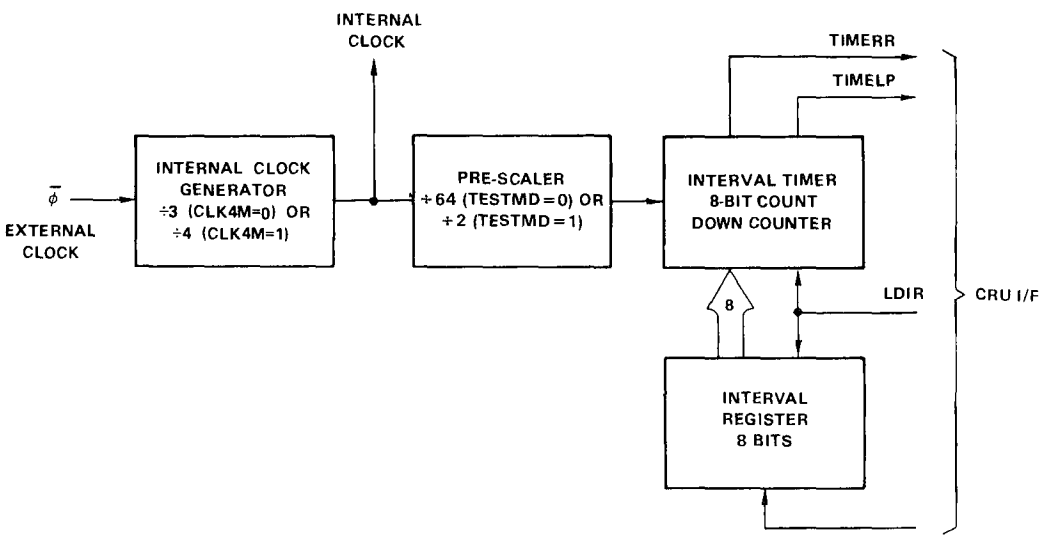


FIGURE 22. INTERVAL TIMER BLOCK DIAGRAM

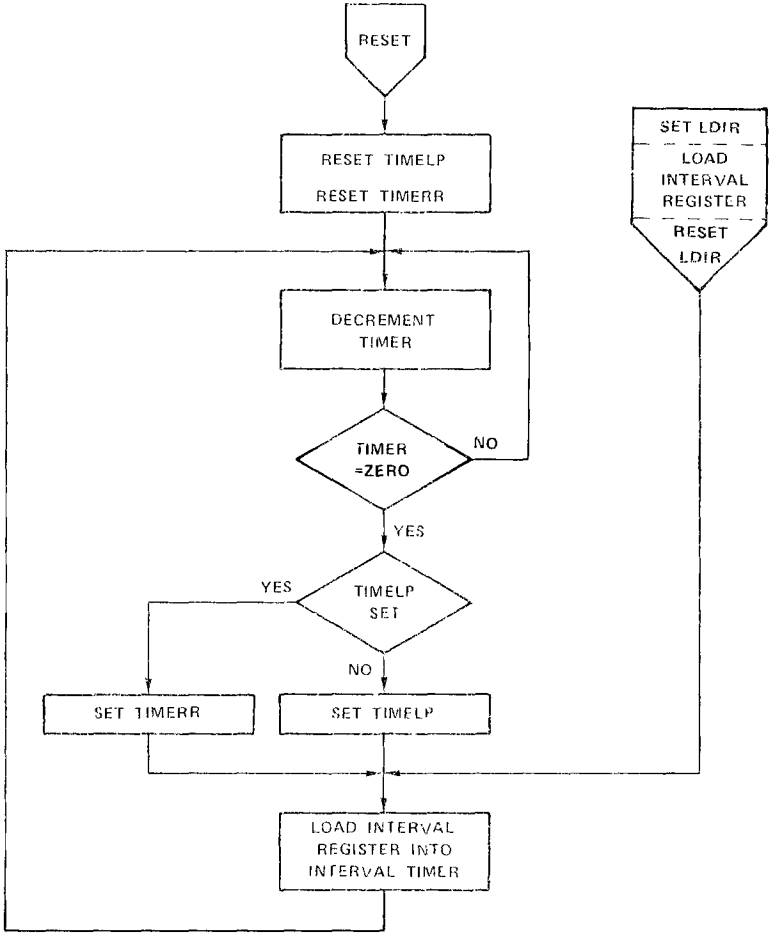


FIGURE 23. INTERVAL TIMER OPERATION

2.5.1 Time Interval Programming

8

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 (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 DSR, CTS, or RTSAUT changes levels

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

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

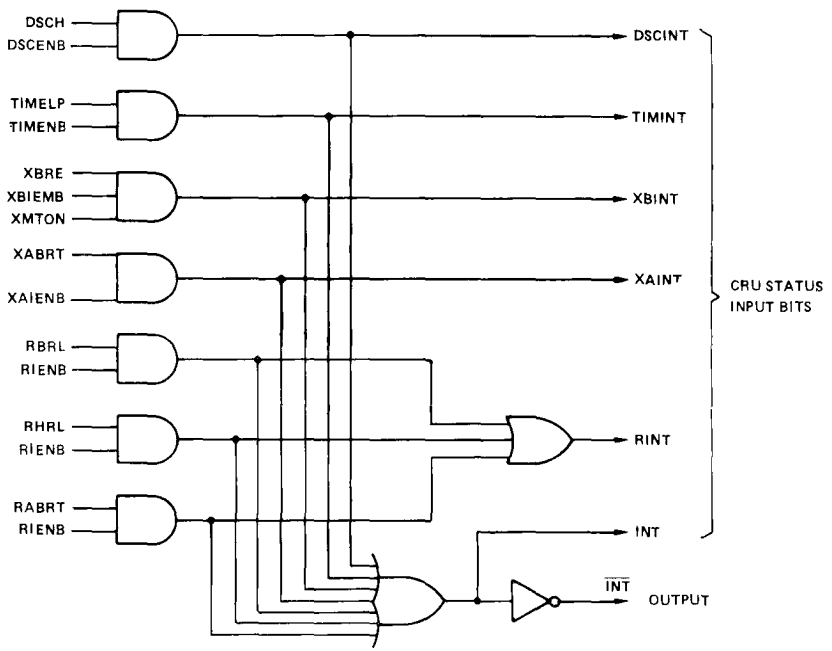
4) XABRT =1. XABRT (transmitter abort) is set in mode 0 and 1 when no data is available for transmission, no provision is made for a fill character, and XMTON is turned ON.

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

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

7) RABRT = 1. RABRT (receive 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



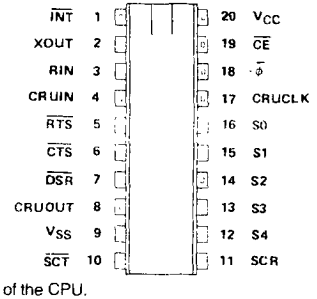
NOTE: See Tables 1 and 5 for input and output signal definitions.

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 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
S3	13	IN	function.
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 bit 0 is 0040_{16}, and the ϕ 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.

►8

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.

```
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.

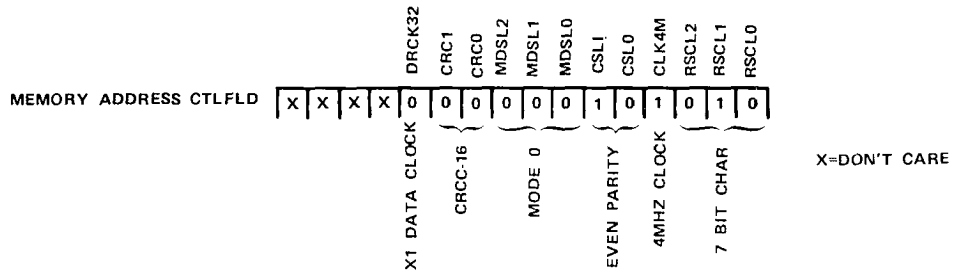
The receiver should be initialized with the CLRRCV command after the control register is loaded to ensure that the receiver logic will assemble the first received character at the proper length.

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

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.

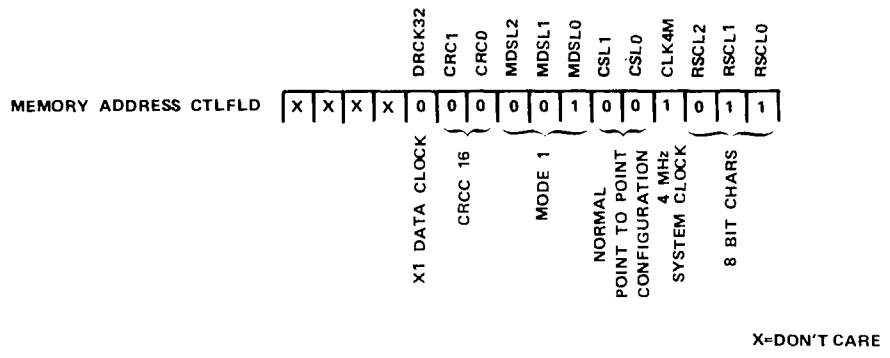
```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
LDSYN2  EQU   27
XPRNT   EQU   23
        .
        .
        .
        LI    R12,>40      Initialize CRU Base.
        SBO   RESET        Reset device and set LDCTRL.
        LDCR  @ CTLFLD,12  Load Control Register and Reset LDCTRL.
        SBZ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

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, user software will then handle message preparation, transmission, reception, and accountability, while the TMS 9903 message link handles synchronization and control.

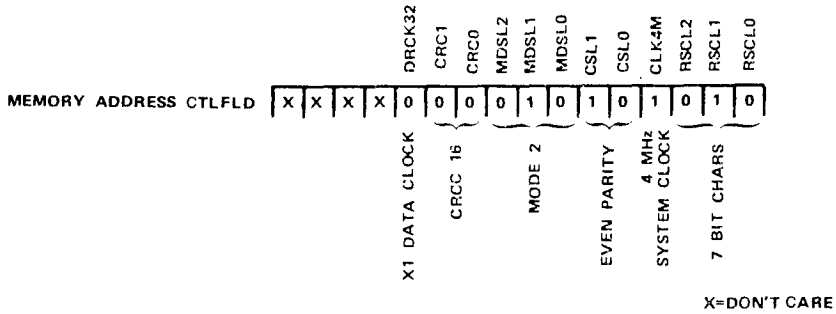
```
RESET   EQU   31
CLRRCV  EQU   30
CLRXMT  EQU   30
LDSYN2  EQU   27
CLXCRC  EQU   29
CLRCRC  EQU   29
LXCRC   EQU   24
LRCRC   EQU   12
        .
        .
        .
        LI    R12,>40
        SBO   RESET      Reset Device
        LDCR  @ CTLFLD,12 Load Control Register
        SBZ   CLRRCV  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

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
```

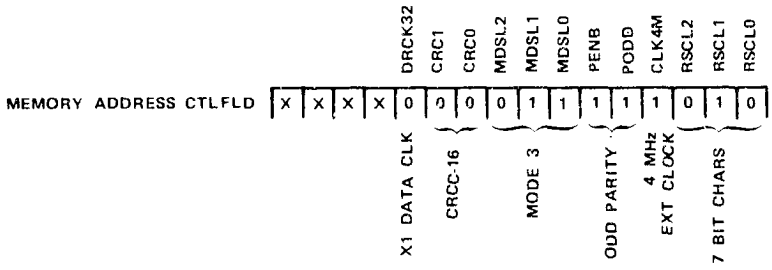


3.1.4 Mode 3 Operation

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 transmit 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
```



X=DON'T CARE

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.

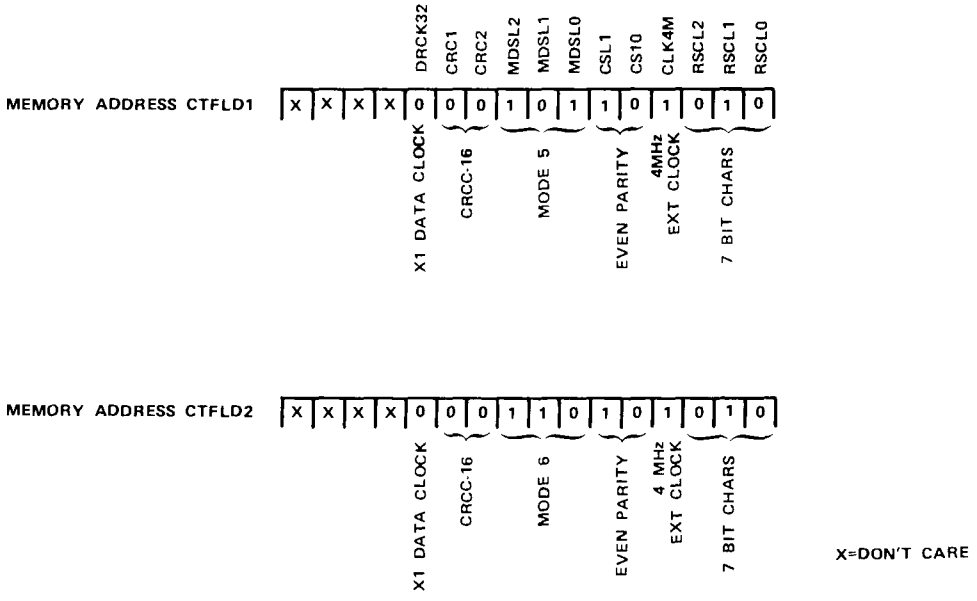
3.1.5 Mode 5 and 6 Operation

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
CTFLD2   DATA   >01AA
```

8◀



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          19₁₆  25₁₀. 25 × 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
          .
          .
          .
```

▶8

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 Transmit 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 Transmit CRC
        SBZ     LXCRC       Reset LXCRC
        SWPB    R3
        TB      XBRE
        JNE     $-1
        LDCR    R3,8
        SBZ     LXBC
        SWPB    R3
        TB      XBRE
        JNE     $-1
        LDCR    R3,8
        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 automatic 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 transmit buffer and to update simultaneously the transmit 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 transmit buffer. After the last character is transmitted, the accumulated CRC is read from the SCC and transmitted, and XMTON is reset. The transmitter and RTSAUT become inactive upon completion of transmission of the last character. Note that $\overline{RTS}$ can be CPU-controlled by setting and resetting RTS (bit address 17). This disables the RTSAUT signal until the transmitter is reset by the RESET or CLRXMT command.

8◀

*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          Test for RBRL = 1
        JNE     RCVLP
        STCR    *R2,8       Store Character
        SBZ     18          Reset RBRL
        SBO     12          Set LRCRC to
        LDCR    *R2,8       Update Receive CRC Register
        SBZ     12          Reset LRCRC
        DEC     R3          Decrement Count
        JEQ     RCVEND      End if Count = 0
        CB      *R2+,R4     Compare to EOB Character and Increment Point
        JNE     RCVLP       Loop If Not Complete
RCVEND  TB      21          Test For RBRL = 1
        JNE     RCVEND
        STCR    R1,8        Store Transmitted CRC Value
        SBZ     18          Reset RBRL
        SWPB    R1          Swap CRC Bytes
        TB      21          Test for RBRL = 1
        JNE     $-1
        STCR    R1,8        Store Transmitted CRC Value
        SBZ     18          Reset RBRL
        SBO     12          Set LRCRC to
        STCR    R6,0        Read Receive 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 receive CRC register. Note that in mode 1 operation the RCVEND instructions to read the received CRC could be replaced with:

▶8

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

3.4 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

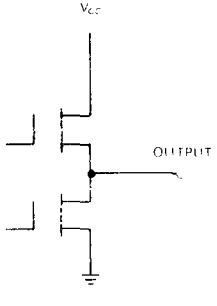
```
        SBO     13          Set Load Control Flag
        LDCR    @INTVL2,8   Load Register
        SBZ     13          Reset Flag
        .
        .
        .
INTVL2  BYTE    10240/64
```

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.

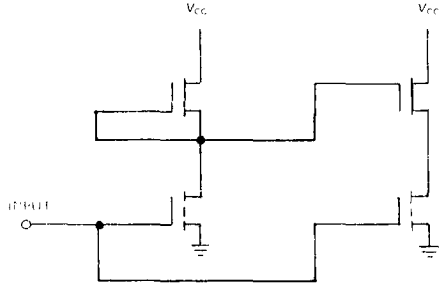
```
        BLWP    @ITVCHG     Call Subroutine
        .
        .
        .
ITVCPC  LIMI    0           Mask All Interrupts
        MOV     @24(R13),R12 Load CRU Base Address
        SBO     13          Set Flag
        LDCR    @INTVL2,8   Load Register
        SBZ     13          Reset Flag
        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

EQUIVALENT OF OUTPUTS



EQUIVALENT OF INPUTS



8

4. TMS 9903 ELECTRICAL SPECIFICATIONS

4.1 ABSOLUTE MAXIMUM RATING 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 20 V
Continuous power dissipation 0.7W
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	5.25	V
Supply voltage, V_{SS}		0		V
High-level input voltage, V_{IH}	2.2	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
I_I	Input current (any input)	V_I = 0 V to V_{CC}	10		10	μA
V_{OH}	High-level output voltage	I_{OH} = −100 μA I_{OH} = −400 μA	2.4 2	3		V
V_{OL}	Low-level output voltage	I_{OL} = 3.2 mA			0.4	V
$I_{CC(av)}$	Average supply current from V_{CC}	Operating at $t_{c(\phi)}$ = 250 ns, T_A = 25°C			100	mA
C_i	Capacitance, any input	f = 1 MHz, all other pins at 0 V			15	pF

4.4 TIMING REQUIREMENTS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

PARAMETER		MIN	TYP	MAX	UNIT
$t_{c(\phi)}$	Clock cycle time		333		ns
$t_{r(\phi)}$	Clock rise time		12		ns
$t_{f(\phi)}$	Clock fall time		12		ns
$t_{w(\phi H)}$	Pulse width, clock high		240		ns
$t_{w(\phi L)}$	Pulse width, clock low		55		ns
$t_{w(CC)}$	CRUCLK pulse width		100		ns
$t_{su(ad)}$	Address setup time, CRUOUT before CRUCLK		220		ns
$t_{su(CF)}$	Chip enable setup time before CRUCLK		180		ns
$t_{h(ad)}$	Address hold time, CE and CRUOUT after CRUCLK		80		ns
$t_{h(CI)}$	Hold time, CRUIN after address		20		ns
$t_{d(ad\,CI)}$	Delay time, address to CRUIN valid		400		ns

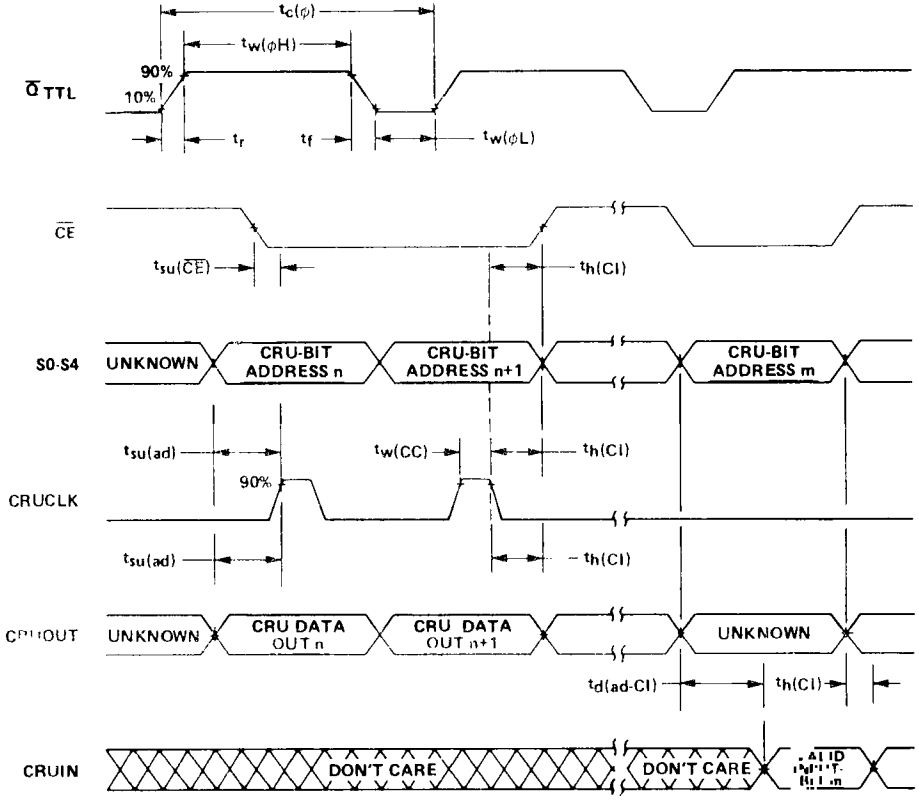
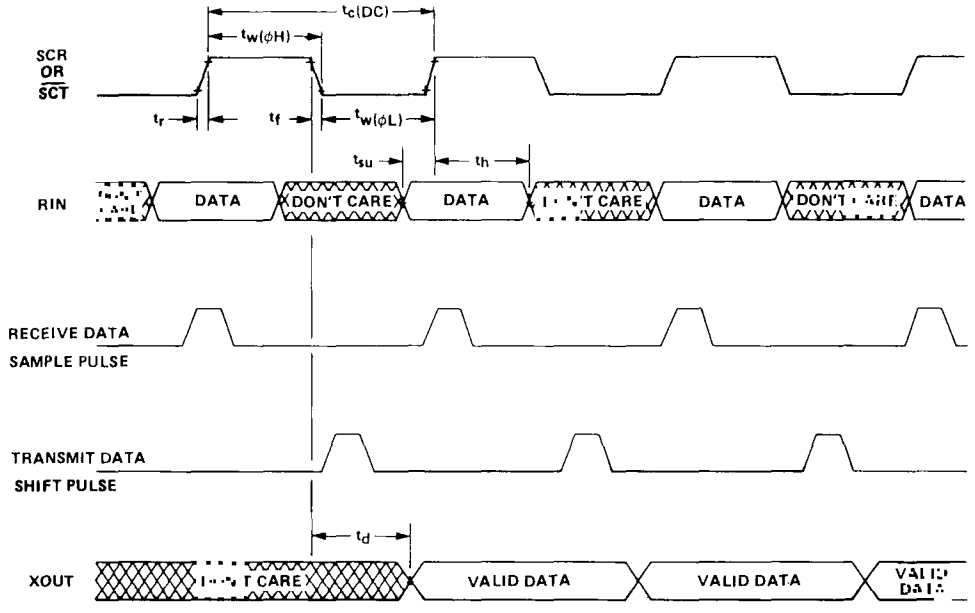


FIGURE 25. TIMING DIAGRAM



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

▶8

FIGURE 26. RECEIVE/TRANSMIT DATA CLOCK TIMING DIAGRAM

TIM 9904
FOUR-PHASE CLOCK GENERATOR/DRIVER

1. INTRODUCTION

1.1 Description

The TIM 9904 four-phase clock generator/driver (SN74LS362) is a 20-pin dual-in-line package peripheral device designed for use with the Texas Instruments TMS 9900 microprocessor family and other microprocessors. The TIM 9904 internal oscillator can be controlled by a fundamental or overtone crystal, or capacitor and a tank circuit, or an external oscillator. The TIM 9904 is fabricated using low-power Schottky technology and is available in both plastic and ceramic packages.

1.2 Key Features

- Clock generator/driver for the TMS 9900 or other microprocessors
- MOS and TTL four-phase outputs
- Self-contained oscillator can be crystal- or tank-controlled
- External oscillator can be used
- Clocked D-type flip-flop with Schmitt-trigger input for reset signal synchronization.
- Standard 20 pin plastic and ceramic package

2. ARCHITECTURE

The TIM 9904 clock generator/driver (Figure 1) comprises an oscillator, a divide-by-four counter, a second divide-by-four-counter with gating to generate four clock phases, high-level (12-volt) output drivers, low-level (5-volt) complementary output drivers, and a D-type flip-flop controlled by an external signal and a $\phi 3$ clock. The four high-level clock phases provide clock inputs to a TMS 9900 (or other) microprocessor. The four complementary TTL-level clocks can be used to time memory or other logic functions in a TMS 9900 computer system. The D-type flip-flop can be used, for example, to provide a reset signal to a TMS 9900, timed by $\phi 3$, on receipt of an input to the FFD input from power turn-on or a manual switch closure. Other applications are possible. A safety feature incorporated in the ϕ outputs causes the ϕ outputs to go low if an open occurs in the V_{CC} supply common to TIM 9904 and TMS 9900, thus protecting the TMS 9900.

The frequency of the internal oscillator can be established by a quartz crystal or a capacitor and LC circuit. Either a fundamental or overtone crystal may be used. The LC circuit connected to the tank inputs selects the desired crystal overtone or establishes the internal oscillator frequency when a capacitor is used instead of a crystal. An LC circuit must always be used at the tank inputs when using the internal oscillator. An external oscillator may be used, if desired.

8◀

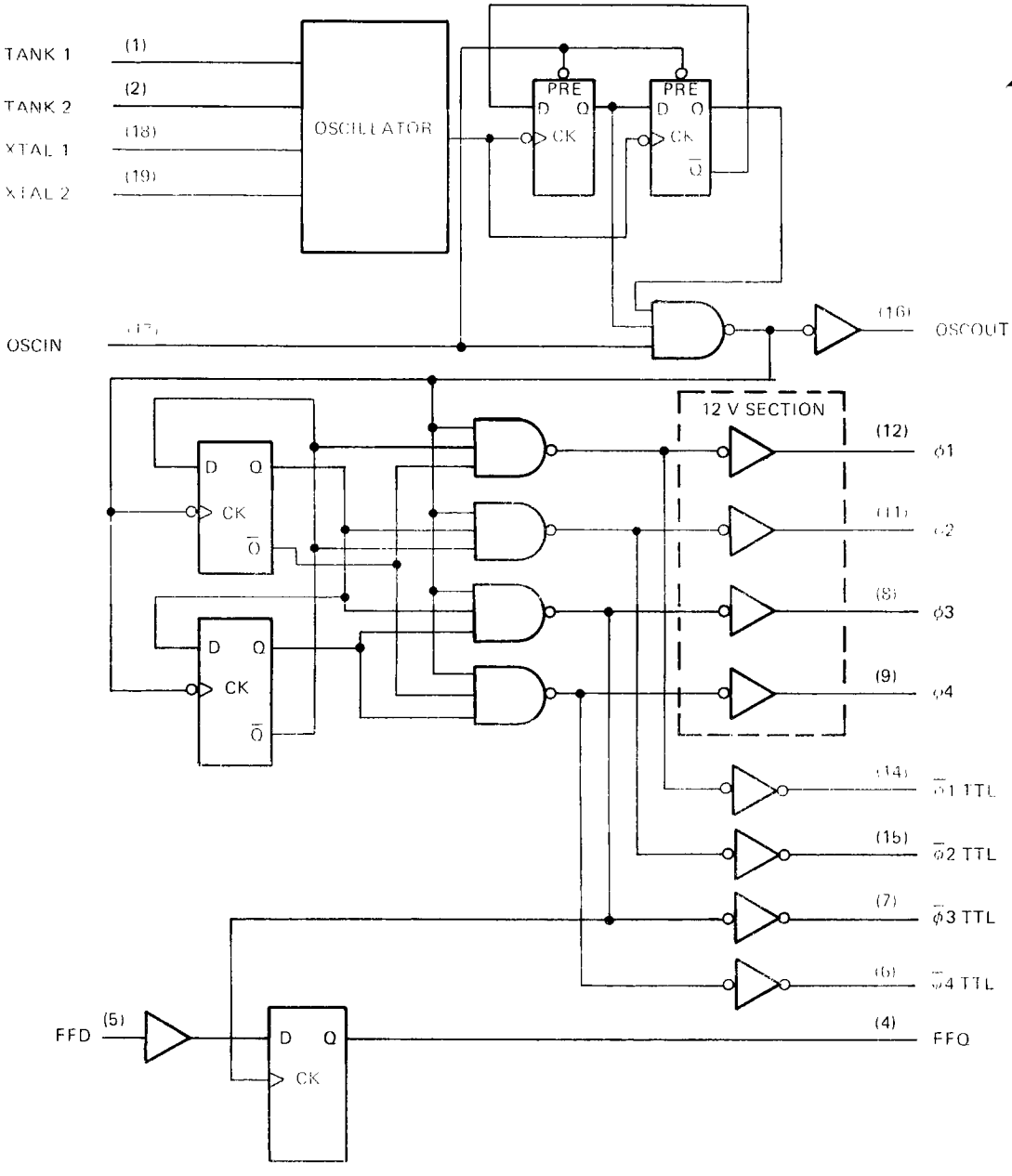


FIGURE 1—TIM 9904 CLOCK GENERATOR/DRIVER FUNCTION BLOCK DIAGRAM

3. DEVICE OPERATION

Connected to a TMS 9900 as shown in Figure 2, the TIM 9904 oscillator operates with a quartz crystal and an LC circuit connected to the tank terminals. For operation of the TMS 9900 microprocessor at 3 MHz, the frequency reference requires a resonant frequency of 48 MHz (16 x 3 MHz). The quartz crystal used as a frequency reference should be designed for series-mode operation with a resistance in the 20- to 75-ohm range, and be capable of a minimum 2-mW power dissipation. Typical frequency tolerance is ±0.005 percent. For 48-MHz operation a third-overtone crystal is used. The inductance L connected across the tank terminals should be 0.47 μH ± 10 percent, and the capacitance C (including board capacity) should be 22 pF ± 5 percent. The LC circuit should be tuned to the third-overtone crystal frequency for best results. The tank circuit should be physically located as close as possible to the TMS devices 9904.

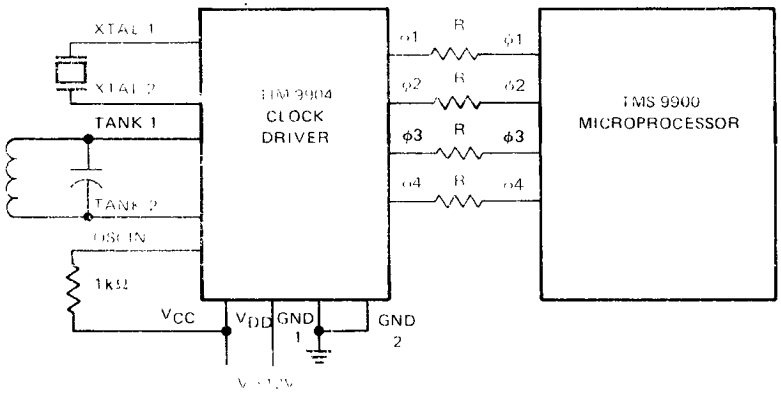


FIGURE 2—TIM 9904, CRYSTAL-CONTROLLED OPERATION

A 0.1-μF capacitor can be substituted for the quartz crystal. With a capacitor rather than a crystal, the LC tuned circuit establishes the operating frequencies. LC component values for operation at any frequency can be computed from $f_{osc} = 1/(2\pi\sqrt{LC})$ where f_{osc} is the oscillator frequency, L is the inductance value in henries, and C is the capacitance value in farads.

When the internal oscillator is used, OSCIN should be connected to V_{CC} through a resistor (1 kΩ nominal), and an LC tank circuit must be connected to the tank inputs except when a fundamental crystal is being used. An external oscillator can be used by connecting it to OSCIN and disabling the internal oscillator by connecting the crystal terminals to V_{CC} and leaving the tank inputs open. The external oscillator must have a frequency four times the desired output clock frequency and a 25 percent duty cycle. The first low-level external clock pulse will preset the divide-by-four counter, allowing the external oscillator signal to directly drive the phase generator. Figure 3 is a timing diagram illustrating operation with an external oscillator.

Resistors between the TIM 9904 ϕ1, ϕ2, ϕ3, and ϕ4 outputs and the corresponding clock input terminals of the TMS 9900 should be in the 10- 20-ohm range (see Figure 2). The purpose of the resistors is to minimize overshoot and undershoot. The required resistance value is dependent on circuit layout; clock signal interconnections should be as short as possible.

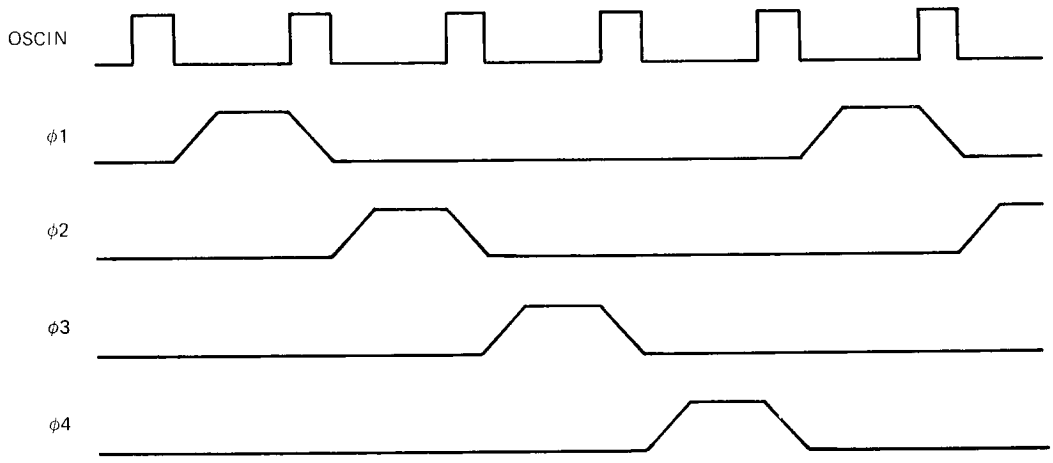


FIGURE 3–EXTERNAL OSCILLATOR TIMING FOR USE WITH TIM 9904

The D-type flip-flop associated with TIM 9904 pins FFD and FFQ can be used to provide a power-on reset and a manual reset to the TMS 9900 as shown in Figure 4. A Schmitt-trigger circuit driving the D input generates a fast-rising waveform when the input voltage rises to a specific value. At power turn-on, voltage across the 0.1 μF capacitor in Figure 4 will rise towards V_{CC}. This circuit provides a delay that resets the TMS 9900 after V_{CC} has stabilized. An optional manual reset switch can be connected to the delay circuit to reset the TMS 9900 at any time. The TMS 9900 HOLD signal could alternately be actuated by FFD.

The ground terminals GND1 and GND2 normally should be connected together and to system ground.

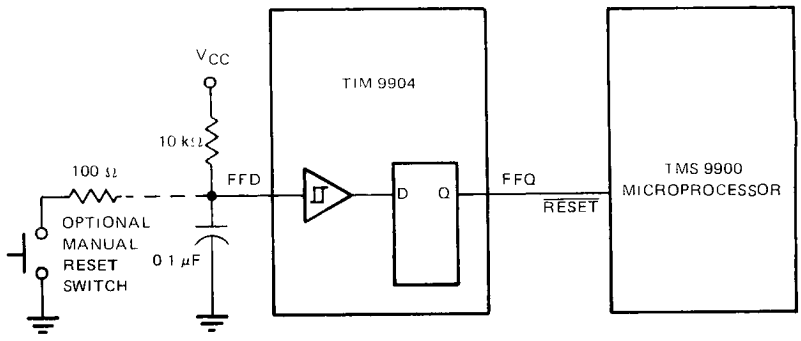


FIGURE 4–POWER-ON RESET

4. DEVICE APPLICATION

4.1 Modes of Operation

The TIM 9904 may be used in one of the following modes to provide clocking for the TMS 9900 or other microprocessor:

- *Overtone operation* — overtone crystal; tank-circuit bandpass filters the selected harmonic
- *Fundamental operation* — fundamental crystal; tank circuit not required
- *Tank-controlled operation* — no crystal; frequency determined as resonant frequency of the tank circuit
- *Externally-controlled operation* — internal oscillator disabled; TTL input signal determines frequency.

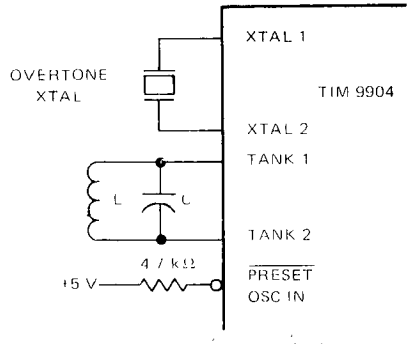
4.1.1 Overtone Operation

Overtone operation is used when crystal-stabilized, high-frequency ($f_{cy} > 1.5$ MHz) clocking is required. The crystal is operated at a harmonic of its fundamental frequency, and the tank-circuit bandpass filters the crystal frequency so as to select the desired harmonic. For example, if 3-MHz operation is required (f_{cy} = 3 MHz), f_{osc} must be 48 MHz. Since fundamental crystals are generally not available with frequencies above approximately 24 MHz, a 48-MHz, third-overtone crystal may be employed. The tank circuit should have a resonant frequency of 48 MHz to bandpass filter the third overtone.

The resonant frequency is determined by the equation:

$$f_{res} = \frac{1}{2\pi\sqrt{LC}}$$

The PRESET/OSCIN input is held at high. Figure 5 typifies the connection of components for overtone operation.



$f_{OSC} = f_{XTAL} = 16\, f_{cy} = \frac{1}{2\pi\sqrt{LC}}$

$f_{OSC} = 4\, f_{OSC\ OUT}$

0.5 MHz < f_{cy} < 3MHz

FIGURE 5–OVERTONE OPERATION

4.1.2 *Fundamental Operation*

If a crystal is available with a fundamental frequency 16 times the required f_{cy}, a tank circuit is not required and the TANK1 and TANK2 inputs are connected to each other through a 100-ohm resistor, as shown in Figure 6. The PRESET/OSCIN input is held at high level.

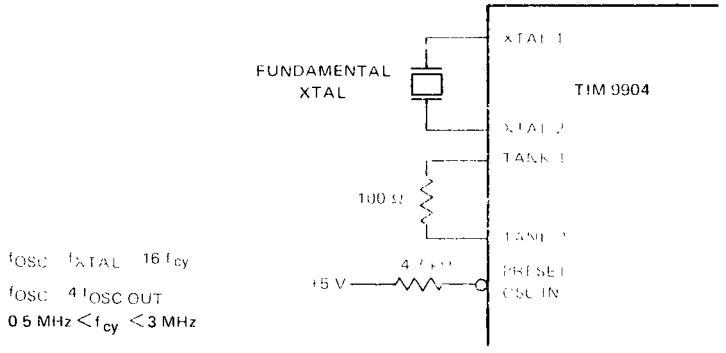


FIGURE 6–FUNDAMENTAL-FREQUENCY CRYSTAL OPERATION

4.1.3 *Tank Controlled Operation*

For applications in which crystal quality stability is not needed, the crystal may be replaced with a 0.1-μF capacitor and $f_{osc} = f_{res} = 1/(2\ \pi \sqrt{LC})$. Slight variations with changing V_{CC} and temperature can be expected. Tank-controlled operation interconnections are shown in Figure 7.

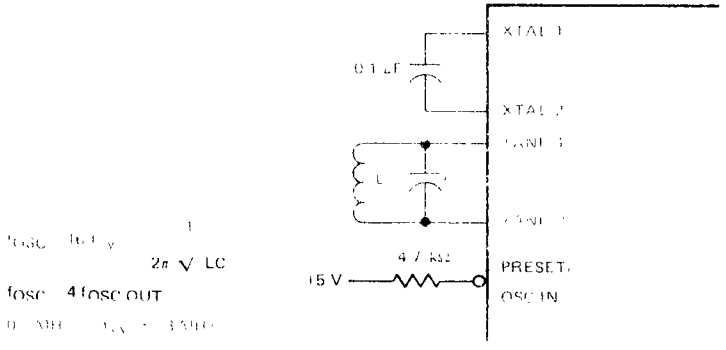
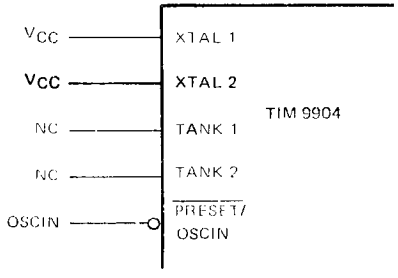


FIGURE 7–TANK-CONTROLLED FREQUENCY OPERATION

4.1.4 Externally Controlled Operation

If a TTL signal is available with the appropriate frequency and waveform, such signal may be connected to the $\overline{\text{PRESET}}$/OSCIN input of the TIM 9904 as shown in Figure 8. The internal oscillator is disabled by leaving tank inputs open and by connecting the crystal inputs to V_{CC}.



$f_{OSCIN} = f_{cy}$
$f_{OSCIN} = f_{OSCOUT}$, OSCOUT = $\overline{OSCIN}$
0.5 MHz $< f_{cy} <$ 3 MHz

FIGURE 8–EXTERNALLY-CONTROLLED OPERATION

4.2 Component Selection

The criterion for selecting the values of the discrete components to be used with the TIM 9904 are recommended in this section.

4.2.1 Crystal

The following crystal specifications are suggested.

- Series resonant, 20- to 75- ohm series resistance, 2-mW maximum power dissipation.
- $f_{XTAL} = 16\, f_{cy}$
- For f_{cy} = 3 MHz, specify 48 MHz, third overtone
- Suggested stability: 0.05 percent from 0° to 70°C.

4.2.2 Tank Circuit

Because the value of the capacitance will be in the picofarad range, board capacity must be considered when selecting component values for the LC tank circuit. The board capacitance (C_B) will be additive to the device capacitance (C_D), as shown in Figure 9. Board capacitance may be computed in the following manner:

(1) Connect devices to the TIM 9904 as shown in Figure 5. To ensure that $f_{cy} \leq 3$ MHz, select values for L and C_D such that

$$\frac{1}{2\pi\sqrt{LC_D}} \leq 48 \text{ MHz}$$

(2) Measure the frequency (f_{CY}) of one of the TTL clock outputs ($\overline{\phi 1}$, $\overline{\phi 2}$, $\overline{\phi 3}$, or $\overline{\phi 4}$).

(3) Since $f_{osc} = 16 f_{cy} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{L(C_B + C_D)}}$

C_B can be determined from the equation:

$$C_B = \frac{1}{L \cdot (32\pi F_{cy})^2} - C_D$$

For example, assume that L = 0.5 μH and C_D = 22 pF:

$$\frac{1}{2\pi\sqrt{LC_D}} = 47.987 \text{ MHz} < 48 \text{ MHz}$$

F_{cy} is determined to be 2.413 MHz. Therefore:

$$C_B = \frac{1}{L \cdot (32\pi F_{cy})^2} - C_D = \frac{1}{0.5\text{x}10^{-6} \cdot (32\pi \cdot 2.413\text{x}10^{6})^2} - 22 \text{ x } 10^{-12} \text{ farad}$$

C_B = 11.98 pF $\approx$ 12 pF.

In order to obtain $f_{res} \approx 48$ MHz using 0.5 μH, $C = C_B + C_D = 22$ pF; thus, C_D must be 10 pF with a board capacitance of 12 pF. When using the tank circuit in overtone operation, the f_{res} should be within 5 percent of f_{osc}, requiring that the product of LC should be within 10 percent of the ideal values for $f_{res} = f_{osc}$. This may be accomplished by using devices with nominal values so that $f_{osc} = 1/(2\pi\sqrt{LC})$, and with 5 percent tolerances.

For the above example with C_B = 12 pF:

L = 0.5 μH ± 5 percent

C = 10 pF ± 5 percent

thus providing a comfortable margin for deviations of component value on a production basis.

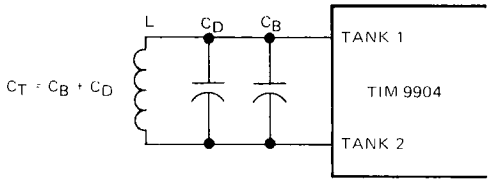


FIGURE 9–EFFECT OF BOARD ON TANK CIRCUIT RESONANT FREQUENCY

4.2.3 *Series Resistors*

Resistors with values on the order of 10 to 22-ohms should be installed between the ϕ1-ϕ4 outputs of the TIM 9904 and the corresponding inputs of TMS 9900. These serve two purposes:

- Reduce overshoot and ringing
- Protect the drivers from overvoltage and undervoltage signals.

Connect the resistors as illustrated in Figure 9.

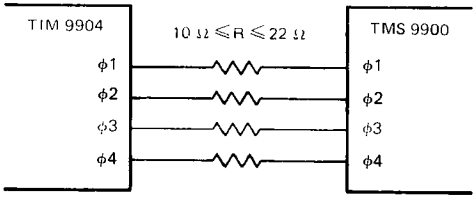
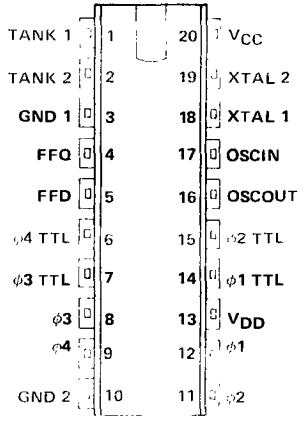


FIGURE 10–SERIES MOS CLOCK RESISTORS

8

4.3 TIM 9904 Terminal Assignments

SIGNATURE	PIN	I/O	DESCRIPTION
TANK 1	1	I	Tank circuit connection
TANK 2	2	I	Tank circuit connection
GND 1	3		Ground reference
FFQ	4	O	Output of D flip-flop
FFD	5		D Input to Schmitt triggered flip-flop
$\overline{\phi 4}$ TTL	6	O	TTL Phase 4 inverted
$\overline{\phi 3}$ TTL	7	O	TTL Phase 3 inverted
$\phi 3$	8	O	MOS Phase 3
$\phi 4$	9	O	MOS Phase 4
GND 2	10		Ground reference
$\phi 2$	11	O	MOS Phase 2
$\phi 1$	12	O	MOS Phase 1
V_{DD}	13	I	Supply voltage 12 V nominal
$\overline{\phi 1}$ TTL	14	O	TTL Phase 1 inverted
$\overline{\phi 2}$ TTL	15	O	TTL Phase 2 inverted
OSCOUT	16	O	Oscillator output
OSCIN	17	I	TTL external oscillator input
XTAL1	18	I	Crystal
XTAL2	19	I	Crystal
V_{CC}	20	I	Supply voltage 5 V nominal



5. ELECTRICAL SPECIFICATIONS

5.1 Absolute Maximum Ratings Over Operating Free-Air Temperature Range (Unless Otherwise Noted)

Supply voltage: V_{CC} (see Note 1) ..... 7 V
V_{DD} (see Note 1) ..... 13 V
Input voltage: OSCIN ..... 5.5 V
FFD ..... −0.5 V to 7 V
Operating free-air temperature range ..... 0°C to 70°C
Storage temperature range ..... −65°C to 150°C

NOTE 1 Voltage values are with respect to the network ground terminals connected together

5.2 Recommended Operating Conditions

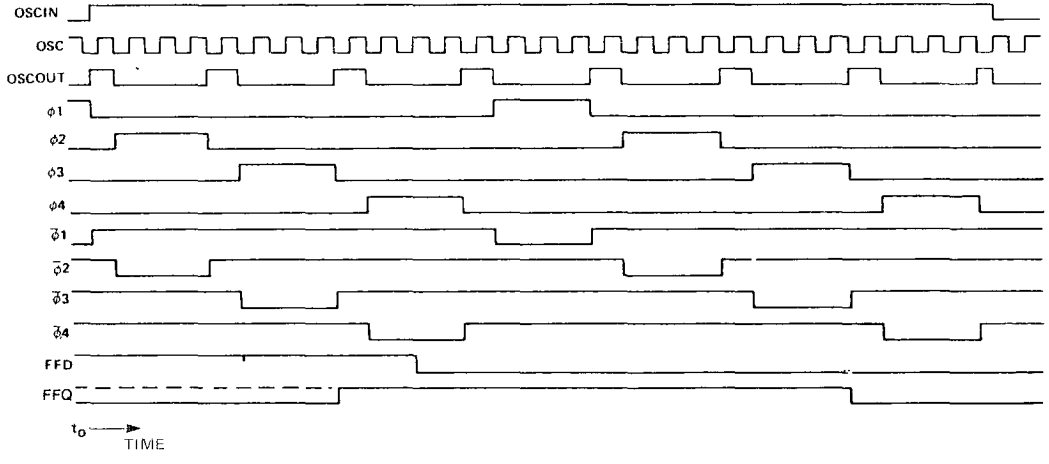
		MIN	NOM	MAX	UNIT
Supply voltages	V_{CC}	4.75	5	5.25	V
	V_{DD}	11.4	12	12.6	V
High-level output current, I_{OH}	$\phi1, \phi2, \phi3, \phi4$			−100	μA
	All others			−400	μA
Low-level output current, I_{OL}	$\phi1, \phi2, \phi3, \phi4$			4	mA
	All others			8	mA
Internal oscillator frequency, f_{osc}			48	54	MHz
External oscillator pulse width, $t_{w(osc)}$		25			ns
Setup time, FFD input (with respect to falling edge of $\phi3$), t_{su}		50			ns
Hold time, FFD input (with respect to falling edge of $\phi3$), t_h		−30			ns
Operating free-air temperature, T_A		0		70	°C

5.3 Electrical Characteristics Over Recommended Operating Free-Air Temperature Range (Unless Otherwise Noted)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{IH}	High-level input voltage				2			V
V_{IL}	Low-level input voltage	FFD					0.5	V
		OSCIN					0.8	
$V_{T+} - V_{T-}$	Hysteresis	FFD			0.4	0.8		V
V_{IK}	Input clamp voltage		V_{CC} = 4.75 V, V_{DD} = 11.4 V, I_I = −18 mA				−1.5	V
V_{OH}	High-level output voltage	$\phi1, \phi2, \phi3, \phi4$	V_{CC} = 4.75 V, V_{DD} = 11.4 V to 12.6 V	I_{OH} = −100 μA	V_{DD}−2	V_{DD}−1.5	V_{DD}	V
		Other outputs		I_{OH} = −400 μA	2.7	3.4		
V_{OL}	Low-level output voltage	$\phi1, \phi2, \phi3, \phi4$	V_{CC} = 4.75 V, V_{DD} = 11.4 V	I_{OL} = 4 mA		0.25	0.4	mA
		Other outputs		I_{OL} = 4 mA		0.25	0.4	
				I_{OL} = 8 mA		0.35	0.5	
I_I	Input current at maximum input voltage	FFD	V_{CC} = 5.25 V, V_{DD} = 12.6 V	V_I = 7 V			0.1	mA
		OSCIN		V_I = 5.5 V			0.3	
I_{IH}	High-level input current	FFD	V_{CC} = 5.25 V, V_{DD} = 12.6 V, V_I = 2.7 V				20	μA
		OSCIN					60	
I_{IL}	Low-level input current	FFD	V_{CC} = 5.25 V, V_{DD} = 12.6 V, V_I = 0.4 V				−0.4	mA
		OSCIN					−3.2	
I_{OS}	Short-circuit output current‡	All except $\phi1, \phi2, \phi3, \phi4$	V_{CC} = 5.25 V		−20		−100	mA
I_{CC}	Supply current from V_{CC}		V_{CC} = 5.25 V, FFD and OSCIN at GND, Outputs open			105	175	mA
I_{DD}	Supply current from V_{DD}		V_{CC} = 5.25 V, V_{DD} = 12.6 V, FFD and DSCIN at GND, Outputs open			12	20	mA

†All typical values are at V_{CC} = 5 V, V_{DD} = 12 V, T_A = 25°C.

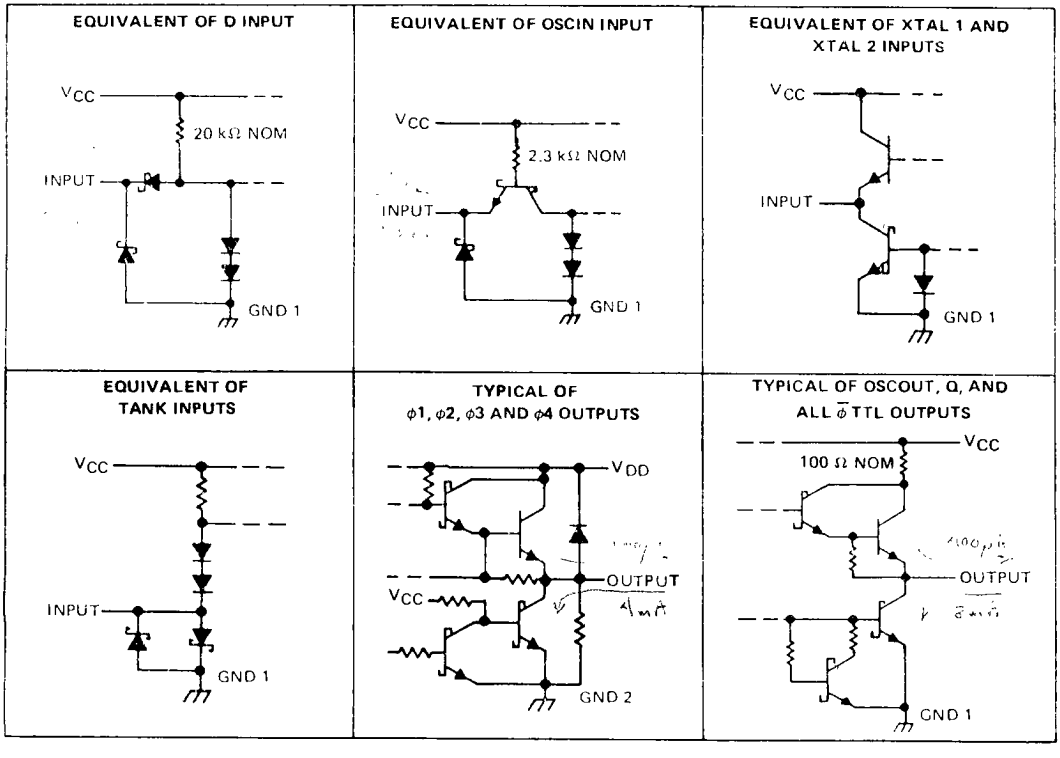
‡Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second. Outputs $\phi1$, $\phi2$, $\phi3$, and $\phi4$ do not have short circuit protection.



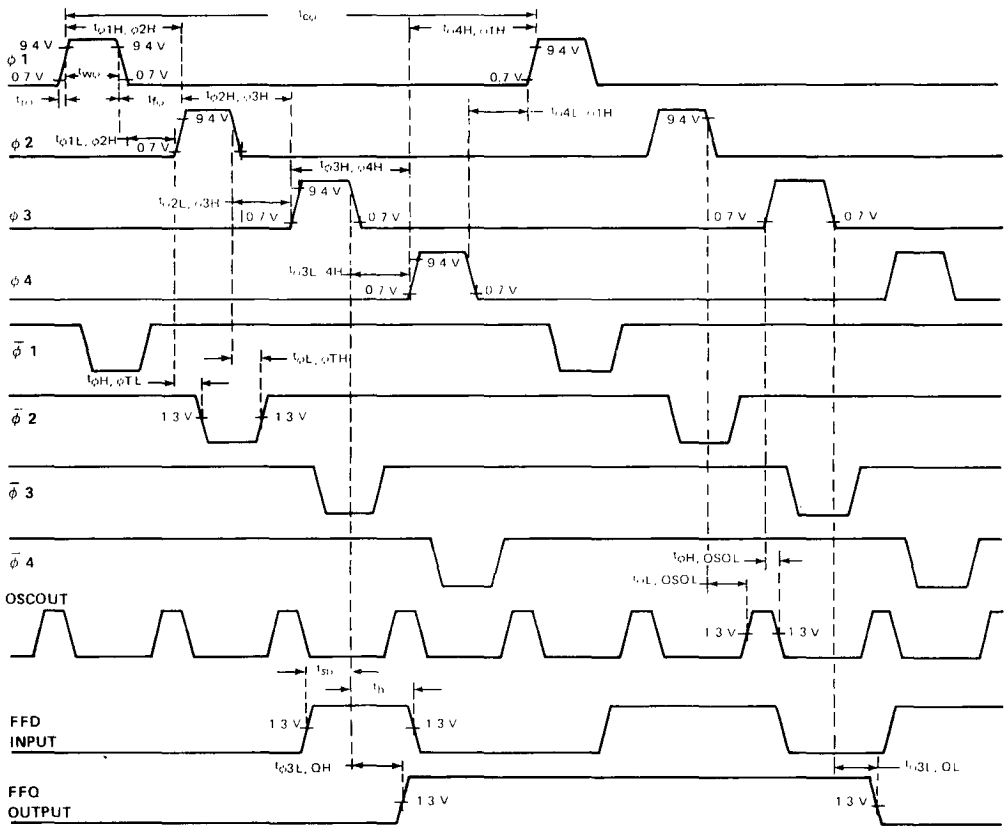
TYPICAL PHASE RELATIONSHIPS OF INPUTS AND OUTPUTS (INTERNAL OSC)

5.4 Switching Characteristics, $T_A = 25°C$, $V_{CC1} = 5\ V$, $V_{CC2} = 12\ V$, $f_{osc} = 48\ MHz$

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{out}	Output frequency, any ϕ or $\bar{\phi}$ TTL			3		MHz
f_{out}	Output frequency, OSCOUT			12		MHz
$t_{c(\phi)}$	Cycle time, any ϕ output		330	333	340	ns
$t_{r(\phi)}$	Rise time, any ϕ output		−5		20	ns
$t_{f(\phi)}$	Fall time, any ϕ output		10	14	20	ns
$t_{w(\phi)}$	Pulse width, any ϕ output high		40	55	70	ns
$t_{\phi 1L, \phi 2H}$	Delay time, $\phi 1$ low to $\phi 2$ high		0	5	15	ns
$t_{\phi 2L, \phi 3H}$	Delay time, $\phi 2$ low to $\phi 3$ high		0	5	15	ns
$t_{\phi 3L, \phi 4H}$	Delay time, $\phi 3$ low to $\phi 4$ high		0	5	15	ns
$t_{\phi 4L, \phi 1H}$	Delay time, $\phi 4$ low to $\phi 1$ high	Output loads.	0	5	15	ns
$t_{\phi 1H, \phi 2H}$	Delay time, $\phi 1$ high to $\phi 2$ high	$\phi 1, \phi 3, \phi 4$ 100 pF to GND	73	83	96	ns
$t_{\phi 2H, \phi 3H}$	Delay time, $\phi 2$ high to $\phi 3$ high	$\phi 2$: 200 pF to GND	73	83	96	ns
$t_{\phi 3H, \phi 4H}$	Delay time, $\phi 3$ high to $\phi 4$ high	Others: $R_L = 2\ k\Omega$,	73	83	96	ns
$t_{\phi 4H, \phi 1H}$	Delay time, $\phi 4$ high to $\phi 1$ high	$C_L = 15\ pF$	73	83	96	ns
$t_{\phi H, \phi TL}$	Delay time, ϕ_n high to $\bar{\phi}_n$ TTL low		−14	−4	6	ns
$t_{\phi L, \phi TH}$	Delay time, ϕ_n low to $\bar{\phi}_n$ TTL high		−29	−19	−9	ns
$t_{\phi 3L, QH}$	Delay time, $\phi 3$ low to FFQ output high		−18	−8	2	ns
$t_{\phi 3L, QL}$	Delay time, $\phi 3$ low to FFQ output low		−19	−9	1	ns
$t_{\phi L, OSOH}$	Delay time, ϕ low to OSCOUT high		−30	−20	−10	ns
$t_{\phi H, OSOL}$	Delay time, FFQ high to OSCOUT low		−27	−17	−7	ns



SCHEMATICS OF INPUTS AND OUTPUTS



SWITCHING CHARACTERISTICS, VOLTAGE WAVEFORMS

8

- Three-State Versions of '151, 'LS151, 'S151
- Three-State Outputs Interface Directly with System Bus
- Perform Parallel-to-Serial Conversion
- Permit Multiplexing from N-lines to One Line
- Complementary Outputs Provide True and Inverted Data
- Fully Compatible with Most TTL and DTL Circuits

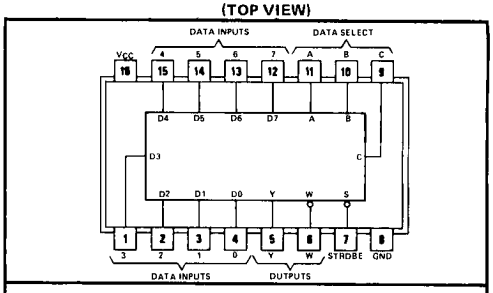
TYPE	MAX NO. OF COMMON OUTPUTS	TYPICAL AVG PROP DELAY TIME (D TO Y)	TYPICAL POWER DISSIPATION
SN54251	49	17 ns	250 mW
SN74251	129	17 ns	250 mW
SN54LS251	49	17 ns	35 mW
SN74LS251	**129**	**17 ns**	**35 mW**
SN54S251	39	8 ns	275 mW
SN74S251	129	8 ns	275 mW

description

These monolithic data selectors/multiplexers contain full on-chip binary decoding to select one-of-eight data sources and feature a strobe-controlled three-state output. The strobe must be at a low logic level to enable these devices. The three-state outputs permit a number of outputs to be connected to a common bus. When the strobe input is high, both outputs are in a high-impedance state in which both the upper and lower transistors of each totem-pole output are off, and the output neither drives nor loads the bus significantly. When the strobe is low, the outputs are activated and operate as standard TTL totem-pole outputs.

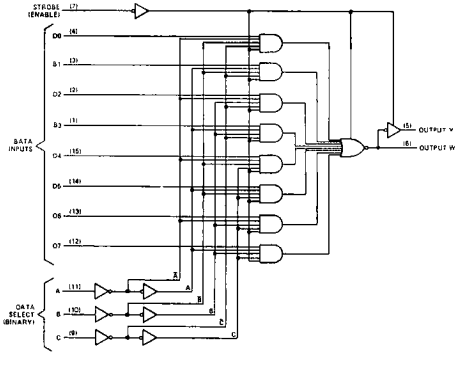
To minimize the possibility that two outputs will attempt to take a common bus to opposite logic levels, the output control circuitry is designed so that the average output disable time is shorter than the average output enable time. The SN54251 and SN74251 have output clamp diodes to attenuate reflections on the bus line.

SN54251, SN54LS251, SN54S251 . . . J OR W PACKAGE
SN74251, SN74LS251, SN54S251 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

functional block diagram



FUNCTION TABLE

INPUTS				OUTPUTS	
SELECT			STROBE		
C	B	A	S	Y	W
X	X	X	H	Z	Z
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = high logic level, L = low logic level
X = irrelevant, Z = high impedance (off)
D0, D1 . . . D7 = the level of the respective D input

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 7 V
Off-state output voltage . . . . . 5.5 V
Operating free-air temperature range: SN54LS251 . . . . . −55°C to 125°C
SN74LS251 . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54LS251			SN74LS251			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−1			−2.6	mA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		SN54LS251			SN74LS251			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage			2			2			V
V_{IL}	Low-level input voltage					0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN,	I_I = −18 mA			−1.5			−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = MAX,	V_{IH} = 2 V, I_{OH} = MAX	2.4	3.4		2.4	3.1		V
V_{OL}	Low-level voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = VIL max	I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
			I_{OL} = 8 mA					0.35	0.5	
I_{OZ}	Off-state (high-impedance-state) output current	V_{CC} = MAX, V_{IH} = 2 V	V_O = 2.7 V			20			20	μA
			V_O = 0.4 V			−20			−20	
I_I	Input current at maximum input voltage	V_{CC} = MAX,	V_I = 7 V			0.1			0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX,	V_I = 2.7 V			20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX,	V_I = 0.4 V			−0.4			−0.4	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX		−30		−130	−30		−130	mA
I_{CC}	Supply current	V_{CC} = MAX, See Note 3	Condition A		6.1	10		6.1	10	mA
			Condition B		7.1	12		7.1	12	

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.
NOTE 3: I_{CC} is measured with the outputs open and all data and select inputs at 4.5 V under the following conditions:
A. Strobe grounded.
B. Strobe at 4.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

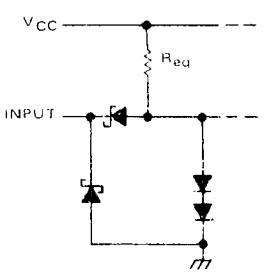
PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A, B, or C (4 levels)	Y	C_L = 15 pF, R_L = 2 kΩ, See Note 4		29	45	ns
t_{PHL}					28	45	
t_{PLH}	A, B, or C (3 levels)	W			20	33	ns
t_{PHL}					21	33	
t_{PLH}	Any D	Y			17	28	ns
t_{PHL}					18	28	
t_{PLH}	Any D	W			10	15	ns
t_{PHL}					9	15	
t_{ZH}	Strobe	Y			30	45	ns
t_{ZL}					26	40	
t_{ZH}	Strobe	W			17	27	ns
t_{ZL}					24	40	
t_{HZ}	Strobe	Y	C_L = 5 pF, R_L = 2 kΩ, See Note 4		30	45	ns
t_{LZ}					15	25	
t_{HZ}	Strobe	W			37	55	ns
t_{LZ}					15	25	

¶ t_{PLH} ≡ Propagation delay time, low-to-high-level output
t_{PHL} ≡ Propagation delay time, high-to-low-level output
t_{ZH} ≡ Output enable time to high level
t_{ZL} ≡ Output enable time to low level
t_{HZ} ≡ Output disable time from high level
t_{LZ} ≡ Output disable time from low level

NOTE 4: See load circuits and waveforms on page 3-11.

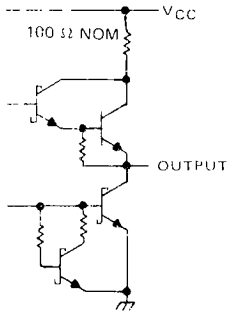
schematics of inputs and outputs

EQUIVALENT OF EACH INPUT



A, B, C, S R_{eq} = 20 kΩ NOM
D0 thru D7 R_{eq} = 17 kΩ NOM

TYPICAL OF BOTH OUTPUTS

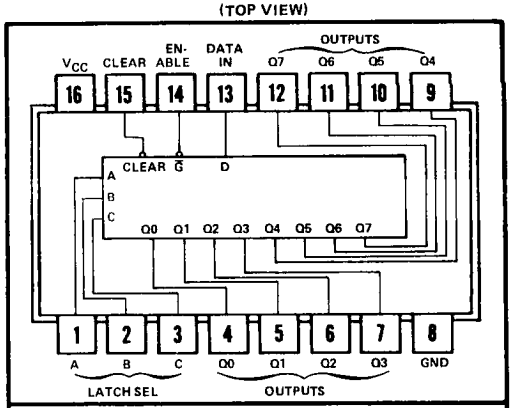


- 8-Bit Parallel-Out Storage Register Performs Serial-to-Parallel Conversion With Storage
- Asynchronous Parallel Clear
- Active High Decoder
- Enable/Disable Input Simplifies Expansion
- Direct Replacement for Fairchild 9334
- Expandable for N-Bit Applications
- Four Distinct Functional Modes
- Typical Propagation Delay Times:

	'259	'LS259
Enable-to-Output . . .	12	17
Data-to-Output	12	18
Address-to-Output . .	16	20
Clear-to-Output	16	20

- Fan-Out
  - I_{OL} (Sink Current)
    - '259 . . . . . 16 mA
    - SN54LS259 . . . . . 4 mA
    - SN74LS259 . . . . . 8 mA
  - I_{OH} (Source Current)
    - '259 . . . . . −0.8 mA
    - 'LS259 . . . . . −0.4 mA
- Typical I_{CC}
  - '259 . . . . . 60 mA
  - 'LS259 . . . . . 22 mA

SN54259, SN54LS259 . . . J OR W PACKAGE
SN74259, SN74LS259 . . . J OR N PACKAGE
(TOP VIEW)



logic: see function table

description

These 8-bit addressable latches are designed for general purpose storage applications in digital systems. Specific uses include working registers, serial-holding registers, and active-high decoders or demultiplexers. They are multifunctional devices capable of storing single-line data in eight addressable latches, and being a 1-of-8 decoder or demultiplexer with active-high outputs.

Four distinct modes of operation are selectable by controlling the clear and enable inputs as enumerated in the function table. In the addressable-latch mode, data at the data-in terminal is written into the addressed latch. The addressed latch will follow the data input with all unaddressed latches remaining in their previous states. In the memory mode, all latches remain in their previous states and are unaffected by the data or address inputs. To eliminate the possibility of entering erroneous data in the latches, the enable should be held high (inactive) while the address lines are changing. In the 1-of-8 decoding or demultiplexing mode, the addressed output will follow the level of the D input with all other outputs low. In the clear mode, all outputs are low and unaffected by the address and data inputs.

FUNCTION TABLE

INPUTS CLEAR	INPUTS $\overline{G}$	OUTPUT OF ADDRESSED LATCH	EACH OTHER OUTPUT	FUNCTION
H	L	D	Q_{i0}	Addressable Latch
H	H	Q_{i0}	Q_{i0}	Memory
L	L	D	L	8-Line Demultiplexer
L	H	L	L	Clear

LATCH SELECTION TABLE

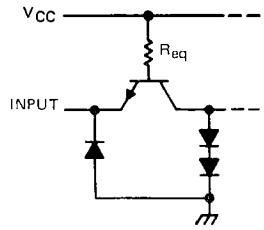
SELECT INPUTS C	B	A	LATCH ADDRESSED
L	L	L	0
L	L	H	1
L	H	L	2
L	H	H	3
H	L	L	4
H	L	H	5
H	H	L	6
H	H	H	7

H = high level, L = low level
D = the level at the data input
Q_{i0} = the level of Q_i (i = 0, 1, . . . 7, as appropriate) before the indicated steady-state input conditions were established.

schematic of inputs and outputs

'259

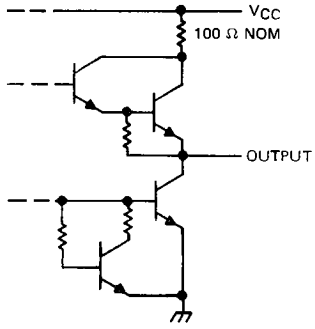
EQUIVALENT OF EACH INPUT



Latch select, data in, or clear. R_{eq} = 4 kΩ NOM
Enable: R_{eq} = 2.2 kΩ NOM

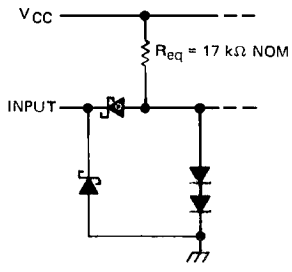
'259

TYPICAL OF ALL OUTPUTS



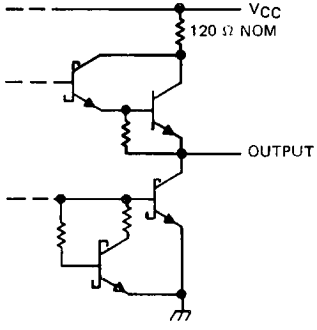
'LS259

EQUIVALENT OF EACH INPUT



'LS259

TYPICAL OF ALL OUTPUTS



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage (see Note 1)		7 V
Input voltage: SN54259, SN74259		5.5 V
SN54LS259, SN74LS259		7 V
Operating free-air temperature range:	SN54259, SN54LS259	−55°C to 125°C
	SN74259, SN74LS259	0°C to 70°C
Storage temperature range		−65°C to 150°C

NOTE 1. Voltage values are with respect to network ground terminal.

8

recommended operating conditions

		SN54LS259			SN74LS259			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				−400			−400	μA
Low-level output current, I_{OL}				4			8	mA
Width of clear or enable pulse, t_w		15			15			ns
Setup time, t_{su}	Data	15↑			15↑			ns
	Address	15↑			15↑			
Hold time, t_h	Data	0↑			0↑			ns
	Address	0↑			0↑			
Operating free-air temperature, T_A		−55		125	0		70	°C

↑The arrow indicates that the rising edge of the enable pulse is used for reference.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†			SN54LS259			SN74LS259			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage				2			2			V
V_{IL}	Low-level input voltage						0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN,	I_I = −18 mA				−1.5			−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max,	V_{IH} = 2 V, I_{OH} = −0.4 mA		2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max,	V_{IH} = 2 V,	I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
				I_{OL} = 8 mA					0.35	0.5	
I_I	Input current at maximum input voltage	V_{CC} = MAX,	V_I = 7 V				0.1			0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX,	V_I = 2.7 V				20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX,	V_I = 0.4 V				−0.4			−0.4	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX			−20		−100	−20		−100	mA
I_{CC}	Supply current	V_{CC} = MAX,	See Note 2			22	36		22		mA

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time, and duration short-circuit should not exceed one second.
NOTE 2: I_{CC} is measured with the inputs grounded and the outputs open.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PHL}	Clear	Any Q	C_L = 15 pF, R_L = 2 kΩ, See Note 3		17	27	ns
t_{PLH}	Data	Any Q			20	32	ns
t_{PHL}					13	21	
t_{PLH}	Address	Any Q			24	38	ns
t_{PHL}					18	29	
t_{PLH}	Enable	Any Q			22	35	ns
t_{PHL}					15	24	

t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 3: Load circuit is shown on page 3-11.

'147, 'LS147

- Encodes 10-Line Decimal to 4-Line BCD
- Applications Include:

  Keyboard Encoding
  Range Selection

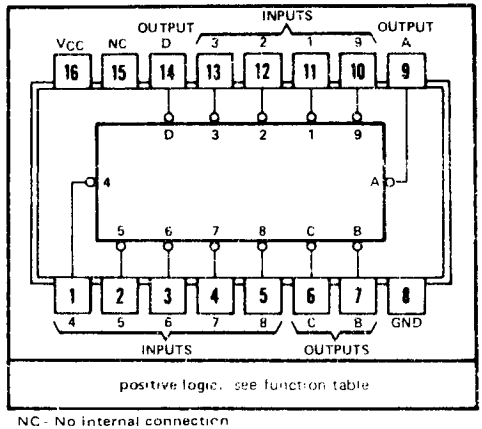
'148, 'LS148

- Encodes 8 Data Lines to 3-Line Binary (Octal)
- Applications Include:

  N-Bit Encoding
  Code Converters and Generators

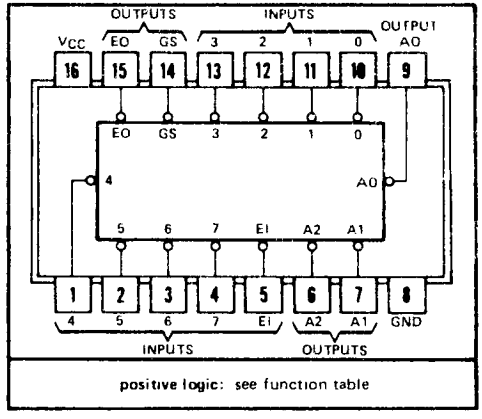
TYPE	TYPICAL DATA DELAY	TYPICAL POWER DISSIPATION
'147	10 ns	225 mW
'148	10 ns	190 mW
'LS147	15 ns	60 mW
'LS148	15 ns	60 mW

SN54147, SN54LS147 . . . J OR W PACKAGE
SN74147, SN74LS147 . . . J OR N PACKAGE
(TOP VIEW)



NC - No internal connection

SN54148, SN54LS148 . . . J OR W PACKAGE
SN74148, SN74LS148 . . . J OR N PACKAGE
(TOP VIEW)



description

These TTL encoders feature priority decoding of the inputs to ensure that only the highest-order data line is encoded. The '147 and 'LS147 encode nine data lines to four-line (8-4-2-1) BCD. The implied decimal zero condition requires no input condition as zero is encoded when all nine data lines are at a high logic level. The '148 and 'LS148 encode eight data lines to three-line (4-2-1) binary (octal). Cascading circuitry (enable input EI and enable output EO) has been provided to allow octal expansion without the need for external circuitry. For all types, data inputs and outputs are active at the low logic level. All inputs are buffered to represent one normalized Series 54/74 or 54LS/74LS load, respectively.

'147, 'LS147
FUNCTION TABLE

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H = high logic level, L = low logic level, X = irrelevant

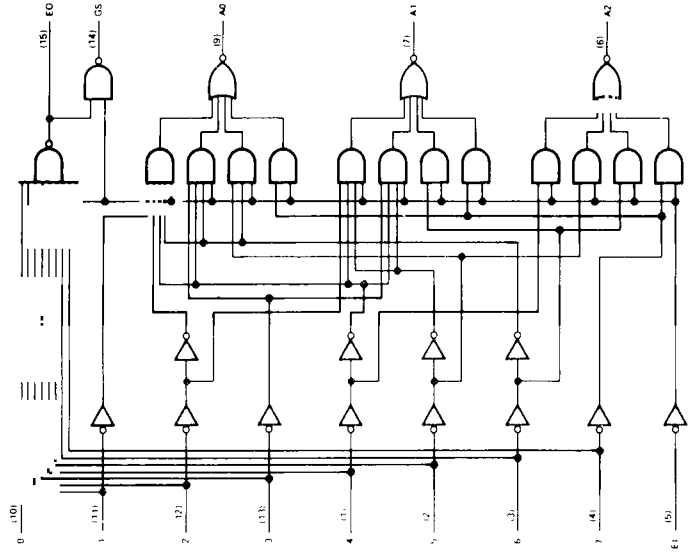
'148, 'LS148
FUNCTION TABLE

INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

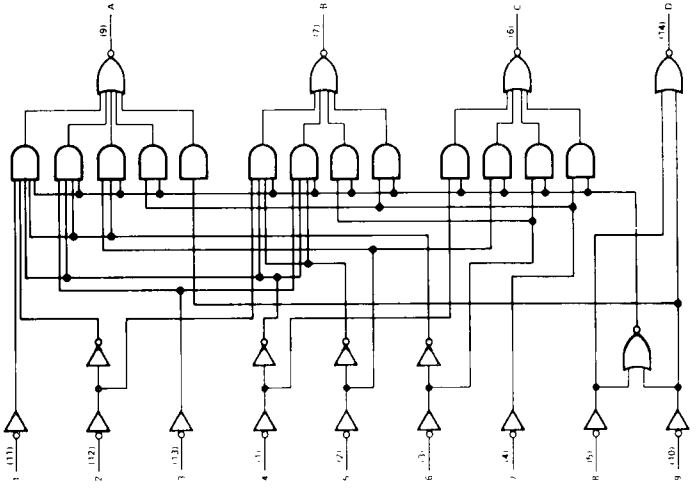
8

functional block diagrams

'148, 'LS148



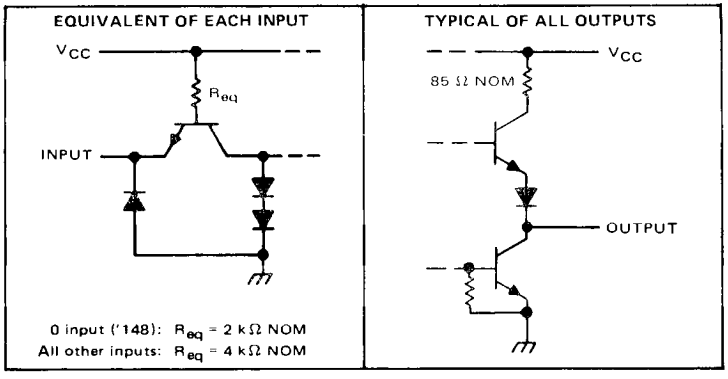
'147, 'LS147



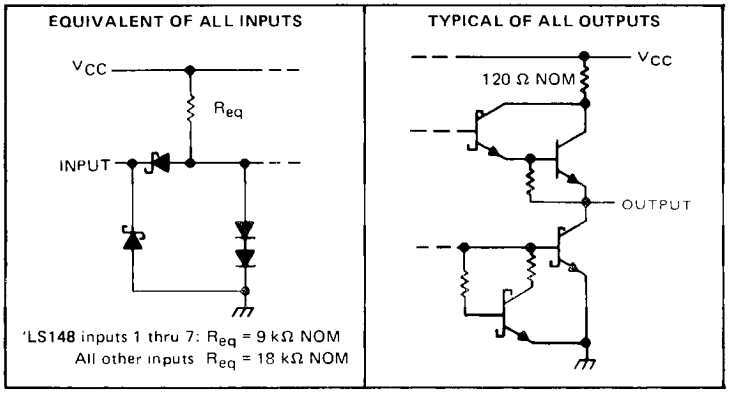
8

schematics of inputs and outputs

'147, '148



'LS147, 'LS148



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage: '147, '148 . . . . . 5.5 V
'LS147, 'LS148 . . . . . 7 V
Interemitter voltage: '148 only (see Note 2) . . . . . **5.5 V**
Operating free-air temperature range: SN54', SN54LS Circuits . . . . . −55°C to 125°C
SN74', SN74LS Circuits . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTES: 1. Voltage values, except interemitter voltage, are with respect to network ground terminal.
2. This is the voltage between two emitters of a multiple-emitter transistor. For '148 circuits, this rating applies between any two of the eight data lines, 0 through 7

8◀

recommended operating conditions

	SN54'			SN74'			SN54LS'			SN74LS'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−800			−800			−400			−400	μA
Low-level output current, I_{OL}			16			16			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS†		'147 MIN	'147 TYP‡	'147 MAX	'148 MIN	'148 TYP‡	'148 MAX	UNIT
V_{IH}	High-level input voltage				2			2			V
V_{IL}	Low-level input voltage						0.8			0.8	V
V_{IK}	Input clamp voltage		V_{CC} = MIN, I_I = −12 mA				−1.5			−1.5	V
V_{OH}	High-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = −800 μA		2.4	3.3		2.4	3.3		V
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 16 mA			0.2	0.4		0.2	0.4	V
I_I	Input current at maximum input voltage		V_{CC} = MAX, V_I = 5.5 V				1			1	mA
I_{IH}	High-level input current	0 input	V_{CC} = MAX, V_I = 2.4 V							40	μA
		Any input except 0					40			80	
I_{IL}	Low-level input current	0 input	V_{CC} = MAX, V_I = 0.4 V							−1.6	mA
		Any input except 0					−1.6			−3.2	
I_{OS}	Short-circuit output current§		V_{CC} = MAX		−35		−85	−35		−85	mA
I_{CC}	Supply current		V_{CC} = MAX, See Note 3	Condition 1		50	70		40	60	mA
				Condition 2		42	62		35	55	mA

NOTE 3: For '147, I_{CC} (condition 1) is measured with input 7 grounded, other inputs and outputs open; I_{CC} (condition 2) is measured with all inputs and outputs open. For '148, I_{CC} (condition 1) is measured with inputs 7 and EI grounded, other inputs and outputs open; I_{CC} (condition 2) is measured with all inputs and outputs open.

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡All typical values are at V_{CC} = 5 V, T_A = 25°C.

§Not more than one output should be shorted at a time.

SN54147, SN74147 switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	WAVEFORM	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any	Any	In-phase output	C_L = 15 pF, R_L = 400 Ω, See Note 4		9	14	ns
t_{PHL}						7	11	
t_{PLH}	Any	Any	Out-of-phase output			13	19	ns
t_{PHL}						12	19	

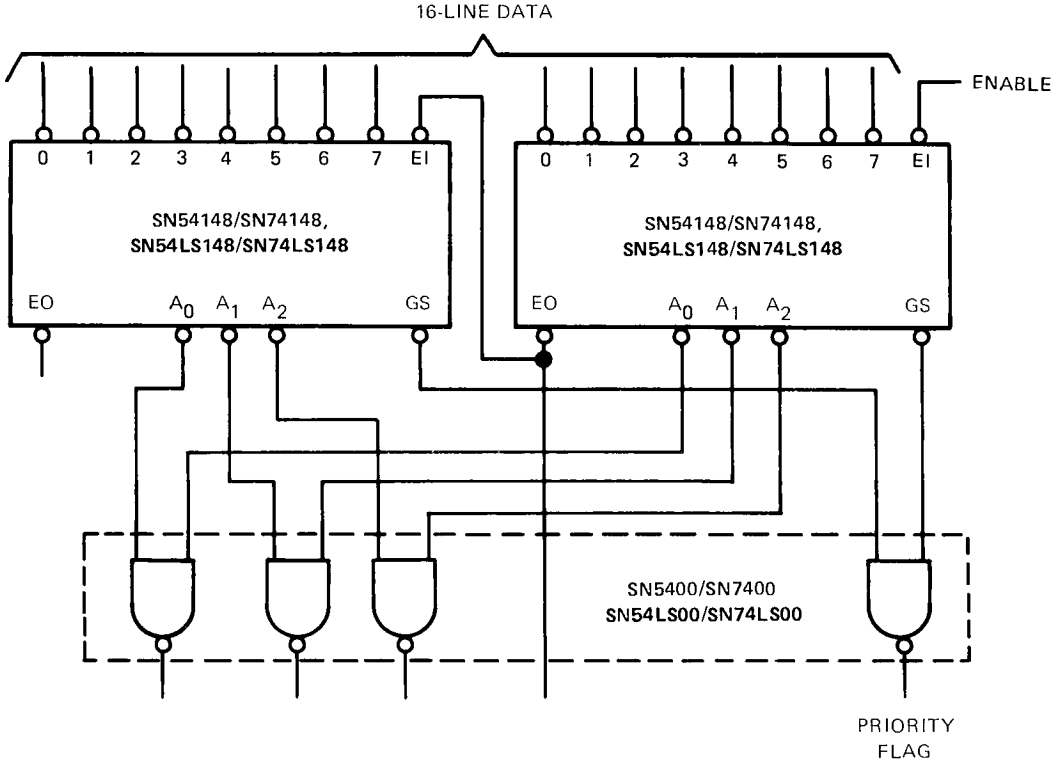
SN54148, SN74148 switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	WAVEFORM	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	0 thru 7	A0, A1, or A2	In-phase output	C_L = 15 pF, R_L = 400 Ω, See Note 4		10	15	ns
t_{PHL}						9	14	
t_{PLH}	0 thru 7	A0, A1, or A2	Out-of-phase output			13	19	ns
t_{PHL}						12	19	
t_{PLH}	0 thru 7	EO	Out-of-phase output			6	10	ns
t_{PHL}						14	25	
t_{PLH}	0 thru 7	GS	In-phase output			18	30	ns
t_{PHL}						14	25	
t_{PLH}	EI	A0, A1, or A2	In-phase output			10	15	ns
t_{PHL}						10	15	
t_{PLH}	EI	GS	In-phase output			8	12	ns
t_{PHL}						10	15	
t_{PLH}	EI	EO	In-phase output			10	15	ns
t_{PHL}						17	30	

¶t_{PLH} = propagation delay time, low to high level output
t_{PHL} = propagation delay time, high to low level output

NOTE 4: Load circuits and waveforms are shown on page 3-10.

TYPICAL APPLICATION DATA

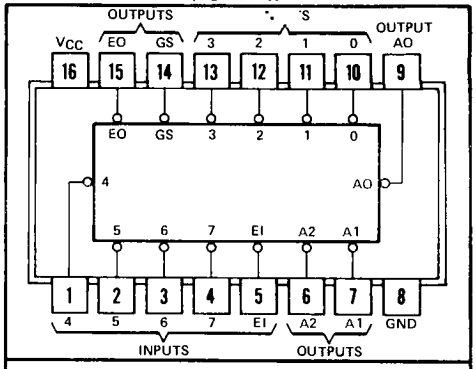


Full 4-bit binary 16-line-to-4-line encoding can be implemented as shown above. The enable input must be low to enable the function. Decoding with 2-input NAND gates produces true (active-high) data for the 4-line binary outputs. If active-low data is required, the SN5408/SN7408 or SN54LS08/SN74LS08 AND gate may be used, respectively.

8

- **3-State Outputs Drive Bus Lines Directly**
- **Encodes 8 Data Lines to 3-Line Binary (Octal)**
- **Applications Include:**
  N-Bit Encoding
  Code Converters and Generators
- **Typical Data Delay . . . 15 ns**
- **Typical Power Dissipation . . . 60 mW**

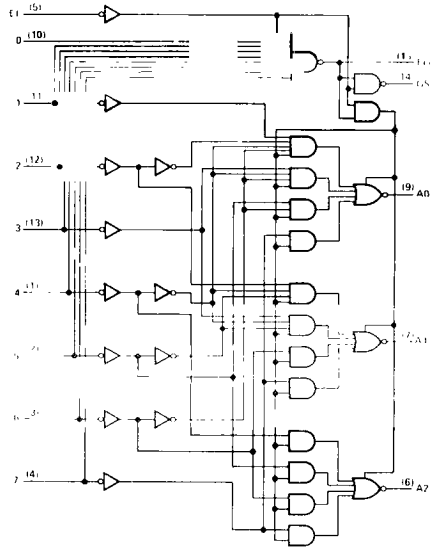
SN54LS348 . . . J OR W PACKAGE
SN74LS348 . . . J OR N PACKAGE
(TOP VIEW)



description

These TTL encoders feature priority decoding of the inputs to ensure that only the highest-order data line is encoded. The 'LS348 circuits encode eight data lines to three-line (4-2-1) binary (octal). Cascading circuitry (enable input EI and enable output EO) has been provided to allow octal expansion. Outputs A0, A1, and A2 are implemented in three-state logic for easy expansion up to 64 lines without the need for external circuitry. See Typical Application Data.

functional block diagram

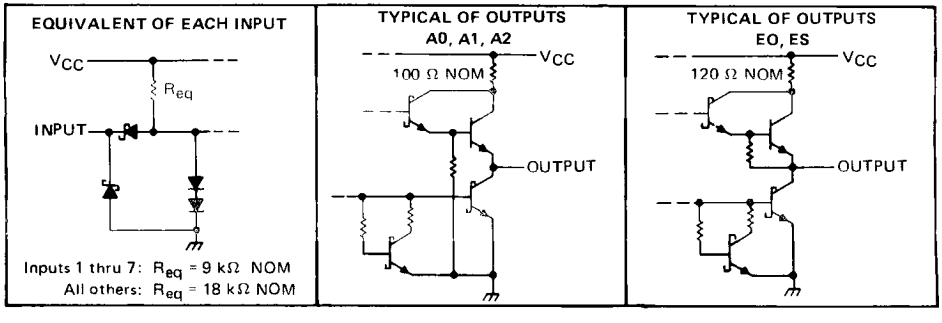


FUNCTION TABLE

INPUTS								OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	Z	Z	Z	H	H
L	H	H	H	H	H	H	H	H	Z	Z	Z	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

H = high logic level, L = low logic level, X = irrelevant
Z = high-impedance state

schematic of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 7 V
Operating free-air temperature range: SN54LS348 . . . . . −55°C to 125°C
SN74LS348 . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54LS348			SN74LS348			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}	A0, A1, A2			−1			−2.6	mA
	EO, GS			−400			−400	µA
Low-level output current, I_{OL}	A0, A1, A2			12			24	mA
	EO, GS			4			8	mA
Operating free-air temperature, T_A		−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS†		SN54LS348			SN74LS348			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage				2			2			V
V_{IL}	Low-level input voltage						0.7			0.8	V
V_{IK}	Input clamp voltage		V_{CC} = MIN,	I_I = −18 mA			−1.5			−1.5	V
V_{OH}	High-level output voltage	A0, A1, A2	V_{CC} = MIN, V_{IH} = 2 V,	I_{OH} = −1 mA	2.4	3.1					V
				I_{OH} = −2.6 mA				2.4	3.1		
		EO, GS	V_{IL} = V_{IL}max	I_{OH} = −400 µA	2.5	3.4		2.7	3.4		
V_{OL}	Low-level Output voltage	A0, A1, A2	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max	I_{OL} = 12 mA		0.25	0.4		0.25	0.4	V
				I_{OL} = 24 mA					0.35	0.5	
		EO, GS		I_{OL} = 4 mA		0.25	0.4		0.25	0.4	
				I_{OL} = 8 mA					0.35	0.5	
I_{OZ}	Off-State (high-impedance state) output current	A0, A1, A2	V_{CC} = MAX, V_{IH} = 2 V	V_O = 2.7 V			20			20	µA
				V_O = 0.4 V			−20			−20	
I_I	Input current at maximum input voltage	Inputs 1 thru 7	V_{CC} = MAX,	V_I = 7 V			0.2			0.2	mA
		All other inputs					0.1			0.1	
I_{IH}	High-level input current	Inputs 1 thru 7	V_{CC} = MAX,	V_I = 2.7 V			40			40	µA
		All other inputs					20			20	
I_{IL}	Low-level input current	Inputs 1 thru 7	V_{CC} = MAX,	V_I = 0.4 V			−0.8			−0.8	mA
		All other inputs					−0.4			−0.4	
I_{OS}	Short-circuit output current§	Outputs A0, A1, A2	V_{CC} = MAX		−30		−130	−30		−130	mA
		Outputs EO, GS			−20		−100	−20		−100	
I_{CC}	Supply current		V_{CC} = MAX, See Note 2	Condition 1		13	25		13	25	mA
				Condition 2		12	23		12	23	

NOTE 2: I_{CC} (condition 1) is measured with inputs 7 and EI grounded, other inputs and outputs open. I_{CC} (condition 2) is measured with all inputs and outputs open.

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions

‡All typical values are at V_{CC} = 5 V, T_A = 25°C.

§Not more than one output should be shorted at a time.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	WAVEFORM	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	0 thru 7	A0, A1, or A2	In-phase output	C_L = 45 pF, R_L = 667 Ω, See Note 3		11	17	ns
t_{PHL}						20	30	
t_{PLH}	0 thru 7	A0, A1, or A2	Out-of-phase output			23	35	ns
t_{PHL}						23	35	
t_{PZH}	EI	A0, A1, or A2				25	39	ns
t_{PZL}						24	41	
t_{PLH}	0 thru 7	EO	Out-of-phase output	C_L = 15 pF R_L = 2 kΩ, See Note 3		11	18	ns
t_{PHL}						26	40	
t_{PLH}	0 thru 7	GS	In-phase output			38	55	ns
t_{PHL}						9	21	
t_{PLH}	EI	GS	In-phase output			11	17	ns
t_{PHL}						14	36	
t_{PLH}	EI	EO	In-phase output			17	21	ns
t_{PHL}						25	40	
t_{PHZ}	EI	A0, A1, or A2		C_L = 5 pF R_L = 667 Ω		18	27	ns
t_{PLZ}						23	35	

¶ t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
t_{PZH} = output enable time to high level
t_{PZL} = output enable time to low level
t_{PHZ} = output disable time from high level
t_{PLZ} = output disable time from low level
NOTE 3: Load circuits and waveforms are shown on page 3-11.

TYPICAL APPLICATION DATA

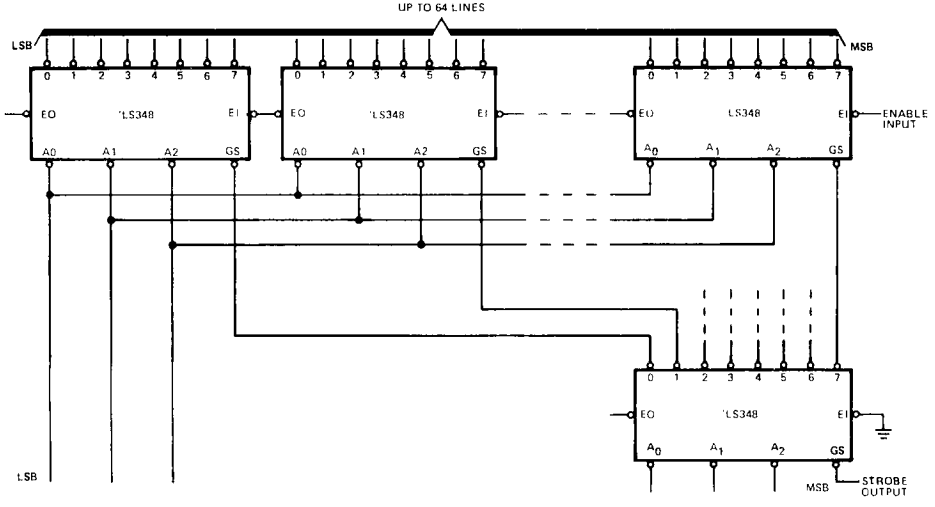


FIGURE 1–PRIORITY ENCODER WITH UP TO 64 INPUTS.

TMS 9909 NL, JL
FLOPPY DISK CONTROLLER

990/9900 FAMILY MICROCOMPUTER COMPONENTS

- Supports up to 4 Double-Sided Drives
- Single Density (FM) or
- Double-Density (MFM or M²FM)
- IBM 3740- and 2D-compatible and custom formats
- Programmable Stepper-Motor and Data Transfer Rates
- Write Precompensation
- 5-Inch or 8-Inch Diskettes
- Soft- and Hard-Sector Compatible
- Internal Phase Acquisition and
- Address Mark Detection

DESCRIPTION

The TMS 9909 Floppy Disk Controller (FDC) is designed to provide complete subsystem integration of a floppy diskette mass storage capability. The FDC is a general purpose peripheral device for microprocessor systems and is programmable by the CPU for data encoding formats, number and type of diskette drives, etc. This FDC programmability offers control for the interface between most host systems and virtually any floppy disk drive produced.

The FDC performs the following functions:

- Step to any track on the diskette
- Format tracks (initialize)
- Read and write diskette data
- Send status to host system.

The TMS 9909 Floppy Disk Controller is designed to provide high-level processing features for data transfer using single- and double-density formats. Integration of the FDC system is state-of-the-art, producing maximum performance with minimized hardware complexity, low component count and reduced system cost.

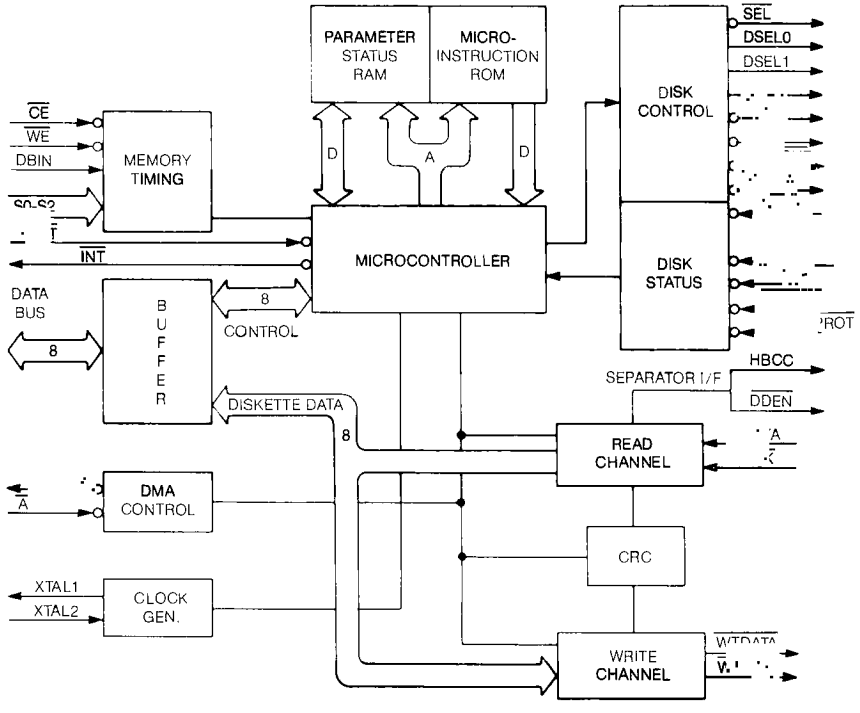


Figure 1. Functional Block Diagram

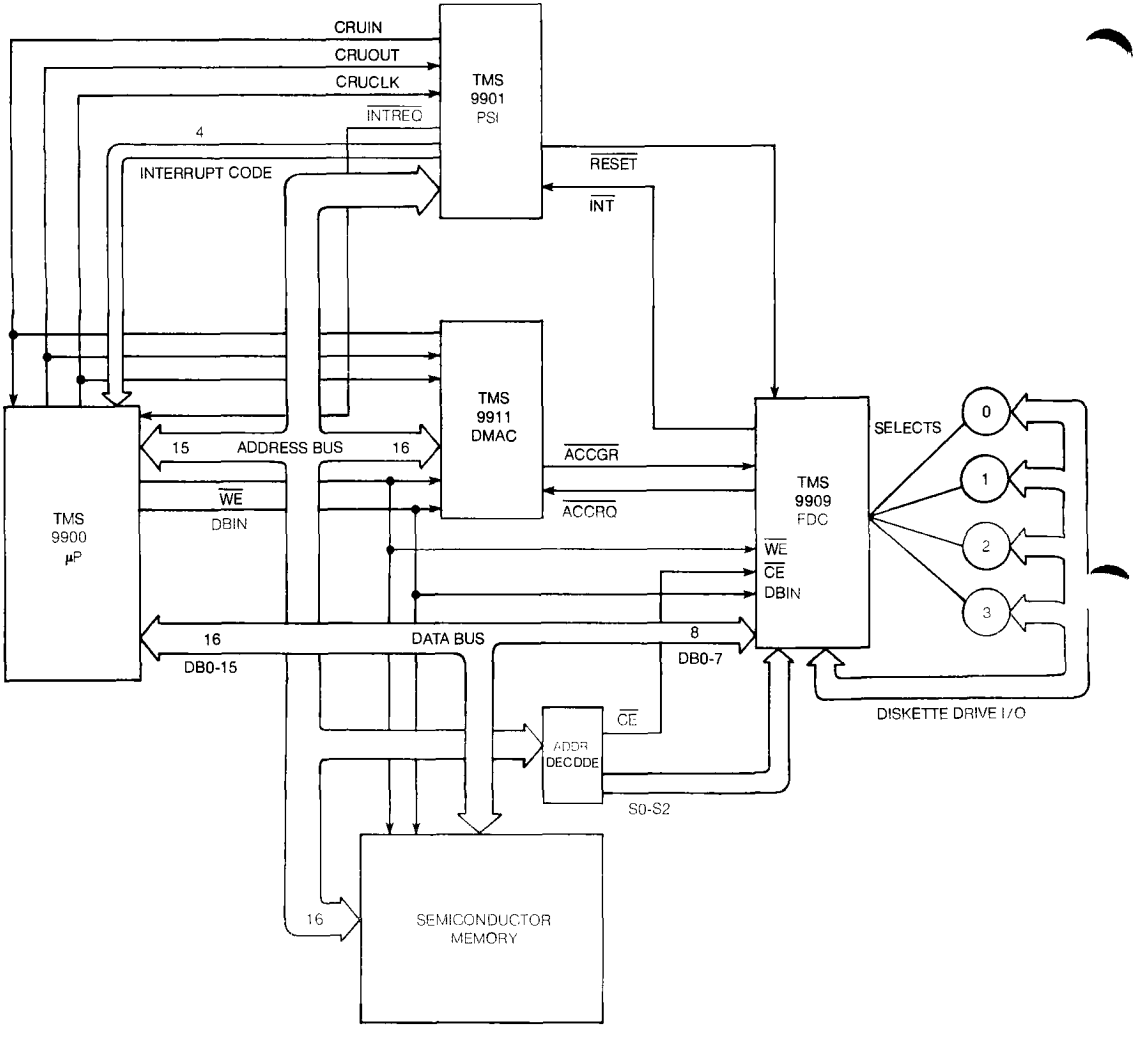
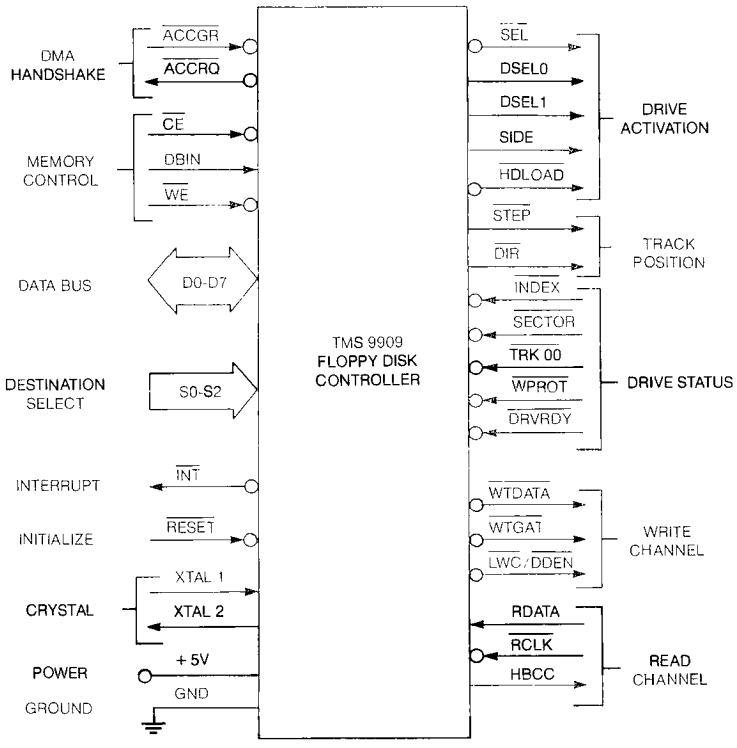


Figure 2—TMS 9900 Microcomputer System incorporating the TMS 9909 Floppy Disk Controller

PIN
DESCRIPTION



TMS 9909 PIN FUNCTIONS

Signature	*I/O*	*Description*
$\overline{\text{ACCRQ}}$	O	To activate the DMA channel, the FDC asserts ACCESS REQUEST.
$\overline{\text{ACCGR}}$	I	The DMA channel responds with ACCESS GRANT when transfer is to begin.
$\overline{\text{CE}}$	I	CHIP ENABLE serves to enable command and parameter input and status output to the host system.
DBIN	I	Data Bus In specifies the direction of data flow between the host system and FDC.
$\overline{\text{WE}}$	I	WRITE ENABLE pulses provide a window for data input to the FDC.
Data Bus DB0-7	I/O	The data bus is used to transfer information between the host system's data bus and the FDC's command, parameter and status registers.
S0-S2	I	To access data the host system must select a desination register using S0-S2.
$\overline{\text{INT}}$	O	Upon completion of an operation or detection of an error, the FDC issues an interrupt to the host.
$\overline{\text{RESET}}$	I	When the system powers up, a pulse on the reset line puts the FDC in its initialized state.
XTAL1	O	A 6-MHz crystal is connected between XTAL1 and XTAL2 to generate timing for the TMS 9909.
XTAL2	I	Alternately, a 6-MHz reference can be connected to XTAL2 with the XTAL1 pin unconnected.
+5V	—	FDC power supply.
GND	—	System ground connection.

Signature	*I/O*	*Description*
$\overline{\text{LWC}}/\overline{\text{DDEN}}$	O	LOW WRITE CURRENT is active when the track written is greater than 43, (programmable) for drives which reduce current on the inner tracks.
		DOUBLE DENSITY is output to the data separator during read operations.
$\overline{\text{STEP}}$	O	STEP pulses are issued by the FDC to move the selected drive's read/write head in the specified
$\overline{\text{DIRECTION}}$	O	DIRECTION, which is a level to define STEP-IN or STEP-OUT.
$\overline{\text{SEL}}$	O	SELECT is activated when DSEL0 and 1 contain an active drive address.
DSEL0	O	DRIVE SELECT 0 and 1 are encoded lines for up to four system drives to activate their control
DSEL1	O	and I/O lines.
SIDE	O	SIDE select determines the side of a dual-sided diskette used.
		Drive Status Lines Go Active On the Following Conditions:
$\overline{\text{INDEX}}$	I	The INDEX hole in the diskette is aligned with the diskette's index hole sensor.
$\overline{\text{SECTOR}}$	I	For hard-sectored diskettes, the SECTOR hole in the diskette is aligned with the diskette drive's sector hole sensor.
$\overline{\text{TRK00}}$	I	Selected drive's head is at its maximum radius (TRACK ZERO).
$\overline{\text{WPROT}}$	I	WRITE PROTECT, when active (low) indicates a read-only diskette has been selected.
$\overline{\text{DRVRDY}}$	I	DRIVE READY, indicates drive is powered up, door is closed, and diskette is properly installed.
RDATA	I	Data from the disk comes into the FDC via the READ DATA line.
$\overline{\text{RCLK}}$	I	READ CLOCK is a pulse synchronized to diskette clock and data half bit-cells.
HBCC	O	The HALF BIT-CELL CLOCK at two times the transfer rate, is used to start the data separator near data synchronization. Also, HBBC can be used to change write precompensation timing.
$\overline{\text{WTDATA}}$	O	The data to be written on the floppy disk is transmitted on the WRITE DATA line.
$\overline{\text{WTGAT}}$	O	WRITE GATE is true during data output to the diskette drive.

PROGRAM DESCRIPTION

The TMS 9909 Floppy Disk Controller is designed for ease of use. To execute a given command, the user writes a command code and parameter list to the FDC's eight-bit data port. When the last parameter has been transferred, the FDC begins command execution as an independent processor.

Setup for FDC commands proceeds as follows. The TMS 9909 is allocated eight memory addresses by the host system, all of which decode into a common chip enable. Selection of the base address with no offset (i.e., S0, S1, S2 = 000) gives the host write-only access to the COMMAND REGISTER and read-only access to the STATUS REGISTER. To initiate a command, the host checks the FDC's status for an idle condition, then writes a COMMAND CODE to the COMMAND REGISTER. Next a list of parameters is written into FDC RAM at S0-S2 address >0. After transfer of parameters is complete, command execution begins.

Track accessing is very flexible with the TMS 9909. Programmable stepper-motor control rates are loaded into the FDC during the RECALIBRATE and ASSIGN RATES command. RECALIBRATE moves the heads of selected drives to track 00. Read, write and write format commands include a parameter indicating the new physical track to seek.

During READ and WRITE sequences, the FDC transfers data between the selected diskette drive and system memory. Memory addressing and timing for data transfer are supervised by a Direct Memory Access Controller like the TMS 9911. The number of sectors of diskette data transferred is programmable from one up to an entire track. DMA handshake is activated for each byte of READ or WRITE data, every 16μs at a 500 k bits per second transfer rate.

Formatting diskettes is another capability of the TMS 9909. After the host CPU writes the command FORMAT TRACK and its associated parameters, the FDC activates its DMA handshake to acquire the contents and length of each field of the user's format. Since the host system provides the length and contents of each format field, specifying an address mark and a fill byte or all sector contents, the user has unlimited flexibility in selecting a format. Formatting IBM-compatible single- and double-density diskettes is straightforward, as is customizing formats to provide optimized interleaving for real-time applications.

An eight-bit status register can be read at the address of the command input register at any time by the CPU. The primary status register provides operating status including:

- Command execution in progress
- Diskette drive not ready
- Selected drive at track 00
- CRC error
- Data underflow/overflow
- ID not found, etc.

After the command in progress has completed its task, the contents of the FDC's internal RAM may be read (in the same way as parameters were written earlier) to provide detailed status information.

Completion of the command in progress, available to host system software in the status register, is signalled in hardware by the INTERRUPT pin of the TMS 9909. Any FDC command can be aborted by the host CPU by hardware activation of the RESET pin, or in software by writing an ABORT command to the command register.

The command macros of the TMS 9909 floppy disk controller are sufficiently powerful to fulfill the needs of most microcomputer systems on a stand-alone basis. For large micro/minicomputer systems, the modular commands can be combined for preprocessing by a dedicated microcomputer, such as the TMS 9985, to provide a sophisticated file management subsystem.

TMS 9909 FLOPPY DISK CONTROLLER COMMANDS

Type	*Command*	*Function*
Controller State Commands	Reset Controller	Initialize FDC internal state and output pins
	Abort Execution	Terminate active command. (At sector end for writes)
	Clear Interrupt	Deactivate $\overline{\text{INT}}$ pin, acknowledging command completion
Format Initialization Commands	Assign ID Attributes	Define address marks and contents of ID fields.
	Assign Fill and Sync	Define fill byte (one is written after end of sector) and sync byte (for phase acquisition) for current format.
Drive Control Commands	Recalibrate Drives, Assign Rates	Define stepper rates and restore selected drives to track zero.
	Seek, Check and Read Data	Seek physical track, locate desired ID (unless unformatted, i.e., Read ID) and transfer sectors (hard or soft) of diskette data to host system.
	Seek, Check and Write Data	Seek physical track and write sectors of data on a formatted diskette, hard or soft sector. Low write current and/or precompensation are selectable.
	Seek and Write Format	Seek physical track and write format fields on a whole track or single hard sector.

- Generation of All Memory Control Signals
- Supports 2 Independent DMA Devices
- Cascadeable to Multiple DMA Channels
- Memory Address and Limit Registers for Each Channel
- Automatic Interrupt Generation
- Operates Under Program Control of the Microprocessor Unit.

DESCRIPTION

The TMS9911 DMAC is an LSI member of the 9900 family of microprocessors and support peripherals. The DMAC is used in 9900 microprocessor systems where devices other than a single CPU require direct access to memory. The DMAC generates memory control signals and sequential memory addresses for two DMA channels (i.e., two independent DMA devices), allowing these devices to access memory autonomously with respect to the CPU. Multiple DMAC's may be used to extend the number of DMA channels beyond two. The interfaces of the DMAC to the CPU, system memory, and DMA peripheral devices are defined in such a manner as to require a minimum amount of additional electronics.

By using the CRU for set up and status of the TMS9911, a DMA controller has been configured in a 40-pin package with no data bus at all. The TMS9911 provides an easy-to-use cost effective method for implementing Direct Memory Access for 9900-family peripherals.

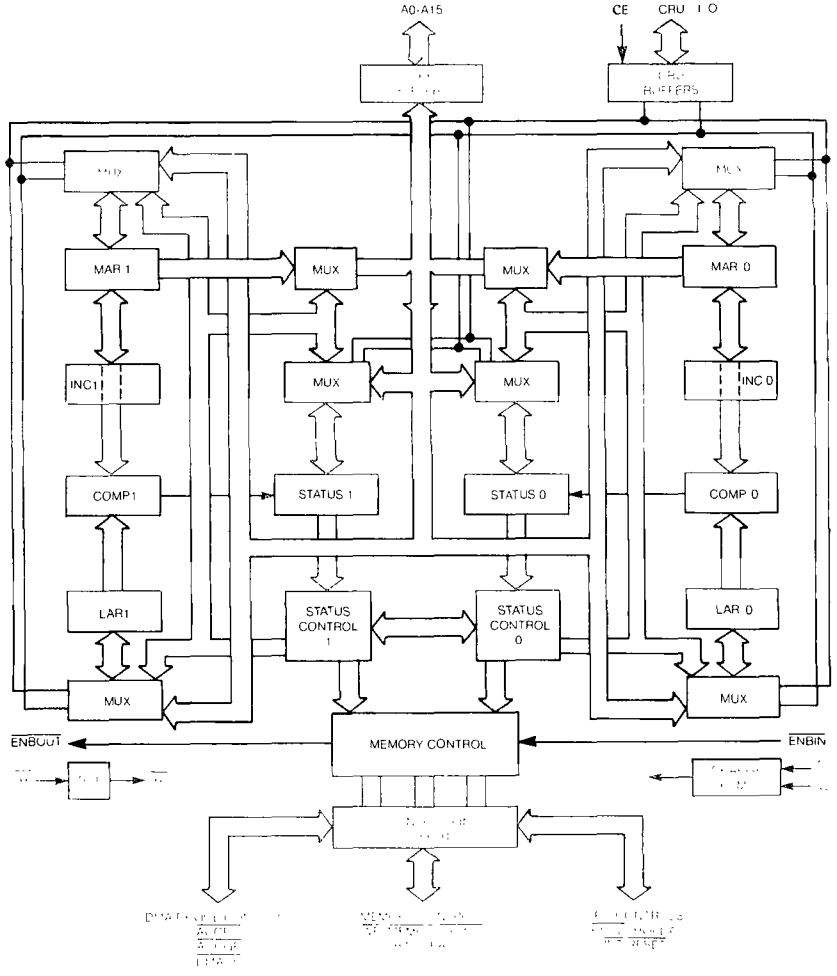


Figure 1. TMS 9911 Functional Block Diagram

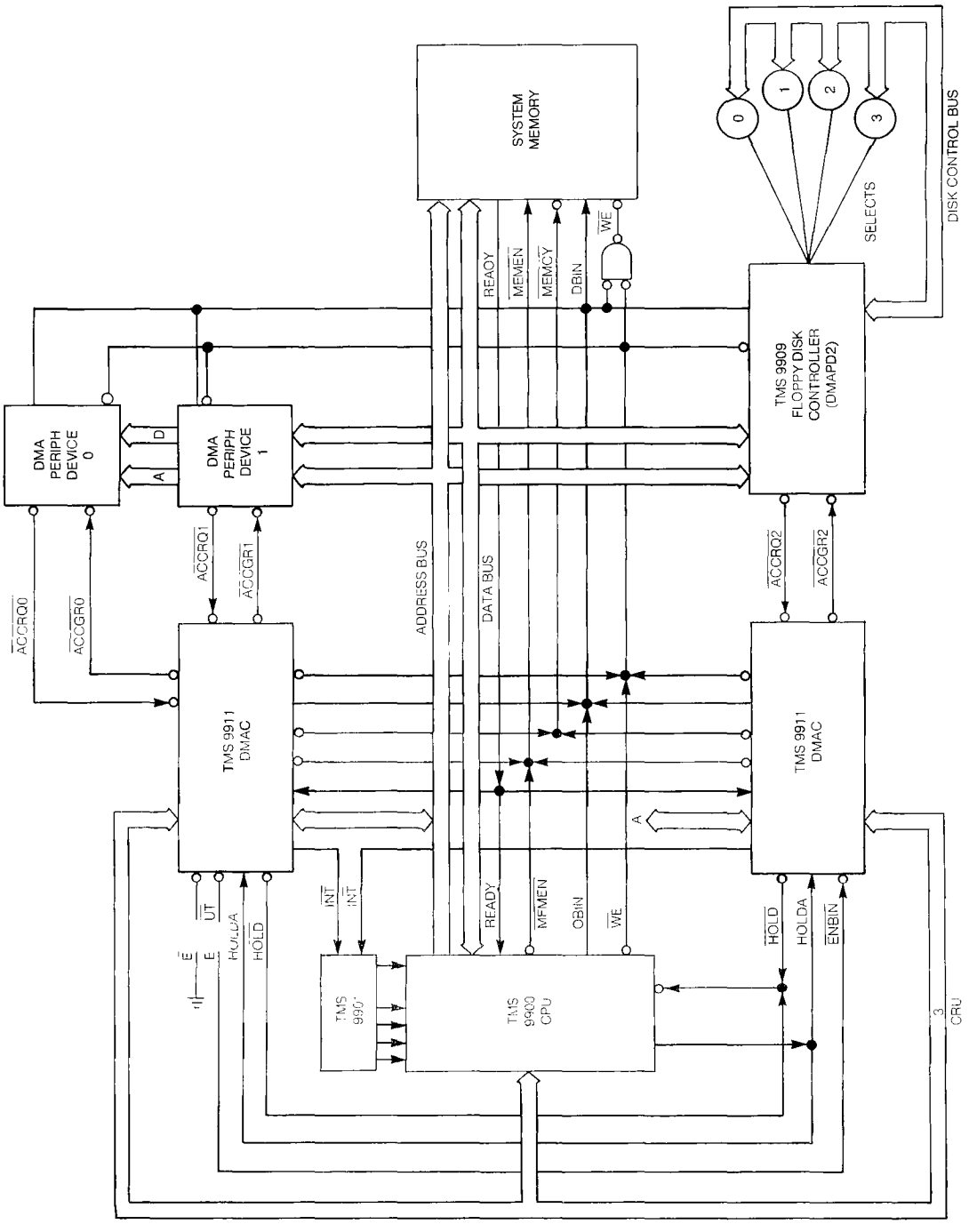
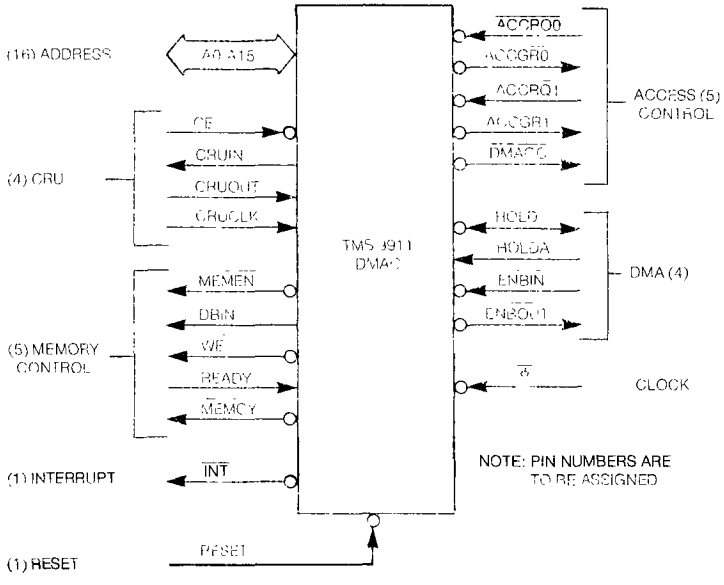


Figure 2. TMS 9900/TMS 9911 System Bus Interconnect



TMS 9911 DMAC Pin Description

TMS 9911 PIN FUNCTIONS

Signature	*I/O*	*Description*
		ADDRESS BUS
A0(MSB)	OUT	A0 through A15 comprise the Address Bus. The address bus outputs the memory address to
A1	OUT	or from which data is to be transferred while the DMAC accesses memory. A0-A15 outputs
A2	OUT	are at high-impedance while the DMAC is not accessing memory. A10-A14 are inputs to the
A3	OUT	DMAC, selecting the address of the bit to or from which the CPU is transferring data via the
A4		CRU. Although A15 is not normally implemented in TMS 9900 systems, this line may be
A5	OUT	used to select which half of the 16-bit data word is to be loaded when the DMA device is
A6	OUT	transferring a single byte.
A7	OUT	
A8	OUT	
A9	OUT	
A10	I/O	
A11	I/O	
A12	I/O	
A13	I/O	
A14	I/O	
A15(LSB)	OUT	
		CRU INTERFACE
$\overline{CE}$	IN	Chip Enable. $\overline{CE}$ is low when the CPU is transferring data to or from the DMAC via the CRU. The other CRU lines are ignored when $\overline{CE}$ is high. $\overline{CE}$ is normally generated by decoding a particular range of CRU addresses from the high order address lines A0-A9, and should not be low when memory accesses are being performed.

Signature	*I/O*	*Description*
CRUIN	OUT	CRU Input Data (to the CPU from the DMAC). CRUIN is at high impedance when $\overline{CE}$ is high. When $\overline{CE}$ is low, CRUIN contains the value of the bit addressed by A10-A14 to be read by the CPU.
CRUOUT	IN	CRU Output Data (from the CPU to the DMAC). CRUOUT contains the value of the datum to be transferred to the DMAC by the CPU during CRU operations.
CRUCLK	IN	CRU Output Data Clock. CRUCLK strobes the datum contained on CRUOUT to the bit addressed by A10-A14 when CE is low.
		MEMORY CONTROL
$\overline{MEMEN}$	OUT	Memory Enable. $\overline{MEMEN}$ is at high impedance except when the DMAC is accessing memory. When $\overline{MEMEN}$ is low, a memory cycle is in progress and no other device may access memory until the cycle is completed.
DBIN	OUT	Memory Data Read Enable. DBIN is at high-impedance except when the DMAC is accessing memory. During DMAC memory cycles DBIN indicates the direction of memory data transfer; i.e. DBIN = 1 for memory read and = 0 for memory write operations.
$\overline{WE}$	OUT	Write Enable. $\overline{WE}$ is at high impedance except when the DMAC is accessing memory. The timing for $\overline{WE}$ is identical to that of the $\overline{WE}$ output of the 9900 CPU's. $\overline{WE}$ performs the function of strobing data from a DMA device into memory during DMA.
READY	IN	Memory Transfer Ready. READY is sampled at the end of each clock cycle during each DMAC memory cycle. If READY = 0, the memory cycle is extended an additional clock cycle and READY is sampled again until READY = 1, at which time the memory cycle continues to completion.
$\overline{MEMCY}$	I/O	Memory Cycle. $\overline{MEMCY}$ is at high impedance except when the DMAC is accessing memory. $\overline{MEMCY}$ is low during all but the last clock cycle of each DMAC memory cycle. As an input, $\overline{MEMCY}$ is used to avoid bus conflicts during DMAC-to-DMAC control transfer.
		DMA DEVICE HANDSHAKE
$\overline{ACCRQ0}$	IN	DMA Device — Access Request. $\overline{ACCRQ0}$ is asserted by the DMA device connected to channel 0 when it wishes to access memory.
$\overline{ACCGR0}$	OUT	DMA Device 0 Access Granted. $\overline{ACCGR0}$ is active while the DMAC is performing a data transfer between memory and the DMA device connected to Channel 0.
$\overline{ACCRQ1}$	IN	DMA Device 1 Access Request. $\overline{ACCRQ1}$ provides the identical function for DMA Device 1 as does $\overline{ACCRQ0}$ for DMA Device 0.
$\overline{ACCGR1}$	OUT	DMA Device 1 Access Granted. ACCGR1 provides the identical function for DMA Device 1 as does ACCGR0 for DMA Device 0.
$\overline{DMACC}$	OUT	DMA accessing memory is active when the system buses are under DMA control. $\overline{DMACC}$ can be used to control the drive direction of bidirec al buffers (when required) such that A10-A14, $\overline{MEMCY}$ and $\overline{HOLD}$ are outputs when $\overline{DMACC}$ is active.

Signature	*I/O*	*Description*
		INTERRUPT OUTPUT
$\overline{INT}$	OUT	Interrupt Output. $\overline{INT}$ is low when either channel has transferred the specified number of bytes of data and the interrupt for that channel is enabled.
		DMA CONTROL
$\overline{HOLD}$	I/O	Hold Request. $\overline{HOLD}$ is an open-drain output which may be tied to other DMACs as an input to the CPU. $\overline{HOLD}$ becomes active when one of the Access Request signals is active and the channel corresponding to the access request is enabled. As an input, $\overline{HOLD}$ is used to set the internal $\overline{HOLD}$ condition by the equation: $\overline{HOLD}$ J(INT) = ACCRQn* (HOLD + $\overline{HOLDA}$).
HOLDA	IN	Hold Acknowledge. HOLDA is asserted by the CPU to indicate that it is entering the Hold state to allow a DMAC to access memory. It becomes active in response to the $\overline{HOLD}$ signal.
$\overline{ENBIN}$	IN	Enable Input. $\overline{ENBIN}$ must be low in order for the DMAC to access memory. When $\overline{ENBIN}$ = 1, a higher priority device is contending for access, thus inhibiting the DMAC from granting access.
$\overline{ENBOUT}$	OUT	Enable Output. $\overline{ENBOUT}$ is active only when $\overline{ENBIN}$ = 0 and neither channel is enabled. and requesting access. $\overline{ENBOUT}$ is connected to the $\overline{ENBIN}$ input of the DMAC of next lower priority.
		CLOCK SIGNALS
$\overline{\phi}$	IN	Clock Input. When the DMAC is used with a TMS 9900 Microprocessor this signal is provided by the $\overline{\phi 1}$ output of the TIM 9904 Clock Generator. In a 9980 system this signal is connected to the CKOUT output of the TMS 9980, and $\overline{HOLD}$ should be sychronized to avoid changing during $\overline{\phi 1}$.
		DEVICE RESET
$\overline{RESET}$	IN	Device Reset. When $\overline{RESET}$ is active, the DMAC is reset to a known state where both DMA channels are disabled. (MAR = LAR = 0, All status bits inactive, CHSEL = CnASEL = 0).
VCC	IN	5 Volts DC ± 5%
GND	IN	0 Volt reference. Pin 21 provides ground for the TMS 9911 logic. Pin 32 is the ground for signal buffers.

8

PROGRAM DESCRIPTION

Each of the 2 channels of the DMAC has two 16-bit registers: a Memory Address Register (MAR), and a Last Address Register (LAR). The Memory Address Register contains the memory address which the next memory cycle by that channel will access. After each memory access is completed, the Memory Address Register is automatically updated to the address for the next memory access. As the Memory Address Register is incremented, it is compared to the value contained in the Last Address Register. If the comparison is true, a status bit and (if enabled) an interrupt to the CPU are activated, indicating that the desired block of data has been transferred.

Each access requires that the DMAC gain control of the System Memory Bus. When the DMAC has control of the bus, no other DMAC or the CPU may perform a memory operation until the DMAC completes its memory cycle. Each memory access is performed by generating the necessary address and control signals to transfer a byte or word of data between system memory and the DMA peripheral device connected to the active channel.

The sequence of operations for using the TMS9911 DMAC is as follows: The host CPU sets up the control registers of the DMAC through the system's serial communication channel, the CRU.

At system power-up, the host must put the DMAC into an initialized state. This can be done with hardware by activating the DMAC's RESET pin, or through software addressing the CRU output bit SWRST (software reset).

The channel to be used for the current DMA transfer is selected by setting the CHSEL (channel select) CRU bit. Two independent channels are implemented on each TMS9911 chip with an automatic priority of CH0>CH1. When more than two DMA channels are required in a system, multiple TMS9911 circuits can be used.with priority established using the $\overline{\text{ENBIN}}/\overline{\text{ENBOUT}}$ chain, as shown in the system diagram.

The DMAC can control the transfer of sixteen-bit words or eight-bit bytes. The mode of operation is selected using the WRDSLn CRU bit. When byte mode is chosen, address bit A15 changes with each transfer. Memory systems which allow byte transfers must implement A15.

Having established the operational mode of this channel, the host CPU sets up the contents of the Memory Address Register and Last Address Register. Access to the MAR or LAR is gained by setting the sense of the CnASEL CRU bit. Thus the user executes a Set Bit One to CnASEL and LDCR for the sixteen bits of the MAR. Execution of another sixteen-bit LDCR sets up the contents of the LAR and the DMAC channel is configured to run.

If an interrupt should be issued by the DMAC when its operation is complete the IENB CRU bit for this channel must be set. Interrupts from TMS9911's are processed by the TMS9901 Programmable Systems Interface as shown in the system diagram. Alternately, interrupts of DMAC's can be wire-ORed and the CPU can poll DMAC channels to locate the active OPCOMP CRU bit.

The DMAC channel cannot begin to function until a final CRU bit, CHENBn, is set. Operation of the DMA channel is then under direct control of the $\overline{\text{ACCRQ}}/\overline{\text{ACCGR}}$ handshake.

Each $\overline{\text{ACCRQ}}$ by the DMA peripheral device causes the DMAC to gain memory bus control through the $\overline{\text{HOLD}}$/HOLDA handshake with the host CPU. $\overline{\text{ACCRQ}}$ may be held active continually by the DMA device, but is normally released after each transfer or block of transfers.

Termination of DMA channel activity can occur several ways. Though the host CPU can reset the DMAC or clear its CHENBn CRU bit, normal DMA termination occurs when the contents of the Memory Address Register equals that of the Last Address Register. Transfer does not occur to the LAR address. LAR is computed for the transfer of n bytes as follows:

$$LAR = MAR + n$$

Hence if MAR = 4 and 6 bytes are to be transferred, LAR = 10 and the last transfer is to address 9.

Status of the TMS9911 can be read through the CRU. CRU status input bits inform the host of the state of CRU output bits and provide a method for checking operation completeness by software.

8

TMS 9914 GENERAL PURPOSE INTERFACE BUS ADAPTER

- IEEE Std. 488-1975 Compatible
- Source and Acceptor Handshake
- Complete Talker and Listener Functions with Extended Addressing
- Controller and System Controller Capability
- Service Request
- Remote and Local with Lockout
- Serial and Parallel Polling
- Device Clear
- Device Trigger
- Compatible with TMS 9911 DMA Controller
- Single +5 V Power Supply
- Interfaces directly to SN75160/1/2 Transceivers

DESCRIPTION

The TMS 9914 General Purpose Interface Bus Adapter is a microprocessor controlled versatile device which enables the designer to implement all of the functions or a subset described in the IEEE Std. 488-1975. Using this standard, a variety of instruments can be interconnected and remotely or automatically programmed and controlled. The TMS 9914 is fabricated with N-channel silicon-gate technology and is completely TTL compatible on all inputs and outputs including the power supply (+5 V). It needs a single phase clock (nominally 5 MHz) which may be independent of the microprocessor system clock and, therefore, it can easily be interfaced with most microprocessors. The general purpose interface bus adapter (GPIBA) performs the majority of the functions contained in IEEE STd. 488-1975 and is versatile enough to allow software implementation of those sections not directly implemented in hardware.

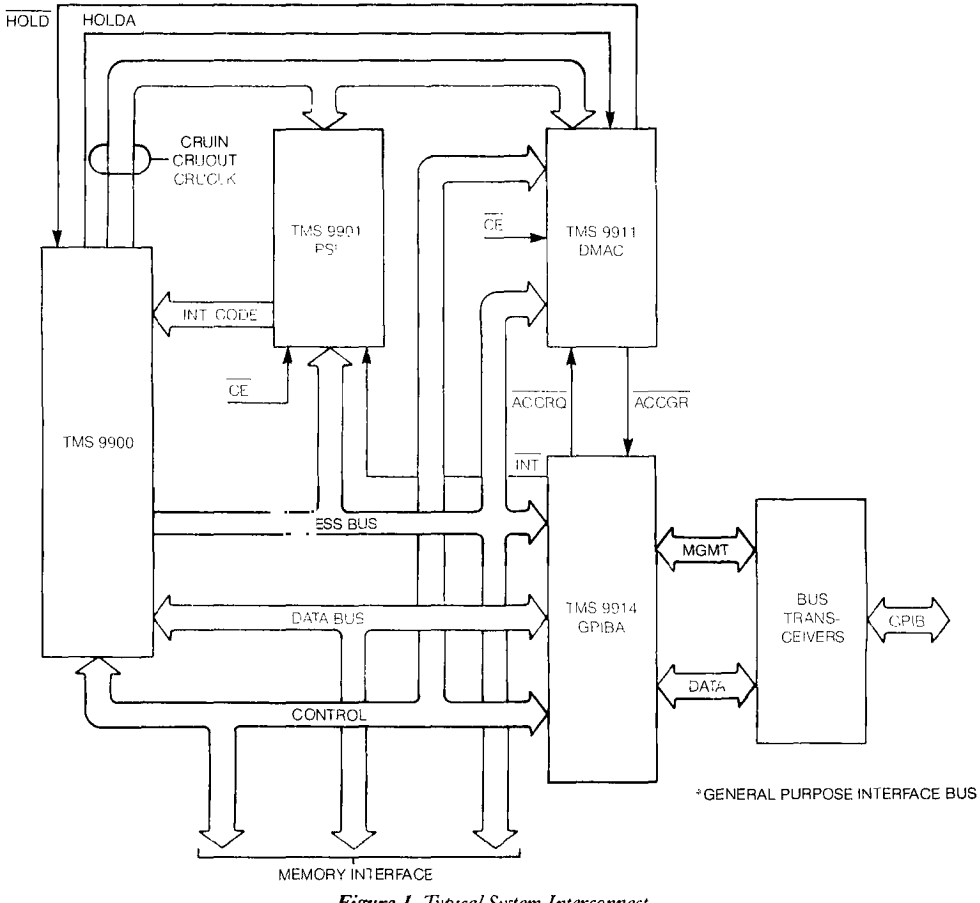


Figure 1. Typical System Interconnect

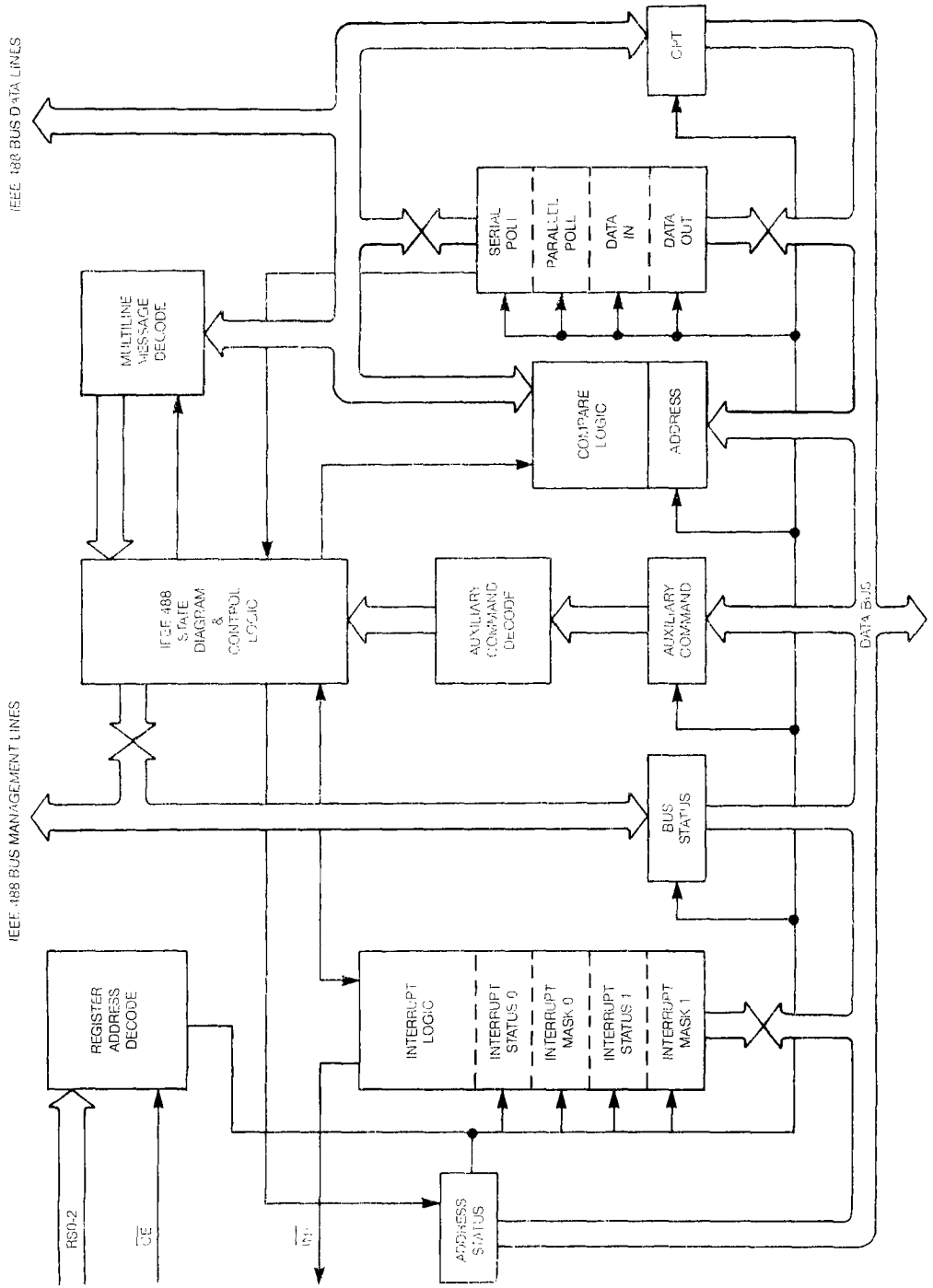


Figure 2. TMS 9914 Simplified Block Diagram

Table 1. Pin Description

Name	*I/O*	*Description*
DI01 through DI08	I/O	DATA I/O lines: allow data transfer between the TMS 9914 and the IEEE 488 data bus.
DAV	I/O	DATA VALID: Handshake Line. Sent by source device to indicate to acceptors that there is valid data on the IEEE bus data lines.
NRFD	I/O	NOT READY FOR DATA: Handshake Line. Sent by the acceptor to the source device to indicate when it is ready for a new byte of data.
NDAC	I/O	DATA NOT ACCEPTED: Handshake Line. Sent by acceptor to source device to indicate when it has accepted the current byte on the data bus.
ATN	I/O	ATTENTION: Management Line. Sent by the controller. When ATN is asserted, the information on the data lines is interpreted as commands, sent by the controller . . . When ATN is false, the data lines carry data.
IFC	I/O	INTERFACE CLEAR. Management Line. Sent by system controller to set the interface system, portions of which are contained in all interconnected devices in a known quiescent state. System controller assumes control. Open drain output with internal pullup.
REN	I/O	REMOTE ENABLE: Management Line. Sent by system controller and is used in conjunction with other messages to select between two alternate sources of programming data, e.g. via interface or front panel. Open drain output with internal pullup.
SRQ	I/O	SERVICE REQUEST: Management Line. Issued by a device on the bus to the controller to indicate a need for service.
EOI	I/O	END OR IDENTIFY: Management Line. If ATN is false, this signal is sent by the "talker" to indicate the end of a multiple byte transfer. If sent by the controller with ATN true, this will perform the parallel polling sequence.
$\overline{\text{CONTROLLER}}$	O	Bus transceiver control line. Indicates that the device is the controller.
TE	O	TALK ENABLE: Bus transceiver control line. Indicates the direction of data transfer on the data bus.
D0 through D7	I/O	Data I/O lines that allow transfer of data between TMS 9914 and the microprocessor.
RS0 through RS2	I	Address lines through which the TMS 9914 registers can be accessed by the microprocessor.
DBIN	I	When true (high) DBIN indicates to the TMS 9914 that the microprocessor is about to read from one of its registers. When false, that the microprocessor is about to write to one of its registers.
$\overline{\text{WE}}$	I	$\overline{\text{WRITE ENABLE}}$: indicates to the TMS 9914 that one of its registers is being written to.
$\overline{\text{CE}}$	I	$\overline{\text{CHIP ENABLE}}$: selects and enables the TMS 9914 for an microprocessor data transfer.
$\overline{\text{INT}}$	O	$\overline{\text{INT}}$: Open drain output. Sent to microprocessor to indicate the occurrence of an event on the bus requiring service.
$\overline{\text{ACCRQ}}$	O	$\overline{\text{ACCESS REQUEST}}$: Signal to TMS 9911 DMA controller requesting DMA.

PIN OUTS TO BE ASSIGNED

NOTE: The names of the IEEE bus lines have been maintained, and are therefore negative logic signals.

TMS 9914 GENERAL PURPOSE INTERFACE BUS ADAPTER

FUNCTIONAL DESCRIPTION

The TMS 9914 interfaces to the CPU with an eight-bit bidirectional data bus, three register select lines, two DMA control lines, reset and interrupt request lines, a DBIN and a $\overline{WE}$ line.

The internal architecture of the TMS 9914 is arranged into 13 registers, there being seven WRITE and six READ registers. Some are actually address ports through which current status can be obtained. Table 2 lists these registers and their addresses. The microporcessor accesses a TMS 9914 register by supplying the correct register address in conjunction with $\overline{WE}$ and DBIN. The $\overline{CE}$ is used to enable the address decode.

Table 2. TMS 9914 Registers and Addresses

NAME	TYPE	RS2	RS1	RS0	DBIN	$\overline{WE}$
INTERRUPT STATUS 0	R	0	0	0	1	1
INTERRUPT MASK 0	W	0	0	0	0	0
INTERRUPT STATUS 1	R	0	0	1	1	1
INTERRUPT MASK 1	W	0	0	1	0	0
ADDRESS STATUS	R	0	1	0	1	1
BUS STATUS	R	0	1	1	1	1
AUXILIARY COMMAND	W	0	1	1	0	0
ADDRESS SWITCH	R	1	0	0	1	1
ADDRESS	W	1	0	0	0	0
SERIAL POLL	W	1	0	1	0	0
COMMAND PASS THROUGH	R	1	1	0	1	1
PARALLEL POLL	W	1	1	0	0	0
DATA IN	R	1	1	1	1	1
DATA OUT	W	1	1	1	0	0

NOTE: The Address Switch register is external to the TMS 9914

In DMA operation the TMS 9911 supplies the memory address but not the peripheral device address (i.e., RS0-2, $\overline{CE}$) are not supplied). When the TMS 9914 sets $\overline{ACCRQ}$ low true, it is either because of a byte input or a byte output, and this will happen whether or not DMA transfer will take place. If in response to $\overline{ACCRQ}$ an $\overline{ACCGR}$ (access granted) is received, the $\overline{ACCRQ}$ will be reset and a DMA transfer will take place between the system memory and either the Data In or Data Out register. If the data transfer is with the microprocessor and if the microprocessor addresses either the Data In or Data Out register, the $\overline{ACCRQ}$ line will be reset. Note that in DMA mode the sense of DBIN is inverted.

Table 3 lists the commands which are directly handled by the TMS 9914, and those which require intervention by the microprocessor for their implementation.

Table 3. Remote Multiple Message Coding

		DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1		Note
Addressed Command Group	ACG	X	0	0	0	X	X	X	X	AC	
Device Clear	DCL	X	0	0	1	0	1	0	0	UC	
Group Execute Trigger	GET	X	0	0	0	1	0	0	0	AC	
Go To Local	GTL	X	0	0	0	0	0	0	1	AC	
Listen Address Group	LAG	X	0	1	X	X	X	X	X	AD	
Local Lock Out	LLO	X	0	0	1	0	0	0	1	UC	
My Listen Address	MLA	X	0	1	L	L	L	L	L	AD	1
My Talk Address	MTA	X	1	0	T	T	T	T	T	AD	2
My Secondary Address	MSA	X	1	1	S	S	S	S	S	SE	3, 4
Other Secondary Address	OSA									SE	4, 5
Other Talk Address	OTA					TAG • MTA				AD	
Primary Command Group	PCG									—	6
Parallel Poll Configure	PPC	X	0	0	0	0	1	0	1	AC	7
Parallel Poll Enable	PPE	X	1	1	0	S	P	P	P	SE	8, 9
Parallel Poll Disable	PPD	X	1	1	1	D	D	D	D	SE	8, 10
Parallel Poll Unconfigure	PPU	X	0	0	1	0	1	0	1	UC	11
Secondary Command Group	SCG	X	1	1	X	X	X	X	X	SE	
Selected Device Clear	SDC	X	0	0	0	0	1	0	0	AC	
Serial Poll Disable	SPD	X	0	0	1	1	0	0	1	UC	
Serial Poll Enable	SPE	X	0	0	1	1	0	0	0	UC	
Take Control	TCT	X	0	0	0	1	0	0	1	AC	12
Talk Address Group	TAG	X	1	0	X	X	X	X	X	AD	
Universal Command Group	UCG	X	0	0	1	X	X	X	X	UC	
Unlisten	UNL	X	0	1	1	1	1	1	1	AD	
Untalk	UNT	X	1	0	1	1	1	1	1	AD	

Symbols: AC — Addressed Command
AD — Address (Talk or Listen)
UC — Universal Command
SE — Secondary (Command or Address)
0 — Logical Zero (high level on IEEE Bus; Low level within 9914).
1 — Logical One (Low level on IEEE Bus; High level within 9914).
X — Don't Care (received message)
X — Must Not Drive (transmitted message)

►8

Notes to Table 3:

1. L L L L L: Represents the coding for the device listen address.
2. T T T T T: Represents the coding for the device talk address.
3. S S S S S: Represents the coding for the device secondary address.
4. Secondary addresses will be handled via address pass through.
5. OSA will be handled as an invalid secondary address pass through by the MPU.
6. PCG = ACG v UCG v LAG v TAG
7. PPC will be handled in software by the MPU via Unrecognized Address Command Group pass through.
8. PPE, PPD will be handled via pass through next secondary feature.
9. S P P P represents the sense and bit for remote configurable parallel poll.
10. D D D D specify don't care bits that must be sent all zeroes, but need not he decoded by receiving device.
11. PPU is handled via Unrecognized Universal Command Group pass through.
12. TCT will be handled via Unrecognized Addressed Command Group pass through. However, in this case, the device must be in TADS before the pass through will occur.

Interrupt Status Registers 0 and 1

INT0	INT1	BI	BO	END	SPAS	RLC	MAC
GET	UUCG	UACG	APT	DCAS	MA	SRQ	IFC

INT0	An interrupt occurred in register 0	GET	A Group Execute Trigger has occurred
INT1	An interrupt occurred in register 1	UUCG	An Undefined Universal Command has been received
BI	A byte has been received		
BO	A byte has been output	UACG	An Undefined Addressed Command has been received. This bit will also be set on receipt of a secondary command when the pts feature in the Auxiliary Command register is utilized.
END	An EOI occurred with ATN false		
SPAS	Serial Poll Active State has occurred with rsv set in the Serial Poll register		
RLC	A REMOTE/LOCAL change has occurred	APT	A secondary address has occurred
		DCAS	Device Clear Active State has occurred
MAC	An address change has occurred	MA	My Address (MLAVMTA)•$\overline{\text{SPSM}}$
		SRQ	A Service Request has been received
		IFC	An IFC has been received

INT0 is the logical OR of each bit of Interrupt Status Register 0 ANDed with the respective bit of Interrupt Mask Register 0. INT1 is the same but applies to Interrupt Mask and Status Register 1. Reading either Interrupt Status Register will also clear it. The INT line will be cleared only when the interrupt status register which caused the interrupt is read.

Interrupt Mask Registers 0 and 1

		BI	BO	END	IFC	RLC	MAC
GET	UUCG	UACG	APT	DCAS	MA	SRQ	SPAS

The Interrupt Mask Registers 0 and 1 correspond to the Interrupt Status Registers 0 and 1 respectively, with the exception of INT0 and INT1.

Address Status Register

REM	LLO	ATN	LPAS	TPAS	LADS V LACS	TADS V TACS	ulpa

The Address Status Register is used to convey the addressed state of the talker/listener and the remote/local and local lockout condition. This information is derived from the TMS 9914 internal logic states at the time of reading. The ulpa bit is used for dual addressing and indicates the state of the LSB of the bus at last primary addressed time.

Bus Status Register

ATN	DAV	NDAC	NRFD	EOI	SRQ	IFC	REN

The Bus Status Register allows the microprocessor to obtain the current status of the IEEE 488 Bus Management Lines.

Auxiliary Command Register

C/S			f4	f3	f2	f1	f0

The Auxiliary Command Register allows control of additional features on chip and provides a means of inputting some of the local messages to the interface functions. Table 4 lists these messages and commands. If C/S = 1, the feature will be set and if C/S = 0, the feature will be cleared. If C/S = NA, it should be sent as zero.

Table 4. Auxiliary Commands

Function	*Mnemonic*	*C/S*	*f4*	*f3*	*f2*	*f1*	*f0*
Chip Reset	rst	0/1	0	0	0	0	0
Release ACDS holdoff	dacr	0/1	0	0	0	0	1
Release RFD holdoff	rhfd	NA	0	0	0	1	0
Holdoff on all data	hdfa	0/1	0	0	0	1	1
Holdoff on EOI only	hdfe	0/1	0	0	1	0	0
Set new byte available false	nbaf	NA	0	0	1	0	1
Force group execute trigger	fget	0/1	0	0	1	1	0
Return to local	rtl	0/1	0	0	1	1	1
Return to local immediate	rtli	0	0	0	1	1	1
Send EOI with next byte	feoi	NA	0	1	0	0	0
Listen only	lon	0/1	0	1	0	0	1
Talk only	ton	0/1	0	1	0	1	0
Take control synchronously	tcs	NA	0	1	1	0	1
Take control asynchronously	tca	NA	0	1	1	0	0
Go to standby	gts	NA	0	1	0	1	1
Request parallel poll	rpp	0/1	0	1	1	1	0
Send interface clear	sic	0/1	0	1	1	1	1
Send remote enable	sre	0/1	1	0	0	0	0
Request contol	rqc	NA	1	0	0	0	1
Release control	rlc	NA	1	0	0	1	0
Disable all interrupts	dai	0/1	1	0	0	1	1
Pass through next secondary	pts	NA	1	0	1	0	0
Set T1 delay	stdl	0/1	1	0	1	0	1

Address Register

edpa	dal	dat	A5	A4	A3	A2	A1

edpa	enable dual primary addressing	dat	disable the talk function
dal	disable the listen function	A1 – A5	primary device address

The Address Switch Register corresponds to the Address Register. A power-up RESET or a rst command with C/S = 1 will leave the chip in a totally idle state. At this point, the Address Switch Register is read and the value is written into the Address Register. The reset condition is then cleared by sending rst with C/S = 0.

Serial Poll Register

S8	rsv	S6	S5	S4	S3	S2	S1

The Serial Poll register is used to establish the status byte that is sent out when the controller conducts a serial poll. Bits 1 through 6 and 8 contain status information, while bit 7, rsv, is used to enable the SRQ line and to indicate to the controller which device(s) was responsible for making a service request.

Command Pass Through Register

DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

The Command Pass Through Register is used to pass through to the microprocessor any commands or secondary addresses that are not automatically handled in the TMS 9914.

Parallel Poll Register

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

This register contains the status bit that is output when the controller conducts a parallel poll.

Data-In Register

DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

The data-in register is used to move data from the interface bus when the chip is addressed as a listener. Upon receipt of a data byte, the chip will hold NRFD true until the microprocessor reads the data-in register, when NRFD will be set false automatically.

Data-Out Register

DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

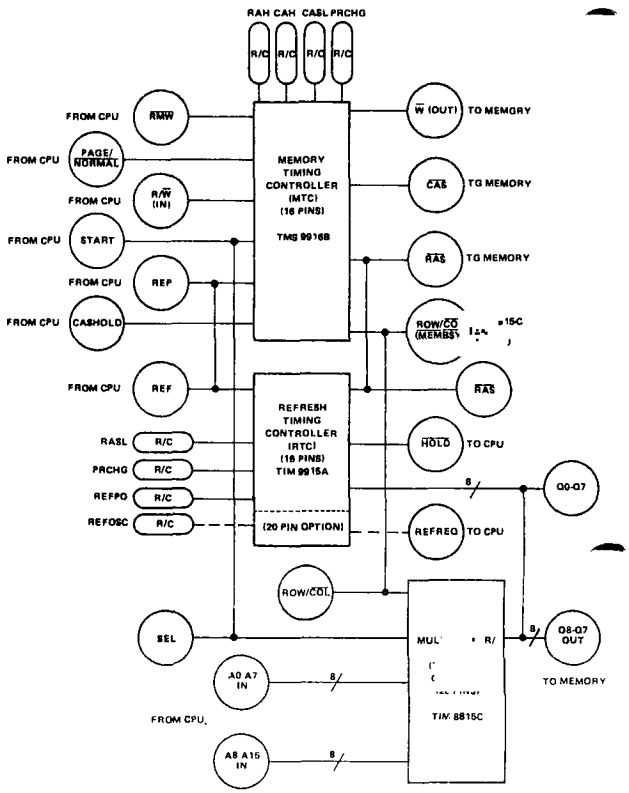
The data-out register is used to move data from the TMS 9914 onto the IEEE std 488-1975 data bus.

After sending a byte out on the bus, the device can take part in a new handshake only after a new byte is placed in the data-out register, when it will be able to send DAV true again.

TIM 9915 - MEMORY TIMING (AND REFRESH) CONTROLLER CHIP SET

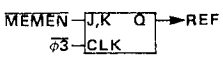
Features of TIM 9915:

- Controls the operation of 4K/16K/64K Dynamic RAMs
- Creates Static RAM Appearance
- Generation and Synchronization of
  - $\overline{RAS}$, $\overline{CAS}$, clocks (and precharge)
  - $\overline{WE}$ Signal
  - Address Multiplexing
  - Refresh (multiple modes)
- Works over wide range of memory speeds
  - Access from 120 ns up to refresh limits
  - Even faster with precision R/C's
- Performs multiple memory cycles
  - Read, Early Write, Read/Write, RMW
- Page-Mode Operation for all memory cycle types
- Selectable Refresh Modes
  - Transparent
  - Cycle Steal
  - Burst Mode
- Refresh Violation Detection
  - Automatic Burst Mode on violation acknowledgement
- Simple Asynchronous START Clock
- Extended $\overline{CAS}$ Data Hold Control via CASHOLD
- All system outputs are
  - Bus Drivers (24mA guaranteed)
  - Tri-State



NOTES: Transparent Refresh Interface for Popular Microprocessors.

TMS 9900



Z80 "MREF" = REF

8080A M1 • T4 = REF

8085 M1 • T4 = REF

DESIGN GOAL

This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

TMS 9927JL,NL
VIDEO TIMER/CONTROLLER

990/9900 FAMILY MICROCOMPUTER COMPONENTS

- Second sourced by SMC as CRT5027
- TTL Compatibility
- Standard and Non-standard CRT Monitor Capability
- Scrolling Capability
- Interlaced and Non-interlaced Formatting
- Fully Programmable Display Format
  - Characters per data row
  - Data rows per frame
  - Raster scans per data row
  - Raster scans per frame
- Fully Programmable Monitor Format
  - Blanking
  - Horizontal sync
  - Vertical sync
  - Composite sync
- Two Programming Methods
  - Processor controlled
  - PROM on data bus
- Generation of Cursor Video
- N-channel Silicon-Gate Technology

DESCRIPTION

The TMS9927 is a single-chip video timer and controller produced using N-channel silicon-gate MOS technology. This 40-pin package contains the logic to generate all the timing signals for display of video data on standard or nonstandard CRT monitors in both interlaced or noninterlaced format. The only function not on the chip is the dot counter; which, due to high video frequencies, cannot easily be implemented with MOS technology. All the inputs and outputs are TTL compatible.

There are nine 8-bit control registers which are user programmable (see *Table 1*). Seven of the control registers are for horizontal and vertical formatting and two are for cursor address.

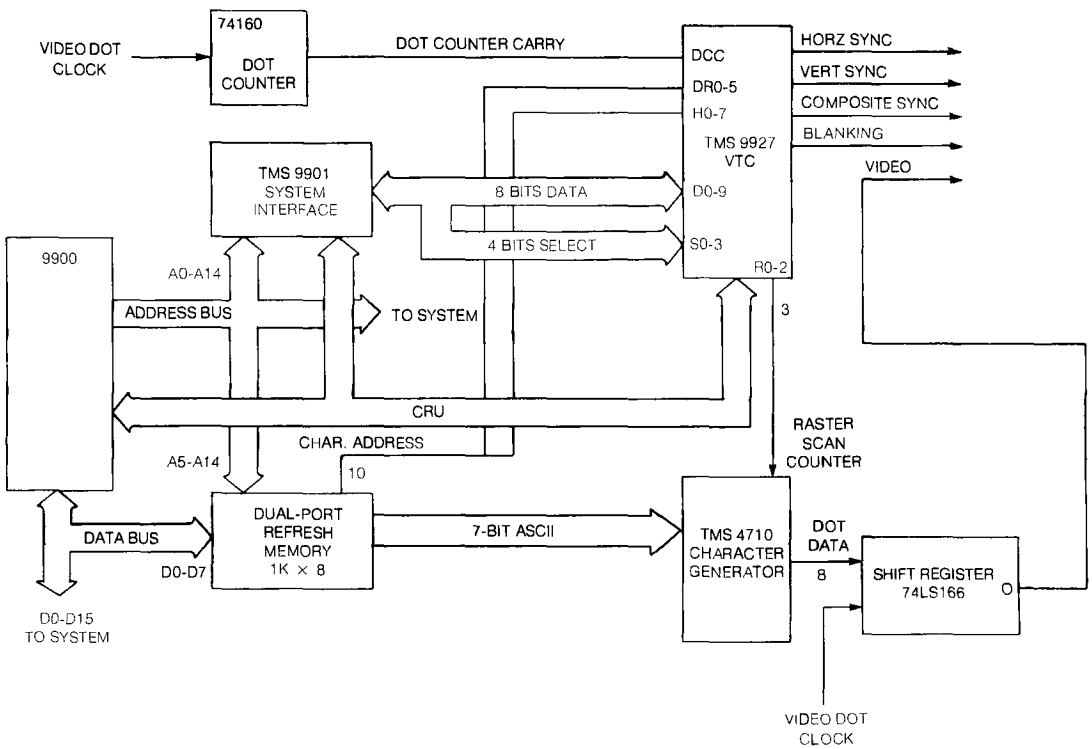


Figure 1. Typical System Interconnect

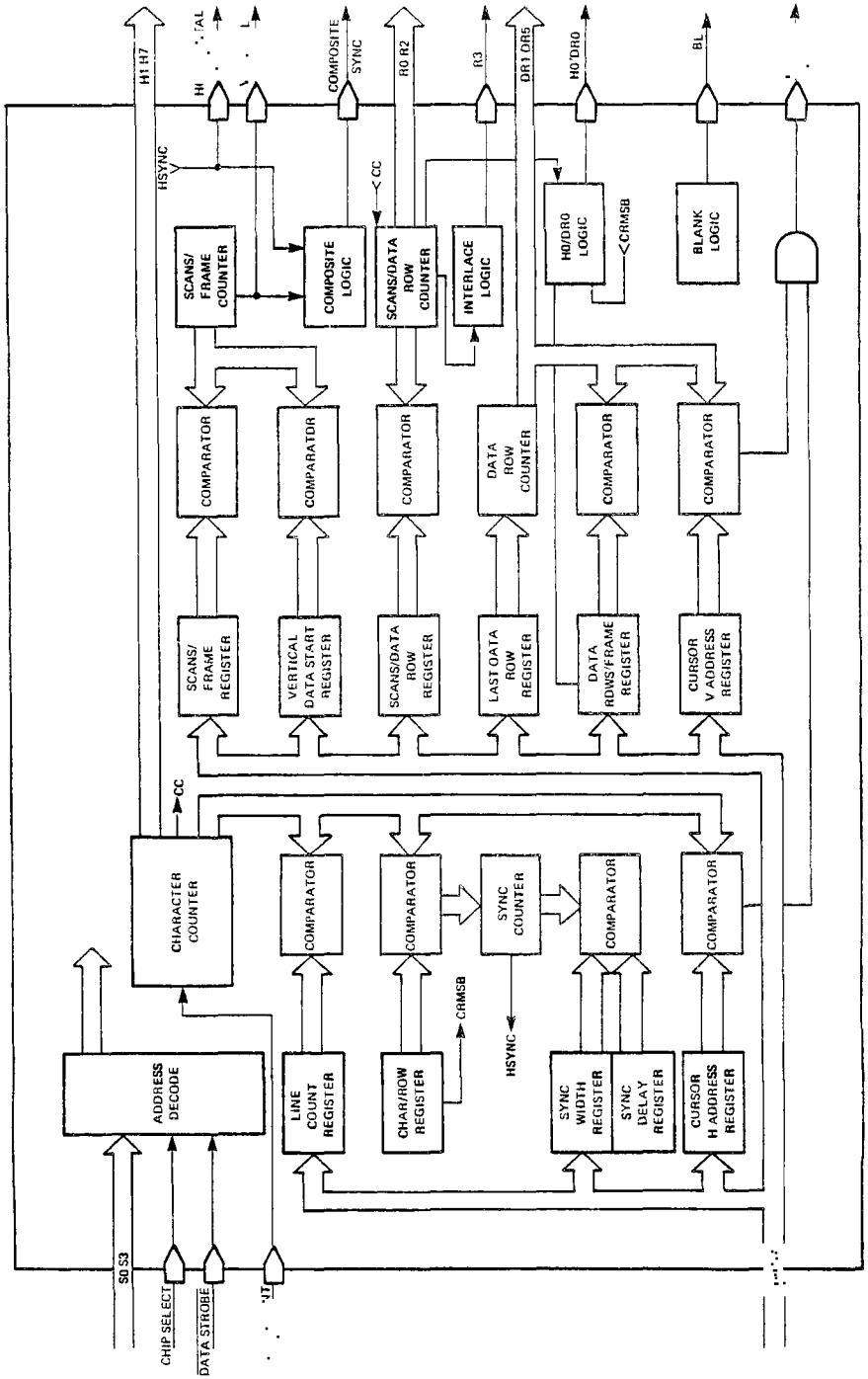
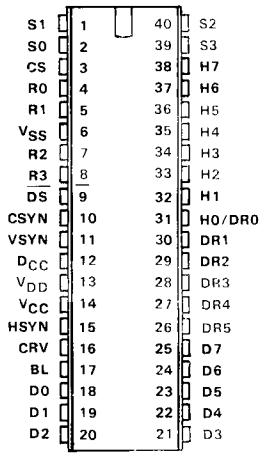


Figure 2. TMS 9927 Architecture.

TMS 9927 PIN FUNCTIONS

Signature	*I/O*	*Description*
D0-7	I/O	Data bus. Input bus for control words from microprocessor or PROM. Bidirectional bus for cursor address.
CS	I	Signals chip that it is being addressed
S0-3	I	Register address bits for selecting one of seven control registers or either of the cursor address registers
$\overline{DS}$	I	Strobes D0-7 into the appropriate register or outputs the cursor character address or cursor line address onto the data bus
DCC	I	Carry from off chip dot counter establishing basic character clock rate.
H7-1	O	Character counter outputs.
R0-2	O	Three most significant bits of the Scan Counter: row select inputs to character generator.
H0/DR0	O	Pin definition is user programmable. Output is MSB of Character Counter if MSB of Characters/Data Row word is a "1", otherwise output is MSB of Data Row Counter
R3	O	Least significant bit of the scan counter. In interlaced mode this bit defines the odd or even field. In this way, odd scan lines of the character font are selected during the odd field and even scans during the even field
DR1-5	O	Data Row counter outputs
BL	O	Defines non active portion of horizontal and vertical scans.
HSYN	O	Initiates horizontal retrace
VSYN	O	Initiates vertical retrace
CSYN	O	Active in non-interlaced mode only. Provides a true RS-170 composite sync waveform.
Vcc	PS	+5 volt Power Supply
V_{DD}	PS	+12 volt Power Supply
V_{SS}	PS	Ground reference



FUNCTIONAL DESCRIPTION

Application Oriented Use

The TMS9927 interfaces to the central processor unit, if one is used, through the communications register unit (CRU) via a TMS9901, as shown in *Figure 2*, or it functions as a mapped memory device. The TMS9901 converts 8 bits of serial CRU data to parallel data to feed the TMS9927 data bus for loading the control registers. The CPU, using the CRU, decodes the high order bits of the address for the TMS9927 chip select and the four low order bits are connected directly to the TMS9927 Video Timer and Controller (VTC) for control register select. The character column (H1-H7) and row lines (DR1-DR5) combine to address the refresh RAM. The refresh RAM outputs the seven-bit ASCII code for the character to be displayed to the character generator, the TMS4710. The character generator uses the raster scan counter (R0-R2) to select which row of the dot matrix to output. A shift register then shifts the dot information out to the video terminal at the dot frequency.

The TMS9927 does have a self-load function as shown in *Figure 3*. It is accomplished by putting the self-load command on the VTC select lines and strobing DATA STROBE ($\overline{DS}$). This causes the TMS9927 to output address information on its row select lines to the control PROM (74S288). The outputs of the control PROM are loaded into the VTC control registers. There are two types of self-load: processor and nonprocessor. The nonprocessor self-load automatically starts the timing chain after load is completed. Processor self-load only causes a self-load and then waits for the start command from the processor. The select signals to the VTC which cause self-load should be applied for the entire duration of self-load.

Table 1. Select Decodes

S0	*S1*	*S2*	*S3*	*Command*	*Description*
0	0	0	0	Load Control Register 0	
0	0	0	1	Load Control Register 1	
0	0	1	0	Load Control Register 2	
0	0	1	1	Load Control Register 3	See *Table 2*
0	1	0	0	Load Control Register 4	
0	1	0	1	Load Control Register 5	
0	1	1	0	Load Control Register 6	
0	1	1	1	Processor Self Load	Command from processor instructing TMS 9927 to enter Self Load Mode
1	0	0	0	Read Cursor Row Address	
1	0	0	1	Read Cursor Character Address	
1	0	1	0	Reset	Resets timing chain to top left of page. Reset is latched on chip by $\overline{DS}$ and counters are held until released by start command.
1	0	1	1	Up Scroll	Increments address of first displayed data row on page. ie; prior to receipt of scroll command—top line = 0, bottom line = 23. After receipt of Scroll Command—top line = 1, bottom line = 0.
1	1	0	0	Load Cursor Character Address	
1	1	0	1	Load Cursor Row Address	
1	1	1	0	Start Timing Chain	Receipt of this command after a Reset or Processor Self Load command will release the timing chain approximately one scan line later. In applications requiring synchronous operation of more than one TMS9927 the dot counter carry should be held low during the $\overline{DS}$ for this command.
1	1	1	1	Non-Processor Self Load	Device will begin self load via PROM when $\overline{DS}$ goes low. The 1111 command should be maintained on S0-3 long enough to guarantee self load (Scan counter should cycle through at least once). Self load is automatically terminated and timing chain initiated when the all "1's" condition is removed, independent of $\overline{DS}$. For synchronous operation of more than one TMS 9927, the Dot Counter Carry should be held low when this command is removed.

Note: During Self Load, the scan counter states corresponding to the nine load command addresses will load the appropriate register. Therefore if reseting of the cursor X and Y position registers is required via self load the PROM words for address 1100 and 1101 should be programmed as all zeros.

▶8

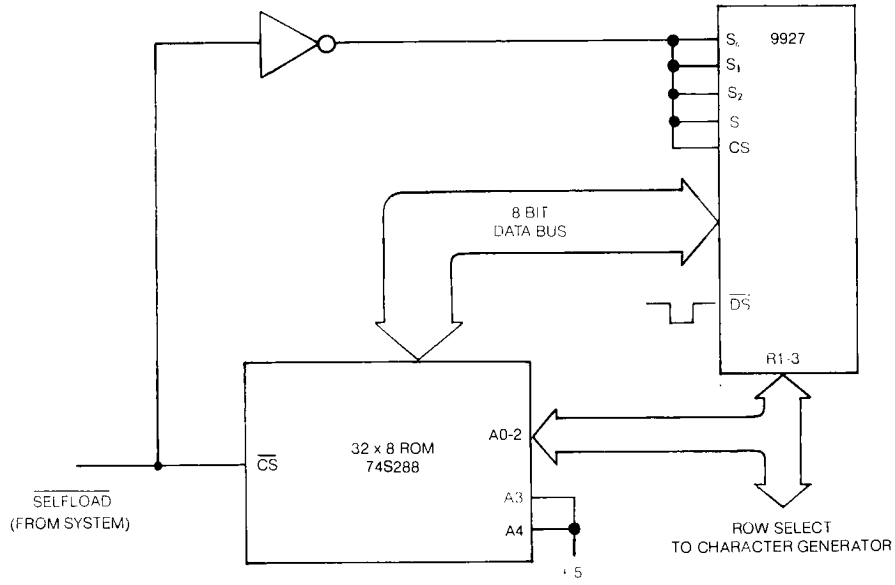


Figure 3. Self-load Command

ARCHITECTURE

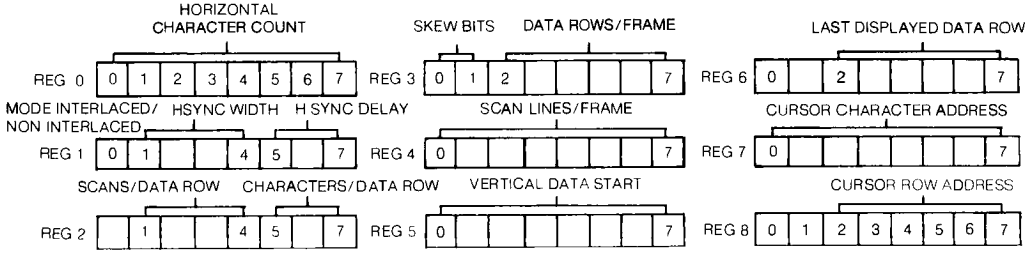
General

The functional block illustrates the architecture of the TMS9927 video timer and controller. The architecture is designed to be as general as possible so that by programming the control registers properly almost any raster scan CRT can be controlled with this chip.

Select Lines

Lines S0-S3 are the select lines. They select the control register for loading via the data bus (DB0-DB7) and also select control functions for the device (see *Table 1*). The bit assignments for the nine control registers are given in *Table 2*. Notice that the cursor line address and the character address can both be read; therefore, the TMS9927 data bus must be bidirectional.

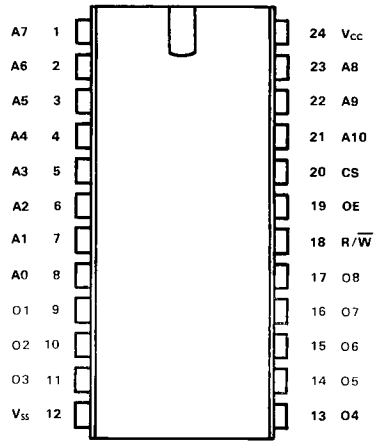
Table 2. TMS 9927 Control Registers.



TMS 9932 JC, NC
COMBINATION ROM/RAM MEMORY

- Single 5 Volt Power Supply, ± 10%
- 15,360 Bits Read Only Memory
- 1024 Bits Random-Access Memory
- All Inputs and Outputs TTL Compatible (No Pull-Up Resistors Needed)
- Maximum Access Time — 400 ns
- Maximum Cycle Time — 400 ns
- Low Power Dissipation of 300 mW (Typical)
- Programmable Chip Select and Output Enable Buffers for Expansion
- N-MOS Silicon-Gate Technology
- Standard 0°C to 70°C Temperature Range

24-Pin Ceramic and Plastic Dual-in-line Packages (Top View)



DESCRIPTION

The TMS 9932 is a 15,360 bit read only memory and a 1024 bit random access memory organized as 1920 words of 8 bit length ROM and 128 words of 8 bit length RAM. The highest 128 addresses will access the RAM, while the lower 1920 addresses access the ROM.

The device is fabricated using N-channel silicon gate technology and is completely static, allowing simple interfacing to bipolar and other MOS circuitry with a minimum system parts count.

All inputs can be driven by 7400 series TTL circuits without the use of any external pull-up resistors. Similarly, each output can drive up to two 7400 series TTL circuits without external resistors.

The data outputs are tri-state for OR tying multiple devices on a common bus. A logical zero on the chip select (CS) or the output enable input (OE) forces the input/output buffers into the high impedance state. Chip select and read/write input (R$\overline{W}$) allow data to be written into the RAM while automatically forcing the I/O buffers into the high impedance state.

The device is supplied in 24 pin dual-in-line plastic and ceramic packages designed for insertion in mounting hole rows on 600-mil centers.

The device is designed for operation over a commercial temperature range from 0°C to 70°C.

OPERATION

Address ($A_0 - A_{10}$)

The 11-bit positive-logic address is decoded on-chip to select one of 2048 words of 8-bit length in the memory array. Addresses 0 to 1919 are ROM addresses; addresses 1920 to 2047 are RAM addresses. A_0 is the least-significant bit and A_{10} is the most-significant bit of the word addresses.

Chip Select (CS)

Chip select can be programmed at the factory at the same time the ROM is programmed to be active with either a high or low input signal. This allows for system expansion to use more than one ROM/RAM circuit. When chip select is disabled, data cannot be written into the RAM and the outputs are in the high-impedance state.

Mode Select (R/$\overline{W}$)

The R/$\overline{W}$ input must be high during read and low during write operations to the RAM. Prior to an address change, R/$\overline{W}$ must be in the read state and must remain in that state for a minimum period to eliminate the possibility of data being written into an unwanted position.

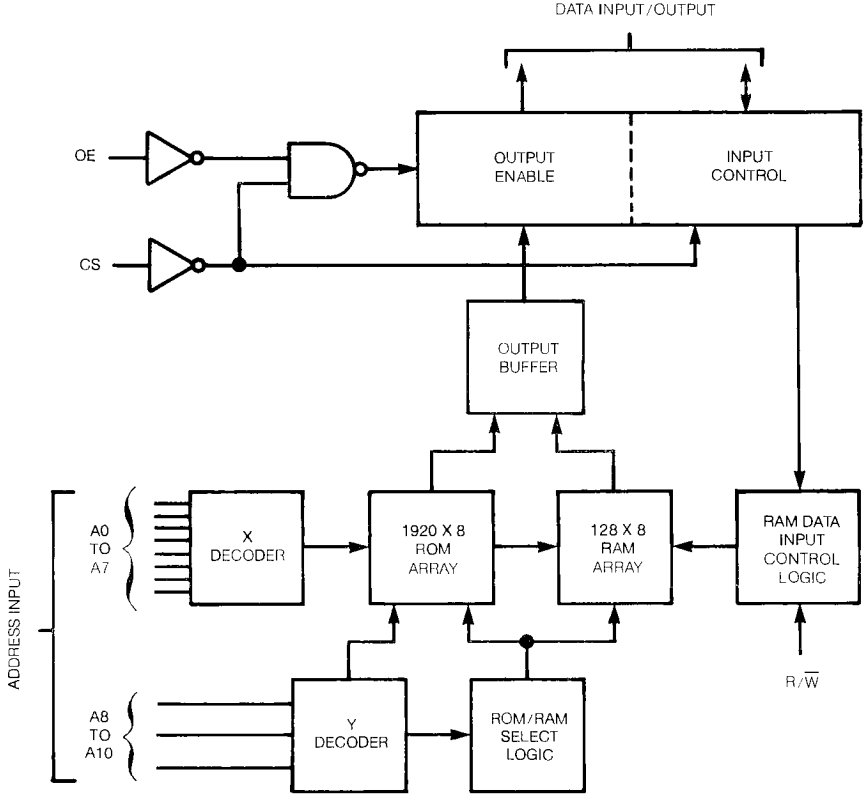
Output Enable (OE)

The output enable input can be programmed, during mask fabrication, to be active with either a high or a low input signal. When output enable is active, all eight outputs are enabled and the eight-bit addressed word can be read. When output enable is not active, all eight outputs are in a high-impedance state.

Data Input/Output (I/O1 – I/O8)

The common input/output terminals are used for both read and write operations. During a write cycle, data must be set up a minimum time before R/$\overline{W}$ goes to the read state (high) to ensure that correct data will enter the addressed memory cell. Also, input data must be held a minimum time after the rise of R/$\overline{W}$.

The output buffers are controlled by output enable and chip select. To read data, output enable and chip select must be valid.



FUNCTION TABLE			
R/$\overline{W}$	CS	OE	OPERATION
L	H	H	Write into RAM (I/O1 to I/O8 = ?)
L	H	L	Write into RAM (I/O1 to I/O8 = Z)
H	H	H	Ream ROM at RAM
X	L	X	Device disabled (I/O1 to I/O8 = Z)
X	X	O	Output disabled (I/O1 to I/O8 = Z)

L = LOW, H = HIGH, X = DON'T CARE, Z = HIGH IMPEDANCE, ? = INDETERMINATE

Functional Block Diagram

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

Supply voltage, V_{CC} (see Note 1) . . . . . −0.5 to 7 V
Input voltage (any input) (see Note 1) . . . . . −0.5 to 7 V
Continuous power dissipation . . . . . 1 W
Operating free-air temperature range . . . . . 0° to 70°C
Storage temperature range . . . . . −55°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.

NOTE 1: Voltage values are with respect to the ground terminal.

Recommended Operating Conditions

PARAMETER	0°C – 70°C			UNIT
	NIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	V
High-level input voltage, V_{IH}	2		V_{CC}	V
Low-level input voltage, V_{IL} (see Note 2)	−0.5		0.8	V
Read cycle time, $t_{c(rd)}$		400		ns
Write cycle time, $t_{c(W)}$		400		ns
Write pulse width, $t_{w(W)}$		250		ns
Address setup time, $t_{su(A)}$		[illegible]		ns
Chip select setup time, $t_{su(CS)}$		[illegible]		ns
Data setup time $t_{su(D)}$		300		ns
Address hold time, $t_{h(A)}$		30		ns
Data hold time, $t_{h(D)}$		30		ns
Operating free-air temperature, T_A	0		70	C

NOTE 2: The algebraic convention where the most negative limit is designated as minimum is used in this data sheet for logic voltage levels only.

Electrical Characteristics Over Recommended Operating Free-Air Temperature Range (Unless Otherwise Noted)

	PARAMETER	TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = −150μA,	V_{CC} = 4.5 V		2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 3.2 mA	V_{CC} = 5.5 V		0.45		V
I_I	Input current	V_I = 0 to 5.25 V			10		μA
I_{OZH}	Off state output current, high-level voltage applied	CS at 2.0 V,	V_O = 4 V		15		μA
I_{OZL}	Off-state output current, low-level voltage applied	CS at 2.0 V,	V_O = 0.45 V		−50		μA
I_{CC}	Supply current from V_{CC}	V_{CC} = 5.5 V, I_O = 0 mA	T_A = 0°C to 70°C		55		mA
			T_A = −40° to 125°C		55		
C_i	Input capacitance	V_I = 0 V, f = 1 MHz	T_A = 25°C		4		pF
C_o	Output capacitance	V_O = 0 V, f = 1 MHz	T_A = 25°C		10		pF

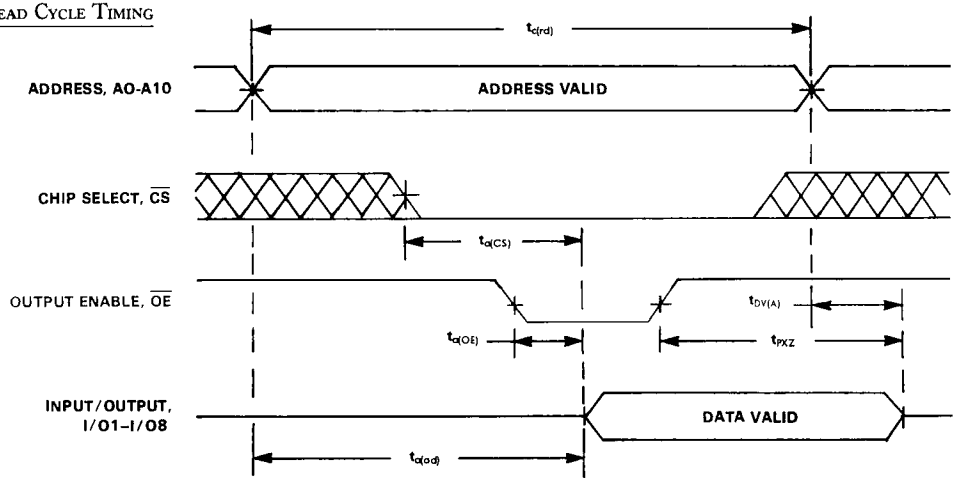
†All typical values are at V_{CC} = 5 V, T_A = 25°C

SWITCHING CHARACTERISTICS OVER RECOMMENDED SUPPLY VOLTAGE RANGE, 2 SERIES 74 TTL LOAD, C_L = 100 pF

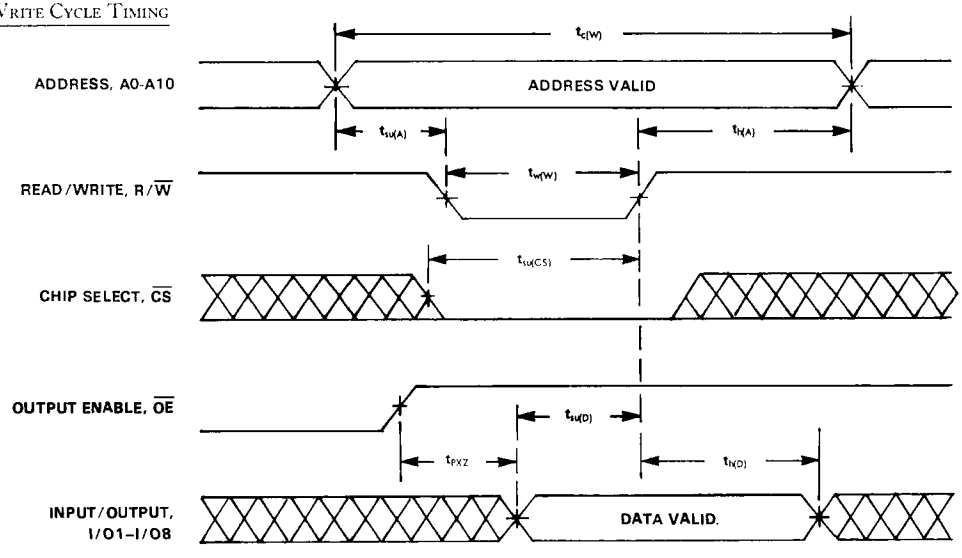
PARAMETER		0° – 70°C MIN	TYP	MAX	UNIT
$t_{a(A)}$	Access time from address			400	ns
$t_{a(CS)}$	Access time from chip select			300	ns
$t_{a(OE)}$	Access time from output enable			100	ns
$t_{DV(A)}$	Previous output data valid after address change			40	ns
t_{pxz}	Output disable time from output enable (see Note 3)			150	ns

NOTE 3: This parameter defines the delay for the I/O bus to enter the input mode

READ CYCLE TIMING



WRITE CYCLE TIMING



NOTE: FOR MEASURING TIMING REQUIREMENTS AND CHARACTERISTICS V_{IH} = 2.0V, V_{IL} = 0.50V, $t_r = t_f$ = 20ns AND ALL TIMING POINTS ARE 50% POINTS.

SOFTWARE PACKAGE

The TMS 9932 is a fixed program memory in which the programming is performed by TI at the factory during the manufacturing cycle to the specific customer inputs supplied in the format shown. The device is organized as 1920 8 bit words with address locations numbered 0 to 1919. Any 8 bit word can be coded as a 2 digit hexadecimal number between 00 and FF. All stored words and addresses in the format are coded in hexadecimal numbers. In coding, all binary words must be in positive logic before conversion to hexadecimal. I/O1 is considered the least significant bit and I/O8 the most significant bit. Addresses input A0 is least significant and A10 is most significant.

Every card should include the TI Customer Device Number in the form MP XXXX-XXX (8 digit number to be assigned by TI) in columns 71 through 80.

Output enable is customer programmable. Every card should include in column 70 a 1 if the output is to be enabled with a high level input at $\overline{OE}$ or a 0 for enabling with a low level input.

The 1920 coded words must be supplied on 120 cards with 16 two digit hex numbers per card.

CARD	COLUMN	
1	1-9	BLANK
	10	:(ASCII character colon)
	11-12	10 (specifies 16 words per card)
	13	BLANK
	14-16	Hex address of 1st word on 1st card (0th word, address normally 000)
	17-18	BLANK
	19-20	0th word in Hex
	•	
	•	
	•	
	49-50	15th word in Hex
	51-69	BLANK
	70	Output Enable ($\overline{OE}$) Active State
	71-80	Customer Device Number
CARD	COLUMN	HEXADECIMAL INFORMATION
120	1-9	BLANK
	10	:(ASCII character colon)
	11-12	10
	13	BLANK
	14-16	Hex address of 1st word on 120th card (1904th word, address normally 770)
	17-18	BLANK
	19-20	1904th word in Hex
	•	
	•	
	•	
	49-50	1919th word in Hex
	51-69	BLANK
	70	Output Enable ($\overline{OE}$) Active State
	71-80	Customer Device Number

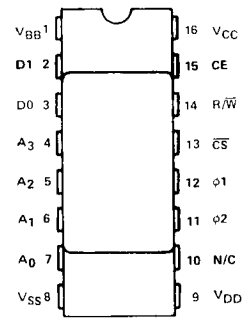
TMS 3064 JL
65,536-BIT CCD MEMORY

- 65,536 x 1 Organization
  (16 Addressable 4096-Bit Loops)
- Performance:

LATENCY TIME AT 5 MHz (MAX)	READ OR WRITE CYCLE (MIN)	READ, MODIFY WRITE CYCLE (MIN)
820 μs	200 ns	300 ns

- Full TTL Compatibility (No Pull-up Resistors Required) on All Inputs Except $\phi 1$, $\phi 2$, and Chip Enable
- Low Power Dissipation:
  280 mW Operating (Typical @ 5 MHz)
  25 mW Recirculating (Typical @ 1 MHz)
  <1 mW Standby (Typical)
- Two-Phase CCD Clocks
- N-Channel Silicon-Gate Technology
- 16-Pin, 400-Mil Dual-in-Line Package

16-PIN CERAMIC
DUAL-IN-LINE PACKAGE

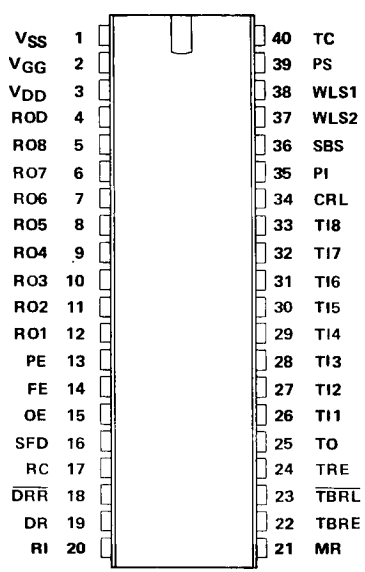


TMS 6011 JC, NC
ASYNCHRONOUS DATA INTERFACE (UART)

MOS
LSI

- Transmits, Receives, and Formats Data
- Full-Duplex or Half-Duplex Operation
- Operation from DC to 200 kHz
- Static Logic
- Buffered Parallel Inputs and Outputs
- Programmable Word Lengths . . . 5, 6, 7, 8 Bits
- Programmable Information Rate
- Programmable Parity Generation/Verification
- Programmable Parity Inhibit
- Automatic Data Formatting
- Automatic Status Generation
- 3-State Push-Pull Buffers
- Low-Threshold Technology
- Standard Power Supplies . . . 5 V, −12 V
- Full TTL Compatibility . . . No External Components

40-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



8

990/9900 FAMILY MICROCOMPUTER COMPONENTS

- SBP/TMS 9900 Series Microprocessor Family Peripheral
- 16 Individual, Single-Bit, Software Configurable I/O Ports
- 20/40 mA Current Sinking Outputs
- 28-Pin Package
- Software Compatible with TMS 9901 when used in conjunction with SBP 9961
- TTL Compatible I/O
- Wide Ambient Temperature Operation
  - SBP 9960CJ: 0°C to +70°C
  - SBP 9960EJ: -40°C to +85°C
  - SBP 9960MJ: -55°C to +125°C
  - SBP 9960NJ: -55°C to +125°C (with high-reliability processing)
- I^2L Technology
  - Constant Current Power Source
  - Fully Static Operation
  - Single Phase Edge-Triggering Clock
  - Wide Temperature Stability

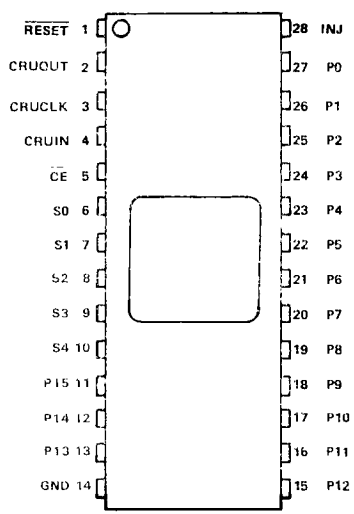
DESCRIPTION

The SBP 9960 CRU I/O Expander is a ruggedized monolithic software-configuration input/output device fabricated with oxide separated Integrated Injection Logic (I^2L) technology. The SBP 9960 provides a flexible and efficient Communications Register Unit (CRU) based interface between the SBP/TMS 9900 series Family of Microprocessors and auxiliary systems functions ranging from bit-oriented sensors and actuatore to byte/word/n-bit-field oriented peripherals.

Under software control, each of the SBP 9960s sixteen individual single-bit I/O ports may be configured to either the input or output mode. I^2L technology enables the SBP 9960s static logic, and TTL compatible I/O, to operate over a very wide ambient temperature range from a single d-c power source with output current sink capability up to 40 mA. When the SBP 9960 is used in conjunction with the SBP 9961 I^2L Interrupt-Controller/Timer, the SBP 9960/SBP 9961 pair form an I^2L systems alternate to the N-channel MOS TMS 9901 Programmable Systems Interface device while maintaining strict compatibility with existing software handlers developed in support of the TMS 9901.

SBP 9960 PIN ASSIGNMENTS AND FUNCTIONS

Signature	*Pin*	*I/O*	*Description*
S0	6	IN	ADDRESS SELECT LINES. The data bit being accessed by the CRU interface is specified by the 5-bit code appearing on S0-S4.
S1	7	IN	
S2	8	IN	
S3	9	IN	
S4	10	IN	
CRUIN	4	OUT	CRU DATA IN (to CPU). Data specified by S0-S4 is transmitted to the CPU by CRUIN. When $\overline{CE}$ is not active, CRUIN is pulled to logic-level high.
CRUOUT	2	IN	CRU DATA OUT (from CPU). When $\overline{CE}$ is active data present on the CRUOUT input will be strobed by CRUCLK and written into the CRU bit specified by S0-S4.
CRUCLK	3	IN	CRU CLOCK (from CPU). CRUCLK specifies that valid data is present on the CRUOUT line.
$\overline{RESET}$	1	IN	POWER-UP RESET. When active (low), $\overline{RESET}$ forces all I/O's (P0-P15) to input mode.
$\overline{CE}$	5	IN	CHIP ENABLE. When active (low), data may be bidirectionally transferred between the SBP 9960 and the CPU.
INJ	28		Supply Current
GND	14		Ground Reference
P0	27	I/O	I/O pins
P1	26	I/O	
P2	25	I/O	
P3	24	I/O	
P4	23	I/O	
P5	22	I/O	
P6	21	I/O	
P7	20	I/O	
P8	19	I/O	
P9	18	I/O	
P10	17	I/O	
P11	16	I/O	
P12	15	I/O	
P13	13	I/O	
P14	12	I/O	
P15	11	I/O	



FUNCTIONAL DESCRIPTION

SBP 9960/CPU INTERFACE

The SBP 9960 communicates with the CPU through the Communications Register Unit (CRU) interface as shown in Figures 1 and 3. The SBP 9960s CRU interface consists of: a) five CRU address select lines (S0-S4), b) a single chip enable ($\overline{CE}$), c) a 9960 to CPU serial data-bit line (CRUIN), d) a CPU to 9960 serial data-bit line (CRUOUT), and e) a CPU to 9960 serial data-bit clock (CRUCLK). When $\overline{CE}$ is activated (logic level low), S0-S4 select a specific single-bit I/O port as indicated in Table 1. In the case of an SBP 9960 write operation, the datum is transferred from the CPU to the SBP 9960 via the CRUOUT line. The CRUOUT datum is strobed into the selected single-bit port by CRUCLK. In the case of a SBP 9960 read operation, the selected single-bit port is sampled by the CPU via the CRUIN line.

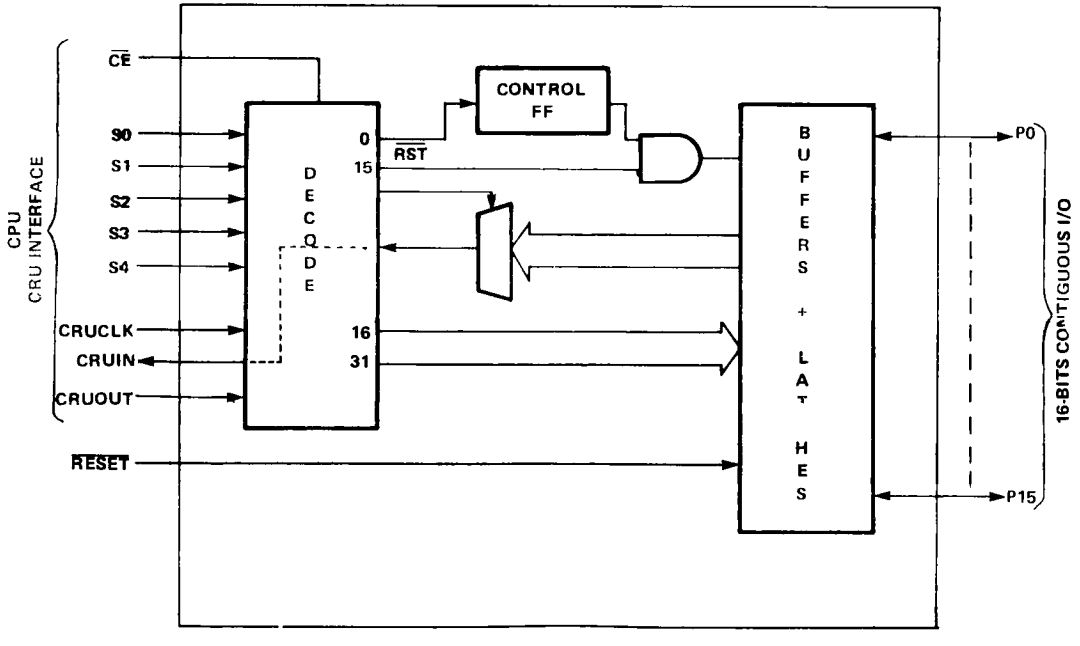


Figure 1.

CRU Bit Assignments

Table 1 describes the SBP 9960's CRU bit assignments. Note that CRU bits 1-14 have been reserved for the SBP 9961 thereby insuring software compatibility between the SBP 9960/SBP9961 pair and the TMS 9901.

Input/Output

One of sixteen SBP 9960, single-bit, combination open-collector-output/resister-divider-input I/O ports is conversationally represented in Figure 2. As a direct result of the open-collector output structure, the data flow direction through the port is determined by the stored logic-level of the associated output-register bit in combination with the data flow direction of the external device serviced by the port. When the ouput-register bit (Q) is at logic-level high, the corresponding package pin (P) is essentially floating and therefore free to be externally pulled to either the high or low logic-level. In other words, when Q is at logic-level high, the ports data flow direction can be either inward, where an external device pulls P to the high or low logic-level; or the data flow direction can be outward, where an external resistor (R) both pulls P to logic-level high and sources current drive into the inputs of external devices. When Q is at logic-level low, the ports unconditional data flow direction is outward, where P has the capacity to sink 20/40*mA of current from external devices. Q can be reset to logic-level low through CPU execution of a SET BIT TO ZERO (SBZ) instruction; Q can be set to logic-level high through: 1) a hardware initiated reset ($\overline{\text{RESET}}$), 2) a software initiated reset ($\overline{\text{RST}}$: CRU BIT 15) preceded by setting the control (CRU BIT 0) to logic-level high, or 3) CPU execution of a SET BIT TO ONE (SBO) instruction. Note that both $\overline{\text{RESET}}$ and RST affect all sixteen single-bit I/O ports while CPU execution of either an SBO or SBZ instruction can be targeted at an individual single-bit port independent of uninvolved ports. Once the data flow direction has been established for each single-bit port, CPU communication with the external devices driven or sensed by each individual port is effected through execution of the CRU instructions: LDCR, STCR, SBO, SBZ, and TM.

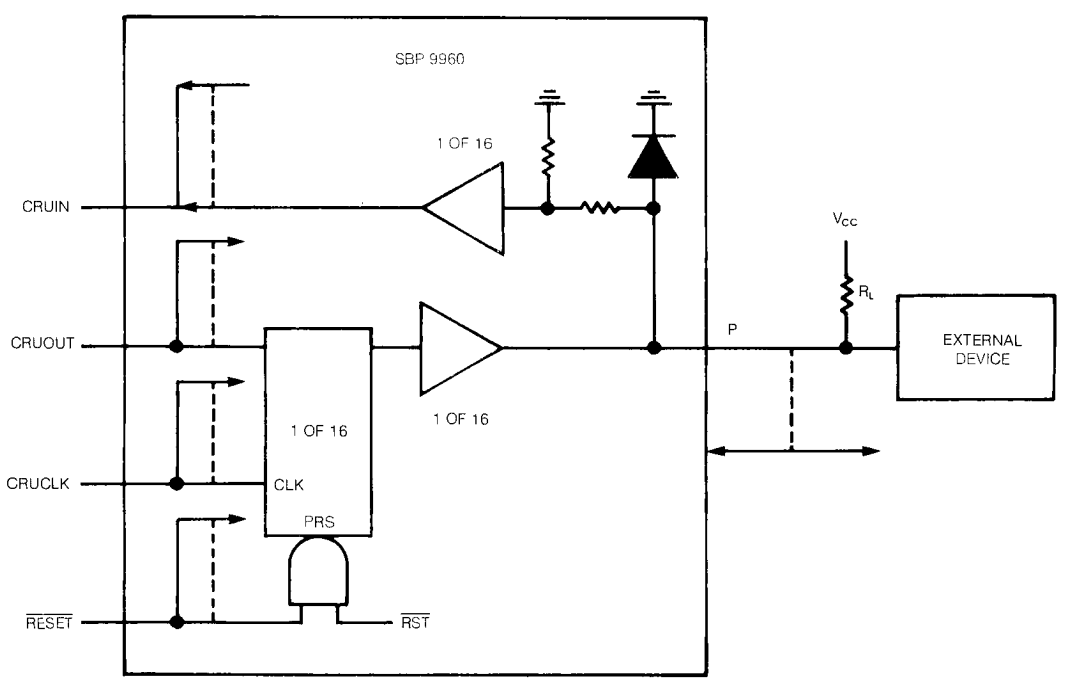


***FIGURE 2.** 1 of 16 Single-Bit I/O Ports*

*Outputs P0, P1, P2, and P3 have extended current sink capability to 40 mA

Table 1. SBP 9960 CRU Bit Assignments

CRU BIT	S0	S1	S2	S3	S4	CRU READ DATA	CRU WRITE DATA
0	0	0	0	0	0	Control Bit	Control Bit
1-14						Note 1	Note 1
15	0	1	1	1	1	"1"	No Operation/ $\overline{RST}$ (2)
16	1	0	0	0	0	P0 Input(3)	P0 Output(4) (5)
17	1	0	0	0	1	P1	P1 (5)
18	1	0	0	1	0	P2	P2 (5)
19	1	0	0	1	1	P3	P3 (5)
20	1	0	1	0	0	P4	P4
21	1	0	1	0	1	P5	P5
22	1	0	1	1	0	P6	P6
23	1	0	1	1	1	P7	P7
24	1	1	0	0	0	P8	P8
25	1	1	0	0	1	P9	P9
26	1	1	0	1	0	P10	P10
27	1	1	0	1	1	P11	P11
28	1	1	1	0	0	P12	P12
29	1	1	1	0	1	P13	P13
30	1	1	1	1	0	P14	P14
31	1	1	1	1	1	P15 Input(3)	P15 Output(4)

NOTES: (1) Bits 1-14 reserved for SBP 9961 Interval Timer/Interrupt Controller
(2) Writing a zero to bit 15 while CONTROL = 1 executes a software reset of the I/O ports.
(3) Data present on the port will be read without affecting the data.
(4) Writing data to the port will both program the port to the output mode and output the data.
(5) These outputs are provided with extended sink current capability to 40 mA.

System Operation

During a typical power-up sequence of a SBP 9960-based system, $\overline{RESET}$ should be activated (logic-level low) to force the SBP 9960 to the state where each of the sixteen individual single-bit I/O ports is in the input mode. System software should then configure each single-bit port as required. If a given port must be reconfigured from the input to output mode after power-up, the associated output-register bit must be set to logic-level high through CPU execution of an SBO instruction.

▶8

SBP 9960/SBP 9961 EMULATION OF THE TMS 9901

Figure 3 shows the system configuration of a SBP 9960 functioning in conjunction with a SBP 9961 in emulation of a TMS 9901. Note the common connection of: a) the individual chip enables, and b) the CRU interface lines. For a complete description of the SBP 9961 and the TMS 9901 refer respectively to the *SBP 9961 Interrupt-Controller/Timer Data Manual* and the *TMS 9901 Programmable Systems Interface Data Manual.*

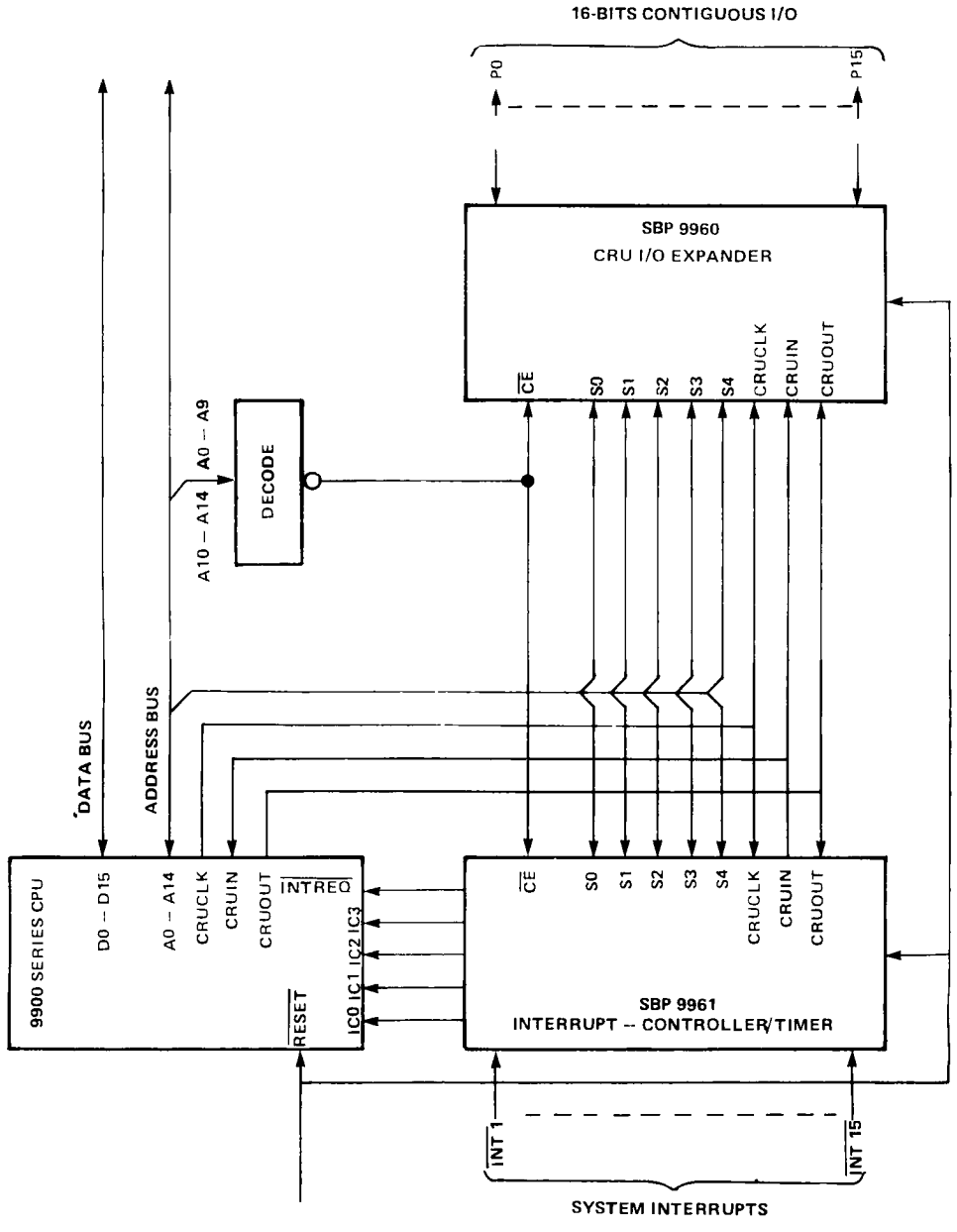


Figure 3. SBP 9960/SBP9961 System Configuration

ELECTRICAL SPECIFICATIONS

Recommended Operating Conditions, Unless Otherwise Noted $I_{cc} = 70$ mA

PARAMETER		MIN	NOM	MAX	UNIT
Supply current, I_{CC}		63	70	77	mA
High-level output voltage, V_{OH}				5.5	V
Low-level output current, I_{OL}				20†	mA
Operating free-air temperature, T_A	SBP 9960MJ, SBP 9960NJ	−55		125	°C
	SBP 9960EJ	−40		85	
	SBP 9960CJ	0		70	

†40 mA on extended drive outputs P0, P1, P2, and P3

Electrical Characteristics (Over Recommended Operating Free-Air Temperature Range, Unless Otherwise Noted)

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_{IK}	Input clamp voltage	I_{CC} = MIN,	I_I = −12 mA			−1.5	V
I_{OH}	High-level output current	I_{CC} = 70 mA, V_{IL} = 0.8 V,	V_{IH} = 2 V, V_{OH} = 5.5 V			400	μA
V_{OL}	Low-level output voltage	I_{CC} = 70 mA, V_{IL} = 0.8 V,	V_{IH} = 2 V, I_{OL} = 20 mA (40 mA§)			0.5	V
I_I	Input current	I_{CC} = 70 mA,	V_I = 2.4 V		225		μA

†For conditions shown as MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at I_{CC} = 70 mA, T_A = 25°C.
§Extended drive outputs only.

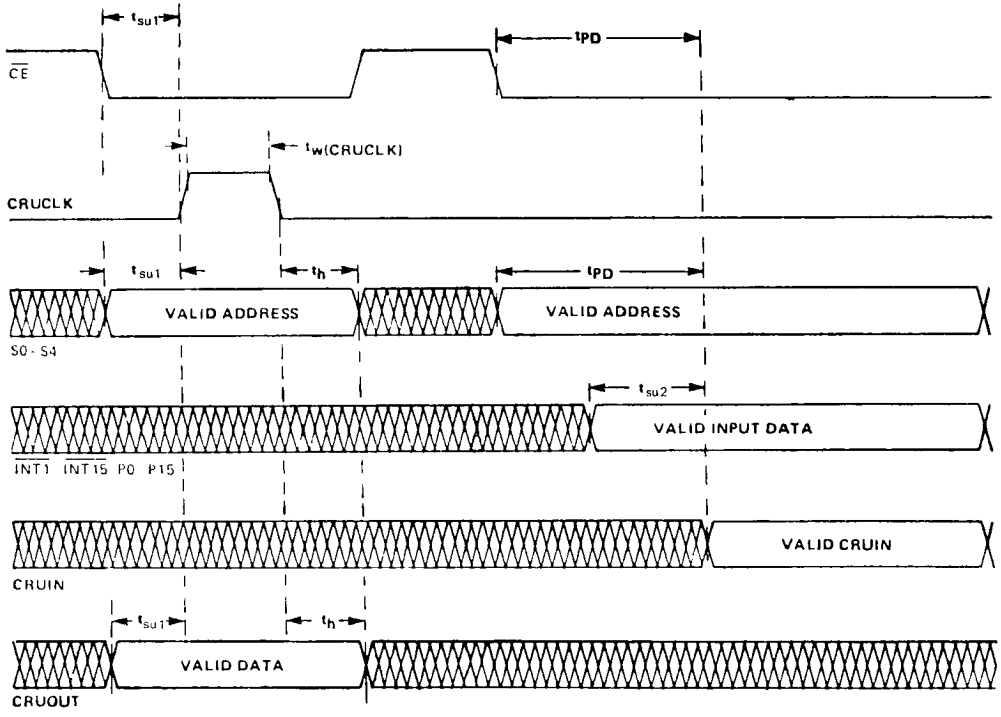
Timing Requirements Over Full Range of Operating Conditions

PARAMETER		MIN	NOM	MAX	UNIT
t_{su1}	Setup time for S0-S4, $\overline{CE}$, or CRUOUT before CRUCLK		200		ns
t_{su2}	Setup time, input before valid CRUIN		200		ns
$t_{w(CRUCLK)}$	CRU clock pulse width		100		ns
t_h	Hold time for Address or Data		0		ns

Switching Characteristics Over Full Range of Recommended Operating Conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PD}	Propagation delay, S0-S4 or $\overline{CE}$ to valid CRUIN	C_L = 100 pF, R_L = 300 Ω		300		ns

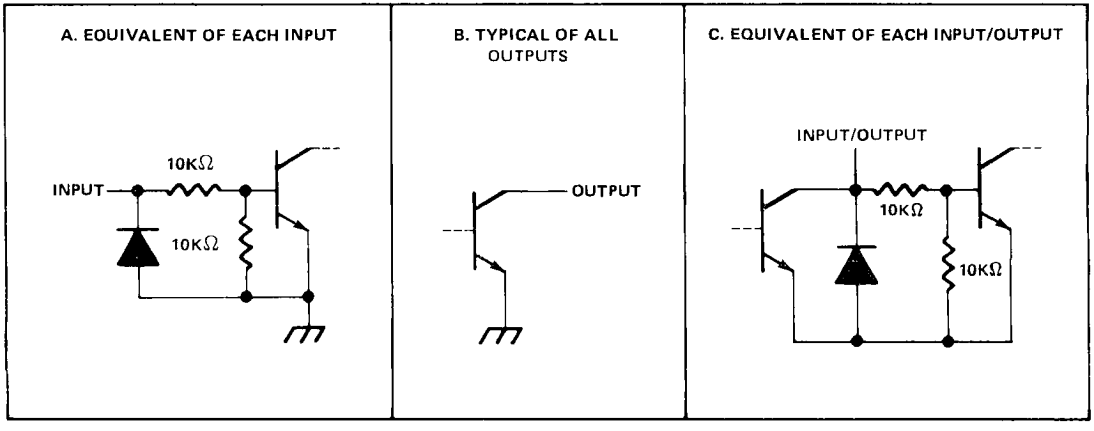
8



NOTE 1: ALL TIMING MEASUREMENTS ARE FROM 10% and 90% POINTS

SWITCHING CHARACTERISTICS

INPUT, OUTPUT, AND INPUT/OUTPUT STRUCTURES



8

990/9900 FAMILY MICROCOMPUTER COMPONENTS

- SBP/TMS 9900 Series Microprocessor Family Peripheral
- 15 Dedicated, Maskable, Prioritized, Encoded Interrupts
- 20 mA Current Sinking Outputs
- 40-Pin Package
- Independently Clocked 14-Bit Interval/Event Timer
- Software Compatible with TMS 9901 when used in conjunction with SBP 9960
- TTL Compatible I/O
- Wide Ambient Temperature Operation
  - SBP 9961CJ: 0°C to +70°C
  - SBP 9961EJ: -40°C to +85°C
  - SBP 9961MJ: -55°C to +125°C
  - SBP 9961NJ: -55°C to +125°C (with high-reliability processing)
- I^2L Technology
  - Constant Current Power Source
  - Fully Static Operation
  - Single Phase Edge-Triggering Clock
  - Wide Temperature Stability

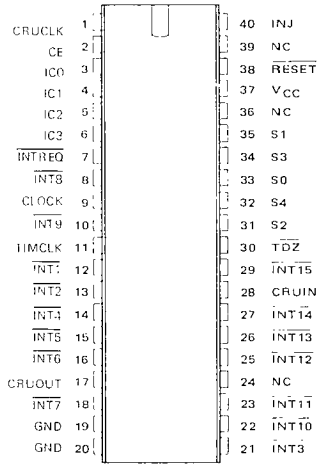
DESCRIPTION

The SBP 9961 Interrupt-Controller/Timer is a ruggedized, monothlithic, Communications Register Unit (CRU) programmable, multifunction systems support device fabricated with oxide separated Integrated Injection Logic (I^2L) technology. The SBP 9961 provides the SBP/TMS 9900 series Family of Microprocessors with a flexible independently clocked interval/event timer plus maskable prioritized interrupt encoding capability. I^2L technology enables the SBP 9961s static logic, and TTL compatible I/O, to operate over a very wide ambient temperature range from a single d-c power source. When the SBP 9961 is used in conjunction with the I^2L SBP 9960 CRU I/O Expander, the SBP 9961/SBP 9960 pair form an I^2L systems alternate to the N-channel MOS TMS 9901 Programmable Systems Interface device while maintaining strict compatibility with existing software handlers developed in support of the TMS 9901.

SBP 9961 PIN ASSIGNMENTS AND FUNCTIONS

Signature	*Pin*	*I/O*	*Description*
S0	33	IN	ADDRESS SELECT LINES. The data bit being accessed by the CRU interface is specified by the 4-bit code appearing on S1-S4. S0 is used as the high order select line when the SBP 9961 is used with the SBP 9960 in emulation of the TMS 9901. Otherwise, tie S0 to logic-level low.
S1	35	IN	
S2	31	IN	
S3	34	IN	
S4	32	IN	
CRUIN	28	OUT	CRU DATA IN (to CPU). Data specified by S0-S4 is transmitted to the CPU by CRUIN. When $\overline{CE}$ is not active, CRUIN is logic-level high.
CRUOUT	17	IN	CRU DATA OUT (from CPU). When $\overline{CE}$ is active, data present on the CRUOUT input will be sampled during CRUCLK and written into the CRU bit specified by S0-S4.
CRUCLK	1	IN	CRU CLOCK (from CPU). CRUCLK specifies that valid data is present on the CRUOUT line.
$\overline{RESET}$	38	IN	POWER-UP RESET. When active (low), $\overline{RESET}$ forces all interrupt masks to "0", and disables the clock.
$\overline{CE}$	2	IN	CHIP ENABLE. When active (low), data transfers may occur between the CPU and the SBP 9961.

SBP 9961
INTERRUPT-CONTROLLER/TIMER



Signature	*Pin*	*I/O*	*Description*
TIMCLK	11	IN	TIMER CLOCK IN. External clock used for the timer decrementer. May be externally tied to the CLOCK input pin.
TDZ	30	OUT	TIMER DECREMENTER EQUALS ZERO. Low active pulse indicating that the timers decrementer contains a value of zero (all logic-level lows).
IC0	3	OUT	INTERRUPT CODE LINES (to CPU). IC0 (MSB) through IC3 output the binary code corresponding to the highest priority enabled interrupt most recently asserted.
IC1	4	OUT	
IC2	5	OUT	
IC3	6	OUT	
INTREQ	7	OUT	INTERRUPT REQUEST (to CPU). When active (low), INTREQ indicates to the CPU that an enabled interrupt has been asserted, prioritized and encoded.
CLOCK	9	IN	CPU SYSTEM CLOCK. Used by the SBP 9961 to synchronize the interrupt interface (INTREQ, IC0-IC3) to the CPU.
INJ	40		Supply Current
GND	19,20		Ground Reference
Vcc	37		Common voltage return/reference for all I/O pull-up resistors.
INT1	12	IN	INTERRUPT INPUTS. When active (low), the signal is ANDed with its corresponding mask bit and if enabled sent to the interrupt control section. INT1 has highest priority.
INT2	13	IN	
INT3	21	IN	
INT4	14	IN	
INT5	15	IN	
INT6	16	IN	
INT7	18	IN	
INT8	8	IN	
INT9	10	IN	
INT10	22	IN	
INT11	23	IN	
INT12	25	IN	
INT13	26	IN	
INT14	27	IN	
INT15	29	IN	

FUNCTIONAL DESCRIPTION

SBP 9961/CPU INTERFACE

The SBP 9961 communicates with the CPU through the Communications Register Unit (CRU) interface as shown in Figures 1 and 4. The SBP 9961s CRU interface consists of: a) five CRU address select lines (S0-S4), b) a single chip enable ($\overline{CE}$), c) a 9961 to CPU serial data-bit line (CRUIN), d) a CPU to 9961 serial data-bit line (CRUOUT), and e) a CPU to 9961 serial data-bit clock (CRUCLK). When $\overline{CE}$ is activated (logic-level low), S0-S4 selects a specific CRU-bit function as indicated in Table 1. In the case of a SBP 9961 write operation, the datum is transferred from the CPU to the SBP 9961 via the CRUOUT line. The CRUOUT datum is strobed into the selected 9961 CRU-bit function by CRUCLK. In the case of a SBP 9961 read operation, the selected CRU-bit function is sampled by the CPU via the CRUIN line.

Table 1. CRU Bit Assignments

CRU BIT	S0	S1	S2	S3	S4	CRU READ DATA	CRU WRITE DATA
0	0(4)	0	0	0	0	Control Bit	Control Bit(1)
1	0	0	0	0	1	$\overline{INT1}$/TIM1(2)	Mask1/TIM1(3)
2	0	0	0	1	0	$\overline{INT2}$/TIM2	Mask2/TIM2
3	0	0	0	1	1	$\overline{INT3}$/TIM3	Mask3/TIM3
4	0	0	1	0	0	$\overline{INT4}$/TIM4	Mask4/TIM4
5	0	0	1	0	1	$\overline{INT5}$/TIM5	Mask5/TIM5
6	0	0	1	1	0	$\overline{INT6}$/TIM6	Mask6/TIM6
7	0	0	1	1	1	$\overline{INT7}$/TIM7	Mask7/TIM7
8	0	1	0	0	0	$\overline{INT8}$/TIM8	Mask8/TIM8
9	0	1	0	0	1	$\overline{INT9}$/TIM9	Mask9/TIM9
10	0	1	0	1	0	$\overline{INT10}$/TIM10	Mask10/TIM10
11	0	1	0	1	1	$\overline{INT11}$/TIM11	Mask11/TIM11
12	0	1	1	0	0	$\overline{INT12}$/TIM12	Mask12/TIM12
13	0	1	1	0	1	$\overline{INT13}$/TIM13	Mask13/TIM13
14	0	1	1	1	1	$\overline{INT14}$/TIM14	Mask14/TIM14
15	0	1	1	1	1	$\overline{INT15}$/$\overline{INTREQ}$	Mask 15

NOTES:
(1) 0 = Interrupt Mode; 1 = TIMCLK Mode.
(2) Data present on $\overline{INT}$ input (or timer value) will be read regardless of mask value.
(3) While in the Interrupt Mode (Control Bit = 0), writing a "1" into a mask will enable interrupt, "0" will disable.
(4) When the SBP 9961/SBP 9960 pair are used in emulation of the TMS 9901, S0 is used to distinguish between activation of the 9961 (S0 = 0) v.s. the 9960 (S0 = 1).

INTERRUPT CONTROL

A block diagram of the SBP 9961s interrupt control section is shown in Figure 2. The interrupt inputs are sampled on the positive-going edge of CLOCK and are ANDed with their respective mask bits. If an interrupt input is active (low) and enabled (MASK = 1), the signal is passed through the priority encoder where the highest priority signal is encoded into a 4-bit binary word as shown in Table 2. This word, along with an interrupt request, is then output to the CPU on the positive-going edge of the next CLOCK.

The output signals will remain valid until either the corresponding interrupt input is removed, the interrupt is disabled (MASK = 0), or a higher priority enabled interrupt becomes active. When the highest priority enabled interrupt is removed, the code corresponding to the next highest priority enabled interrupt is output. If no enabled interrupt is active, $\overline{INTREQ}$ will be pulled to logic-level high with IC0-IC3 retaining the last asserted interrupt code. $\overline{RESET}$ (power-up reset) will force the interrupt code IC0-IC3 to (0,0,0,0) with $\overline{INTREQ}$ pulled high, and will reset all mask bits low (interrupts disabled). Individual interrupts can be subsequently enabled (disabled) by programming the appropriate mask bits. Unused interrupt inputs may be used as data inputs by disabling the interrupt (MASK = 0).

SBP 9961
INTERRUPT-CONTROLLER/TIMER

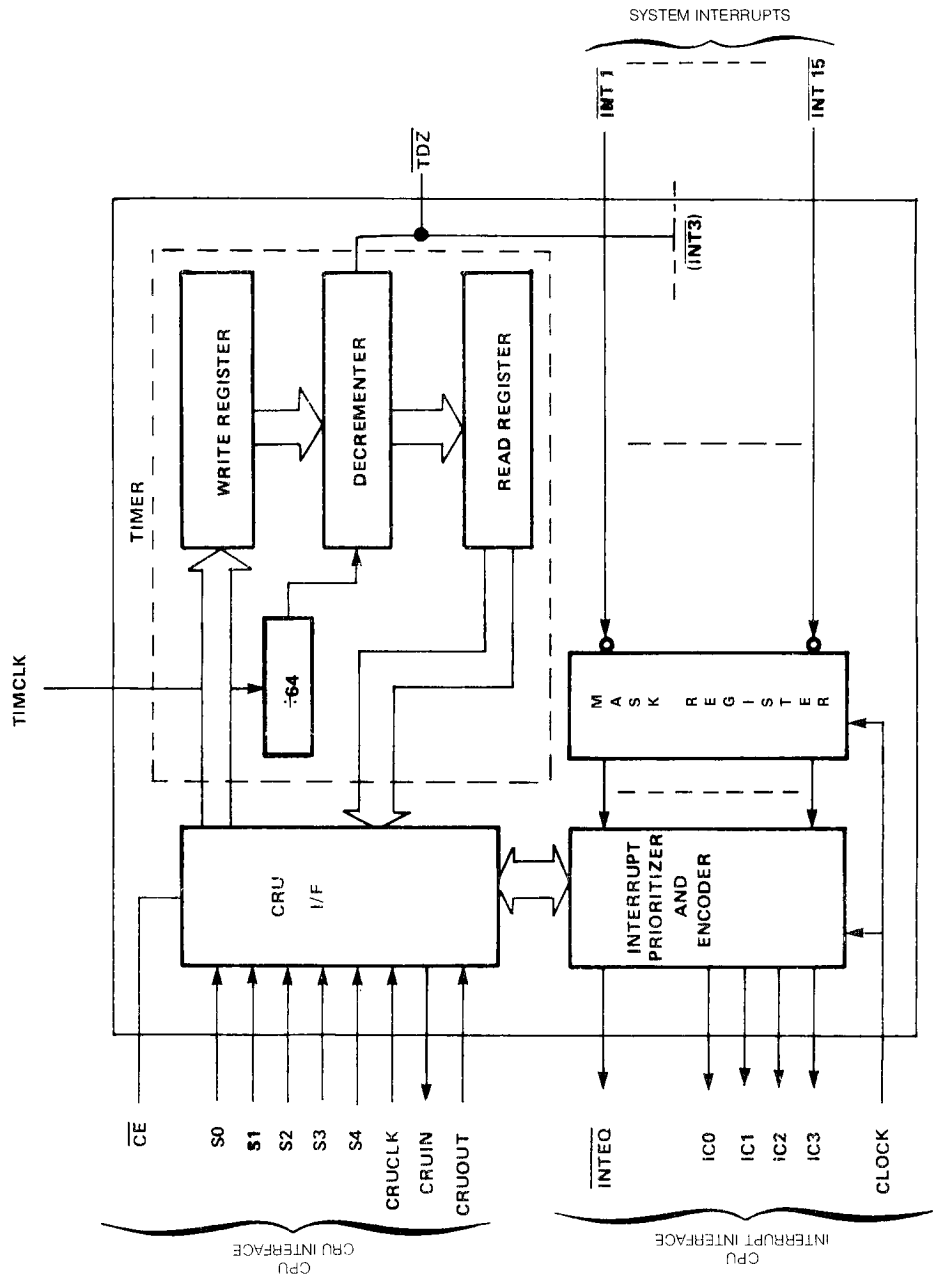


Figure 1. SBP 9961 Block Diagram

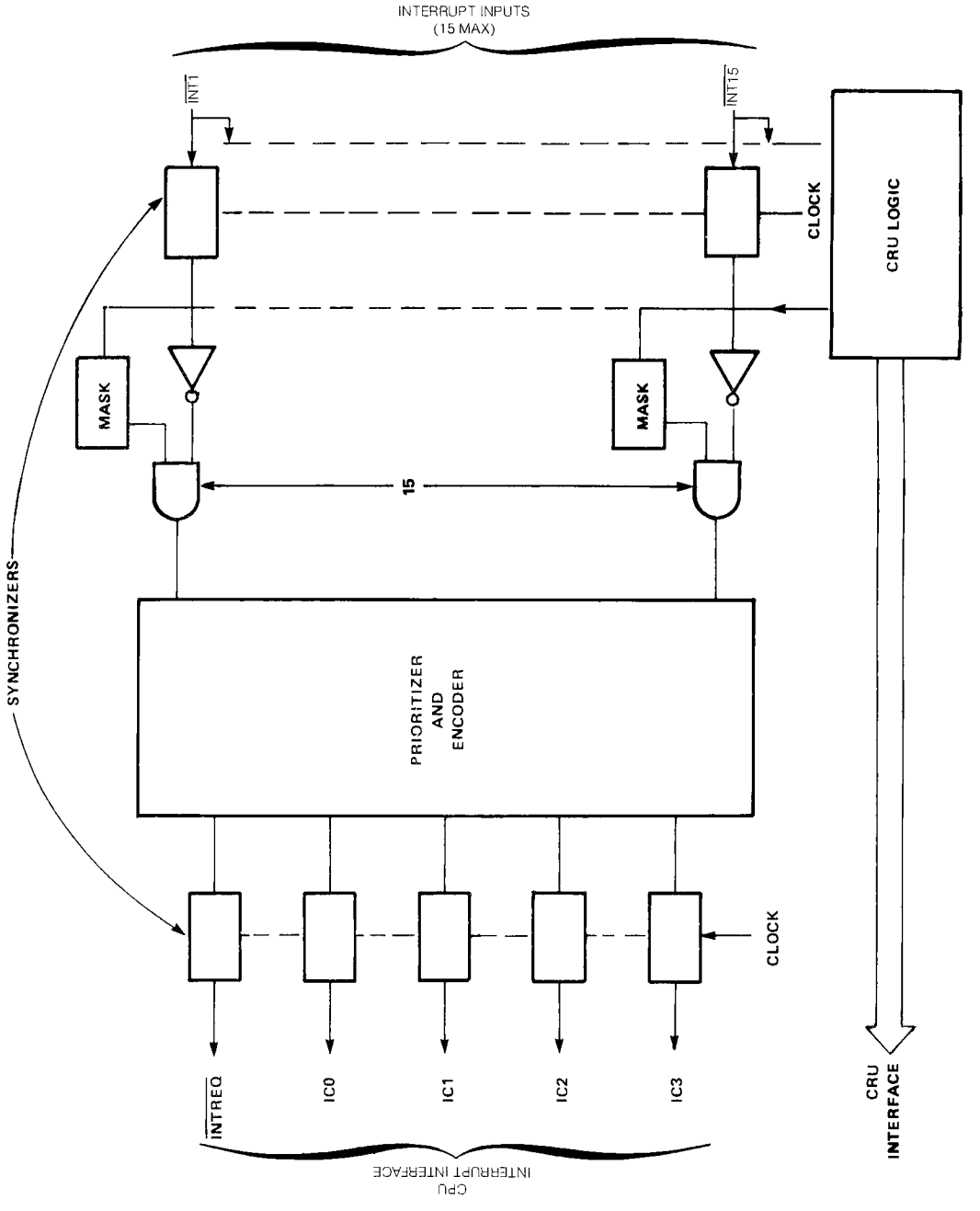


Figure 2. Interrupt Control Logic

Table 2. Interrupt Code Generation

INTERRUPT/STATE	PRIORITY	I_{C0}	I_{C1}	I_{C2}	I_{C3}	[illegible]
$\overline{INT}$ 1	1 (HIGHEST)	0	0	0	1	0
$\overline{INT}$ 2	2	0	0	1	0	0
$\overline{INT}$ 3/TIMER	3	0	0	1	1	0
$\overline{INT}$ 4	4	0	1	0	0	0
$\overline{INT}$ 5	5	0	1	0	1	0
$\overline{INT}$ 6	6	0	1	1	0	0
$\overline{INT}$ 7	7	0	1	1	1	0
$\overline{INT}$ 8	8	1	0	0	0	0
$\overline{INT}$ 9	9	1	0	0	1	0
$\overline{INT}$ 10	10	1	0	1	0	0
$\overline{INT}$ 11	11	1	0	1	1	0
$\overline{INT}$ 12	12	1	1	0	0	0
$\overline{INT}$ 13	13	1	1	0	1	0
$\overline{INT}$ 14	14	1	1	1	0	0
$\overline{INT}$ 15	15 (LOWEST)	1	1	1	1	0
NO INTERRUPT	—	Note 1	Note 1	Note 1	Note 1	1

(1) IC0-IC3 hold the level code of the previous interrupt

Interval Timer

The SBP 9961s interval/event timer, shown in Figure 3, has the following operational features:

a) Independent clock input TIMCLK
b) Programmable 14-bit decrementer
c) Time-reaches-zero level-3 interrupt
d) Timer reaches zero output pulse $\overline{TDZ}$
e) Able to read the current decremented value and therefore fun[illegible]s an event timer
f) Able to determine the SBP 9961s operating mode and value of I[illegible]$\overline{REQ}$.

The SBP 9961 has an independent timer clock input, TIMCLK, which allows the user to define an interval timer clock frequency other than that of the CPU. This, however, does not preclude the user option of connecting TIMCLK to the CPUs CLOCK input and therefore running the interval timer at the CPUs clock frequency. The typical operating range of TIMCLK is 0-5 MHz.

The timer's CRU control bits are shown in Table 1. The SBP 9961 is placed into the timer-access mode by writing a logic-level high to the control bit located at CRU address zero. CRU bits 1-14 are then used to initiate the write-register with the desired start count. Writing a non-zero value to the write-register a) enables the decrementer, b) programs the third [illegible]ity interrupt ($\overline{INT3}$) as the timer interrupt, and c) disables the influence of external interrupts on the I[illegible]nput pin. A single LDCR instruction can be used to accomplish the above initialization operation. After the write-register has been initialized with the desired start count, the timer begins decrementing toward zero. Upon reaching zero, the timer issues the level-3 interrupt, outputs the timer-zero pulse $\overline{TDZ}$, and restarts itself with the write-register value. Since the timer interrupt is latched, clearing that interrupt is accomplished by writing either a logic-level low or high to the respective interrupt mask bit at CRU address three. The CRUCLK that accompanies that write operation is the stimulus which resets the timer's interrupt latch. However, in order to retain the current mask value, the appropriate SBZ or SBO CRU-write instruction must be executed unless the mask value itself is to be changed. At any point in the timer's decrement sequence, a timer restart can be accomplished by either reinitiating the entire write-register with an LDCR instruction, or by writing to any individual write-register bit with an SBZ or SBO instruction.

8◀

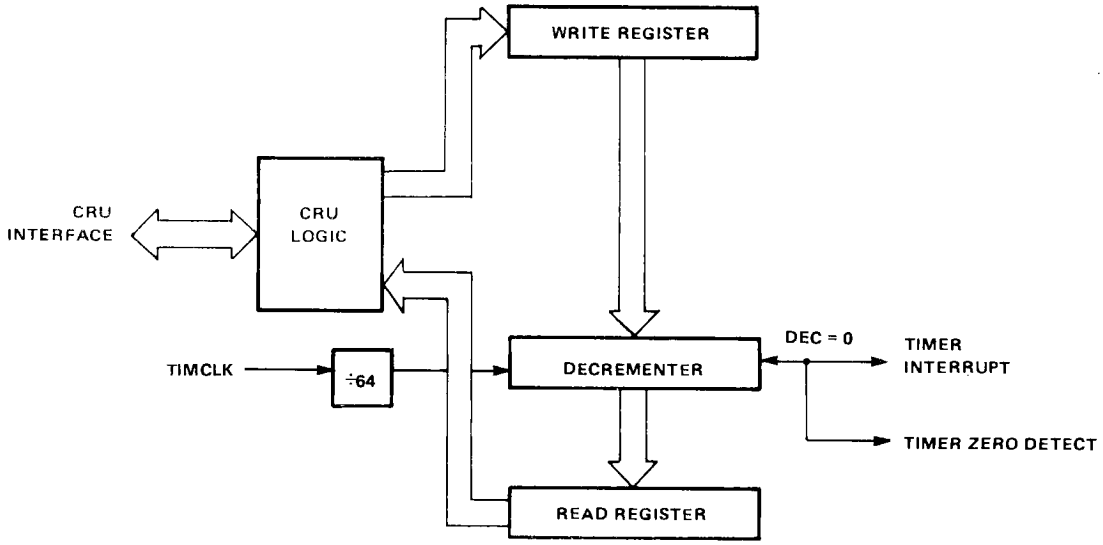


Figure 3. Interval/Event Timer

If the control bit is at logic-level low, the timer's read-register is updated with the current decrementer value after each decrement operation (once every 64 TIMCLK clocks); if the control bit is at logic-level high (timer-access mode), the read-register retains its current value therby ensuring that the read-register is not changed in the event a CRU read operation is executing during a decrement operation. Consequently, the current value of the timer's decrementer can be interrogated by 1) placing the SBP 9961 into the timer-access mode, and 2) performing a CRU-read operation on the timer's read-register through execution of an STCR instruction. The timer, then, can function as an event timer by reading the elapsed time between software events as shown in Table 3. Note that when accessing the timer, all interrupts should be disabled. The timer is disabled by either $\overline{\text{RESET}}$ (power-up reset) or by writing all zeroes to the write-register.

SBP 9961 Status

The SBP 9961s status can be determined by reading the value of the control bit located at CRU address zero. If the control bit has a logic-level low value, then the interrupt masks may be changed and data on the interrupt inputs may be read. However, access to the interval timer is inhibited. If the control bit has a logic-level high value, then the timer may be initiated, restarted, or read. Also, reading CRU address fifteen gives the status of INTREQ where logic-level low indicates activation.

Table 3. Software Examples

ASSUMPTIONS:

- Total of 6 interrupts are used
- RESET has been applied
- System uses timer at maximum interval

```
                LI      R12, CRUBAS     Setup CRU Base Address to point to 9961
                LDCR    @X,0            Program Timer with maximum interval
SYSTEM          LDCR    @Y,7            Re-enter interrupt mode and enable top 6 interrupts
SETUP           (X)  ---------->FFFF
                (Y)  ---------->7FXX
                .
                .
                .
                .
                .
                BLWP    CLKVCT          Save Interrupt Mask
                .
                .
                .
CLKPC           LIMI    0               Disable Interrupts
                LI      R12, CRUBASE+1  Set up CRU base
                .
                .
                SBO     -1              Set 9961 into timer-access mode
                STCR    R4, 14          Store read register into R4
                .
                .                       Process Timer Value
                .
                .
                SBZ     -1              Re-enter Interrupt Mode (i.e., Exit Timer-Access Mode)
                RTWP                    Restore Interrupt Mask
                .
                .
                .
CLKUCT          DATA    CLKWP,CLKPC
```

SYSTEM OPERATION

During power-up, $\overline{\text{RESET}}$ should be activated (low) to force the SBP 9961 into a known state. $\overline{\text{RESET}}$ will disable all interrupts, disable the timer, and force IC0-IC3 to (0,0,0,0) with $\overline{\text{INTREQ}}$ pulled high. System software should then enable the proper interrupts and program the timer (if used). (See Table 3 for an example.) After initial power up, the SBP 9961 is accessed only as needed to service the timer and enable or disable interrupts.

Figure 4 shows SBP 9961s system configuration. Figure 5 shows the use of a SBP 9961 with a SBP 9960 CRU I/O expander in emulation of the TMS 9901. (See *TMS 9901 Systems Interface Data Manual.*)

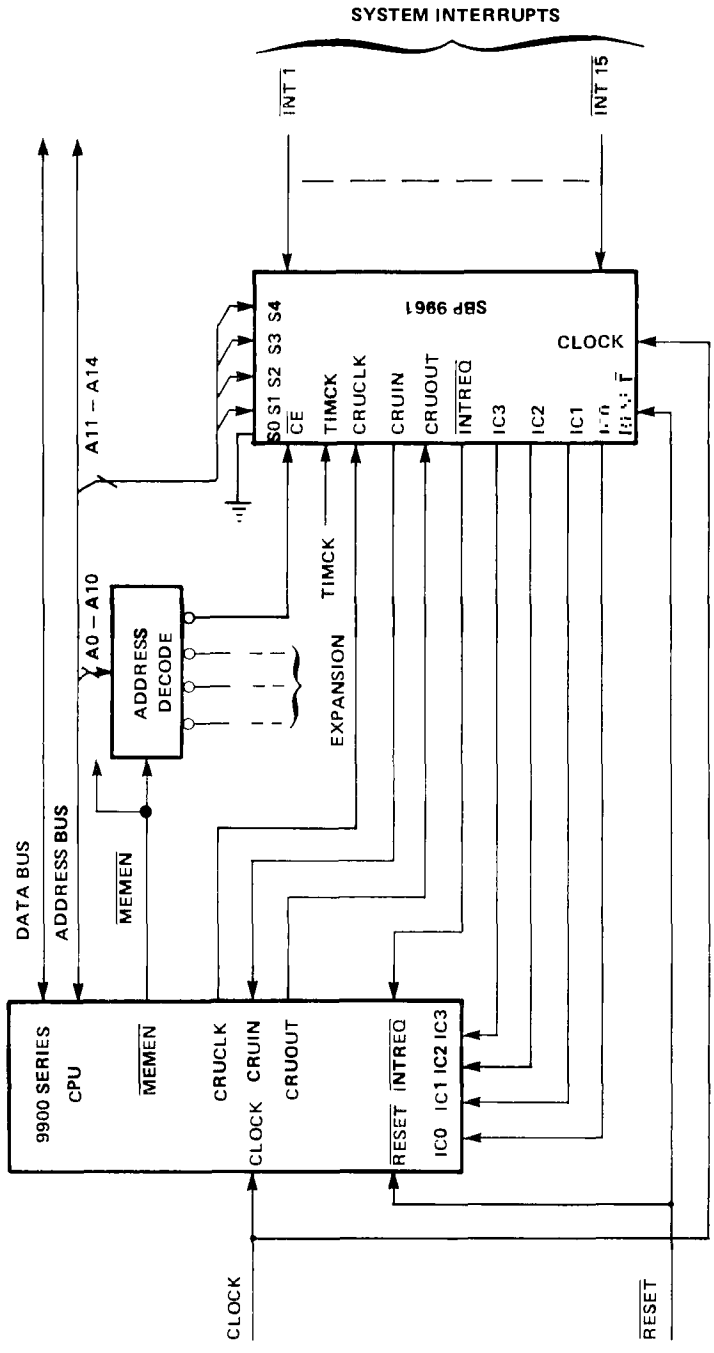


Figure 4. SBP 9961 System Configuration

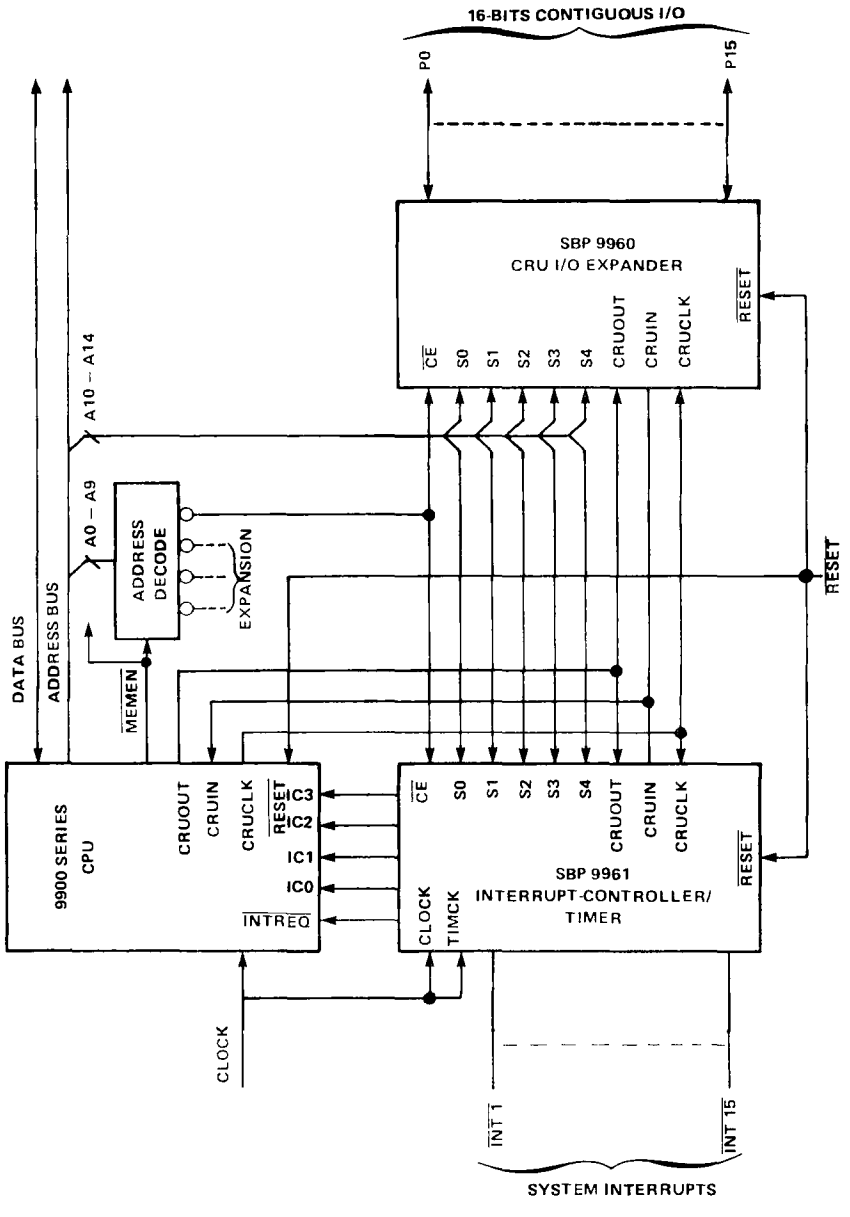


Figure 5. SBP 9961 Configuration with SBP 9960

ELECTRICAL SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS, UNLESS OTHERWISE NOTED I_{CC} = 130mA

PARAMETER		MIN	NOM	MAX	UNIT
Supply current, I_{CC}		115	130	145	mA
High-level output voltage, V_{OH}				5.5	V
Low-level output current, I_{OL}				20	mA
Operating free-air temperature, T_A	SBP 9961MJ, SBP 9961NJ	−55		125	°C
	SBP 9961EJ	−40		85	
	SBP 9961CJ	0		70	

ELECTRICAL CHARACTERISTICS (OVER RECOMMENDED OPERATING FREE-AIR TEMPERATURE RANGE, UNLESS OTHERWISE NOTED)

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_{IK}	Input clamp voltage	I_{CC} = MIN,	I_I = −12 mA			−1.5	V
I_{OH}	High-level output current	I_{CC} = 130 mA, V_{IL} = 0.8 V,	V_{IH} = 2 V, V_{OH} = 5.5 V			400	μA
V_{OL}	Low-level output voltage	I_{CC} = 130 mA, V_{IL} = 0.8 V,	V_{IH} = 2 V, I_{OL} = 20 mA			0.5	V
I_I	Input current	I_{CC} = 130 mA,	V_I = 2.4 V		180		μA

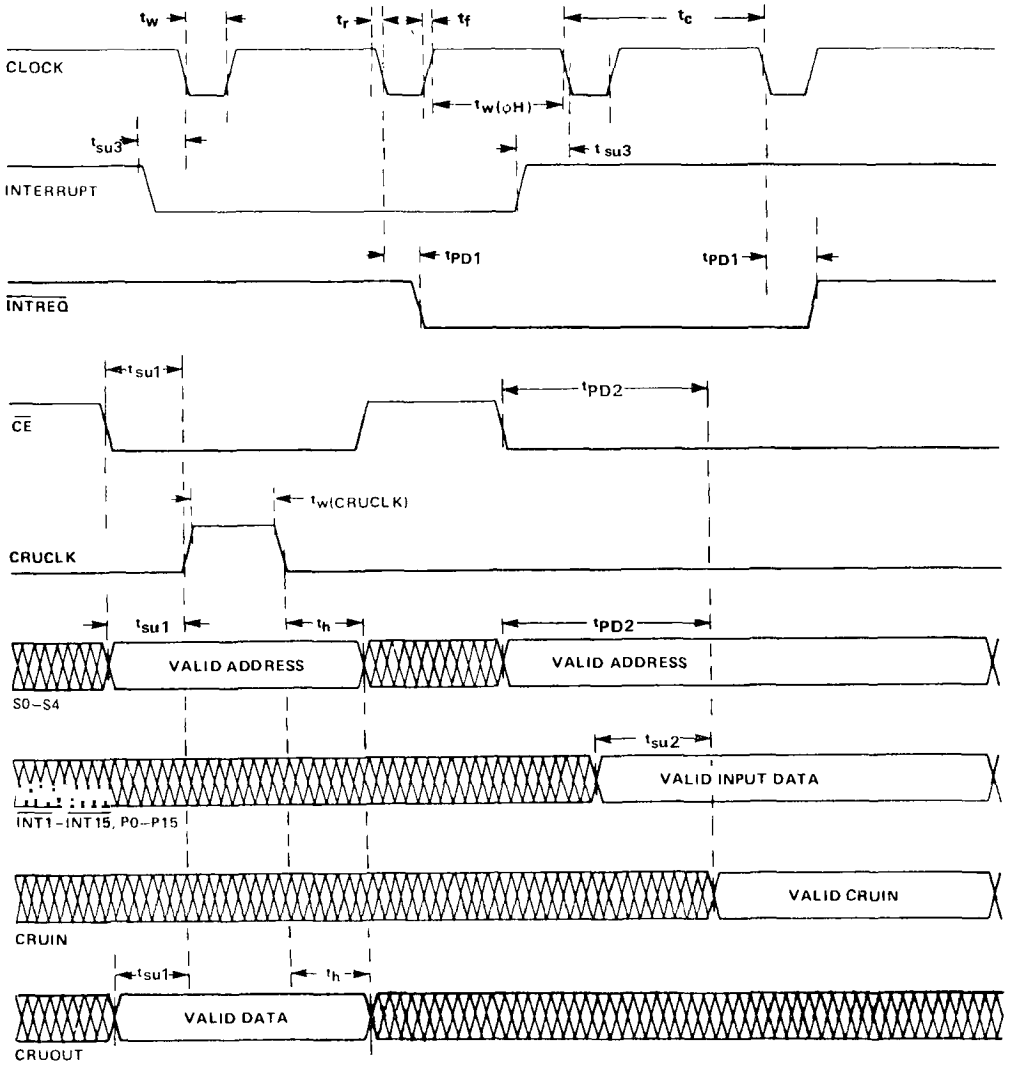
†For conditions shown as MAX, use the appropriate value specified under recommended operating conditions
‡All typical values are at I_{CC} = 130 mA, T_A = 25°C.

TIMING REQUIREMENTS OVER FULL RANGE OF OPERATING CONDITIONS

PARAMETER		MIN	NOM	MAX	UNIT
t_c	Clock cycle time	333			ns
t_r	Clock rise time		10	20	ns
t_f	Clock fall time		10	20	ns
t_{wL}	Clock pulse low width	111			ns
t_{wH}	Clock pulse high width	222			ns
t_{su1}	Setup time for S0-S4, $\overline{CE}$, or CRUOUT before CRUCLK		200		ns
t_{su2}	Setup time, input before valid CRUIN		200		ns
t_{su3}	Setup time, interrupt before clock high		60		ns
$t_{w(CRUCLK)}$	CRU clock pulse width		100		ns
t_h	Address hold time		80		ns
t_{TC}	TIMCLK cycle time		200		ns

SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS

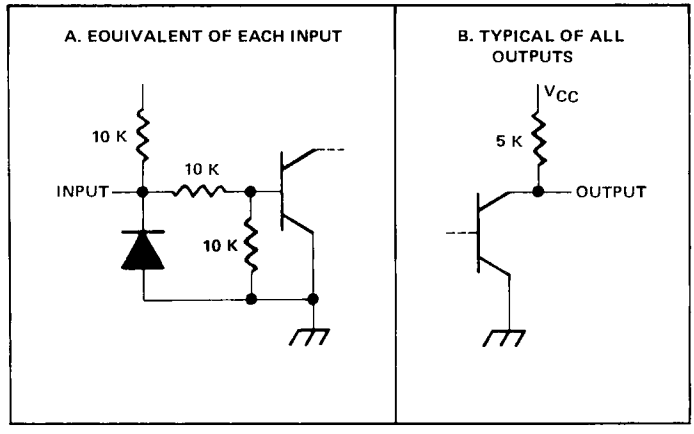
PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PD1}	Propagation delay, ↑ CLOCK to valid $\overline{INTREQ}$, IC0-IC3	C_L = 25 pF, R_L = 5K Ω		150		ns
t_{PD2}	Propagation delay, S0-S4 or $\overline{CE}$ to valid CRUIN	C_L = 25 pF, R_L = 5K Ω		330		ns



NOTE 1: ALL TIMING MEASUREMENTS ARE FROM 10% and 90% POINTS

Switching Characteristics

INPUT AND OUTPUT STRUCTURES



SBP 9964
TIMING CONTROLLER for the SBP 9900A

990/9900 FAMILY MICROCOMPUTER COMPONENTS

- 14-Bit Interval Timer-Event Counter
- $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ Synchronization
- SBP 9900A Clock Generation
- 20-Pin Package
- TTL Compatible Open-Collector I/O

Description

The SBP 9964 is an SBP 9900A peripheral support device which performs general timing and synchronization functions usually implemented with SSI TTL packages.

Internal to the SBP 9964 is a 14-bit interval timer-event counter, an SBP 9900A clock generator and an SBP 9900A $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ signal synchronizer. The interval timer-event counter communicates with the SBP 9900A through the SBP 9900A's Communication Register Unit (CRU) I/O interface. The interval timer-event counter may be efficiently applied to a variety of applications in which the interval between external events, the number of external events, or the initiation of periodic events is desired. $\overline{\text{RESET}}$ and $\overline{\text{LOAD}}$ synchronizers provide for SBP 9900A compatible synchronization of these signals from asynchronously applied external signals.

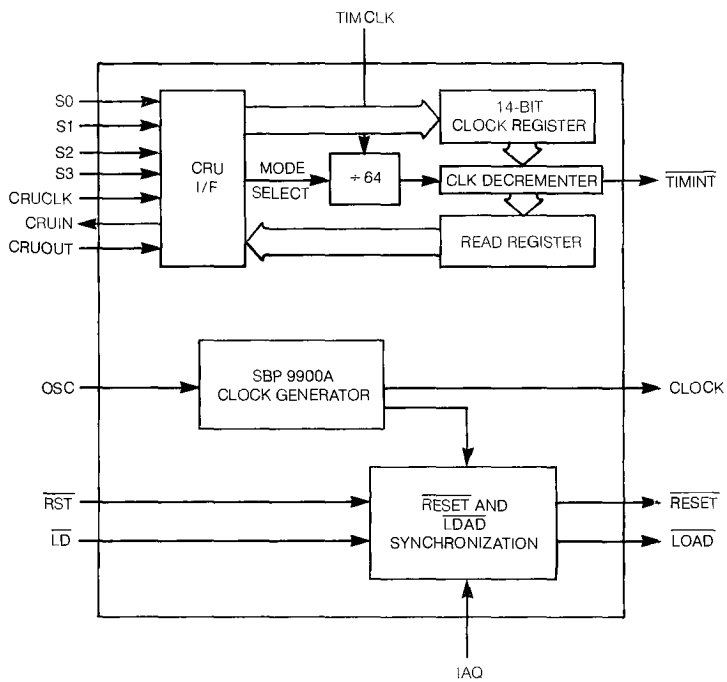


Figure 1. Functional Block Diagram

DESIGN GOAL

This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

990/9900 FAMILY MICROCOMPUTER COMPONENTS

- Microprocessor Memory-Mapped I/O Peripheral Interface
- Dual 8-Bit Input/Output Peripheral Ports
- Internal Mask Registers and Associated Compare Logic for Character/Data Recognition
- TTL Compatible Open-Collector I/O
- 40-Pin Package

Description

The SBP 9965 Peripheral Interface Adapter is a byte oriented, parallel memory-mapped, input/output interface which interfaces to microprocessor CPU's through the memory bus. Two 8-bit I/O ports with independent handshake lines are provided which allow a variety of byte oriented peripheral devices to be efficiently interfaced to the CPU. High data rates are effected through parallel transfers of data between the CPU and the peripheral device.

Two internal mask registers, one associated with each I/O port, may compare logic which flags the CPU whenever an equal condition exists between I/O and mask register data. This feature is useful for byte string searches or control character recognition.

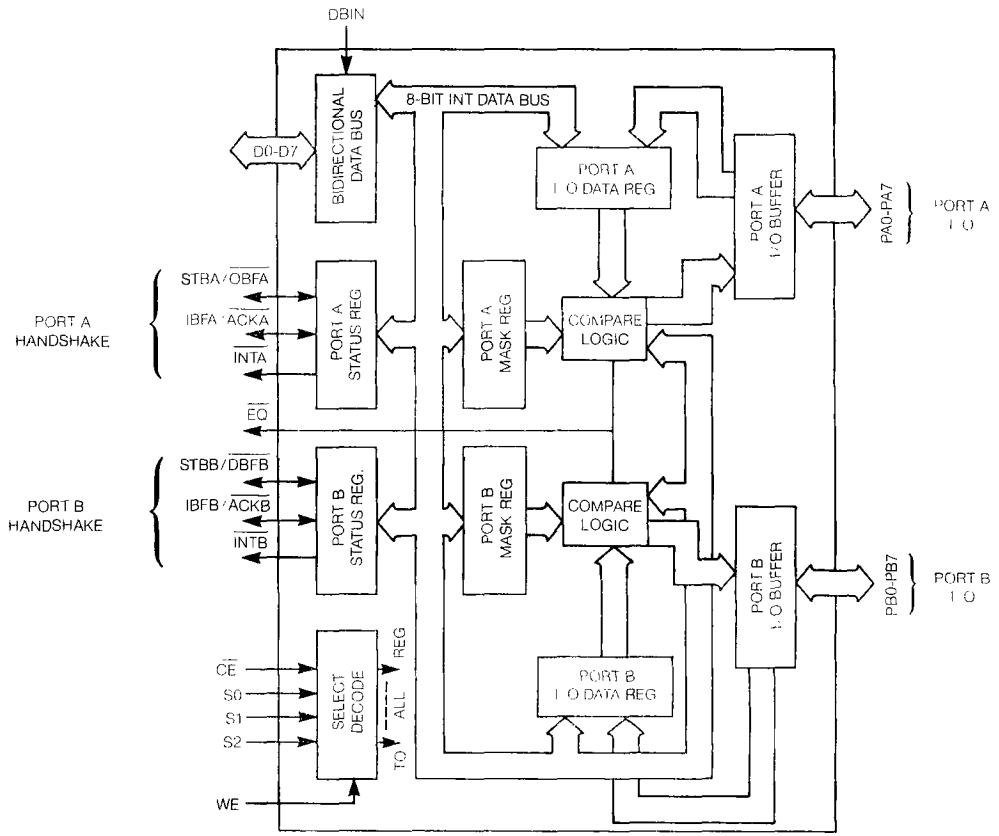
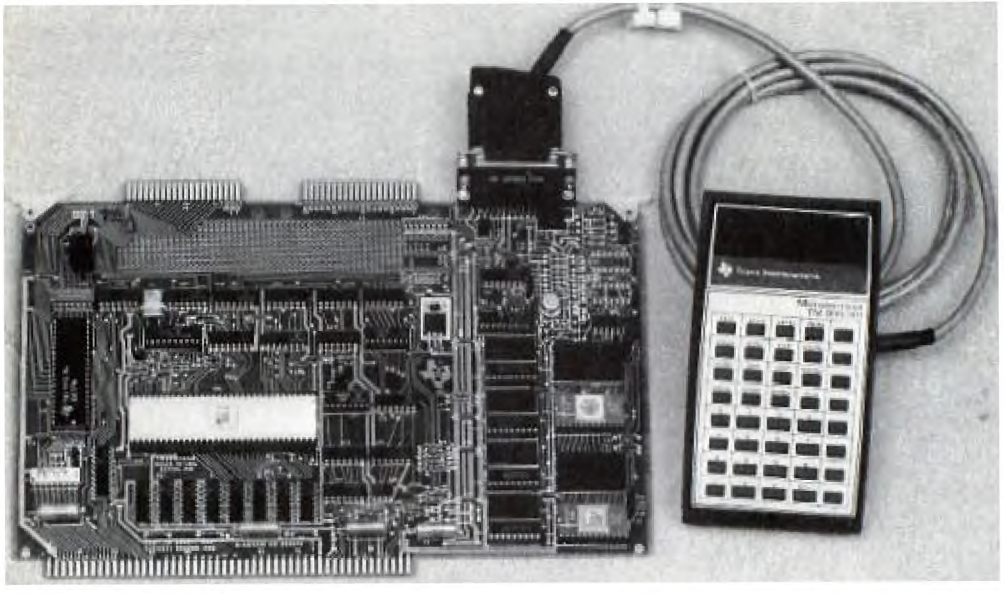


Figure 1. Function Block Diagram

DESIGN GOAL

This document describes the design specifications for a product under development. Texas Instruments reserves the right to change these specifications in any manner, without notice.

TM 990 Series
Microcomputer Modules



TM 990/100M Microcomputer Module
with TM 990/301 Microterminal

GENERAL

The TM 990 microcomputer modules are a series of low cost TMS 9900 family microcomputers, assembled on a single printed circuit board. The TMS 9900 16-bit NMOS microcomputer and the TMS 9980, a software compatible 8-bit data bus microprocessor, will be the CPU's for the initial board systems. The TM 990 microcomputers offer a new level of hardware capability incorporating all of the powerful TMS 9900 LSI components. Whether a single CPU unit with on board memory and self contained I/O or an expanded multiboard system is needed, the application can be implemented at the lowest possible systems cost. With its broad line of semiconductor products, Texas Instruments manufactures most of the components utilized on the modules and thus is able to exercise quality control before and after the assembly of the system. This and high volume production ensure that the highest level of reliability and the maximum possible cost savings are passed along to the OEM.

At the heart of any of these systems is, of course, the software compatibility of the entire 990/9900 family. This product line fits squarely into that level of hardware/software integration between the flexibility of the TMS 9900 family component level application and the complete prepackaged system nature of the 990/4 and 990/10 minicomputers. The common instruction set enables the engineer to base hardware decisions solely on considerations of time, design, effort, and cost goals without concern for future compatibility of products or the ability to change his level of integration of the current product should initial considerations change.

The CPU boards, in particular, provide ready-to-use units for the evaluation of 9900 family component and software capability, especially when incorporated with the TM 990/301 microterminal. The OEM will find that these modules provide the best path to market in a time-critical application or in an application whose volume is difficult to assess. In most cases the modules will continue to be the best systems answer for such applications, but if it should not, TI will supply all necessary schematics and artwork to assist in the transition. The modules are completely supported by the AMPL* prototyping system. This floppy disc system, described more fully in another part of this book, has capabilities which make it invaluable for program development and debug as well as for a final systems test unit.

*Trademark of Texas Instruments Incorporated.

8◂

MICROCOMPUTER MODULES

The TM 990/100, TM 990/101, and TM 990/180 CPU modules come complete with on-board memory and I/O interface. Each 7½ x 11-inch (190 x 279 mm) board comes with 1k x 16 bits of TMS 2708 EPROM capability that can be expanded to 2k x 16 bits using TMS 2708's or 4k x 16 bits utilizing the jumper-selectable TMS 2716 option on the TM 990/100 and TM 990/101 modules. Static RAM capacity is 256 x 16 (1k x 16 for the TM 990/101) expandable to 512 x 16 (2k x 16 for the TM 990/101). Sixteen bits of parallel I/O are implemented on all three CPU's, as is a RS-232 or TTY serial interface. The TMS 9901, which performs the parallel I/O interface, also enables the user to implement the full interrupt capability of the processor. The TMS 9901 and the TMS 9902 (and for the TM 990/101, another TMS 9902 or a TMS 9903), which handle the serial I/O interface, each have programmable interval timers incorporated on-chip, thereby automatically providing the module user with two interval timers for the TM 990/100 and TM 990/180, and three interval timers for the TM 990/101. The TM 990/101 has two serial I/O parts. The bus structure of the CPU modules makes it possible to expand the system beyond the single board level. Memory, I/O, and special purpose controller boards may be added along with the TM 990/510 card cage for larger system applications.

EIA or teletype terminals can be optionally selected by the user and a differential line driver can be added as a factory option. Additionally, the TM 990/301 microterminal is an extremely low cost hexadecimal terminal option. The microterminal will execute the TIBUG monitor commands and can be used as a computer front panel.

TM 990/100M, TM 990/101, TM 990/180M MICROCOMPUTERS

TM 990/100M

- TMS 9900 16-bit CPU
- Up to 512 x 16 bits of RAM, TMS 4042-2 (2111-1)
- Up to 2k words of EPROM using TMS 2708 or 4k words using TMS 2716
- TMS 9901 programmable system interface
- TMS 9902 asynchronous communications controller
- EIA or TTY terminal interface option
- Prototyping area for custom applications
- Fully expandable bus structure
- Designed to fit the TM 990/510 card cage
- TIBUG operating monitor

TM 990/101

- TMS 9900 16-bit CPU
- Up to 2k x 16 bits of RAM, TMS 4045-45
- Up to 2k words of EPROM using TMS 2708 or 4k words using TMS 2716
- DMA to on-board memory
- TMS 9901 programmable system interface
- Two serial I/O ports, using TMS 9902 asynchronous communications controllers
- Three programmable interval timers
- Edge Triggered Interrupt, with software reset
- CRU addressable L.E.D. and DIP switch for custom applications
- Designed to fit the TM 990/510 card cage

TM 990/180

- TMS 9980 16-bit CPU
- Up to 1k x 8 bits of RAM TMS 4045-45
- Up to 4k bytes of EPROM using TMS 2708
- TMS 9901 programmable system interface
- TMS 9902 asynchronous communications controller
- EIA or TTY terminal interface option
- Prototyping area for customer applications
- Fully expandable bus structure
- Designed to fit the TM 990/510 card cage
- TIBUG operating monitor

8

SPECIFICATIONS

	TM 990/100M	*TM 990/101*	*TM 990/180M*
CPU:	TMS 9900	TMS 9900	TMS 9980
Instruction set	69 instructions	69 instructions	69 instructions
Bit operation	8, 16, or 32 bits	8, 16, or 32 bits	8, 16, or 32 bits
System clock	3 MHz	3 MHz	2.5 MHz
Interrupts	16 levels—15 may be external	16 levels—15 may be external	6 levels—4 may be external
Interval timers	Two (in TMS 9901 and TMS 9902)	Three (in TMS 9901, TMS 9902, and TMS 9903, or in TMS 9901 and two TMS 9902's)	Two (in TMS 9901 and TMS 9902)
	RESOLUTION / MAXIMUM INTERVAL	RESOLUTION / MAXIMUM INTERVAL	RESOLUTION / MAXIMUM INTERVAL
TMS 9901	21.3 μs / 349 ms	21.3 μs / 349 ms	25.6 μs / 414 ms
TMS 9902	64 μs / 16.4 ms	64 μs / 16.4 ms	76.8 μs / 19.7 ms
TMS 9903		64 μs / 16.4 ms	
Memory:	16-bit word configuration	16-bit word configuration	8-bit byte configuration
On-board EPROM/ROM	1k words, expandable to 4k	1k words, expandable to 4k	2k bytes, expandable to 4k
On-board RAM	256 words, expandable to 512	1k words, expandable to 2k	512 bytes, expandable to 1k
Off-board expansion	Up to 32 k words	Up to 32k words	Up to 16k bytes
Input/Output			
Parallel:	16 lines, expandable to 4k	16 lines (7 dedicated and 9 that may be programmed as inputs, outputs, or interrupts) expandable to 4k	16 lines, expandable to 2k
Serial:	Asynchronous Controller, TMS 9902 5-8 bits/character Programmable data rate, stop bits, parity	Serial Port A: Asynchronous Controller, TMS 9902 Serial Port B: Asynchronous Controller, TMS 9902, or Synchronous Controller, TMS 9903 5-8 bits/character Programmable data rate, stop bits, parity	Asynchronous controller, TMS 9902 5-8 bits/character Programmable data rate, stop bits parity
Baud rates (bps)	75 300 2400 19,200 110 600 4800 38,400 150 1200 9600	110 600 4800 19,200 150 1200 9600 38,400 300 2400	75 300 2400 19,200 110 600 4800 38,400 150 1200 9600
Interfaces			
Bus: Data and address	3-state, TTL compatible	3-state, TTL-compatible	3-state, TTL compatible
Control	TTL-compatible	TTL-compatible	TTL-compatible
Parallel I/O and interrupts	TTL-compatible	TTL-compatible	TTL-compatible
Serial I/O	RS-232, 20-mA current loop, or differential line driver	Port A: RS-232C, 20-mA current loop, or RS-232C Multidrop Port B: RS-232C terminal, or modem with optional cable	RS-232, 20-mA current loop, or differential line driver

TM 990/100M, TM 990/101, TM 990/180M MICROCOMPUTERS

	TM 990/100M	*TM 990/101*	*TM 990/180M*
Expansion Prototyping Area	Space for one 40-pin DIP and four 16-pin DIP's	Not applicable	Space for one 40-pin DIP and four 16-pin DIP's
Software	TI BUG monitor self-contained in EPROM (TM990/401-1)	TIBUG monitor self-contained in EPROM (TM 990/401-3)	TIBUG monitor self-contained in EPROM (TM 990/401-2)
Mating Connectors			
Bus: 100-pin, 0.125-inch (3,18-mm) centers	TI H4311211-50 (Solder tail), TI H431111-50 (Wire wrap), or Viking 3VH50		
Parallel I/O: 40-pin, 0.100-inch (2,54-mm) centers	TI H421121-20, 3M 3464-0001, or Viking 3VH20/1JN5		
Serial I/O: 25-pin (male)	Cannon DB-25P, Cinch DB-25P, or ITT DB-25P		
Power Requirements		1K RAM 1K EPROM / 2K RAM 2K EPROM	
	5V ± 3% 1.3A 12V ± 3% 0.2A −12V ± 3% 0.1A	5V ± 3% 1.6A / 1.8A 12V ± 3% 0.2A / 0.4A −12V ± 3% 0.2A / 0.3A −5V regulated on board from −12V	5V ± 3% 1.3A 12V ± 3% 0.2A −12V ± 3% 0.1A
Operating Temperature Range	0°C to 70°C	0°C to 70°C	0°C to 70°C
Physical Characateristics:			
Width	11 inches (279 mm)	11 inches (279 mm)	11 inches (279 mm)
Height	7.5 inches (190 mm)	7.5 inches (190 mm)	7.5 inches (190 mm)
Depth		0.5 inch (12.7 mm)	
Board thickness	0.062 inch (1,58 mm)	0.062 inch (1,58 mm)	0.062 mm (1,58 mm)
Weight		1 pound (0.45 kg)	
Terminations	4.7 kΩ	10 kΩ	4.7 kΩ
I/O and Interrupt	5V	5V	5V
READY B and HOLD B		220 Ω 5V 330Ω	

ORDERING INFORMATION

TM 990/100M-1 — TMS 9900 microcomputer board with TIBUG monitor in two TMS 2708 EPROM's and EIA or TTY serial I/O jumper option.

TM 990/100M-2 — TMS 9900 microcomputer board with unprogrammed TMS 2708 EPROM's and EIA or differential line driver jumper option.

TM 990/100M-3 — TMS 9900 microcomputer board with fully expanded memory (four TMS 2716 EPROM's and eight TMS 4042-2 RAM's) and EIA or differential line driver jumper option.

TM 990/180M-1 — TMS 9980 CPU board with TIBUG monitor in two TMS 2708 EPROM's and EIA or TTY serial I/O jumper option.

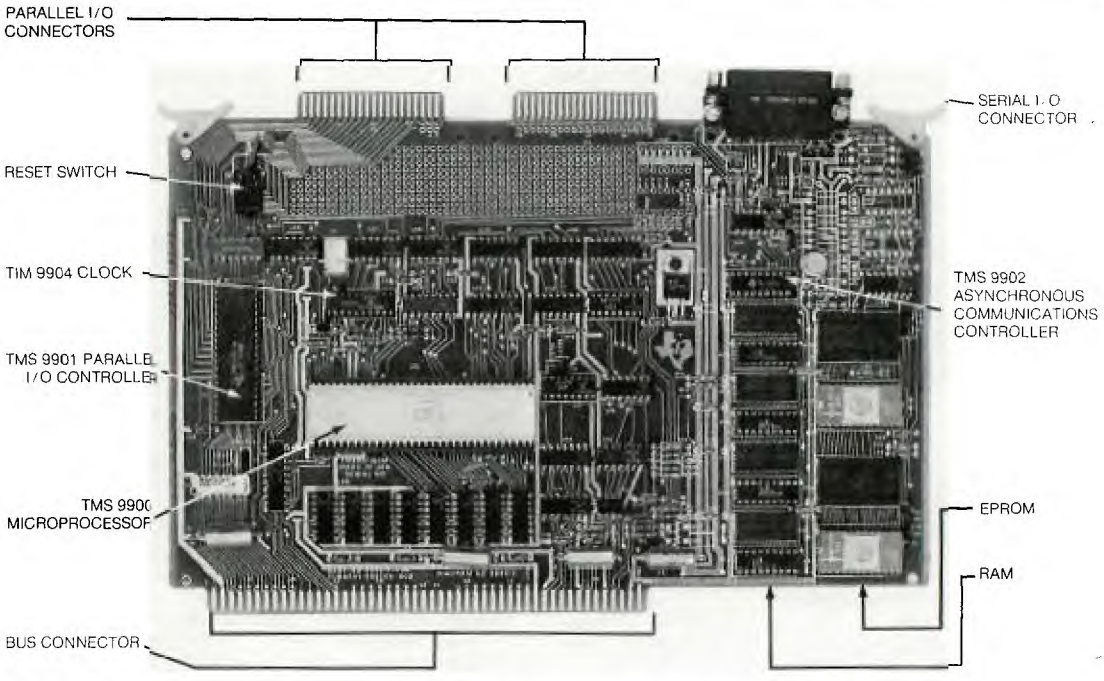
TM 990/180M-3 — TMS 9980 CPU board with four unprogrammed TMS 2708 EPROM's, eight TMS 4042 RAM's, and EIA or differential line driver jumper option.

TMS990/101M OPTIONS

TM990/101-1 TMS 9900 microcomputer board with TIBUG monitor in two TMS 2708 EPROM's and TTY, EIA and microterminal on the local serial port. The remote serial port supports synchronous/asynchronous communications.

TM990/101-2 TMS 9900 microcomputer board with unprogrammed TMS 2708 EPROM's and multidrop, EIA and microterminal options on the local serial port. The remote serial port supports synchronous/asynchronous communications.

TM990/101-3 TMS 9900 microcomputer board with fully expanded memory (four TMS 2716 EPROM's and eight TMS 4045-45 RAM's) and TTY, EIA and microterminal options on the local serial port. The remote serial port supports synchronous/asynchronous communications.



The TM 990/100M is an assembled, tested microcomputer module utilizing the powerful, NMOS 16-bit, TMS 9900 microprocessor as its CPU. With RAM and ROM/EPROM included on board as well as programmable serial and parallel I/O, the TM 990/100M is a powerful single-board microcomputer. Since all address, data, and control lines are brought to the board connectors, the board can be expanded to use the entire capabilities of the TMS 9900 in larger systems.

OPERATION

The TM 990/100M microcomputer is a software compatible member of the TMS 9900/990 family. The TMS 9900 is used as a CPU to provide 16 bits of processing power with a minicomputer instruction set which includes multiply and divide. The TM 990/100M module is designed for 3 MHz operation, utilizing the full 16 levels of prioritized interrupts and the advanced memory-to-memory architecture of the TMS 9900. Additionally, the bus structures are set up to take advantage of the full 64K byte memory addressing capability of the 9900 and the nonmultiplexed memory, I/O and interrupt buses.

MEMORY

The on-board memory includes both an EPROM/ROM section and a static RAM section. Four sockets are available for TMS 2708, TMS 2716 EPROM or TMS 4700, TMS 4732 ROM operation. The assembled price includes two TMS 2708's, or 1K words. Using the available jumper option, all four sockets can be populated with TMS 2716's, providing a maximum on-board EPROM capability of 4K words. The static RAM area consists of two 256-word banks of memory. Four TMS 4042-2 (TMS 2111-1) are populated and four additional sockets are included. The cycle time of this memory section is 0.667 microseconds. The address map is shown in *Figure 1;* the minimum area of EPROM RAM area may not be used for off-board expansion. DMA control lines are also accessible on the bus.

INTERRUPTS AND TIMERS

Fifteen maskable interrupts plus the reset and load trap vectors are implemented. *Table 1* shows the implementation. The TMS 9901 handles all 15 external interrupts which can be generated from either the bus connector or the I/O bus. The TMS 9901 enables each level to be individually maskable under program control. Additionally, level 3 can be programmed to use the interval timer in the TMS 9901. Level 4 can be generated from the TMS 9902 as an interval timer or for three other serial interface conditions (see the *TMS 9902 Data Manual*). Two programmable timers, therefore, are available on board.

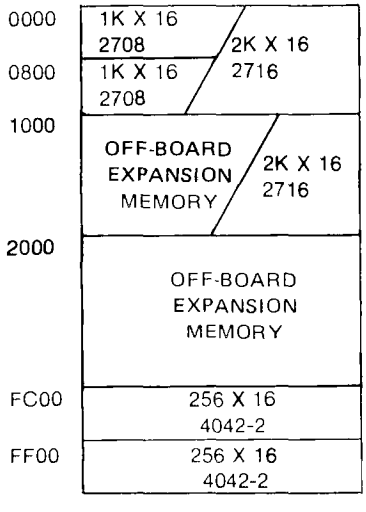


Figure 1. *Memory Address Map.*

I/O

The serial I/O and the parallel I/O are handled over the dedicated I/O bus of the TMS 9900, the communications register unit (CRU). *Table 2* lists the address assignments within the dedicated 4K CRU address space. The TMS 9902 acts as the controller for this asynchronous serial interface. The character length, baud rate (75 to 38,400), and parity and stop bits are programmable. Three optional types of interface are supported:

- EIA
- 20 mA neutral current loop TTY
- Private wire differential line driver/receiver.

The TM 990/100M board is delivered complete with a 25-pin RS-232 type female connector, and is jumper selectable to support EIA or TTY operation. The differential line driver is normally unpopulated (see *Options*). Also, the TMS 9903 synchronous communications controller can be utilized since the TMS 9902/9903 are socket compatible.

INTERRUPT LEVEL	FUNCTION
0	Reset or PRES
1	External Device
2	External Device
3	Clock or External
4	Serial Int. or Ext.
5-15	External Devices
Load	Restart

Table 1. Interrupts.

BASE ADDRESS (REGISTER 12)	CRU BIT NUMBER	FUNCTION
0080_{16}	$40_{16} \rightarrow 5F_{16}$	On-Board Serial I/O Port (TMS 9902)
0100_{16}	$80_{16} \rightarrow 9F_{16}$	On-Board 16 I/O Parallel Interface, Interrupt Status Register, Interrupt Mask Register, and Interval Timer (TMS 9901)
0000_{16}	$00_{16} \rightarrow 3F_{16}$	Reserved for On-Card Expansion
$00C0_{16}$	$60_{16} \rightarrow 7F_{16}$	
0140_{16}	$A0_{16} \rightarrow FF_{16}$	
2000_{16}	$1000_{16} \rightarrow FFF_{16}$	Off-Board CRU

Table 2. CRU Address Map.

The parallel I/O is handled by the TMS 9901; 16 parallel lines are all interfaced to the top edge connector which has 40 pins on 0.100 inch (2.54 mm) centers. Additionally, eight parallel lines are interfaced to the bus connectors. The programmable features of the TMS 9901 permit configuring these lines as I/O lines or interrupts (refer to the *TMS 9901 Data Manual*). All I/O lines are equipped with pullup resistors.

TIBUG

The TIBUG monitor TM 990/401-1 is normally supplied preprogrammed in the populated TMS 2708 EPROM's (see *Options*). Its operation is described in the *User's Manual* or the TM 990 Series literature.

PROTOTYPING AREA

The prototyping area is large enough to accommodate one 40-pin DIP (0.6 inch 15,24 mm centers) plus four 16-pin DIP's (0.3 inch 7,62 mm centers).

OPTIONS

The TM 990/100M-1 board is equipped with two TMS 2708's preprogrammed with the TIBUG monitor, and the serial I/O is jumper selectable as EIA port or a TTY interface. The TM 990/100M-2 board is populated with two blank EPROM's, and a private wire differential line driver interface is populated instead of the TTY interface. The TM 990/100M-3 board is delivered with the maximum memory expansion (512 words of RAM and 4K words of unprogrammed EPROM) and the differential line driver. Other software or accessories, such as the line by line assembler and the microterminal, may be ordered under separate part numbers.

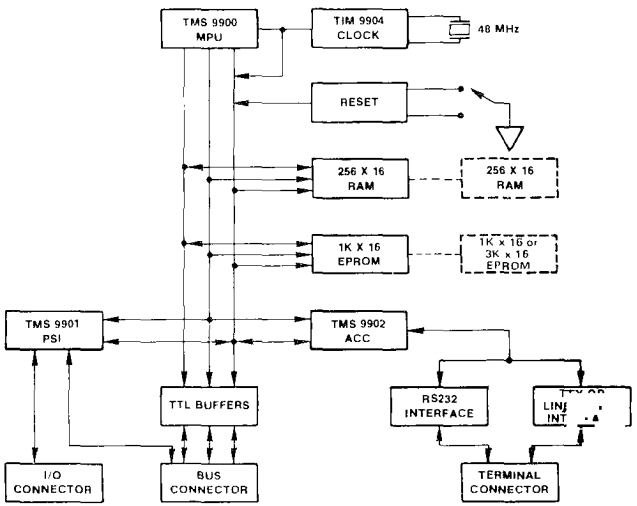
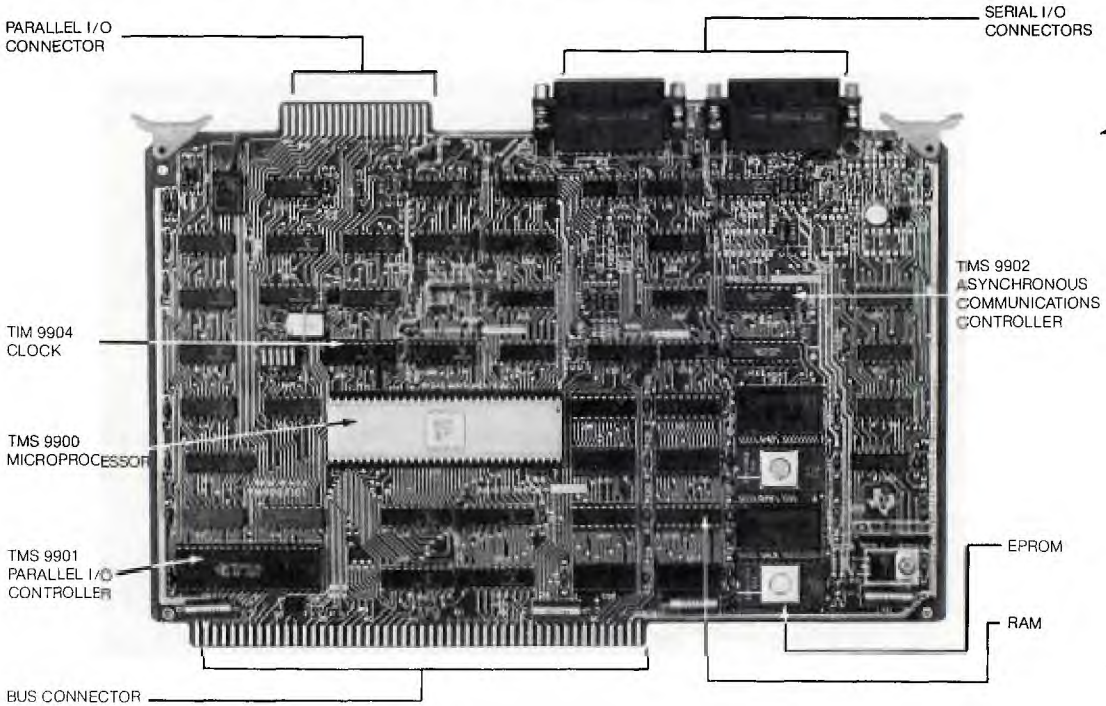


Figure 2. TM 990/100M Block Diagram



The TM 990/101 microcomputer is a member of Texas Instruments line of OEM computer products which take full advantage of Texas Instruments broad based semiconductor technology to provide economical, computer-based solutions for OEM, applications. The CPU, clock, memory, I/O, and bus interface are all contained on a single 7½ x 11 inch (190,5 x 279 mm) printed circuit board.

OPERATION

The TMS 9900 microprocessor is the heart of the TM 990/101. This 16-bit CPU features a memory to memory architecture and a minicomputer instruction set which includes hardware multiply and divide. A total of eight addressing modes, including indirect and pre-indexed addressing, provide powerful software capabilities while the TMS 9900's two-address architecture makes a memory-to-memory add possible without having to load register pairs with addresses or using a dedicated accumulators. The TMS 9900's instruction set is upward compatible with the TI 990 minicomputer family. The TMS 9900 addresses 32K 16-bit words of memory. In addition to DMA and memory mapped I/O, the TMS 9900 performs I/O functions on a separate data structure called the Communications Register Unit (CRU). The CRU consists of 4096 output bits and 4096 input bits. Each bit is separately addressable. Five instructions enable the programmer to perform both single and multibit CRU operations.

MEMORY

The TM 990/101 microcomputer contains up to 4K bytes of static RAM (TMS 4045) on-board. All positions are socketed. Sockets are provided for up to 8K bytes of EPROM (TMS 2716). Convenient jumper options also allow utilization of TMS 2708 1K x 8 bit EPROM's. Provisions are included for deconfiguration of either or both on-board RAM and on-board EPROM, if desired, when used with other TM 990 series Memory Expansion boards. A jumper-selectable wait state for the EPROM is also included.

All memory device locations are socketed. A PROM controls memory address decoding. The RAM is decoded as one bank but the two EPROM pairs are decoded as separate banks, allowing custom placement of EPROM. Such a custom decoding scheme can be done by obtaining a blank SN74S287 PROM, and programming it properly.

CONFIGURATION	MEMORY MAP	ALTERNATE MEMORY MAP
RAM, bank 2	$F000_{16}$-F7FE	0000_{16}-07FE$_{16}$
RAM, bank 1	$F800_{16}$-FFFE$_{16}$	0800_{16}-0FFE$_{16}$
EPROM, all TMS 2708*	0000_{16}-0FFE$_{16}$	$F000_{16}$-FFFE$_{16}$
EPROM, all TMS 2716*	0000_{16}-1FFE$_{16}$	$E000_{16}$-FFFE$_{16}$

*Jumper selectable

INTERRUPTS AND TIMERS

Seventeen interrupt inputs are available on the TM 990/101. All interrupts trap through vectors in memory. Two interrupts are non-maskable interrupts while the others are maskable. There are three interrupt sources on board: Serial I/O Port A, Serial I/O Port B, and the TMS 9901 interval timer. Interrupt 6 may be triggered on either a positive or negative transition. All other interrupts are active low. The 15 maskable interrupts are also automatically prioritized by the microprocessor.

INTERRUPT	LEVEL	VECTOR	DESCRIPTION
PRES	0	0000-0002_{16}	Unmaskable, active low.
RESTART	LOAD	$FFFC_{16}$-FFFE$_{16}$	Unmaskable, active low. May be activated by software (LREX).
INT1-INT5	1-5	0004_{16}-0016_{16}	Maskable, dedicated, active low.
INT6	6	0018_{16}-$001A_{16}$	Maskable, dedicated (+) or (-) edge detect or active low.
INT7-INT15	7-15	$001C_{16}$-$003E_{16}$	Maskable, active low. May be programmed as interrupt, input, or output.

8

I/O

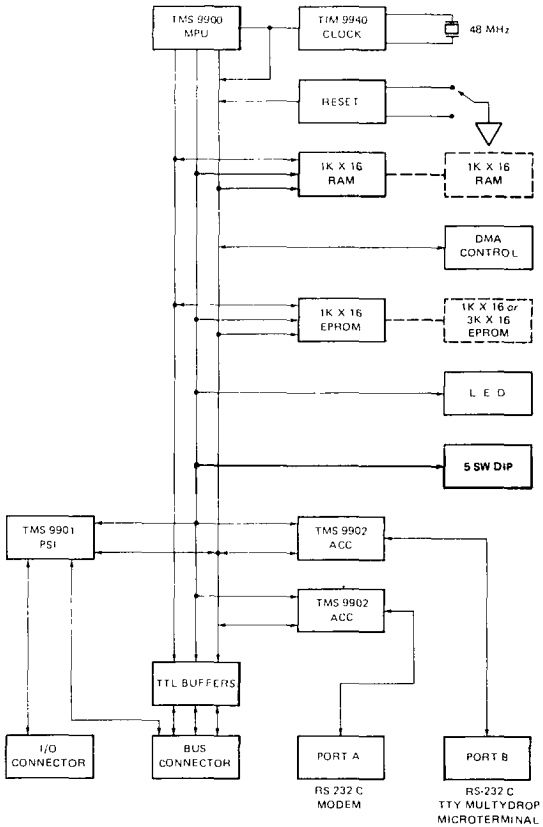
The TM 990/101 contains up to 16 programmable I/O lines controlled by a TMS 9901 Programmable Systems Interface. Seven lines are dedicated I/O lines while 9 lines may be programmed as either I/O or interrupt inputs. The 16 I/O lines appear on a 40-pin edge connector which mates with ribbon cable or round cable.

Two serial interfaces are available on the TM 990/101. Each port is controlled by a TMS 9902 Asynchronous Communications Controller. Serial communications rates of up to 38,400 baud may be maintained. Port A is compatible with the serial I/O port on the TM 990/100 microcomputer. Port A supports EIA compatible terminals as well as 20 mA neutral current loop teletypes. Port A also supports TI's TM 990/301 Microterminal. A version of the TM 990/101 supports a differential line driver-receiver communications interface in place of the TTY interface. This multidrop interface supports 9600 baud serial communications at distances of up to 10,000 ft. on shielded twisted pairs. Serial Port B supports communications with EIA compatible terminals as well as popular modems such as Bell Type 103J and 212A, using an optional modem cable.

TM 990/101 memory and I/O capacity may be increased by adding Texas Instruments standard I/O and memory expansion cards. Memory may be expanded to 60K bytes by the addition of the TM 990/201 memory expansion boards, leaving 4K bytes open for memory mapped I/O. Parallel I/O and interrupt expansion capability may be increased by the addition of the TM 990/310 48-I/O Data Module.

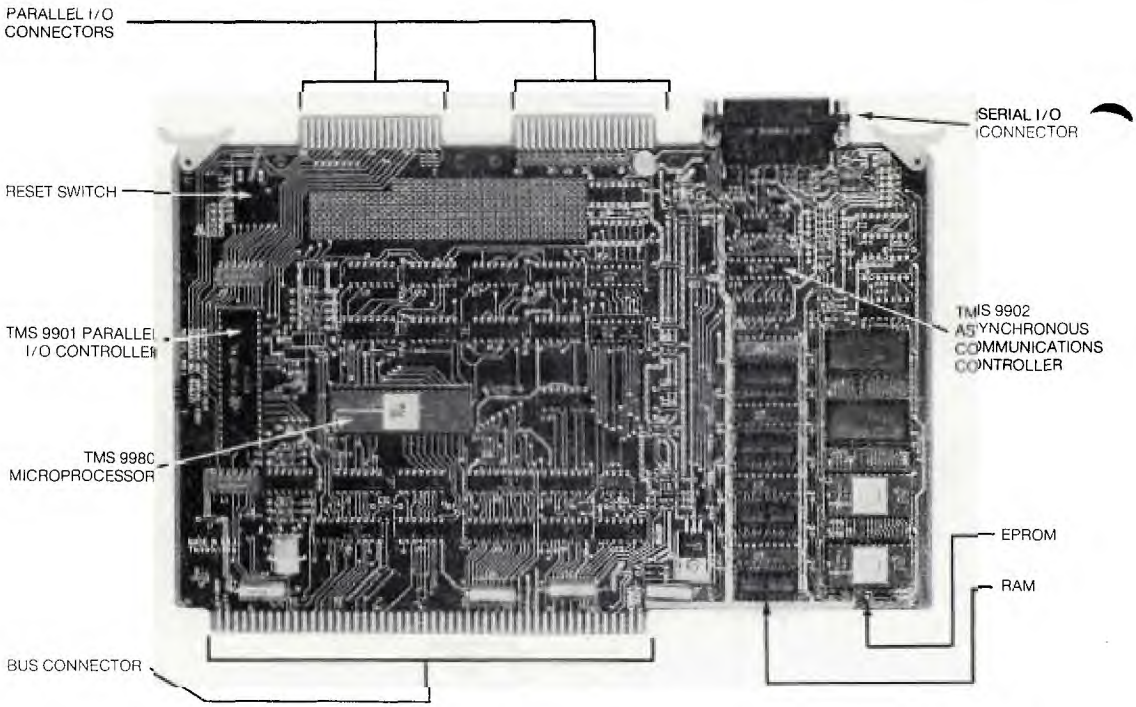
The development cycle for TM 990/101 based products may be significantly reduced by using Texas Instruments Advanced Microprocessor Prototyping System (AMPL). TMS 9900 emulation as well as 10 MHz trace capability are featured. This floppy disk based software development system permits programs to be edited, assembled, linked, loaded, and executed much faster than conventional paper tape or cassette based systems. TMS 9900 emulation allows development and debugging of software directly on the TM 990/101 while monitoring and controlling this environment from the AMPL prototyping system.

▶8



TM 990/101 Block Diagram

8◀



The TM 990/180M is an assembled, tested microcomputer module utilizing the NMOS 16-bit TMS 9980 microprocessor as its CPU. The TMS 9980 utilizes an eight bit data bus which may be the most cost effective solution for smaller byte-dedicated operations. With RAM and ROM/EPROM included on board as well as programmable serial and parallel I/O, the TM 990/180M is a powerful single-board microcomputer. Since all address, data, and control lines are brought to the board connectors, the board can be expanded to use the entire capabilities of the TMS 9980.

OPERATION

The TM 990/180M microcomputer module is a software compatible member of the TMS 9900/990 family. The TMS 9980 is used as a CPU to provide 16 bits of processing power with a minicomputer instruction set which includes multiply and divide. The TM 990/180M module is designed for 2.5 MHz operation, utilizing the full six levels of prioritized interrupts and the advanced memory-to-memory architecture of the TMS 9980. Additionally, the bus structures are set up to take advantage of the full 16K byte memory addressing capability of the TMS 9980 and the nonmultiplexed memory, I/O and interrupt buses. The bus structure is compatible with the other boards of the TM 990 series such as the TM 990/100M board.

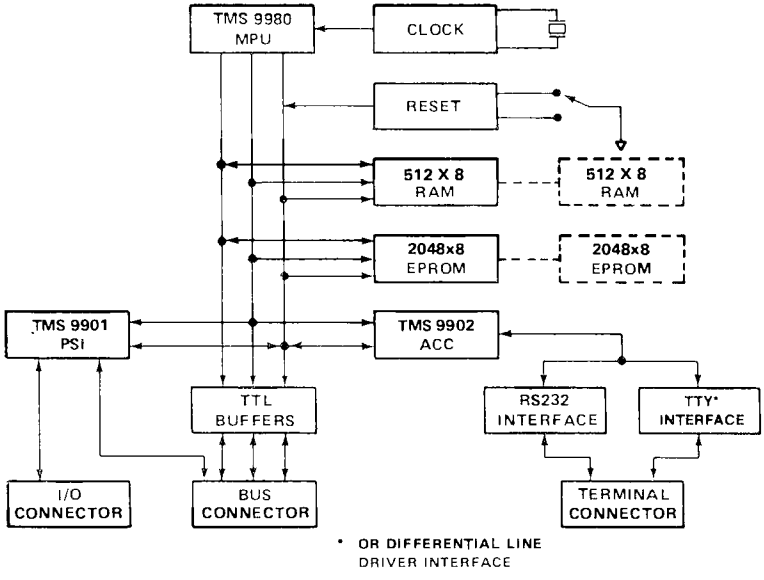


Figure 2. TM 990/180M Block Diagram

MEMORY

The on-board memory includes both an EPROM/ROM section and a static RAM section. Four sockets are available for TMS 2708 EPROM or TMS 4700 ROM operation. The assembled price includes two TMS 2708's or 2K bytes. The static RAM area consists of four 256 byte banks of memory. Four TMS 4042-2 (TMS 2111-1) are populated, and four more sockets are included. The cycle time of this memory section is 1.33 microseconds. The memory address map is shown in *Figure 1.*

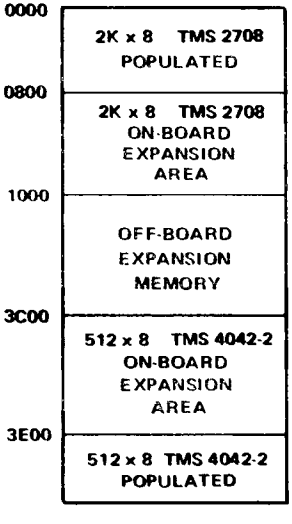


Figure 1. Memory Address Map

8

INTERRUPTS AND TIMERS

Four maskable interrupts plus the reset and load trap vectors are implemented. *Table 1* shows the implementation. The TMS 9901 handles all four external interrupts which can be generated from either the bus connector or the I/O bus. The TMS 9901 enables each level to be individually maskable under program control. Additionally, level 3 can be programmed to use the interval timer in the TMS 9901. Level 4 can be generated as an interrupt from the TMS 9902. One of the functions that will cause this interrupt is the interval timer. Two programmable timers, therefore, are available on board.

LEVEL FROM TMS 9901	TRAP VECTOR LOCATION	FUNCTION
1	0000_{16}	Reset pushbutton or PRES from the chassis backplane connector
2	$3FFC_{16}$	Software (LREX) or RESTART from the chassis backplane connector
3	0004_{16}	Real Time Clock (TMS 9901) or external device.
4	0008_{16}	Serial interface (TMS 9902) or external device
5	$000C_{16}$	External device
6	0010_{16}	External device

Table 1. *TMS 9980 Interrupts*

INPUT/OUTPUT

The Serial I/O and the parallel I/O are handled over the dedicated I/O bus of the TMS 9900 or the communications register unit (CRU). Table 2 lists the address assignments within the dedicated 4K CRU address space. The TMS 9902 acts as the controller for this asynchronous serial interface. The character length, baud rate (75 to 38,400), parity and stop bits are programmable. Three optional types of interface are supported:

▶8
- EIA
- 20 mA neutral current loop TTY
- Private wire differential line driver/receiver.

BASE ADDRESS	CRU BIT NUMBER	FUNCTION
0000_{16}	$000_{16} \longrightarrow 03F_{16}$	Reserved for on-board CRU Expansion
0080_{16}	$040_{16} \longrightarrow 05F_{16}$	On-board serial I/O (TMS 9902)
$00C0_{16}$	$060_{16} \longrightarrow 07F_{16}$	Reserved for on-board CRU Expansion
0100_{16}	$080_{16} \longrightarrow 09F_{16}$	On-board parallel I/O interface interrupt status register, interrupt mask register, interval timer (TMS 9901)
0140_{16}	$0A0_{16} \longrightarrow 0FF_{16}$	Reserved for on-board CRU Expansion
0200_{16}	$100_{16} \longrightarrow 3FF_{16}$	Off-board CRU

Table 2. CRU Address Map

The TM 990/180M board is delivered complete with a 25-pin RS-232 type female connector, and is jumper selectable to support EIA or TTY operation. The differential line driver is normally unpopulated (see *Options*). Also, the TMS 9903 synchronous communications controller can be utilized, since the TMS 9902/9903 are socket compatible.

The parallel I/O is handled by the TMS 9901; 16 parallel lines are all interfaced to the top edge connector which has 40 pins on 0.100 inch (2,54 mm) centers. Additionally, eight parallel lines are interfaced to the bus connectors. The programmable features of the TMS 9901 permit these lines to be set up as I/O lines or interrupts (refer to the *TMS 9901 Data Manual*). All I/O lines are equipped with pullup resistors.

TIBUG

The TIBUG monitor TM 990/401-2 is normally supplied preprogrammed in the populated TMS 2708 EPROM's (see Options). Its operation is described in the TIBUG *User's Manual* or the TM 990-series literature.

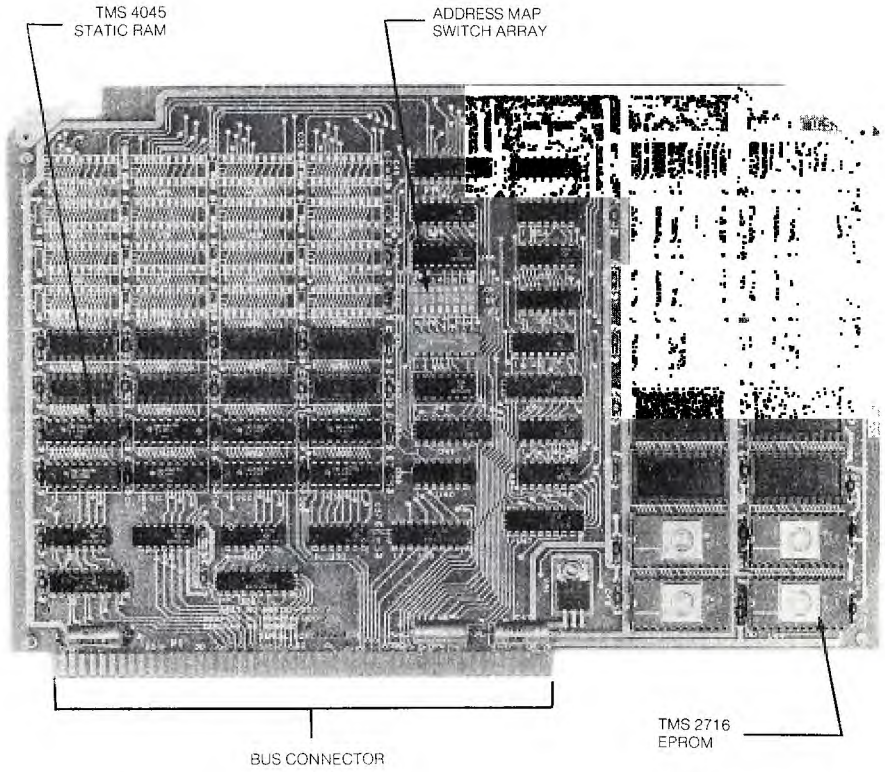
PROTOTYPING AREA

The prototyping area is large enough to accommodate one 40-pin DIP (0.6 inch 15,24 mm centers) plus four 16-pin DIP's (0.3 inch 7,62 mm centers).

OPTIONS

The TM 990/180M-1 board is equipped with two TMS 2708 EPROM's preprogrammed with the TIBUG monitor, and the serial I/O is jumper selectable as an EIA port or a TTY interface. The TM 990/180M-3 board is populated with four TMS 2708 EPROM's (unprogrammed), eight TMS 4042-2 RAM's, and a private wire differential line driver interface instead of the TTY interface. Other software or accessories, such as the line by line assembler and the microterminal, may be ordered under separate part numbers.

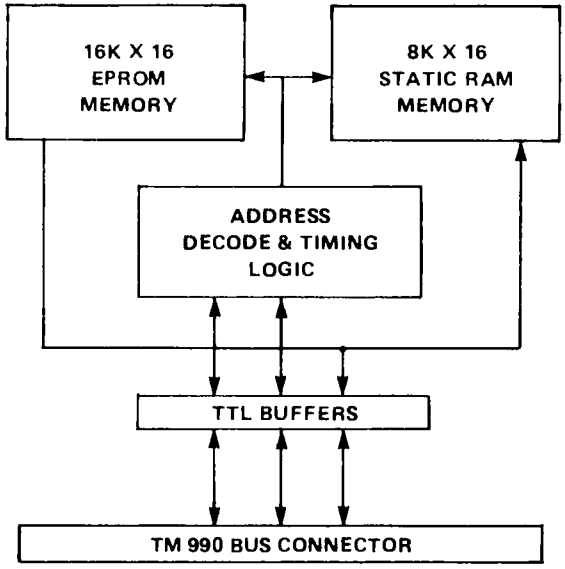
TM 990/201 MEMORY EXPANSION BOARD



The TM 990/201 is an assembled, tested, memory expansion board designed for use with TMS 9900-based microcomputer modules such as the TM 990/100M. The TM 990/201 contains both static RAM and EPROM memory, expandable to a maximum configuration of 8K x 16 bit words of RAM and 16K x 16 bit words of EPROM. The TM 990/201 does not support the TMS 9980-based TM 990/180M microcomputer.

FEATURES

- Bus-compatible with the TM 990/100M microcomputer module
- 4K words TMS 2716 EPROM, expandable to 16K words
- 2K words TMS 4045 static RAM, expandable to 8K words
- 1 microsecond cycle time (3MHz)
- TTL-compatible interface
- Designed to fit the TM 990/510 card cage.



DESCRIPTION

The TM 990/201 memory expansion board is a member of Texas Instruments' line of OEM computer products which takes advantage of Texas Instruments' broad based semiconductor technology to provide economical, computer based solutions for OEM applications. The memory expansion board is contained on a 7½ x 11 inch printed circuit board which is fully compatible with the TM 990 board format.

The TM 990/201 features up to 8K x 16 bits of static RAM and up to 16K x 16 bits of EPROM. The static RAM array is composed of Texas Instruments TMS 4045, 1K x 4 static memory devices. The EPROM array comprises Texas Instruments TMS 2716, 2K x 8 EPROM devices. The static RAM array is arranged into four banks of memory, each 2K x 16. The EPROM array is likewise arranged into eight banks, each 2K x 16. Both memory arrays are socketed for convenient memory expansion. The TM 990/201 -41 is half socketed, and the TM 990/201 -42 and TM 990/201 -43 are fully socketed.

▶8

The TM 990/201 memory controller logic provides the timing and memory mapping functions necessary to interface the TM 990/201 to 16-bit TM 990/1XX series microcomputers. The memory map is switch selectable for both the RAM and EPROM arrays. Sixteen convenient memory map configurations are possible for each array, and the maps are configured on 2K word address boundaries. The map logic also is designed to accommodate customized memory maps.

TM 990/201 MEMORY EXPANSION BOARD

The TM 990/201 -4X family of memory expansion boards is populated with TMS 4045-45 static RAM's and TMS 2716 EPROM's. Both devices offer 450 nsec access time; consequently, each memory cycle to the TM 990/201 is extended one clock cycle by the insertion of a WAIT state. If faster static RAM's are utilized in the RAM array, the WAIT state in RAM memory cycles can be conveniently removed using only a jumper.

OPTIONS

The TM 990/201 is available in the following three versions.

MODEL NO.	MEMORY POPULATED		MEMORY EXPANSION AREA (EXTRA SOCKETS PROVIDED)	
	EPROM	RAM	EPROM	RAM
TM 990/201-41	4K x 16	2K x 16	4K x 16	2K x 16
TM 990/201-42	8K x 16	4K x 16	8K x 16	4K x 16
TM 990/201-43	16K x 16	8K x 16	—	—

MEMORY CONFIGURATION

Figures 1 and *2* show the memory configurations of RAM and EPROM available on the TM 990/201 memory expansion board.

8

A0-A3 (HEX)	HEX MEMORY ADDRESS	MICROCOMPUTER MEMORY MAP /100
0	0000-0FFF	EPROM
1	1000-1FFF	EPROM (EXPAN.)
2	2000-2FFF	
3	3000-3FFF	
4	4000-4FFF	
5	5000-5FFF	
6	6000-6FFF	
7	7000-7FFF	
8	8000-8FFF	
9	9000-9FFF	
A	A000-AFFF	
B	B000-BFFF	
C	C000-CFFF	
D	D000-DFFF	
E	E000-EFFF	MAPPED I/D
F	F000-FFFF	RAM

SWITCH NO.	DIP SWITCH CODES*																HEX
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
5	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
6	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	
7	ON	ON	ON	ON	OFF	OFF	OFF	OFF	ON	ON	ON	ON	OFF	OFF	OFF	OFF	
8	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	

*OFF = 1, ON = 0

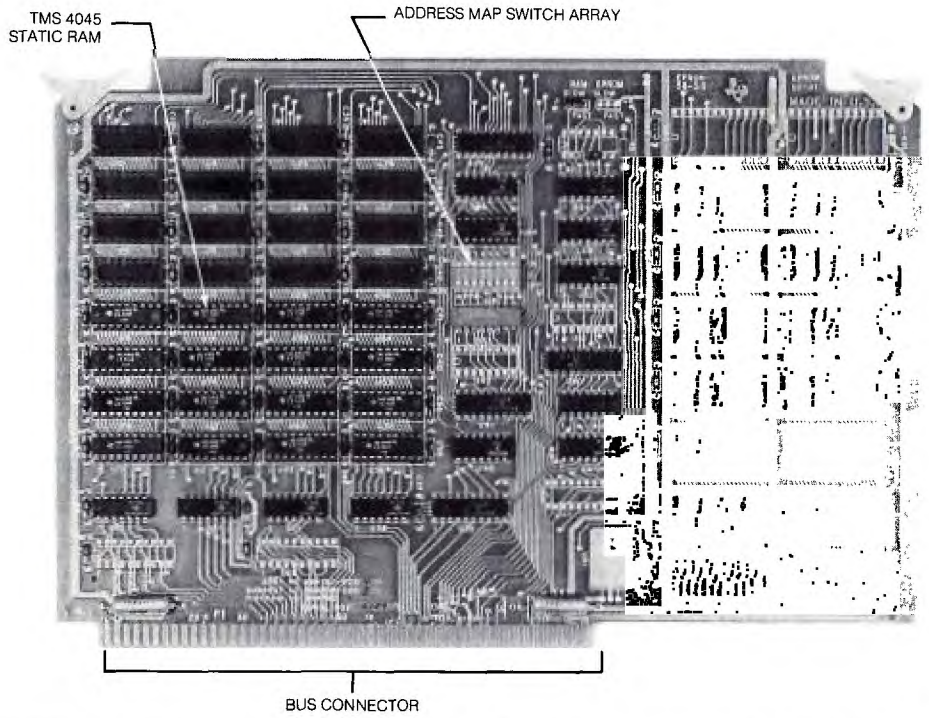
Figure 1. TM 990/201 Ram Decode Configurations

8

A0-A3 (HEX)	HEX MEMORY ADDRESS	MICROCOMPUTER MEMORY MAP /100	SWITCH NO.	SWITCH CODES* 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	HEX
			1	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
			2	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	
			3	ON	ON	ON	ON	OFF	OFF	OFF	OFF	ON	ON	ON	ON	OFF	OFF	OFF	OFF	
			4	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
0	0000-0FFF	EPROM																		
1	1000-1FFF	EPROM (EXPAN.)																		
2	2000-2FFF																			
3	3000-3FFF																			
4	4000-4FFF																			
5	5000-5FFF																			
6	6000-6FFF																			
7	7000-7FFF																			
8	8000-8FFF																			
9	9000-9FFF																			
A	A000-AFFF																			
B	B000-BFFF																			
C	C000-CFFF																			
D	D000-DFFF																			
E	E000-EFFF	MAPPED I/O																		
F	F000-FFFF	RAM																		

*OFF = 1, ON = 0

Figure 2. TM 990/201 Eprom Decode Configurations

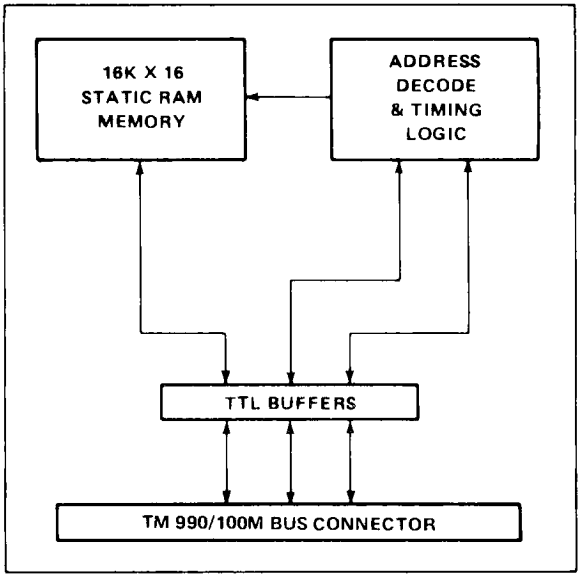


The TM 990/206 is an assembled, tested, RAM expansion memory board designed for use with TMS 9900-based microcomputer modules such as the TM 990/100M. The TM 990/206 contains static RAM memory devices up to a maximum configuration of 8K x 16 words. The TM 990/206 is similar to the popular TM 990/201 memory board, but only the RAM section is populated. The TM 990/206 does not support the TMS 9980-based TM 990/180M microcomputer.

FEATURES:

- Bus compatible with the TM 990/100M microcomputer module
- 4K words of TMS 4045 static RAM, expandable to 8K words
- 1μsec cycle time (3 MHz)
- TTL-compatible interface
- Designed to fit the TM 990/510 card cage.

TM 990/206 EXPANSION MEMORY BOARD



DESCRIPTION

The TM 990/206 expansion memory board is a member of Texas Instruments' line of OEM computer products which take advantage of Texas Instruments' broad based semiconductor technology to provide economical, computer based solutions for OEM applications. The memory expansion board is contained on a 7½ x 11 inch printed circuit board which is fully compatible with the TM 990 board format.

The TM 990/206 features up to 8K x 16 static RAM. The RAM array is composed of Texas Instruments TMS 4045-45 1K x 4 static memory devices. The array is configured into four banks of memory, each bank consisting of 2K words. The RAM array is fully socketed for convenient memory expansion.

The memory controller logic provides the timing and memory mapping functions necessary to interface the TM 990/206 to 16-bit TM 990/1XX series microcomputers. Sixteen convenient, switch selectable, memory map configurations are possible. All maps are configured on 2K word address boundaries.

The TM 990/206-4X family of memory expansion boards is populated with TMS 4045-45 static RAM's, featuring an access time of 450 nsec. For operation with a TM 990/100M microcomputer, each memory cycle to the TM 990/206 is extended one clock cycle by the insertion of a WAIT state. If faster static RAM's are utilized, the WAIT state can be conveniently removed with a jumper.

OPTIONS

The TM 990/206 is available in two versions:

Model	*RAM Population*	*Expansion Area Additional Sockets*
TM 990/206-41	4K×16	4K×16
TM 990/206-42	8K×16	—

MEMORY CONFIGURATION

Figure 1 shows the possible memory configurations for the RAM areas available on the TM 990/206.

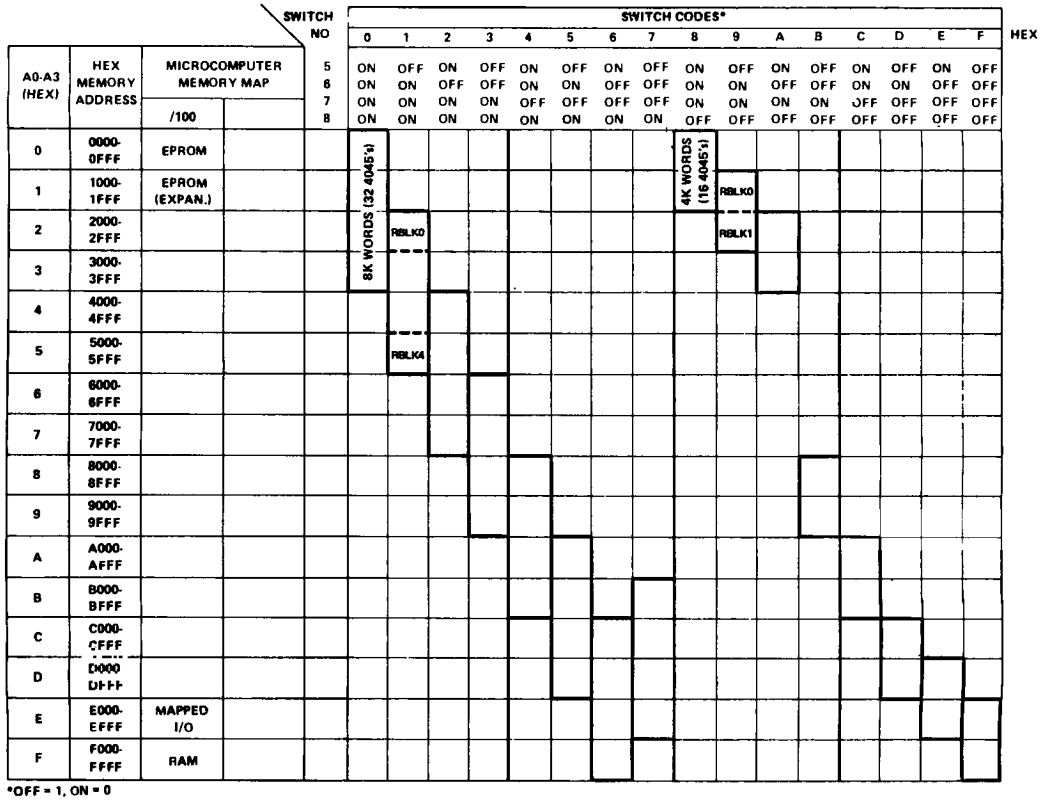


Figure 1. *TM 990/206-4X Ram Decode Configuration*

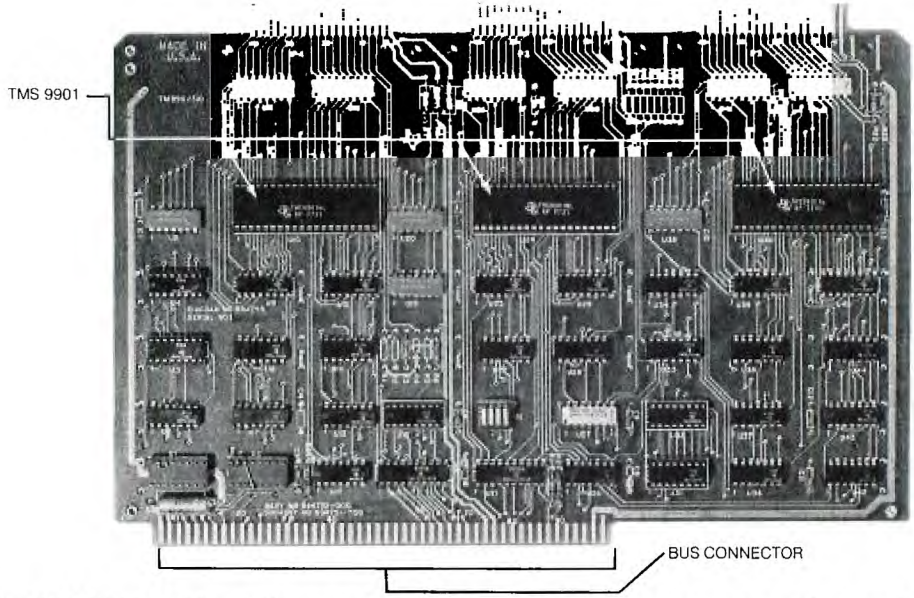
TM 990/206 EXPANSION MEMORY BOARD

SPECIFICATIONS

	TM 990/201			*TM 990/206*	
Memory Configuration	TMS 4045-45, 1K x 4 static RAM TMS 2716, 2K x 8 EPROM			TMS 4045-45, 1K x 4 static RAM	
Typical Power Requirements for Various Model Numbers	−41	−42	−43	−41	−42
5V ±5%	1.0A	1.4A	2.15A	1.3A	2.15A
12V ±5%	0.16A	0.225A	0.475A	Not required	Not required
−12V ±5%	0.05A	0.125A	0.225A	Not required	Not required
Cycle Time Memory Device Access Time	450 ns	300 ns		200 ns	150 ns
Memory Cycle Time @ 3 MHz	1.0 μs	0.667 μs		0.667 μs	0.667 μs
Bus Interface Data and Address Control	Three-state TTL-compatible TTL-compatible				
Mating Connector 100 pin, 0.125 inch (3,175 mm) centers	TI H431111-50 (wire wrap), TI H431121-50 (solder tail), or Viking 3VH50/ICN5 (pierced tail)				
Operating Temperature Range	0°C to 70°C				
Physical Characteristics Width Height Board thickness Component height	11 inches (279,4 mm) 7½ inches (190,5 mm) 0.062 inch (1,575 mm) 0.40 inch (10,16 mm)				

ORDERING INFORMATION

TM 990/201-41	4K x 16 EPROM, 2K x 16 RAM, half socketed
TM 990/201-42	8K EPROM, 4K RAM, fully socketed
TM 990/201-43	16K EPROM, 8K RAM, fully socketed
TM 990/206-41	4K x 16 RAM, sockets for 8K x 16 memory
TM 990/206-42	8K x 16 RAM, fully socketed



The TM 990/310 is a fully assembled, fully tested, input/output expansion module compatible with all TM 990 family microcomputer modules. The TM 990/310 offers a maximum I/O expansion capability of 48 I/O points, programmable as either inputs or outputs.

FEATURES

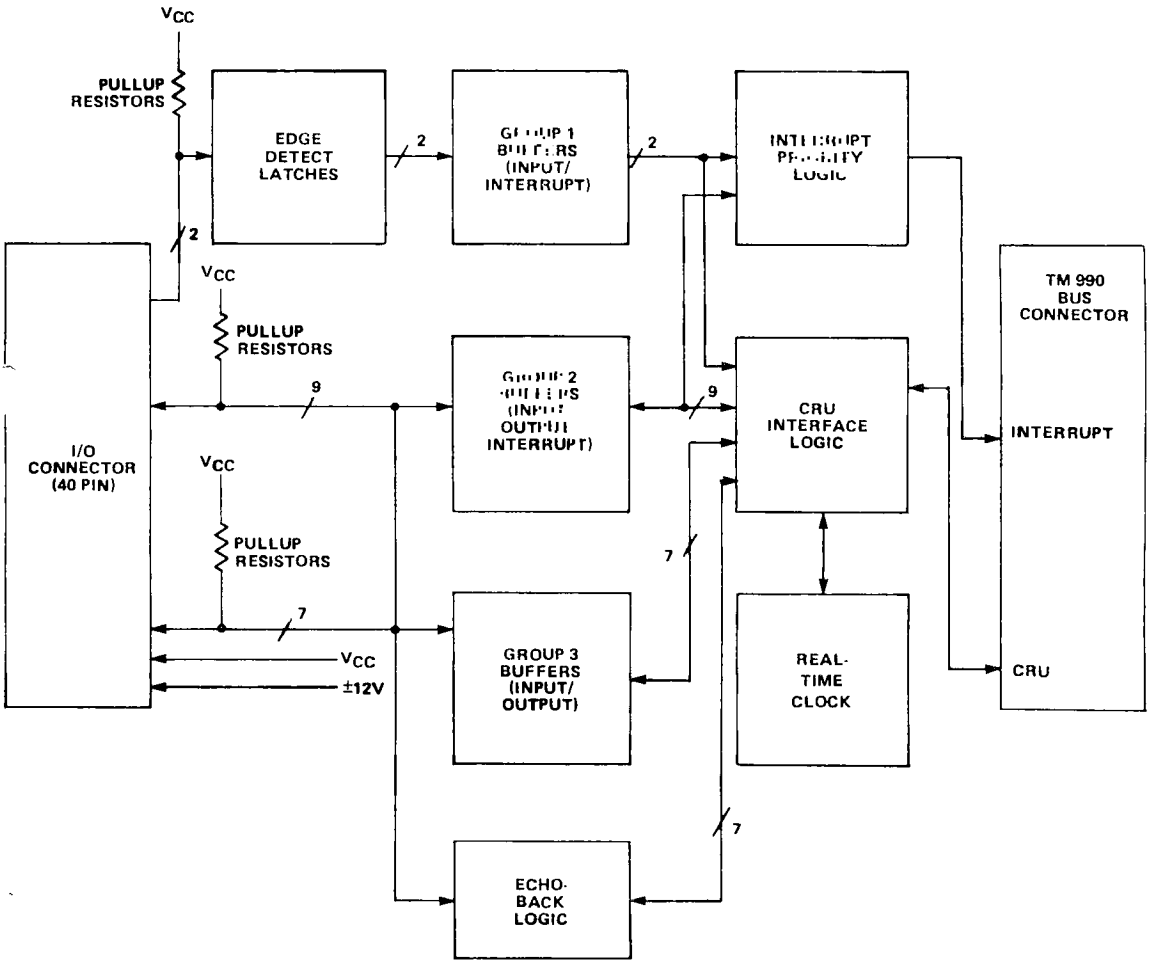
- Compatible with the TM 990 microcomputer module CRU bus
- Designed to fit the TM 990/510 card cage
- Inputs/outputs are TTL-compatible
- May be used with solder, wire wrap, or ribbon cable edge connectors
- Up to 27 I/O lines may be programmed as prioritized, unlatched interrupts
- Three (+) and three (-) edge-triggered and latched, prioritized interrupt inputs are provided (in addition to 48 I/O lines)
- Contains three real-time clocks (or event timers)
- I/O lines are provided with echo-back feature

▶8

OPERATION

The TM 990/310 input/output expansion module is implemented using the TM 990 printed circuit format. The TM 990/310 uses three TMS 9901 LSI, programmable, systems interface chips to control I/O. The extreme versatility and low cost of the TM 990/310 module makes it usable in a wide variety of I/O applications. Inputs and outputs may be mixed in any proportion, and any number of interrupts may be utilized, up to a maximum of 33. The interrupt priority encoding scheme also permits use of the module as an interrupt expander for the TM 990/100M microcomputer family.

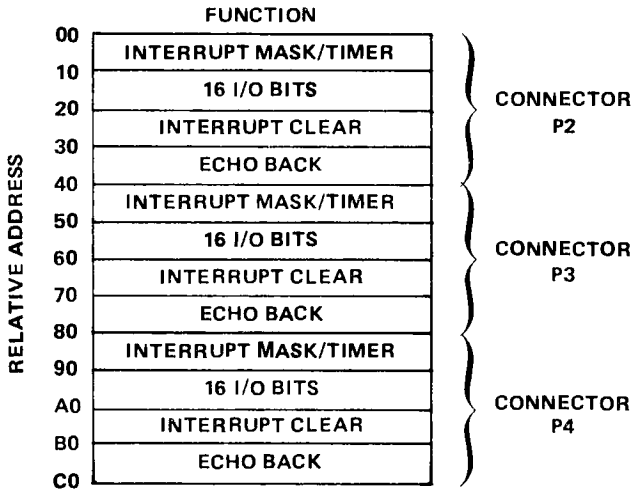
The TM 990/310 expansion module contains three I/O logic groups, each of which interfaces to separate connectors with 16 I/O lines. (Signal and ground are routed for each I/O line, and each line is equipped with pullup resistors.) Each I/O group may be programmed as 16 inputs, 16 outputs, nine interrupts, or any combination thereof. Each of the output lines is equipped with an echo-back feature which enables the user to read back each bit as it is written to a given output point. In addition, each connector contains a rising edge detect interrupt input and a falling edge detect interrupt input, along with +5 volts, +12 volts, and −12 volts power supply connections.



TM 990/310 Expansion Module, 16 I/O Lines, Logic Group Block Diagram (One of three groups)

ADDRESSING

The TM 990/310 I/O expansion module is addressed via the dedicated CRU interface over the system bus connector. Each I/O bit can be addressed individually; or up to 16 parallel ports can be addressed. Each 16-bit I/O line logic group has an addressing block of 64 bits, and each group can be stacked back to back. Each connector appears exactly the same; the functions and relative addresses for one TM 990/310 is shown below. The CRU address map permits addressing of 4K individual addresses. Any CRU bit beginning with 100_{16} can be addressed; the first FF_{16} bits are dedicated to the microcomputer module.

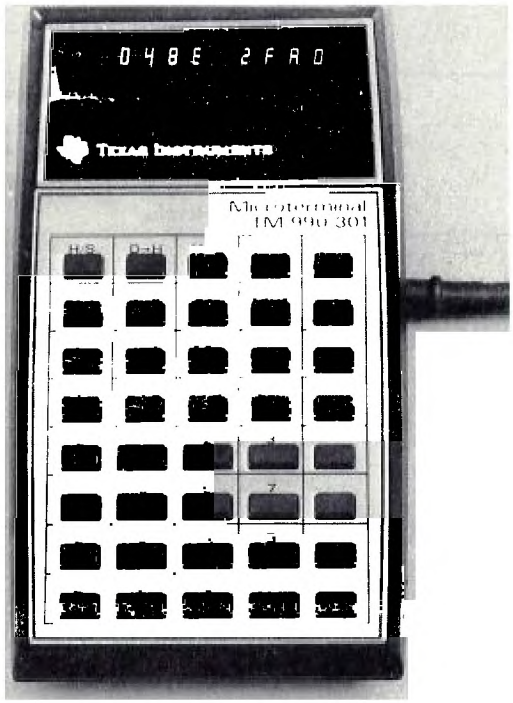


CRU ADDRESS MAP

▶8

SPECIFICATIONS

Input/Output	48 bits programmed as inputs, outputs, or up to 27 unlatched interrupts
Interrupts	33 maximum [six are (+) or (−) edge-detect latches] output of priority encoders may be jumpered to three levels of the 15 external TM 990 interrupts levels.
Interval Timers	Three 14-bit timers
Resolution	21.3μs
Maximum interval (for 3 MHz CPU clock)	349μs
Input Levels	
High-level input voltage	2.0 V nominal
Low-level input voltage	0.8 V nominal
Maximum input voltage range	−0.3 V to +10 V
Input current	10 kΩ (±10%) pullup to V_{cc}
Edge Detect Interrupts	
Positive-going threshold voltage	1.9 V maximum
Negative-going threshold voltage	0.5 V minimum
Hysteresis	0.4 V minimum, 0.8 V typical
Maximum input voltage range	−0.3 V to +5.5 V
High-level input current	−1.29 mA maximum at 2.7 V
Low-level input current	−2.85 mA maximum at 0.4 V
Outputs	
High-level output voltage	2.4 V minimum at −200 μA 2.0 V minimum at −600 μA
Low-level output voltage	0.4 V maximum at 2.2 mA
Input/Output Connectors	
40 Pin (3 each)	TI H31120 (wire wrap), Viking 3 VH20/1JN5 (solder tail), 3M 3464-0001 (ribbon cable), or equivalents.
100 pin	TI H43111150 (wire wrap), TI H431121-50 (solder tail), Viking 3 VH50/ICN5 (pierced tail), or TM 990/510 card cage
Power Requirements	5 V ±5%, 800 mA (typical)
Temperature Range	
Operating	0°C to 70°C
Storage	−65°C to 150°C
Physical Characteristics	
Width	11 inches (279,4 mm)
Height	7½ inches (190,5 mm)
Board thickness	0.062 inch (1,575 mm)
Component height	0.5 inch (12,7 mm) maximum



FEATURES

- Hexadecimal pushbutton keyboard
- Register, Memory, or CRU Display and entry keys
- Operations under TIBUG monitor
- 4 digit hexadecimal display of address and data
- Execution, single instruction and conversion keys

▶8 The TM 990/301 is a microterminal designed to interface with the TM 990 series of microcomputer modules. The microterminal's communications link to the TM 990 CPU module is via the EIA type cable and the serial terminal interface. The TM 990/301 performs the front panel functions of the microcomputer system, giving the programmer the ability to display and change register and memory information. This low cost terminal offers the capability to enter short programs in hexadecimal or alter a section of a longer sequence.

OPERATION

The TM 990/301 operates under control of the TM 990/401 TIBUG monitor. The data rate utilized is 110 baud. Once the CPU board is initialized, depressing the clear (CLR) key transfers TIBUG monitor control to the microterminal. The TIBUG software will enter a wait routine unless performing a function defined by the microterminal. Depressing the run (RUN) key will cause the CPU module to begin program execution and it will ignore other key depressions until the halt (HALT/SIE) key is depressed. If the CPU is halted, depression of the single instruction execution key (HALT/SIE) causes execution of the next instruction.

The display of the microterminal is divided into two 4 hexadecimal digit banks. The left bank displays address register information and the right bank displays data registers (see *Figure 1*).

KEY	***FUNCTION***
CLR	Clear — blank all displays — initialize software
RUN	Run — TM 990 CPU begins program execution; "RUN" is displayed in data digits.
HALT/SIE	Halt/Single Instruction Execution — If in run mode, halts CPU execution — address of next instruction displayed in address digits. If CPU is halted, one single instruction will be executed. Address display indicates address of next instruction; data display indicates contents of that location.
O—F	Hexadecimal digits (0—15) — data entry. F/- also indicates negative.
EPC	Enter Program Counter — Enter 4 digits, key depressions alters active program counter, data display indicates entered value.
DPC	Display Program Counter — Active PC register indicated in data display.
EST	Enter Status Register — Enter 4 digits — key depressions alters active status register data display indicates entered value.
DST	Display Status Register — Active status register indicated in data display.
EWP	Enter Workspace Pointer — Enter 4 digits — key depression alters active workspace pointer — data display indicates entered value.
DWP	Display Workspace Pointer — Active WP indicated in data displayed.
EMA	Enter Memory Address — Enter 4 digits — key depression will shift display of digit from data display to address display. Contents of the new memory address will then be indicated in the data display.
EMD	Enter Memory Data — After EMA function has been executed, enter 4 digits — the data display indicates the new data — key depression alters the data at the displayed memory address.

KEY	*FUNCTION*
EMDI	Enter Memory Data/Increment — Functions the same as EMD — after key depression of EMDI and data update the address display will automatically increment by 2 and the new addresses contents will be indicated by the data display. To increment the address register without altering data contents, depress EMDI key without entering new digit information.
DCRU	Display CRU Data — Enter 4 digits — the first digit specifies the CRU bit count; the remaining 3 digits specify the CRU address. Key depression shifts the entered digits to the address display and indicates the contents of that address in the data display. All 16 bits will be displayed.
ECRU	Enter CRU Data — After DCRU function has been executed, enter 4 digits — the new data is now indicated by the data display — key depression alters the data at the specified CRU address — only the number of bits specified will be altered.
H—→D	Hexadecimal to Decimal Conversion — Enter 4 digits — key depression will indicate the decimal equivalent in 5 rightmost display digits.
D—→H	Decimal to Hexadecimal Conversion — Enter 6 digits — the first digit designates the sign F/ − =negative, 0=positive) the remaining 5 are decimal data — key depression displays hexadecimal equivalent in 4 right digits.

▶8

SPECIFICATIONS

Display

8 digit hexadecimal display
4 left digits indicate address register
4 right digits indicate data register

Keyboard

16 data keys
16 function keys
8 keys not connected

Interface: Serial Asynchronous Interface
Signals Include
/HALT
TERMINAL DATA IN
TERMINAL DATA OUT
+12 V
GND
−12 V
+5

Power Requirements: Supplied through cable
+12 V @ 50 mA
−12 V @ 20 mA
− 5 V @ 150 mA

Operating Temperature Range
0°C to 50°C

Physical Dimensions
Height: 5.8 inches
Width: 3.2 inches
Thickness: 1.38 inches
Cable Length: 6 ft.

Ordering Information
TM 900/301

Microterminal compatible with all TM 990 series Microcomputer modules.

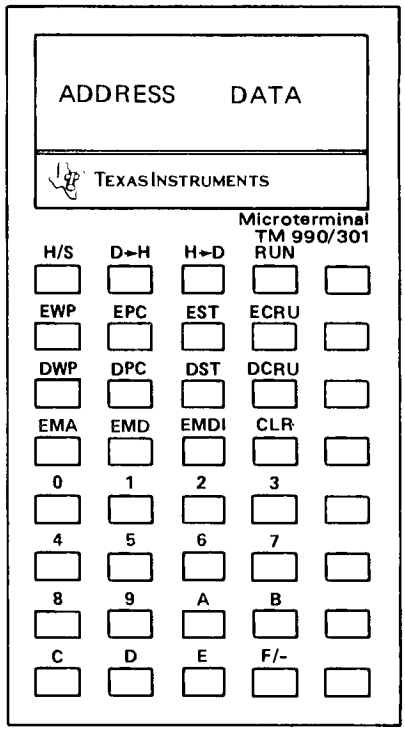


Figure 1. Microterminal Keyboard and Display.

8

TIBUG

TM 990/401 (TIBUG) is a comprehensive, interactive debug monitor which is included in the basic price of the CPU modules. (The OEM optionally may order the board with blank EPROM). TIBUG includes 13 user commands plus six user accessible utilities. TIBUG operates with 110, 300, 1200, and 2400 baud terminals. The user's manual for each CPU board includes a complete description of the use of TIBUG as well as a complete listing of the monitor. The TM 990/100M board TIBUG software is slightly different from the TM 990/180M board TIBUG software. Therefore, the TM 990/401-1 software is compatible with the TM 990/100M board, and the TM 990/401-2 software is compatible with the TM 990/180M board, and the TM 990/401-3 is compatible with the TM 990/101M board.

TIBUG Commands

B Executive and breakpoint on a specified address.
C Inspect/change the communications register unit.
D Dump memory to cassette or paper tape in 990 compatible tag format compatible with PX990, the TMS 9900 software development system.
E Execute
F Find word/byte
H Hexadecimal arithmetic
L Load memory from cassette or paper tape (compatible with TMS 9900 prototyping system formats).
M Inspect/change memory
R Inspect/change user program counter, workspace pointer, and status register.
S Execute user program in single/multiple steps.
W Inspect/change user register file.

User Accessible Utilities

- Read a character from the UART
- Write a character to the UART
- Hexadecimal numeric input
- Four-digit hexadecimal numeric output
- Single-digit hexadecimal numeric output
- ASCII message output

Line By Line Assembler

TM 990/402 is a line by line assembler supplied preprogrammed into a TMS 2708 EPROM Kit for immediate use in the system. These EPROM's insert into the extra sockets not required for the TIBUG monitor. It is an extremely useful tool for assembly language input of short programs or easy patching of longer programs.

The line by line assembler can be directly accessed from the TIBUG monitor by utilizing the "R" command and entering the proper program counter value. A complete User's Manual is included with the TMS 990/402 and a source listing can also be obtained. The TM 990/402-1 software is compatible with both the TM 990/100M and TM 990/101M board, and the TM 990/402-2 software is compatible with the TM 990/180M board.

Line By Line Assembler Input Commands

$ Convert symbolic constants from ASCII to hexadecimal and store in memory.
+,- Enter numeric constant
/ Change program counter
ESC Return control to monitor

Assembly Instruction.

Enter the instruction mnemonic and operand field — all allowable TMS 9900 instructions and addressing modes are recognized. Displacements are allowed on either an absolute basis or relative with respect to the current instruction designated by $-n. The displacement range is +254, -256 bytes.

Assembler Error Messages

*S Syntax error
*D Displacement error jump target address
*R Range error — current field has erroneous input

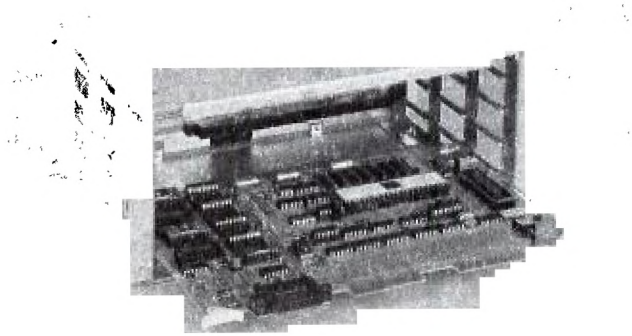
8

CARD CAGE

The TM 990/510 is an OEM card cage with four slots on 1 inch centers. The backpanel contains the address bus, data bus, interrupt and control lines to permit memory, I/O and DMA expansion of CPU modules. A 10-terminal barrier strip is mounted on the backpanel to permit connection of the following signals as the system requires:

- Reset
- Restart
- Power Down Interrupt
- ± 5V
- ± 12V
- ± 15 V
- GND

The outside dimensions of the OEM card cage are 5 inches (127 MM) high, 12.5 inches (317,5mm) wide, and 8 inches (203,2 mm) deep.

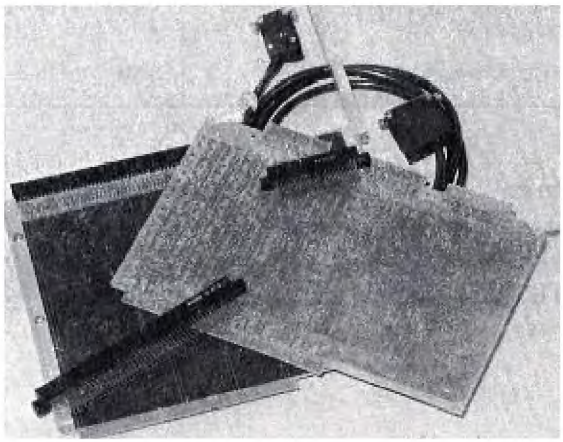


TM 990/510 OEM Card Cage

8

EXTENDER BOARD

The TM 990/511 is a connector bus compatible extender board. The extender board has a printed circuit tab and a connector with card guide at its edges.



PROTOTYPING BOARD

The TM 990/512 is a universal prototyping card. The printed circuit board is designed to accommodate 0.3, 0.4, 0.6 and 0.9 inch (7,62, 16,16, 15,24, and 22,86 mm) wide, dual in line, IC packages or their equivalent soldertail or wirewrap sockets. The TM 990/512 has GND and 5 V planes, two power strips for ± 12 V, plus two power strips for user-selectable voltage option.

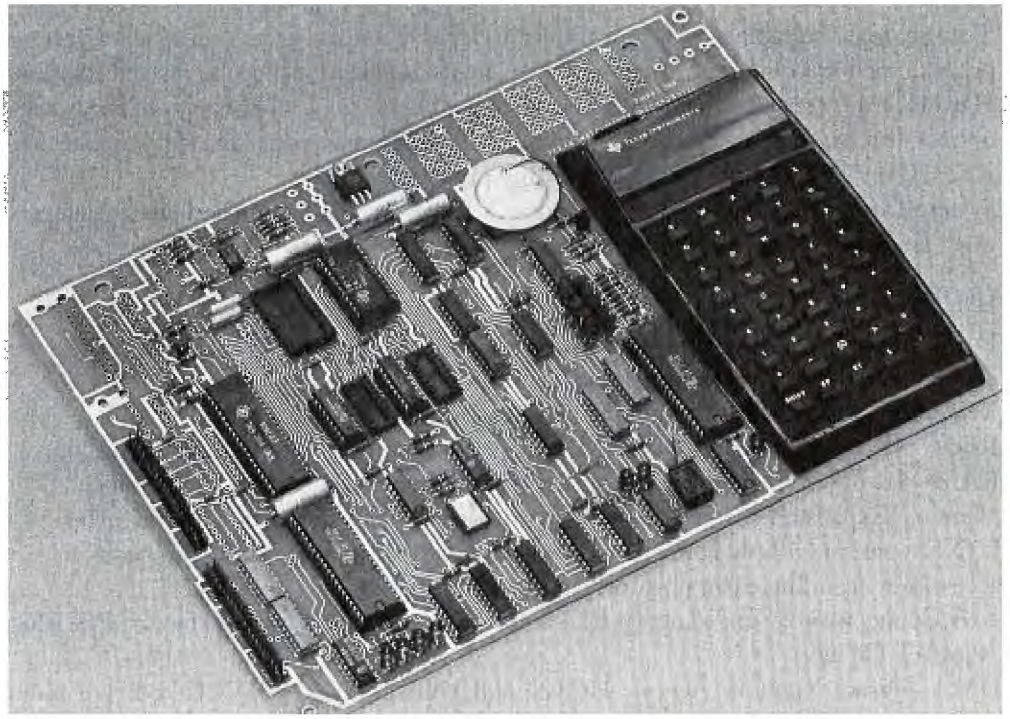
CONNECTORS AND CABLES

The bottom edge bus connector is a 100-pin printed circuit tap on 0.125 inch (3,175 mm) centers, compatible with the TM 990 bus specification. Two top edge, 40-pin, printed circuit tab connectors for I/O or interrupt are on 0.10 (2,54 mm) centers. Additionally, the CPU modules come mounted with a 25-pin EIA connector, and the universal prototyping boards have space for mounting the same. The TM 990/501 is a connector kit offered for all of the connectors. The parts supplied in this kit are readily available, multisource connectors. The connectors available through TI as part of the TM 990/501 kit are

QTY	*Description*	*Part Number*
1	25 pin male connector	AMP DB-25P or ITT Cannon DB-25P
1	40 pin solder eyelet female edge connector	Viking 3VH20/1JN5 or TI H421121-20
1	100 pin solder wire wrap female edge connector	Viking 3VH50/1CN5 or TI H431111-50
1	Hood for 25 pin connector	AMP 206478-3

Various cable kits are also offered. TM 990/502 is a general purpose EIA cable compatible with terminals which have female RS-232 connectors. TM 990/503 is an EIA cable compatible with 743 KSR or 745 terminals. The TM 990/504 is a standard current loop TTY cable. The TM 990/505 is compatible with the 733 ASR.

TM990/189 UNIVERSITY MICROCOMPUTER BOARD



TM990/189 University Microcomputer Board

The TM990/189 is a self-contained microcomputer system (available in kit form or fully assembled and tested) which is designed primarily as a learning tool for the engineer, student or hobbyist. It can be used as an aid in the instruction of microcomputer fundamentals, machine and assembly level language programming and microcomputer interfacing as well as demonstrating the power of the 9900 family 16-bit architecture. The board utilizes the powerful NMOS 16 bit TMS 9980 microprocessor as its CPU. Additionally, this extremely economical board has the following exciting features:

8

SOFTWARE

- Unibug—user interactive software debug monitor, ROM resident.
- Line by Line Assembler with forward references—assembles 9900 instructions into machine level code and allows for the use of labels.

TEXTS

- Tutorial text—suitable as a 3-hour university course outline or as a stand-alone self-paced programmed learning text. Includes experiments, applications, problems and solutions.
- Hardware reference manual—describes theory of operations, connection of hardware options (for example, expansion memory, I/O expansion, audio cassette interface, and RS232C or TTY options) and kit assembly procedures.
- System Design Handbook—design handbook featuring application and design references to all members of the TMS 9900 family.

HARDWARE

- Power Supply—included as a standard or available as an option.
- Display—10 digit seven segment display.
- Keyboard—45 keys, full alpha-numeric keyboard.
- ROM—4K bytes of dedicated ROM, sockets for additional 2k on-board expansion.
- RAM—1K bytes RAM including sockets for additional 1K on-board expansion.
- Memory Expansion—50-pin connector provided to expand memory to 16K bytes.
- I/O Expansion—50-pin connector is provided with all standard CPU bus signals accessible, including extra pins for user-defined functions.
- EIA Connector—25-pin standard EIA connector is provided for interface to RS232C or 20 mA TTY loop.
- DMA—Direct Memory Access. HOLD and HOLD ACKNOWLEDGE are brought out to external pins.
- Visual Indicators—7 LED's are provided: 3 dedicated and 4 user-defined.
- Acoustical Indicator—Piezoelectric disk for audio reference.
- Audio Cassette—single audio cassette interface.
- I/O—16 I/O lines are provided.

ORDERING INFORMATION

TM990/189 K	Kit form less power supply
TM990/189 M	Assembled and tested microcomputer module.
TM990/519	Power supply for TM990/189 K or TM990/189 M including all interconnecting cables.

▶8

Memory

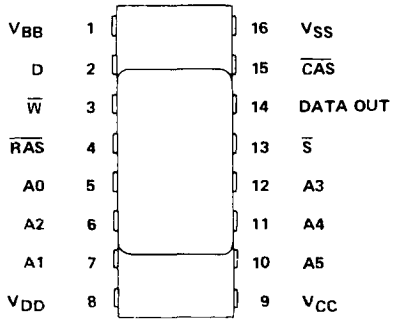
TMS 4027 JL, NL
4096-BIT DYNAMIC RANDOM-ACCESS MEMORY

- 4096 X 1 Organization
- Industry Standard 16-Pin 300-Mil Package Configuration
- 10% Tolerance on All Supplies
- All Inputs Including Clocks TTL Compatible
- Three-State Fully TTL-Compatible Output Latched and Valid Into Next Cycle
- 3 Performance Ranges:

	ACCESS TIME ROW ADDRESS (MAX)	ACCESS TIME COLUMN ADDRESS (MAX)	READ OR WRITE CYCLE (MIN)	READ, MODIFY-WRITE† CYCLE (MIN)
TMS 4027-15	150 ns	100 ns	320 ns	330 ns
TMS 4027-20	200 ns	135 ns	375 ns	420 ns
TMS 4027-25	250 ns	165 ns	375 ns	480 ns

- Page-Mode Operation for Faster Access Time
- Low-Power Dissipation
  - Operating 460 mW (max)
  - Standby 27 mW (max)
- 1-T Cell Design, N-Channel Silicon-Gate Technology
- Refresh time: 2 ms

16-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



PIN NAMES

A0–A5	Address Inputs
$\overline{CAS}$	Column address strobe
D	Data input
DATA OUT	Data output
$\overline{RAS}$	Row address strobe
$\overline{S}$	Chip select
$\overline{W}$	Write enable
V_{BB}	–5 V power supply
V_{CC}	+5 V power supply
V_{DD}	+12 V power supply
V_{SS}	0 V ground

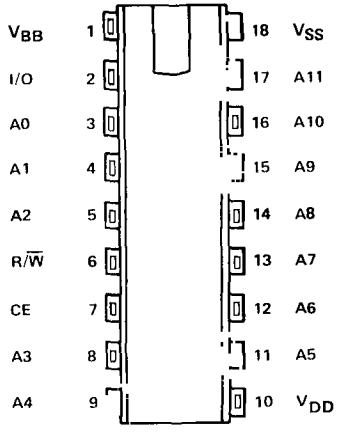
TMS 4050 JL, NL
4096-BIT DYNAMIC RANDOM-ACCESS MEMORIES

- 4096 x 1 Organization
- 18-Pin 300-Mil Package Configuration
- Multiplexed Data Input/Output
- 3 Performance Ranges:

	ACCESS TIME (MAX)	READ OR WRITE CYCLE (MIN)	READ, MODIFY WRITE CYCLE (MIN)
TMS 4050	300 ns	470 ns	730 ns
TMS 4050-1	250 ns	430 ns	660 ns
TMS 4050-2	200 ns	400 ns	600 ns

- Full TTL Compatibility on All Inputs (No Pull-up Resistors Needed)
- Registers for Addresses Provided on Chip
- Open-Drain Output Buffer
- Single Low-Capacitance Clock
- Low-Power Dissipation
  - 420 mW Operating (Typical)
  - 0.1 mW Standby (Typical)
- N-Channel Silicon-Gate Technology
- Refresh time: 2 ms or less

18-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



VDD +12V power supply
VSS 0V ground
VBB – 5V power supply

8

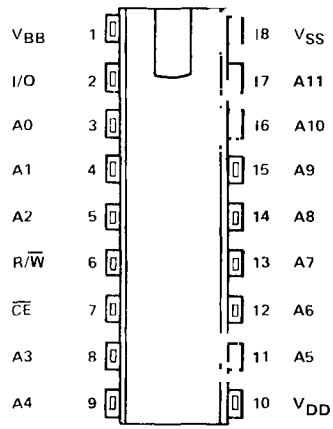
TMS 4051 JL, NL
4096-BIT DYNAMIC RANDOM ACCESS MEMORIES

- 4096 x 1 Organization
- 18-Pin 300-Mil Package Configuration
- Single Low-Capacitance TTL-Compatible Clock
- Multiplexed Data Input/Output
- 2 Performance Ranges:

	ACCESS TIME (MAX)	READ OR WRITE CYCLE (MIN)	READ, MODIFY WRITE CYCLE (MIN)
TMS 4051	300 ns	470 ns	730 ns
TMS 4051-1	250 ns	430 ns	660 ns

- Full TTL Compatibility on All Inputs (No Pull-up Resistors Needed Except with $\overline{CE}$)
- Registers for Addresses Provided on Chip
- Open-Drain Output Buffer
- Low-Power Dissipation
  - 460 mW Operating (Typical)
  - 60 mW Standby (Typical)
- N-Channel Silicon-Gate Technology
- Refresh time: 2 ms or less

18-PIN CERAMIC AND PLASTIC DUAL-IN-LINE PACKAGES (TOP VIEW)



VDD +12V power supply
VSS 0V ground
VBB − 5V power supply

TMS 4060 JL, NL
4096-BIT DYNAMIC RANDOM-ACCESS MEMORIES

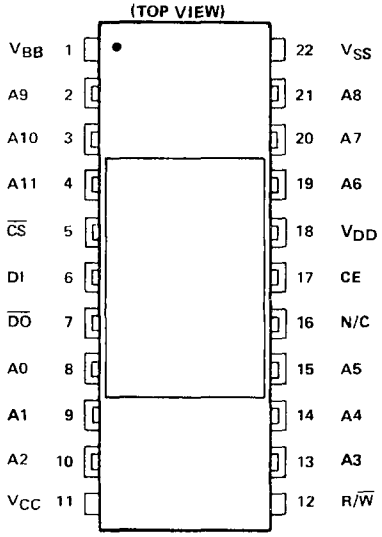
MOS
LSI

- 4096 x 1 Organization
- 3 Performance Ranges:

	ACCESS TIME (MAX)	READ OR WRITE CYCLE (MIN)	READ, MODIFY WRITE CYCLE (MIN)
TMS 4060	300 ns	470 ns	710 ns
TMS 4060-1	250 ns	430 ns	640 ns
TMS 4060-2	200 ns	400 ns	580 ns

- Full TTL Compatibility on All Inputs Except CE (No Pull-Up Resistors Needed)
- Low Power Dissipation
  - 400 mW Operating (Typical)
  - 0.2 mW Standby (Typical)
- Single Low-Capacitance Clock
- N-Channel Silicon-Gate Technology
- 22-Pin 400-Mil Dual-in-Line Package
- Refresh time: 2 ms or less

22-PIN CERAMIC AND PLASTIC DUAL-IN-LINE PACKAGES (TOP VIEW)



Vcc + 5V power supply
Vss 0V ground
VDD +12V power supply
VBB − 5V power supply

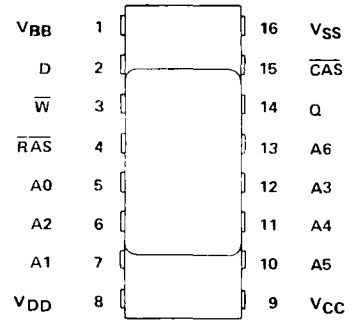
TMS 4116 JL
16,384-BIT DYNAMIC RANDOM-ACCESS MEMORY

- 16,384 X 1 Organization
- 10% Tolerance on All Supplies
- All Inputs Including Clocks TTL Compatible
- Unlatched Three-State Fully TTL-Compatible Output
- 3 Performance Ranges:

	ACCESS TIME ROW ADDRESS (MAX)	ACCESS TIME COLUMN ADDRESS (MAX)	READ OR WRITE CYCLE (MIN)	READ, MODIFY-WRITE† CYCLE (MIN)
TMS 4116-15	150 ns	100 ns	375 ns	375 ns
TMS 4116-20	200 ns	135 ns	375 ns	375 ns
TMS 4116-25	250 ns	165 ns	410 ns	515 ns

- Page-Mode Operation for Faster Access Time
- Common I/O Capability with "Early Write" Feature
- Low-Power Dissipation
  - Operating 462 mW (max)
  - Standby 20 mW (max)
- 1-T Cell Design, N-Channel Silicon-Gate Technology
- 16-Pin 300-Mil Package Configuration
- Refresh time: 2 ms or less

16-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)

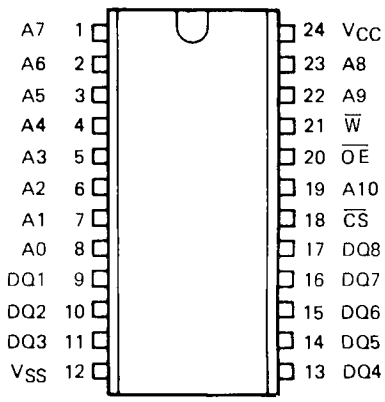


PIN NOMENCLATURE			
A0-A6	Address Inputs	$\overline{W}$	Write Enable
$\overline{CAS}$	Column address strobe	V_{BB}	−5 V power supply
D	Data input	V_{CC}	+5 V power supply
Q	Data output	V_{DD}	+12 V power supply
$\overline{RAS}$	Row address strobe	V_{SS}	0 V ground

8

- **2K × 8 Organization**
- **Single +5 V Supply (±10% Tolerance)**
- **Fully Static Operation (No Clocks, No Refresh)**
- **JEDEC Proposed Standard Pinout**
- **24-Pin 600 Mil Package Configuration**
- **Plug-in Compatible with 16K 5V EPROMs**
- **8-Bit Output for Use in Microprocessor-Based Systems**
- **Max Access/Min Cycle Times Down to 150 ns**
- **Tri-State Outputs with $\overline{CS}$ for Or-ties**
- **$\overline{OE}$ Eliminates Need for External Bus Buffers**
- **Common I/O Capability**
- **All Inputs and Outputs Fully TTL Compatible**
- **Fanout to Series 74, Series 74S, or Series 74LS TTL Loads**
- **N-Channel Silicon-Gate Technology**
- **Power Dissipation Under 495 mW Max**
- **Guaranteed dc Noise Immunity of 400 mV with Standard TTL Loads**

TMS 4016
24-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A0-A10	Addresses
DQ1-DQ8	Data In/Data Out
$\overline{CS}$	Chip Select
$\overline{OE}$	Output Enable
$\overline{W}$	Write Enable
V_{SS}	Ground
V_{CC}	+5 V Supply

description

The TMS 4016 static random-access memory is organized as 2048 words of 8 bits each. Fabricated using proven N-channel, silicon-gate MOS technology, the TMS 4016 operates at high speeds and draws less power per bit than 4K static RAMs. It is fully compatible with Series 74, 74S, or 74LS TTL. Its static design means that no refresh clocking circuitry is needed and timing requirements are simplified. Access time is equal to cycle time. A chip select control is provided for controlling the flow of data-in and data-out and an output enable function is included in order to eliminate the need for external bus buffers.

Of special importance is that the TMS 4016 static RAM has the same standardized pinout as TI's compatible EPROM family. This, along with other compatible features, makes the TMS 4016 directly plug-in compatible with the TMS 2516 (or other 16K 5V EPROMs). No modifications are needed. This allows the microprocessor system designer complete flexibility in partitioning his memory board between read/write and non-volatile storage. A more detailed explanation of this compatibility is given on the reverse side.

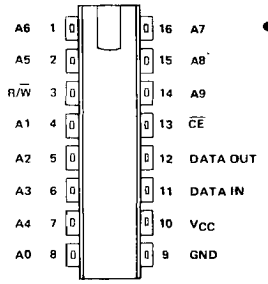
PREVIEW
This document contains the design specifications for a product under development. Specifications may be changed in any manner without notice.

TMS 4033, 4034, 4035, 4039, 4042, 4043, JL, NL
1024-BIT AND TMS 4036 NL 512-BIT STATIC RAMs

- No clocks — No refresh
- Input/Output fully TTL compatible
- Three-state output for OR-Tie capability
- Single 5-volt supply
- Simple, fully decoded addressing
- Reliability, same process as TI's industry standard 4K RAMs
- Economy, high volume production techniques and choice of plastic or ceramic packaging
- Super low standby power (typical 2-3 mW)
- Wide range of speeds for design optimization
- Easy to use 8-bit byte organization (64 x 8) plus industry standard 256 x 4 and 1024 x 1

TMS 4033, TMS 4034, TMS 4035

16 PIN CERAMIC AND PLASTIC DUAL IN-LINE PACKAGES (TOP VIEW)

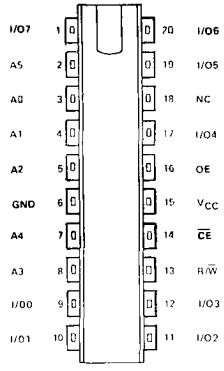


- 1024 x 4 x 1 Organization
- 225 mW Typical power dissipation

Part Number	Max Access/ Min Cycle
TMS 4033/2102 1 JL, NL	450 ns
TMS JL, NL	650 ns
TMS L, NL	1000 ns

TMS 4036

20 PIN PLASTIC DUAL IN LINE PACKAGES (TOP VIEW)

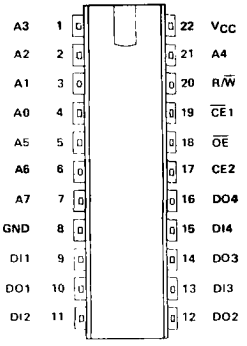


- 64 x 8 Organization
- 250 mW Typical power dissipation
- On-chip multiplexed I/O for microprocessor-oriented systems

Part Number	Max Access/ Min Cycle
TMS 4036 NL	1000 ns
TMS 4036 1 NL	650 ns
TMS 4036 2 NL	450 ns

TMS 4039

22 PIN CERAMIC AND PLASTIC DUAL IN LINE PACKAGES (TOP VIEW)

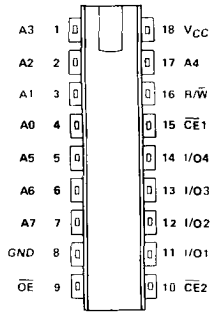


- 256 x 4 Organization
- 175 mW Typical power dissipation
- Separate input and output
- 2 chip enables for maximum control

Part Number	Max Access/ Min Cycle
TMS 4039/2101 JL, NL	1000 ns
TMS 4039 1/2101 2 JL, NL	650 ns
TMS 4039 2/2101 1 JL, NL	450 ns

TMS 4042

18 PIN CERAMIC AND PLASTIC DUAL IN LINE PACKAGES (TOP VIEW)

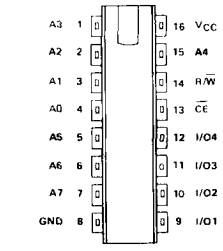


- 256 x 4 Organization
- 175 mW Typical power dissipation
- Common I/O for Bus-oriented systems

Part Number	Max Access/ Min Cycle
TMS 4042/2111 JL, NL	1000 ns
TMS 4042 1/2111 2 JL, NL	650 ns
TMS 4042 2/2111 1 JL, NL	450 ns

TMS 4043

16 PIN CERAMIC AND PLASTIC DUAL IN LINE PACKAGES (TOP VIEW)



- 256 x 4 Organization
- 175 mW Typical power dissipation
- Provides common I/O and highest packing density

Part Number	Max Access/ Min Cycle
TMS 4043/2112 JL, NL	1000 ns
TMS 4043 1/2112 2 JL, NL	650 ns
TMS 4043 2 JL, NL	450 ns

8

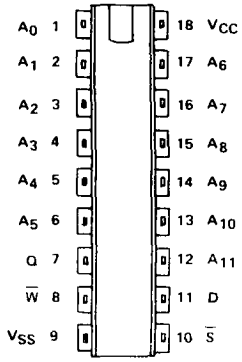
- **4096 × 1 Organization**
- **Single +5 V Supply (±10% Tolerance)**
- **High Density 300-mil 18- and 20-Pin Packages**
- **Fully Static Operation (No Clocks, No Refresh, No Timing Strobe)**
- **4 Performance Ranges:**

	ACCESS TIME (MAX)	READ OR WRITE CYCLE (MIN)
TMS 4044/L44-45, TMS 4046/L46-45	450 ns	450 ns
TMS 4044/L44-25, TMS 4046/L46-25	250 ns	250 ns
TMS 4044/L44-20, TMS 4046/L46-20	200 ns	200 ns
TMS 4044-15, TMS 4046-15	150 ns	150 ns

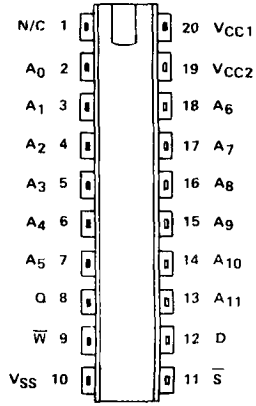
- **400 mV Guaranteed DC Noise Immunity with Standard TTL Loads — No Pull-Up Resistors Required**
- **Common I/O Capability**
- **3-State Outputs and Chip Select Control for OR-Tie Capability**
- **Fan-Out to 2 Series 74, 1 Series 74S, or 8 Series 74LS TTL Loads**
- **Low Power Dissipation**

	MAX (OPERATING)	MAX (STANDBY)
TMS 4044	440 mW	156 mW
TMS 40L44	275 mW	96 mW
TMS 4046	440 mW	13 mW
TMS 40L46	275 mW	13 mW

TMS 4044/TMS 40L44
18-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



TMS 4046/TMS 40L46
20-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



PIN NAMES

A_0-A_{11}	Addresses
D	Data In
Q	Data Out
$\overline{S}$	Chip Select
V_{CC} (TMS 4044/L44)	+5 V Supply
V_{CC1} (TMS 4046/L46)	+5 V Supply (array only)
V_{CC2} (TMS 4046/L46)	+5 V Supply (periphery only)
V_{SS}	Ground
$\overline{W}$	Write Enable

▸8

TMS 40L45 JL, NL; TMS 40L47 JL, NL
1024-WORD BY 4-BIT STATIC RAMs

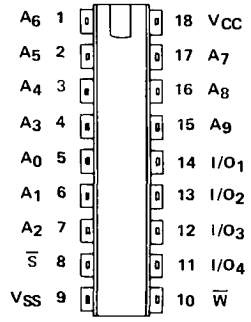
- 1024 x 4 Organization
- Single 10% Tolerance 5-V Supply
- High Density 300-mil 18- and 20-Pin Packages
- Fully Static Operation (No Clocks, No Refresh, No Timing Strobe)
- 3 Performance Ranges:

	ACCESS TIME (MAX)	READ OR WRITE CYCLE (MIN)
TMS 40L45-25, TMS 40L47-25	250 ns	250 ns
TMS 40L45-30, TMS 40L47-30	300 ns	300 ns
TMS 40L45-45, TMS 40L47-45	450 ns	450 ns

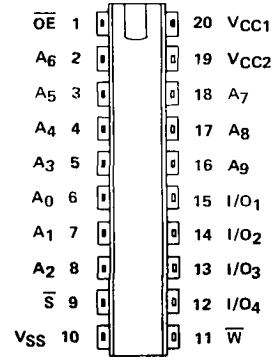
- 400-mV Guaranteed Noise Immunity With Standard TTL Loads – No Pull-Up Resistors Required
- Common I/O With Three-State Outputs and Chip Select Control for OR-Tie Capability
- Fan-Out to 1 Series 74 or 74S TTL Load – No Pull-Up Resistors Required
- Low Power Dissipation
  - 250 mW *Typical
  - 370 mW *Maximum
- Standby Power Dissipation (TMS 40L47)
  - 12 mW Typical
  - 24 mW Maximum

PIN NAMES	
A_0-A_9	Addresses
I/O_1-I/O_4	Data input/output
$\overline{OE}$	Output Enable
$\overline{S}$	Chip Select
V_{CC} (TMS 40L45)	+5-V Supply
V_{CC1} (TMS 40L47)	+5-V Supply (array only)
V_{CC2} (TMS 40L47)	+5-V Supply (periphery only)
V_{SS}	Ground
$\overline{W}$	Write Enable

TMS 40L45
18-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



TMS 40L47
20-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



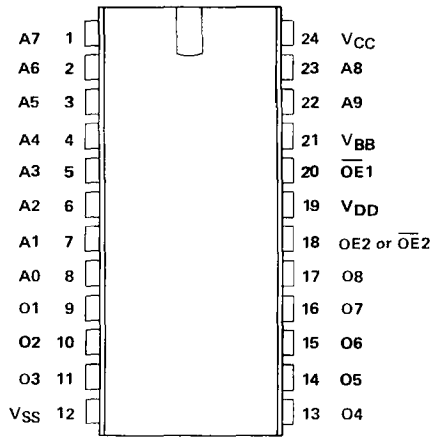
8

TMS 4700 JL, NL
1024-WORD BY 8-BIT READ-ONLY MEMORY

MOS
LSI

- 1024 x 8 Organization
- All Inputs and Outputs TTL-Compatible
- Maximum Access Time . . . 450 ns
- Minimum Cycle Time . . . 450 ns
- Typical Power Dissipation . . . 310 mW
- 3-State Outputs for OR-Ties
- Output Enable Control
- Silicon-Gate Technology
- 8-Bit Output for use in Microprocessor Based Systems
- Pin-compatible with TMS 2708, TMS 27L08, and Intel 2308
- **Inputs require external pull-up resistors for TTL-compatibility**

24-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



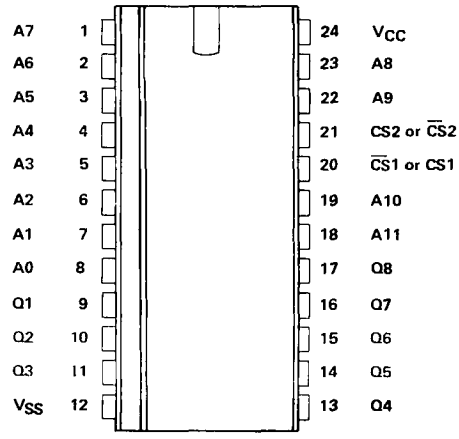
V_{BB}	−5V power supply
V_{CC}	+5V power supply
V_{DD}	+12V power supply
V_{SS}	0V ground

TMS 4732 JL, NL
4096-WORD BY 8-BIT READ-ONLY MEMORY

MOS
LSI

- 4096 x 8 Organization
- All Inputs and Outputs TTL-Compatible
- Fully Static (No Clocks, No Refresh)
- Single 5 V Power Supply
- Maximum Access Time . . . 450 ns
- Minimum Cycle Time . . . 450 ns
- Typical Power Dissipation . . . 580 mW
- 3-State Outputs for OR-Ties
- Pin Compatible with TMS 4700, TMS 2708 and Intel 8316B
- Two Output Enable Controls for Chip Select Flexibility
- N-Channel Silicon-Gate Technology

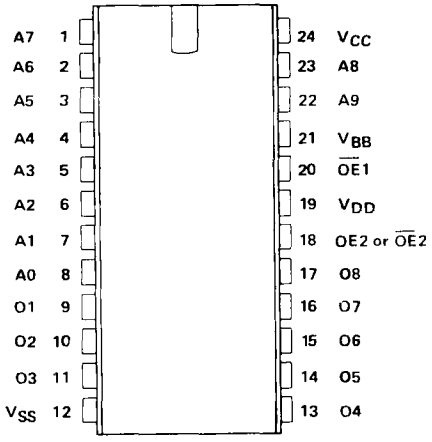
24-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



TMS 4710 JL, NL COMPLETE ASCII CHARACTER SET GENERATOR 5x7 CHARACTER, 8x8 BLOCK

- TMS 4710 (Standard TMS 4700 8K ROM)
- Full Upper and Lower Case ASCII Character Generator
- Ideal for Video Terminal Applications
- Fully Static Operation
- Block Size 8 x 8
- Character Size 5 x 7
- 1024 x 8 Organization
- All Inputs and Outputs TTL-Compatible
- Maximum Access Time . . . 450 ns
- Minimum Cycle Time . . . 450 ns
- Typical Power Dissipation . . . 310 mW
- 3-State Outputs for OR-Ties
- Output Enable Control
- Silicon-Gate Technology
- 8-Bit Output for use in Microprocessor Based Systems

24-PIN CERAMIC AND PLASTIC
DUAL-IN-LINE PACKAGES
(TOP VIEW)



8

TYPES SBP 8316, SBP 9818
16,384 I^2L READ-ONLY MEMORIES

- Mask Programmable I^2L ROM
- Fully TTL Compatible Inputs/Outputs
- Programmable Options Include:
  - User Selectable Speed/Power Operation:
    - Wide Range for Injector Current Supply Operation (SBP 9818)
    - Resistor Options for 5-Volt Supply Operation (SBP 8316)
  - Choice of Outputs:
    - Open-Collector for V_{CC} or INJ Operation
    - Internal 10K Ω Pull-Up Resistors to V_{CC} (SBP 8316)
  - Choose Any Combination of Up to 3 Boolean Variables for Chip Select or 2 Boolean Variables with Latched Outputs
- Industry Standard Pin Assignments in 24-Pin Plastic or C-DIP Packages
- Choice of Temperature Ranges:
  - SBP 8316CN, SBP 9818CN for 0 to 70°C Applications
  - SBP 8316MJ, SBP 9818MJ for −55°C to 125°C Applications
- Single+5-V power supply for the SBP8316
- Injector current of 500 mA maximum for the SPB9818

J OR N PACKAGE
(TOP VIEW)

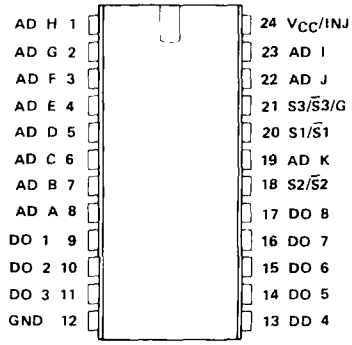
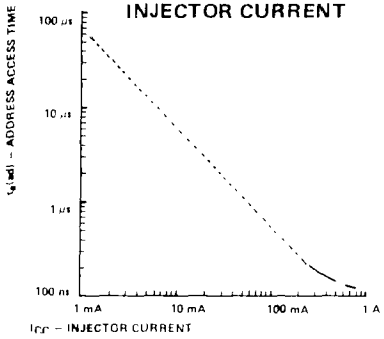


FIGURE 1
ADDRESS ACCESS TIMING
VS.
INJECTOR CURRENT

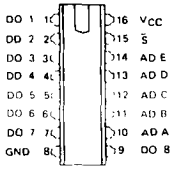


SERIES 54S/74S
PROGRAMMABLE READ-ONLY MEMORIES

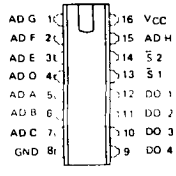
- Titanium-Tungsten (Ti-W) Fuse Links for Fast, Low-Voltage, Reliable Programming
- All Schottky-Clamped PROM's Offer:
  - Fast Chip Select to Simplify System Decode
  - Choice of Three-State or Open-Collector Outputs
  - P-N-P Inputs for Reduced Loading on System Buffers/Drivers
- Single 5-V power supply
- Full Decoding and Chip Select Simplify System Design
- Applications Include:
  - Microprogramming/Firmware Loaders
  - Code Converters/Character Generators
  - Translators/Emulators
  - Address Mapping/Look-Up Tables

TYPE NUMBER (PACKAGES)		BIT SIZE (ORGANIZATION)	OUTPUT CONFIGURATION	TYPICAL PERFORMANCE	
−55°C to 125°C	0°C to 70°C			ADDRESS ACCESS TIME	POWER DISSIPATION
SN54S188(J, W)	SN74S188(J, N)	256 bits	open-collector	25 ns	400 mW
SN54S288(J, W)	SN74S288(J, N)	(32 W x 8 B)	three-state		
SN54S287(J, W)	SN74S287(J, N)	1024 bits	three-state	42 ns	500 mW
SN54S387(J, W)	SN74S387(J, N)	(256 W x 4 B)	open-collector		
SN54S470(J)	SN74S470(J, N)	2048 bits	open-collector	50 ns	550 mW
SN54S471(J)	SN74S471(J, N)	(256 W x 8 B)	three-state		
SN54S472(J)	SN74S472(J, N)	4096 bits	three-state	55 ns	600 mW
SN54S473(J)	SN74S473(J, N)	(512 W x 8 B)	open-collector		
SN54S474(J, W)	SN74S474(J, N)	4096 bits	three-state	55 ns	600 mW
SN54S475(J, W)	SN74S475(J, N)	(512 W x 8 B)	open-collector		

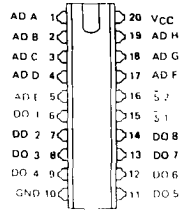
256 BITS (32 WORDS BY 8 BITS) 'S188, 'S288



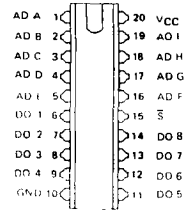
1024 BITS (256 WORDS BY 4 BITS) 'S287, 'S387



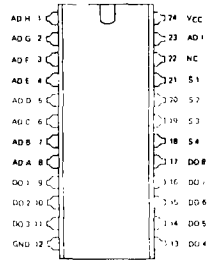
2048 BITS (256 WORDS BY 8 BITS) 'S470, 'S471



4096 BITS (512 WORDS BY 8 BITS) 'S472, 'S473



4096 BITS (512 WORDS BY 8 BITS) 'S474, 'S475



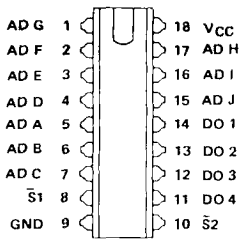
Pin assignments for all of these memories are the same for all packages.

8

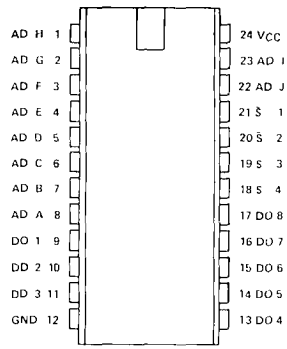
- Titanium-Tungsten (Ti-W) Fuse Links for Fast Low-Voltage Reliable Programming
- Full Decoding and Chip Select Simplify System Design
- Single 5-V power supply
- Fast Chip Select to Simplify System Decode
- Choice of Three-State or Open Collector Outputs
- PNP Inputs for Reduced Loading on System Buffers/Drivers
- Applications Include:
  Microprogramming/Firmware Loaders
  Code Converters/Character Generators
  Translators/Emulators
  Address Mapping/Look-Up Tables

TYPE NUMBER (PACKAGES)		BIT SIZE (ORGANIZATION)	OUTPUT CONFIGURATION	TYPICAL PERFORMANCE		
				ACCESS TIMES		POWER
–55°C to 125°C	0°C to 70°C			ADDRESS	SELECT	DISSIPATION
SN54S476(J)	SN74S476(J,N)	4096 bits	three-state	35 ns	15 ns	475 mW
SN54S477(J)	SN74S477(J,N)	1024 W x 4 B	open-collector			
[illegible]	SN74S[illegible],N)	8192 bits	three-state	45 ns	20 ns	600 mW
SN54S479(J)	SN74S479(J,N)	1024 W x 8 B	open-collector			

SN54S/74S476 3-S OUTPUTS
SN54S/74S477 0-C OUTPUTS
4096 BITS
(1024 WORDS BY 4 BITS)
'S476, 'S477



SN54S/74S478 3-S OUTPUTS
SN54S/74S479 0-C OUTPUTS
8192 BITS
(1024 WORDS BY 8 BITS)
'S478, 'S479



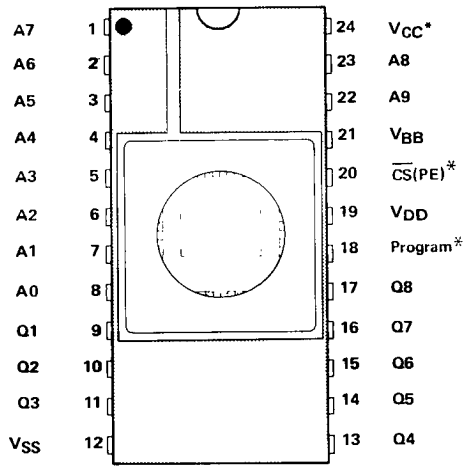
MAXIMUM DELAY TIMES

TYPE	ADDRESS	EN	DISABLE
SN54S'	75 ns	40 ns	40 ns
SN74S'	60 ns	30 ns	30 ns

TMS 2708 JL, TMS 27L08 AND TMS 2716 JL
8K AND 16K ERASABLE PROGRAMMABLE ROMs

- 2708 JL and 27L08 JL – 1024 X 8 Organization
- 2716 JL 2048 X 8 Organization
- All Inputs and Outputs Fully TTL-Compatible
- Static Operation (No Clocks, No Refresh)
- Maximum Access Time . . . 450 ns
- Minimum Cycle Time . . . 450 ns
- 3-State Outputs for OR-Ties
- N-Channel Silicon-Gate Technology
- 8-Bit Output for Use in Microprocessor-Based Systems
- **Low Power**
  TMS 27L08 . . . 245 mW (Typical)
  TMS 2716 . . . 315 mW (Typical)
- 10% Power Supply Tolerance (TMS 27L08 Only)
- Plug-Compatible Pin-Outs Allowing Interchangeability/Upgrade to 16K With Minimum Board Change

24-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



*For 2716 JL Only:

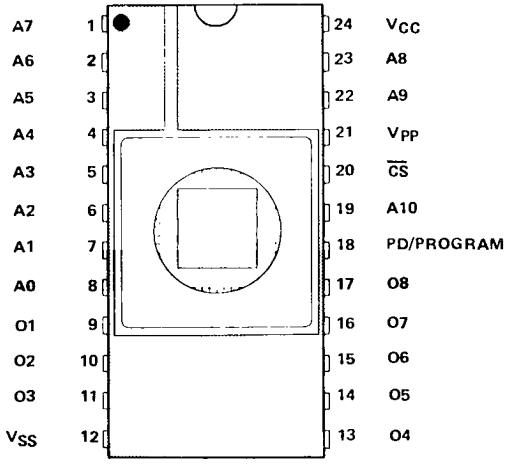
Pin:	
18	$\overline{CS}$ (Program)
20	A10
24	V_{CC} (PE)

V_{BB}	–5V power supply
V_{CC}	+5V power supply
V_{DD}	+12V power supply
V_{SS}	0V ground

TMS 2516 JL
16K ERASABLE PROGRAMMABLE READ-ONLY MEMORIES

- 2048 x 8 Organization
- Single +5 V Power Supply
- All Inputs and Outputs Fully TTL-Compatible
- Maximum Access Time . . . 450 ns
- Minimum Cycle Time . . . 450 ns
- 3-State Outputs for OR-Ties
- 8-Bit Output for Use in Microprocessor Based Systems
- N-Channel Silicon-Gate Technology
- Low Power: 525 mW Maximum Active Power
  132 mW Maximum Standby Power
- Guaranteed d.c. Noise Immunity with Standard TTL Loads – No Pull-Up Resistors Required
- Interchangeable with Intel 2716

24-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)

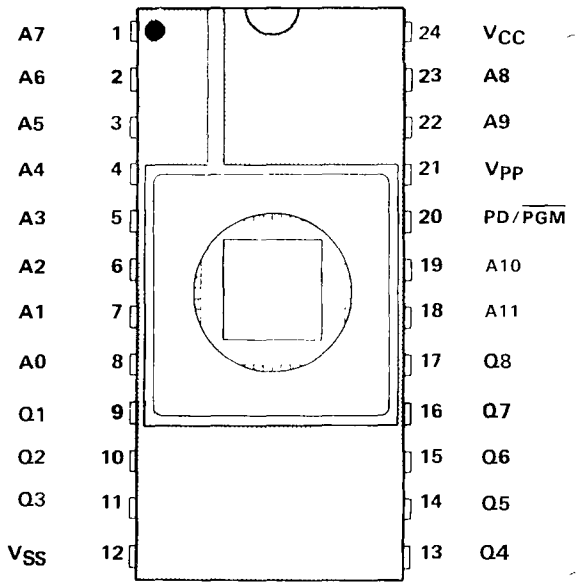


8

TMS 2532 JL
32K ERASABLE PROGRAMMABLE ROMs

- 4096x8 Organization
- Single +5V Power Supply
- Pin Compatible with all 8K and 16K EPROMs
- Plug-in Compatible with the TMS 4732 32K ROM
- All Inputs and Outputs Fully TTL Compatible
- Static Operation (No Clocks, No Refresh)
- Maximum Access Time . . . 450ns
- Minimum Cycle Time . . . 450ns
- 3-State Outputs for OR-Ties
- 8 Bit Output for Use in Microprocessor Based Systems
- N-Channel Silicon-Gate Technology
- Low Power: 840mW Maximum Active
  132mW Maximum Standby
- Guaranteed DC Noise Immunity with Standard TTL Loads—No Pull-Up Resistors Required
- Interchangeable with Intel 2716

24-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)

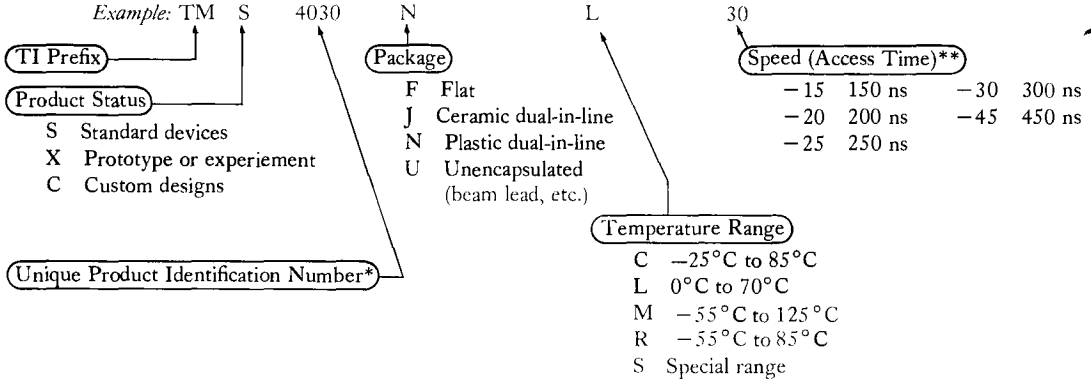


Mechanical Data

8

NUMBERING SYSTEM

Factory orders for circuits should include the complete part-type numbers.



MANUFACTURING INFORMATION

Die-attach is by standard gold silicon eutectic or by conductive epoxy.

Thermal compression gold wire bonding is used on plastic packaged circuits. Typical bond strength is 5 grams. Bond strength is monitored on a lot-to-lot basis. Any bond strength of less than 2 grams causes rejection of the entire lot of devices. On hermetic devices either thermal compression or ultrasonic wire bonding is used. All hermetic MOS LSI devices produced by TI are capable of withstanding 5×10^{-7} atm cc/sec inspection and may be screened to 5×10^{-8} atm cc/sec fine leak, if desired by the customer, for special applications.

All packages are capable of withstanding a shock of 3000 g. All packages except the 64-pin package are capable of passing a 20,000 g acceleration (centrifuge) test at the Y-axis. Final specifications for the 64-pin package are not available at this printing. Pin strength is measured by a pin-shearing test. All pins are able to withstand the application of a force of 6 pounds at 45° in the peel-off direction.

DUAL-IN-LINE PACKAGES

A pin-to-pin spacing of 100 mils has been selected for standard dual-in-line packages.

TI uses two basic types of hermetically sealed ceramic dual-in-line packages. The first type is the side-brazed package cap and tin-plated leads. The second is the cerdip which consists of a ceramic base and cap sealed with a low-temperature glass and tin-plated leads.

*Inclusion of an "L" within the identification number indicates the device operates in the low power range (e.g., 27L08, 40L45).

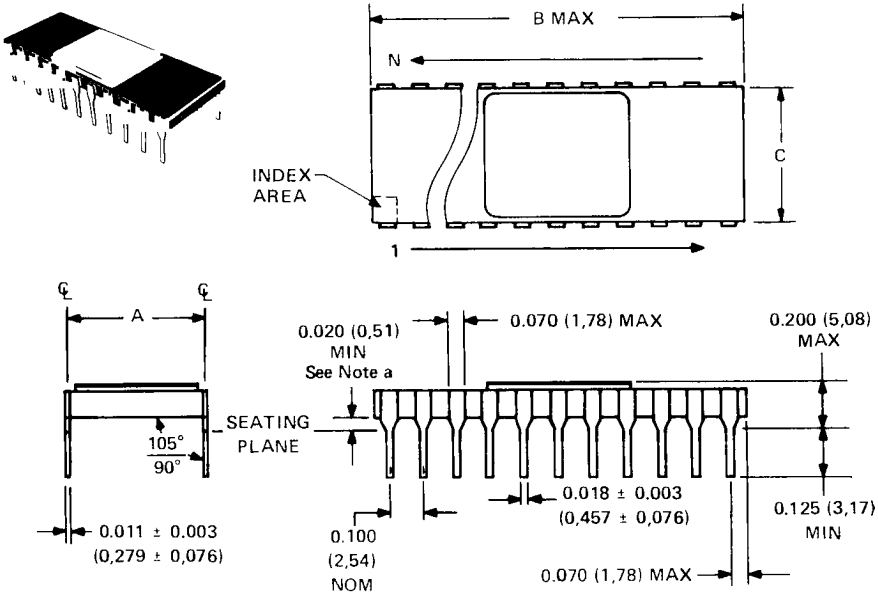
▸8

**On some parts DRAMs -1 250 ns, -2 200 ns SRAMS -1 650 ns, -2 450 ns 9900 FAMILY -30 3MHz, -40 4 MHz

The following dual-in-line packages are available in plastic or ceramic:

Distance Between Rows	*Number of Pins*								
	8	10	16	18	20	22	24	28	40
300 mils	X†	X†	X	X	X				
400 mils						X			
600 mils							X	X	X

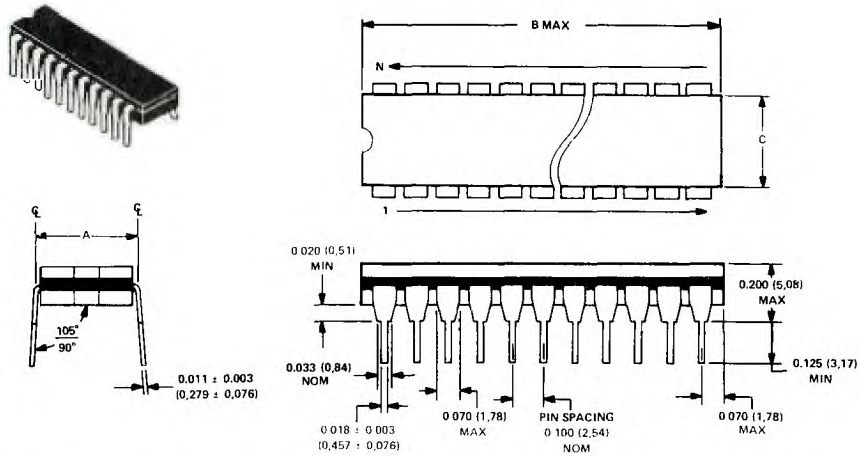
CERAMIC PACKAGES WITH SIDE-BRAZED LEADS AND METAL OR EPOXY OR GLASS LID SEAL



NOTES: a. This minimum spacing is valid for printed circuit board mounting with 0.033 (0,84) diameter holes for the leads.
b. All linear dimensions are in inches and parenthetically in millimeters. Inch dimensions govern.

DIM \ PINS	16	18	20	22	24	28	40
A ± 0.010 (0,26)	0.300 (7,62)	0.300 (7,62)	0.300 (7,62)	0.400 (10,16)	0.600 (15,24)	0.600 (15,24)	0.600 (15,24)
B MAX	0.840 (21,4)	0.910 (23,1)	1.020 (25,9)	1 28,0)	1,290 (32,8)	1.415 (36,0)	2.020 (51,3)
C NOM	0.290 (7,4)	0.290 (7,4)	0.290 (7,4)	(9,9)	0.590 (15,0)	0.590 (15,0)	0.590 (15,0)

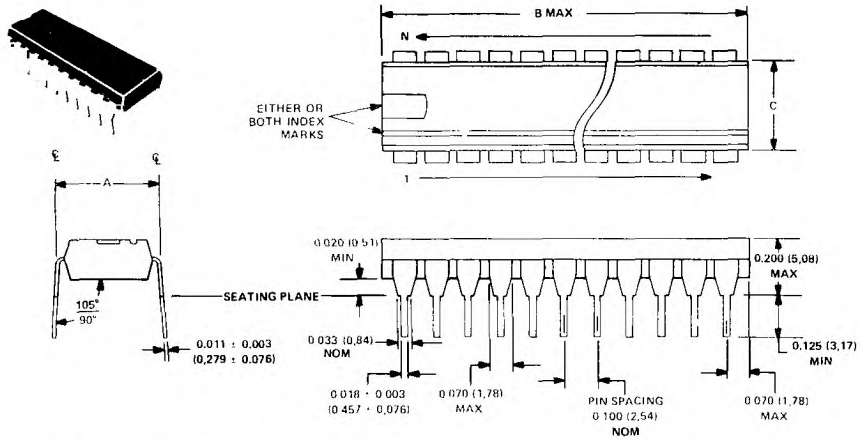
CERDIP PACKAGES



ALL LINEAR DIMENSIONS ARE IN INCHES AND PARENTHETICALLY IN MILLIMETERS. INCH DIMENSIONS GOVERN.

DIM \ PINS	16	18	20	22	24	28	40
A ± 0.010 (0,26)	0.300 (7,62)	0.300 (7,62)	0.300 (7,62)	0.400 (10,16)	0.600 (15,24)	0.600 (15,24)	0.600 (15,24)
B MAX	0.785 (20,0)	0.920 (23,4)	1.070 (27,2)	1.100 (28,0)	1.290 (32,8)	1.460 (37,1)	2.090 (53.1)
C MAX	0.288 (7,3)	0.288 (7,3)	0.288 (7,3)	0.388 (9,86)	0.560 (14,2)	0.560 (14,2)	0.560 (14,2)

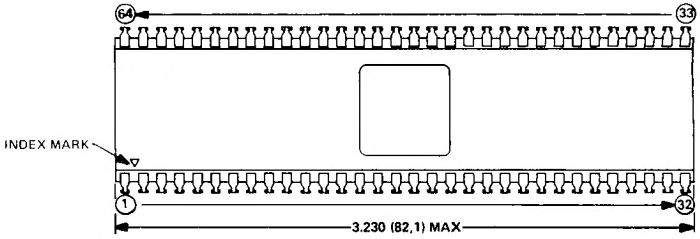
PLASTIC PACKAGES

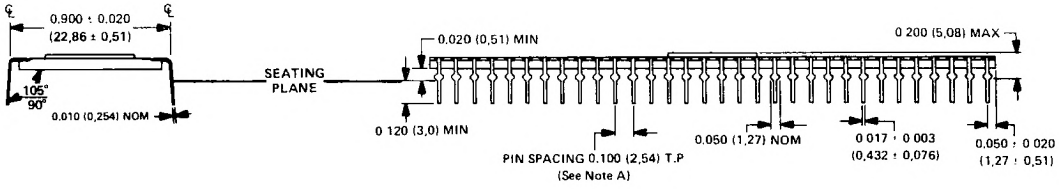


ALL LINEAR DIMENSIONS ARE IN INCHES AND PARENTHETICALLY IN MILLIMETERS. INCH DIMENSIONS GOVERN.

DIM \ PINS	8	16	18	20	22	24	28	40
A ± 0.010 (0,26)	0.300 (7,62)	0.300 (7,62)	0.300 (7,62)	0.300 (	0.400 (10,16)	0.600 (15,24)	0.600 (15,24)	0.600 (15,24)
B MAX	0.390 (9,9)	0.870 (22,1)	0.920 (23,4)	1.070 (	1.100	1.290 (32,8)	1.460 (36,6)	2.090 (53,1)
C NOM	0.250 (6,4)	0.250 (6,4)	0.250 (6,4)	0.265 (6,7)	0.350 (8,9)	0.550 (14,0)	0.550 (14,0)	0.550 (14,0)

CERAMIC PACKAGES WITH TOP-BRAZED OR SIDE-BRAZED LEADS AND METAL OR EPOXY OR GLASS LID SEAL





ALL LINEAR DIMENSIONS ARE IN INCHES AND PARENTHETICALLY IN MILLIMETERS, INCH DIMENSIONS GOVERN.

Software

- TMS 9900 Cross Assembler
- TMS 9900 Simulator
- ROM Utility

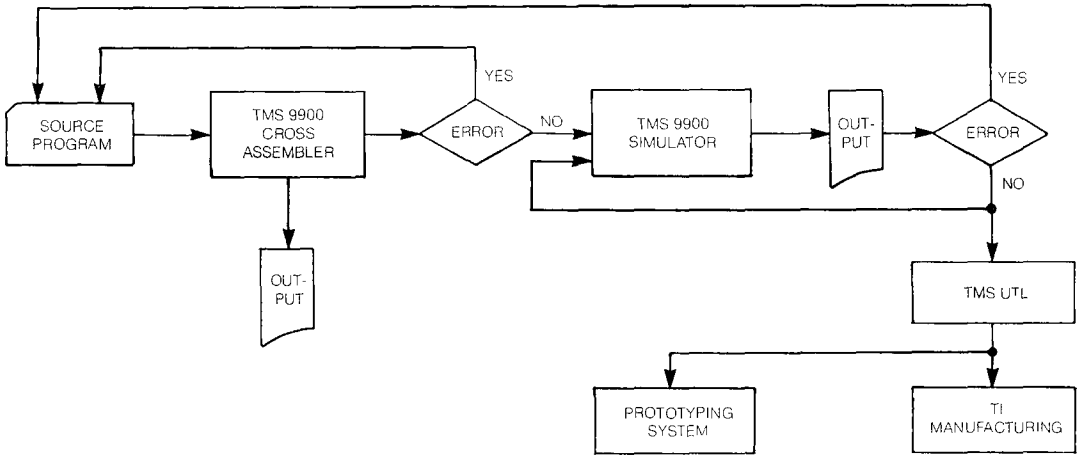


Figure 1

TMS 9900 CROSS ASSEMBLER DESCRIPTION

Before the advent of assemblers and other programming aids, the computer programmer was required to manually generate the particular bit patterns that constitute a program. This tedious task is now performed by symbolic assembler program such as the TMS 9900 Assembler. The Assembler permits the programmer to refer to data, memory addresses and machine actions symbolically when creating a source program. The TMS 9900 Assembly Language source is translated by the TMS 9900 Cross Assembler into relocatable linkable TMS 9900 Object module format. Both the source input and the object output are fully compatible with the FS 990 Prototype Development System and nationally available timesharing services (GE, NCSS, and TYMSHARE).

TMS 9900 SIMULATOR DESCRIPTION

8

The TMS 9900 Simulator is compatible with and has extensions to the Simulator on GE, NCSS, and TYMSHARE. The Simulator runs either in batch mode or in an interactive mode for maximum effectiveness. The microprocessor system simulation is specified by the designer from a keyboard/display device in the interactive mode. The output, such as instruction trace can be viewed on a CRT or printed out when in the batch mode.

The Simulator accepts object modules plus "link-control" statements from the Assembler as shown in Figure 1. Then the load module plus debug and control statements are sent to a "Run Processor" that performs the application program's execution. By executing instructions in software just as the TMS 9900 microprocessor executes instructions in hardware, the program logic is verified and the performance is measured. To set up a target system's characteristics, control language statements initialize memory, I/O ports, I/O linkages, and processor clock frequency. The control language also allows multiple breakpoints, full instruction trace, snapshots, memory and register inspection/changes. Interrupt controls and total run/stop commands are also provided. By exercising total software control, a program can be thoroughly checked before the hardware is running.

TMS UTL, ROM UTILITY DESCRIPTION

When the application program has been satisfactorily verified, the object module is accessed by the ROM Utility program, TMS UTL, for translation into a format acceptable for production of a gate placement program (prepatory to mass production). Alternatively, the utility can generate a BNPF or hexadecimal formatted file that is an input to a PROM programmer (Data I/O, etc.) to produce a PROM or EPROM version of the program. In all, there are 12 acceptable input formats and 12 output formats in support of the TMS 9900 microprocessors. Table I indicates the valid input/output translations supported by the utility for the TMS 9900.

Table I — TMSUTL Format Paths

	AVAILABLE OUTPUT FORMAT							
AVAILABLE INPUT FROM	1	2	3	4	5	6	7	8
1) Hexadecimal #2 (PROM)	YES	YES	YES	YES	NO	NO	YES	YES
2) Hexadecimal #2 (ROM)	YES	YES	YES	YES	NO	NO	YES	YES
3) BNPF	YES	YES	YES	YES	YES	YES	YES	YES
4) SN74S271 & SN74S371 ROM/HILO Format	YES	YES	YES	YES	NO	NO	YES	YES
5) TMS 9900 Standard Absolute Object of Cross Support System (Assembler or Loader/Simulator) & Prototyping System	YES	YES	YES	YES	YES	YES	YES	YES
6) TMS 9900 Compressed Absolute Object of Prototyping System	YES	YES	YES	YES	YES	YES	YES	YES
7) TMS 4700 ROM	YES	YES	YES	YES	NO	NO	YES	YES
8) TMS 4800 ROM	YES	YES	YES	YES	NO	NO	YES	YES

OPERATING ENVIRONMENT

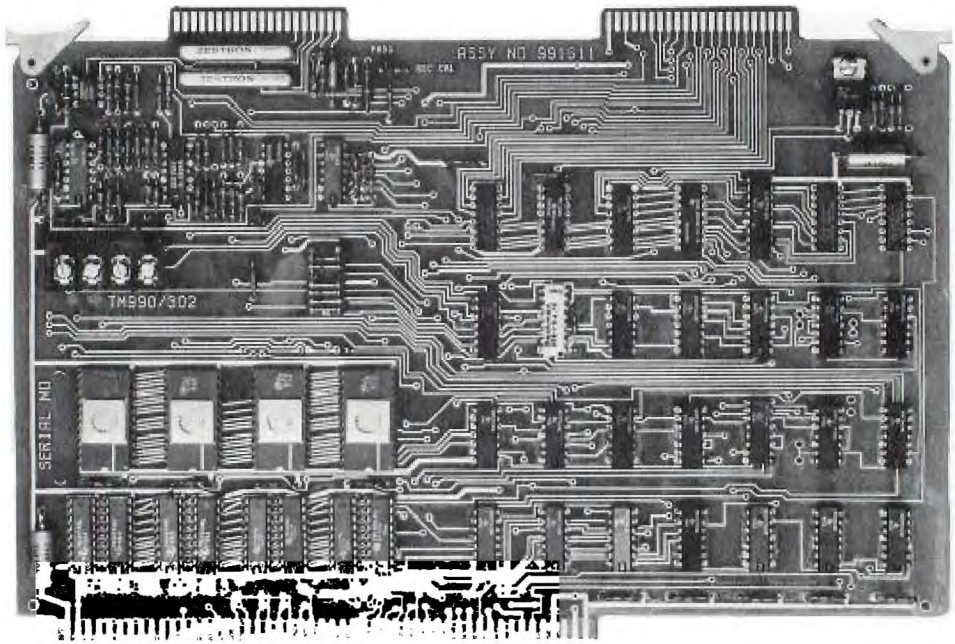
The programs are written to conform to ANSI STANDARD X3.0 (1966) 16-bit FORTRAN and are designed to execute on any minicomputer with the following minimum characterics.

- ANSI STANDARD X3.0 (1966) 16-bit FORTRAN Compiler
- Two's compliment arithmetic
- Disc capacity for up to 7 simultaneously active sequential files, with two being in rewindable and re-readable media
- A 16k to 20k word user program memory partition

PACKAGING

The TMS 9900 Transportable Cross Support package is composed of three distinct products: TMS 9900 Cross Assembler, Simulator, and ROM Utility. The part number for the package is TMSW 101MT. The product name is TMS 9900 *Transportable Cross-Support Software.* TMSW 101T is manufactured only on *half-inch, 9 track PE encoded* (IBM compatible) magnetic tape recorded at *1600 BPI.* The tape is *unlabeled, unblocked,* with *80 ASCII bytes per data record* and contains *128 files.* The first file on the tape is a data file which contains a one-time description for each file on the tape. Each file is terminated by an EOF mark except for the last file which is terminated with a double EOF to indicate *end-of-logical tape.*

Included in the shipping package is a *User Manual* for each of the three programs and an *Installation Manual* (4 manuals, total).



- Dual or Single audio-cassette interface
- EPROM programming options:
  TMS 9940, TMS 2716, TMS 2708,
  TMS 2532, TMS 2516
- Software development aids residing in ROM:
  - Symbolic Assembler
  - Text Editor
  - EPROM programmer
  - Relocating loader
  - I/O Scheduler/Handler
  - Debugger
- Optional POWER BASIC development software residing in EPROM (16k bytes)
- 4K × 16 EPROM or preprogrammed ROM
- 2K × 16 RAM
- Memory expandability for additional performance (TM 990/201 or TM 990/206 memory expansion boards)
- EIA communication with other computers

▶8

DESCRIPTION

The TM 990/302 is an assembled, tested module for developing assembly language software to be used on 990/9900 family microprocessor based systems. The TM 990/302, a bus-compatible member of the TM 990 microcomputer module family, provides dual audio cassette interfaces, both static RAM and ROM memory, and hardware circuitry for the programming of read-only memory devices. Used in conjunction with either the TM 990/100M or TM 990/101M microcomputer modules, the TM 990/302 provides a complete standalone software development system offering support for program generation, editing, assembly, debugging, and EPROM programming at an extremely attractive cost. Figure 1 is a system block diagram of the TM 990/302. The TM 990/100 and TM 990/101 memory map incorporating the Software Development Board ROM and RAM are shown in Figure 2.

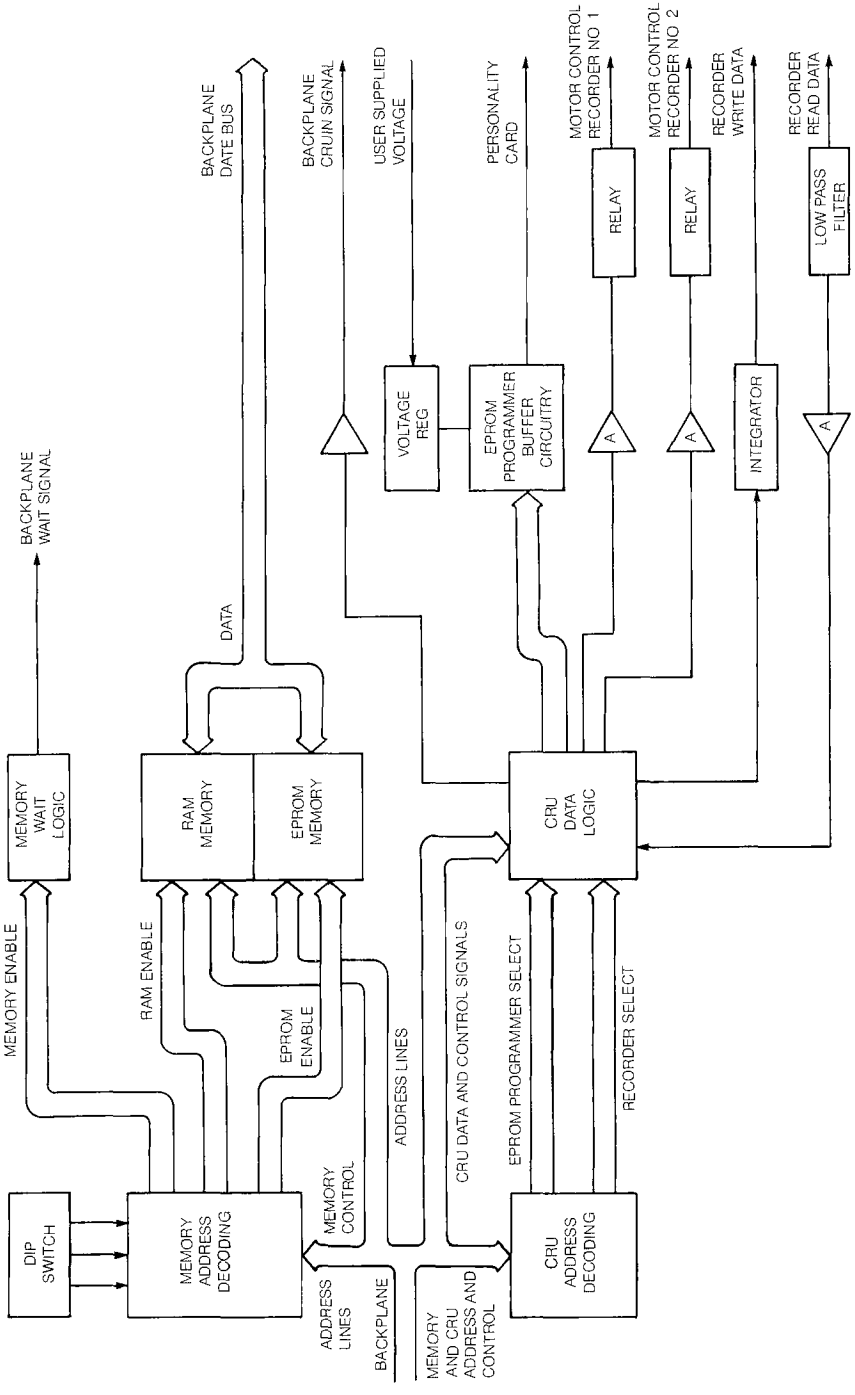


Figure 1. TM 990/302 System Functional Block Diagram

8

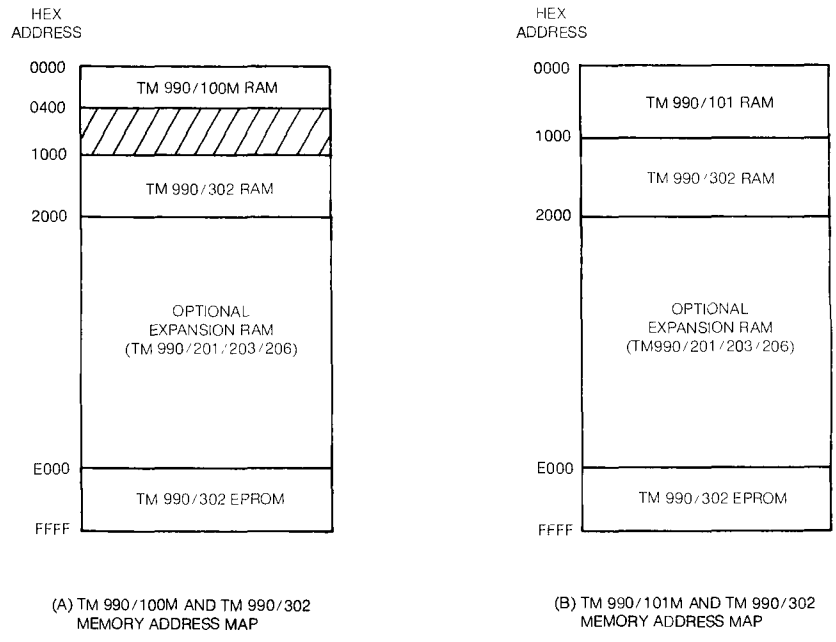


Figure 2. System Memory Map

SOFTWARE

Figure 3 is a typical software development cycle using the TM 990/302 Software Development board.

SOURCE TEXT EDITING

The text editor provides the means for initial source code entry or program update. Initial source inputs will be from the user's terminal. Source programs on audio cassette will be updated with changes made from the user's terminal. The size of the text editor buffer is determined at initialization as a function of the total available RAM.

The text editor operates on the source code text in a line mode. Text editor commands with their respective functions are:

D	Delete lines n thru m
I	Insert at line n with optional line-number autoincrement by m
K	Keep (store) buffer and print new top line in the buffer
G	Get buffer and print new bottom line of the buffer
P	Print lines n thru m
Q	Flush the input file until end of input file and return to executive.
R	Resequence output line numbers, n is initial line number and m is the increment.

To create or update the source program, the text editor provides manipulation of individual lines of code and entry with automatic line number indexing. The designer may delete, insert, print, resequence, and interactively check 9900 instruction syntax from his keyboard. The text editor handles programs of any length by segmenting the code into "buffer" blocks. It controls buffer loading and storage into cassette-tape memory. The buffer is enlarged by plugging in memory-expansion cards, which also expand the amount of target system memory available for execution.

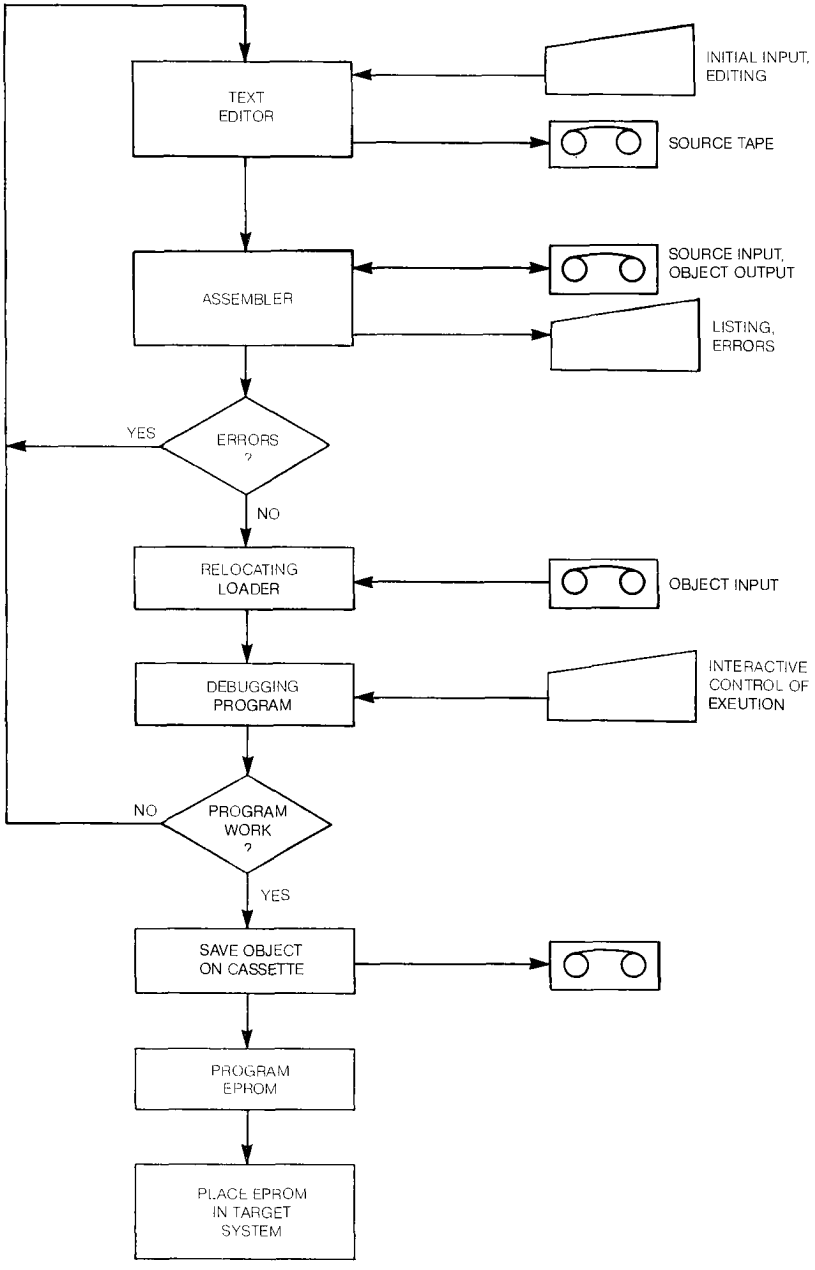


Figure 3. Typical Software Development Execution

ASSEMBLING SOURCE

The next step in program development is a two-pass assembly of TMS 9900, SBP 9900, TMS 9980, TMS 9985, TMS 9940 instruction sets into absolute standard 9900 object code. This two-pass assembler allows four-character symbolic addressing. The assembly listing output, including error messages, is routed to a user chosen device.

DEBUGGING

Seven debug commands aid program development after the loader program puts the assembled object into memory. Multi-step trace, software breakpoints and data inspection changes are featured.

Debug Commands:

SB	Set software breakpoint and execute
IM	Inspect/change memory
IC	Inspect/change CRU
IR	Inspect/change registers
RU	Record Program Execution Path Change
ST	Single step for 1 or more instructions with or without trace
DM	Dump memory to specified cassette in object format

EPROM PROGRAMMING

After debug, the EPROM programmer can be invoked to program EPROM's, read back EPROM's into memory, or compare EPROM contents to memory. Byte and word serial formats are available. The EPROM programmer is able to program the following EPROMS: 2708, 2716, 2516, 2532, and 9940.

HARDWARE FEATURES

- Two audio cassette interfaces
- 4K × 16 programmed ROM's
- 2K × 16 RAM
- One decode PROM for upgrading TM 990/100M microcomputer board RAM maps
- TMS 2716, TMS 2708, TMS 2532, TMS 2516 and TMS 9940 EPROM programming personality card.

When the TM 990/302 board is used as a software development system, the equipment configuration could include:

- TM 990/302 Software Development Board
- A TM 990/100M or TM 990/101 microcomputer board
- TM 990/510/520 (4 or 8 slot) card cage
- User supplied power supply: +12V, +5V, −12V, +30V to +52V (for EPROM Programmer) TM 990/518 Power Supply
- User supplied audio cassette player/recorder
- Power Basic Interpreter (16k byte) optional
- User supplied terminal: current loop TTY, TI silent 700, or equivalent.

▶8

Table 1 lists the hardware characteristics of the development system when used with either of the two microcomputer boards.

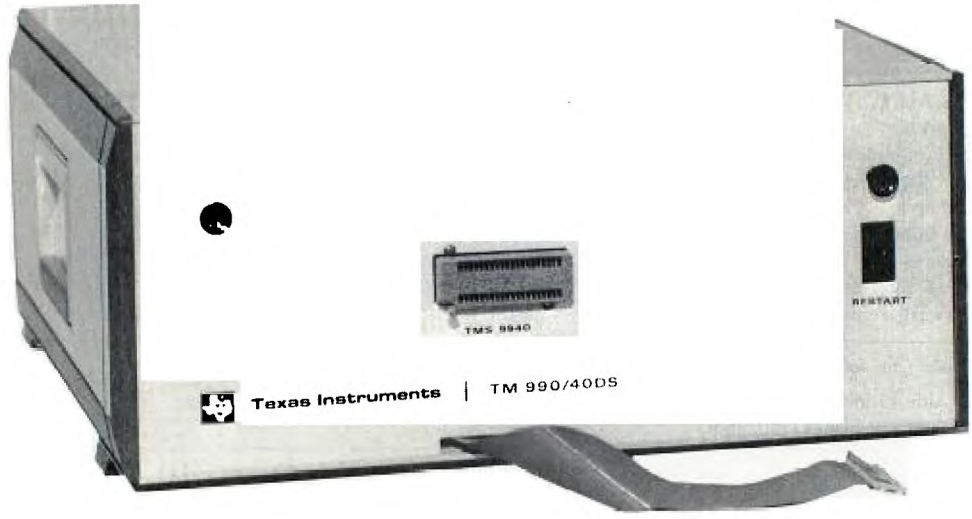
TM 990/302 SOFTWARE DEVELOPMENT BOARD

Table 1. TM 990/302 System Specifications

Item	*TM 990/302 with TM 990/100M*	*TM 990/302 with TM 990/101M*
Microprocessor	TMS 9900, 16 bit	TMS 9900, 16 bit
Timers	2	3
Serial ASCII	1(110-19,200 baud)	2(110-19,200 baud)
I/O ports	1 16-bit port	1 16-bit port
RAM (min-max)	2.25K-2.25K words	4K words
Non-volatile memory	5K-8K words ROM	4K-8K words ROM
Programmable memories	2708, 2716, 2516 2532, 9940	2708, 2716, 2516 2532, 9940
Cassette interfaces	2	2
Operating Temperature	0-55°C	0-55°C
Physical Characteristics Width: 11 inches (279.4mm) Height: 7½ inches (190.5mm)		
Power Requirements +5V 1.5A −12V 50mA +12V 50mA 30-52V 100mA EPROM programming voltage		
Ordering Information TM 990/302 Software Development Board		

8

TM 990/40DS
DESIGN AID FOR TMS 9940 MICROCOMPUTER



- EPROM Programmer for TMS 9940E
- Assembler
- Debug Monitor
- Trial in System Emulation
- Can be used with EPROM and/or Mask ROM Version of TMS 9940

DESCRIPTION

The TM 990/40DS is a low cost development system for the first 16-bit single-component microcomputer, the TMS 9940. To make the TM990/40DS as cost effective as possible, the traditional front panel assembly of lights and switches is not used. Instead an input/output system is provided which enables the programmer to use a standard EIA terminal such as an ASR-33 Teletype or a Texas Instruments Silent 700 terminal. The TM 990/40DS helps the programmer:

- Generate executable software
- Program this software to be on the EPROM version of the TMS 9940, and
- Test the program to be in the user's target system using the Trial In-system Emulation (TISE) Feature.

▶8

SOFTWARE GENERATION

To help generate the software the TM 990/40DS features an assembler and monitor. The Line by Line Assembler (LBLA) is a single pass assembler that assembles the user's program written in TMS 9940 instructions and stores machine code in memory. As each source line is assembled, the resulting machine code is placed in the user's RAM on the TM 990/40DS. The user can then implement the TIBUG II monitor to test and debug the software prior to using the EPROM programmer.

TIBUG II is a debug monitor which provides an interactive interface between the user and the TM 990/40DS. The TIBUG II monitor provides software routines that accomplish special tasks. These routines, listed in Table I, facilitate software development using the TM 990/40DS. All communications with TIBUG II occur via a 20mA current loop or RS-232-C device.

PROGRAMMING THE EPROM OF THE TMS 9940

This is accomplished by three of the TIBUG II commands: PP, CP, and VP. These are described in Table I. These commands allow for the programming of the EPROM from the user's RAM memory of the TM 990/40DS, the copying from the TMS 9940E EPROM to the RAM memory, and the verification of the EPROM and RAM memories.

TRIAL IN-SYSTEM EMULATION (TISE)

Once the user's program has been assembled and debugged using the TM 990/40DS TIBUG II monitor, the program can be tested in the user's target system using the TISE feature of the TM 990/40DS. This feature allows emulation of most of the TMS 9940's operations, utilizing the TM 990/40DS memory. Using TISE, a three-foot 40-conductor cable is connected to the edge connector of the TM 990/40DS. The other end of the cable contains a 40-pin male connector that is plugged into the user's system at the socket that will contain the TMS 9940.

Table I — TIBUG II Commands

Command	*Description*
ZA	Assembler — used to call up the Line by Line Assembler.
LP	Load Object Program from cassette/paper tape into the TM 990/40DS user memory.
DP	Dump the TM 990/40DS memory onto cassette/paper tape in TMS 9900 absolute object format.
SB	Set Breakpoint. This command sets breakpoints that allow programs to be executed from one memory address to another. From 1 to 16 addresses can be entered as breakpoints. When a breakpoint address occurs, execution stops and contents of the Program Counter, Status Register and the Workspace Pointer are displayed.
CB	Clear Breakpoint. Clears breakpoints previously set.
IM	Inspect/Change Memory, Memory Dump. Memory inspect/change "opens" a memory location, displays it, and gives the option of changing the data in the location. Memory dump directs a display of memory contents from "start address" to "stop address".
IR	Inspect/Change Hardware Registers (Workspace Pointer, Program Counter, Status Register). These three registers are displayed and may be changed.
IW	Inspect/Change user workspace. This command is used to display contents of the entire workspace register file or display one register at a time allowing the user to change the register contents.
IC	Inspect/Change CRU. The CRU register is displayed and may be changed. NOTE: The CRU is a bit oriented I/O interface through which both input and output bits can be directly addressed individually or in fields of 1 to 16.
FB	Find Byte in memory. A memory field is searched for a value. The memory addresses that contain the value are printed out.
FW	Find Word in memory. Same as Find Byte in memory except the value is a word.
DH	Decimal to Hexadecimal conversion provides the user the capability of converting decimal numbers to hexadecimal.
HD	Hexadecimal to Decimal conversion, provides user the capability of converting hexadecimal numbers to decimal.
HA	Hexadecimal Arithmetic. Two hexadecimal numbers are entered and their sum and difference are printed out.
PP	Program EPROM. Used to load the TMS 9940's EPROM area with the user's program.

Command	*Description*
CP	Copy EPROM program. The contents of the EPROM section of the TMS 9940 is transferred into the user's RAM area of the TM 990/40DS.
VP	Verify PROM. The contents of the EPROM of the TMS 9940 is compared to the contents of the user's RAM area in the TM 990/40DS. This verifies that the correct program is in the TMS 9940.
MV	Move Block of memory. A starting and ending address is given for the block of memory to be transferred along with the starting address of the destination.
EX	Execute. Following execution of this command, program execution begins at the value presently in the program counter.
RU	Run in multistep mode. From 1 to 64K instruction executions followed with WP, PC, Status Printout.
HE	Help command. This command brings up a listing of all the TIBUG II commands as reference for the user.
TM	Texas Instruments 733 ASR run at 1200 Baud. This command is used to alert TIBUG II that the terminal being used is a 1200 Baud terminal other than a Texas Instruments 733 ASR,
TT	Self Test. The self test is a software routine used to test the TMS 9940s. The routine is loaded into the TMS 9940's RAM space and executed by the TM 990/40DS. Once completed, the program transmits the results of the test to the system terminal.

TM 990/40DS PARTS LIST

The TM 990/40DS consists of 3 boards (described below), a TM 990/510 card cage, power supply, Trial In-System Evaluation (TISE) cable, EPROM programmer cable, a serial interface cable, and a chassis.

Three boards make up the TM 990/40DS System (See Figure 1). First is the TM 990/100M-4 microcomputer board with 1k × 8 bits of RAM and 8k × 8 bits of EPROM. This board is the central processing unit which controls I/O to the emulator boards and also to peripheral devices.

The two emulator boards are the TM 990/901 and TM 990/902 which handle the emulator functions of the TM 990/40DS during software development.

ORDERING INFORMATION TM 990/40DS
DESIGN AID FOR TMS 9940 MICROCOMPUTER

The TM 990/40DS may be ordered through any TI authorized distributors under the following Part Number: TM 990/40DS.

▶8

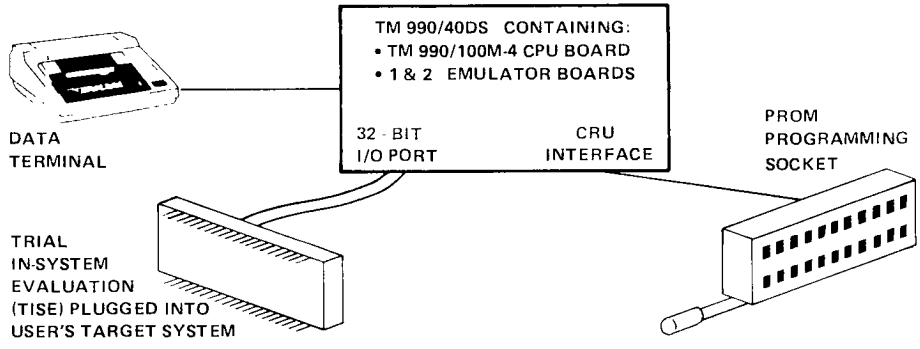


Figure 1. TM 990/40DS System Diagram

TM990/450, 451, 452, TMSW201F POWER BASIC FAMILY

POWER BASIC FEATURES

- Software Based in ROM or Floppy Diskettes
- For Use with Single or Multi-Board Systems
- Bit, Character and Word Oriented I/O
- Multiple Process Execution
- Interprocess Communication through Common Variables
- Automatic Minimum Memory Configuration
- Real Time Clock and Interrupt Processing
- Extended Arithmetic Capability
  - Multi-Dimensional Arrays
  - Multi-Argument Functions
  - 48 Bit Real Precision

A FAMILY OF PRODUCTS

The POWER BASIC Family is a set of software that edits and translates BASIC language statements into 9900 instructions and executes these statements to solve a particular algorithm.

Texas Instruments family of POWER BASIC language interpreters brings the features of BASIC to the industrial user of microprocessors providing a selection of products to meet the requirements for evaluation, development, and application in economical yet versatile packages. These additions continue the development of a comprehensive line of software development tools to support 990/9900 family components and systems to meet a wide range of application requirements.

The POWER BASIC family members are provided in read-only memory devices and on floppy diskettes. They can operate on a variety of hardware configurations ranging from stand-alone component based systems using 9900 family microprocessors to full-feature minicomputers such as the FS990. *Figure 1* illustrates the solution of real world problems in real time provided by a TM990/101M-10 module.

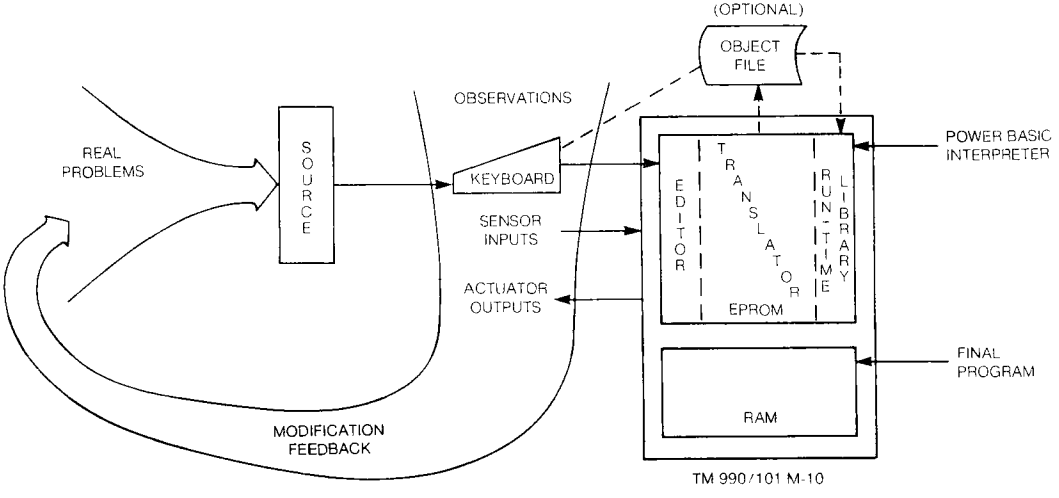


Figure 1. Real World Problem-Solving in Real Time with a TM990/101M-10 Module

APPLICATIONS

POWER BASIC is used to solve problems in data acquisition and control, data communications, information analysis, and sequencing of external events. Current applications include: intrusion alarm monitoring, navigational computation, numerical control, data reduction and analysis, inventory and payroll management, point-of-sale accounting, simulation and forecasting, and data base manipulation. Also used for education in programming and the structure of algorithmic processes.

APPLICATION FEATURES

POWER BASIC for Increased Speed

- TI Breaks the 20 Second Barrier

The 16 bit architecture and instruction set of the 9900 family of microprocessors is well suited for efficient execution of POWER BASIC programs. Results compared with benchmark programs reported in *KILOBAUD* magazine (October, 1977) show POWER BASIC to be faster than all others reported — less than 20 seconds on benchmark 7.

POWER BASIC for Cost Effectiveness

- Single Board, ROM Resident, Configurable

For production systems written in POWER BASIC, a CONFIGURATOR program is provided which analyzes the POWER BASIC application program and produces a minimum memory load module for that application.

Insertion of assembly language subroutines provides additional problem solving flexibility.

POWER BASIC for Flexible I/O

- Direct Manipulation of I/O and Memory

Control of information into and out of a 9900 family microprocessor based system is easily accomplished using a special variety of assignment statements supported by POWER BASIC. Internal and external data and control signals may be received and transmitted by the application program using a special set of POWER BASIC system defined functions. These special functions, as well as assembly language subroutines, support direct manipulation of either CRU or memory-map interfaced devices. In many cases, use of the system defined memory and CRU functions will be adequate substitutes for assembly language, improving the reliability of the application program by using pre-tested software and increasing the productivity of the design engineer.

▶8

POWER BASIC for Multi-Processing (Eval. BASIC)

- Multiple Process Execution and Communication

Within the POWER BASIC command and control structure, special provision has been made for independent execution of programs through a FOREGROUND/BACKGROUND mode switch. Once a program has been started in the FOREGROUND mode, a switch can be activated to provide a new program environment while relegating the currently executing POWER BASIC application program to the BACKGROUND mode of operation. After this has been accomplished, communication between FOREGROUND and BACKGROUND programs is supported through the use of shared dedicated variables. These unique variables may be modified and tested by either program independently, offering full capabilities for interaction and control.

TM990/450, 451, 452, TMSW201F POWER BASIC FAMILY

POWER BASIC FAMILY MEMBERS

Part No.	*Media*	*Name*	*Description*
TM990/450 TM990/101M-10	EPROM device kit TM990/101M	Evaluation POWER BASIC	Reduced memory version (8K Byte) designed to offer evaluation tools for exploring POWER BASIC applications. ROM kit executes stand-alone on TM990/100M, 101M modules.
TM990/451	ROM device kit	Development POWER BASIC	Expanded memory version (12K Byte) providing capability for design, development, debug, and complete programming of POWER BASIC programs. Executes on TM990/302 module interfaced with TM990/100M, 101M CPU modules.
TM990/452	EPROM device kit	Development BASIC Software Enhancement package	Enhancements (4K byte) to Development POWER BASIC. Provides utilities for use with TM990/302 module (EPROM programmer, cassette interfaces, etc.). Executes on TM990/302 module interfaced with TM990/101M CPU or with TM990/100M CPU and TM990/100M CPU and TM990/201 Memory module.
TMSW201F	FS990 diskette	Configurable POWER BASIC	Fully expanded version including complete diskette file support and a CONFIGURATOR program which reduces the size of POWER BASIC programs for execution.

A REAL SOLUTION TO A PROBLEM

To the industrial designer of microprocessor based electronic equipment, Texas Instruments POWER BASIC Family offers a versatile alternative to the use of assembly language in implementing application programs. Designed to provide a selection of products to meet a broad range of feature and cost requirements, POWER BASIC delivers productivity improvements and architecture independence which impact development costs and minimize project risks. Packed with improved features, POWER BASIC makes the solution of complex system problems a straight-forward process, eliminating unnecessary design details, while providing the kind of performance mandated by state-of-the-art semiconductor technology and minicomputer architecture.

Texas Instruments is committed to provide the most advanced microprocessor system development tools, making the 990/9900 family of microprocessors the low risk choice for designers of electronic equipment. The TI POWER BASIC Family reaffirms that commitment and is indicative of the quality support which the user can expect from Texas Instruments in the future.

Additional details on the individual products which comprise the POWER BASIC Family are available from your Texas Instruments sales representative and authorized distributors.