The Engineering Staff of
TEXAS INSTRUMENTS LIMITED
Semiconductor Group

SBP 9900
I^2L 16-Bit
Microprocessor
Data Manual

OCTOBER 1976

TEXAS INSTRUMENTS
LIMITED

Information contained in this publication is believed to be accurate and reliable. However, responsibility is assumed neither for its use nor for any infringement of patents or rights of others that may result from its use. No license is granted by implication or otherwise under any patent or patent right of Texas Instruments or others.

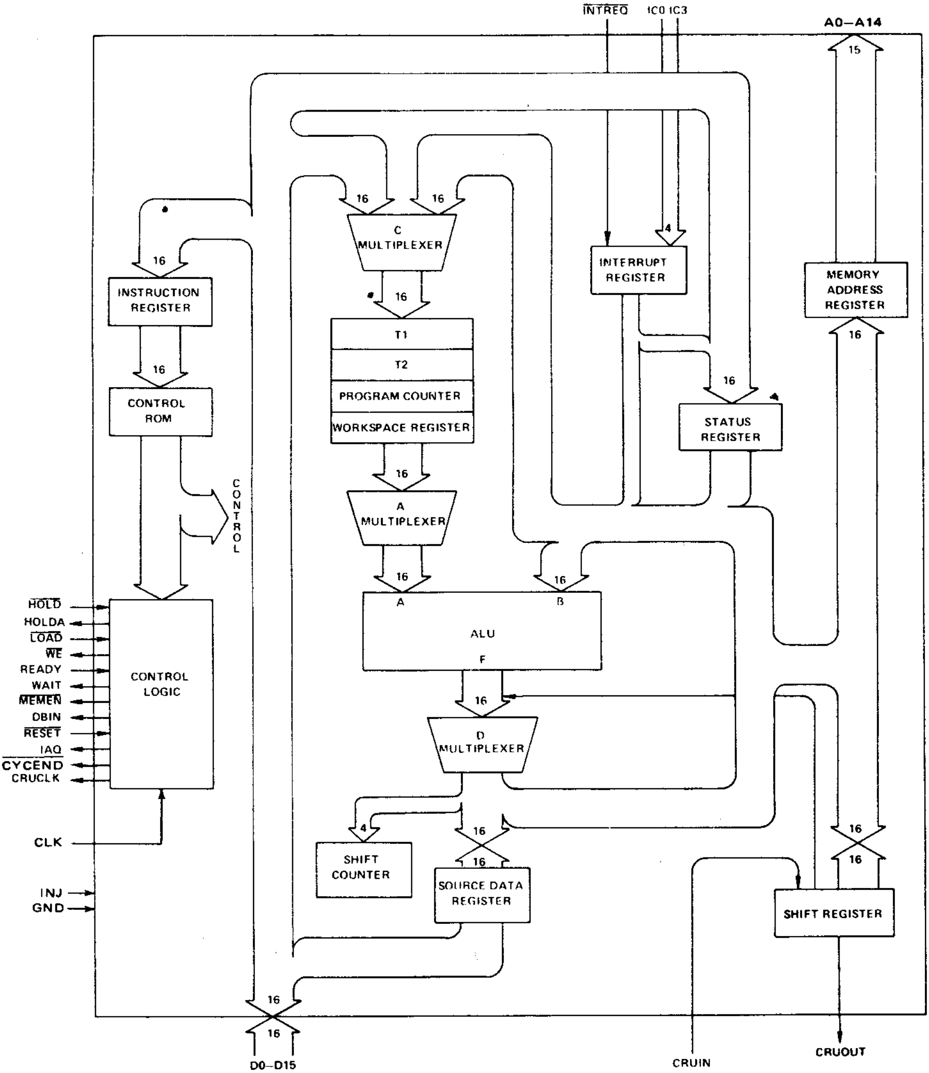
TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES



TENTATIVE DATA SHEET

This document provides tentative information on a new product. Texas Instruments reserves the right to change specifications for this product in any manner without notice.

FIGURE 1 – SBP 9900 ARCHITECTURE

1. INTRODUCTION

1.1 DESCRIPTION

The SBP 9900 microprocessor is a ruggedized monolithic parallel 16-bit Central Processing Unit (CPU) fabricated with Integrated Injection Logic (I^2L) technology. The SBP 9900 combines the properties of I^2L technology with a 16-bit word length, an advanced memory to memory architecture, and a full minicomputer instruction set to extend the end application reach of Texas Instruments 9900 microprocessor family into those applications requiring efficient, stable, reliable performance in severe operating environments. I^2L technology enables the SBP9900 to operate over a −55 to 125°C ambient temperature range from a single d-c power source with user selectable speed/power performance. Static Logic is used throughout with directly TTL compatible I/O permitting use with standard logic and memory devices and thereby eliminating the need for special clock and interface functions. The SBP 9900 is software compatible with other 9900 microprocessor family members and shares a common body of hardware/software with Texas Instruments 990 minicomputer family.

1.2 KEY FEATURES

- Parallel 16-Bit Word Length
- Full Minicomputer Instruction Set Includes Multiply and Divide
- Directly Addresses Up to 65,536 Bytes/32,768 Words of Memory
- Advanced Memory-To-Memory Architecture
- Multiple 16-Word Register Files (Work Spaces) Reside in Memory
- Separate I/O, Memory and Interrupt Bus Structures
- 16 Prioritized Hardware Interrupts
- 16 Software Interrupts (XOPS)
- Programmed and DMA I/O Capability
- Serial I/O Via Communications-Register-Unit (CRU)
- 64-Pin Package
- Software Compatible with TI 9900 Microprocessor/990 Minicomputer Family
- I^2L Technology:
  - −55°C to 125°C Ambient Temperature Range
  - User Selectable Speed/Power Operation
  - 2.6 MHz Nominal Clock at 500 mW
  - Single d-c Power Supply
  - Fully Static Operation
  - Single Phase Clock
  - Directly TTL Compatible I/O (Including Clock)

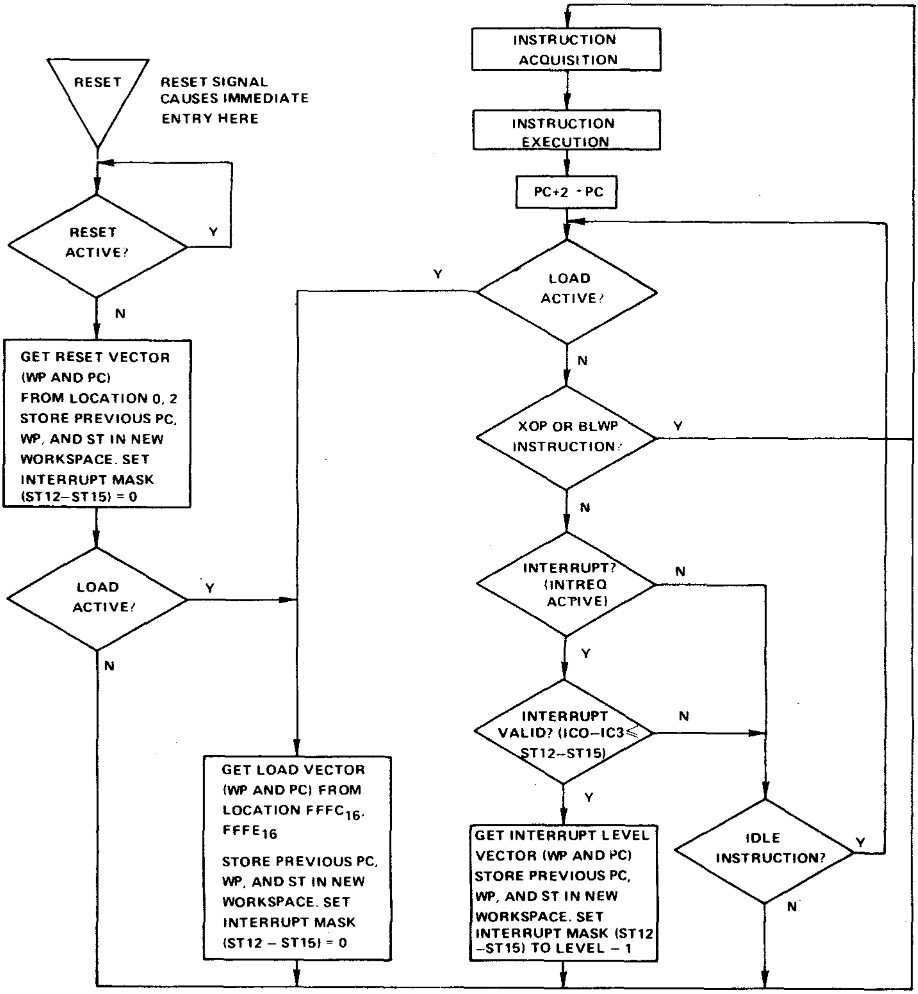

RESET
RESET SIGNAL CAUSES IMMEDIATE ENTRY HERE
RESET ACTIVE?
Y
N
GET RESET VECTOR (WP AND PC) FROM LOCATION 0, 2 STORE PREVIOUS PC, WP, AND ST IN NEW WORKSPACE. SET INTERRUPT MASK (ST12–ST15) = 0
LOAD ACTIVE?
Y
N
GET LOAD VECTOR (WP AND PC) FROM LOCATION FFFC16, FFFE16
STORE PREVIOUS PC, WP, AND ST IN NEW WORKSPACE. SET INTERRUPT MASK (ST12 – ST15) = 0
INSTRUCTION ACQUISITION
INSTRUCTION EXECUTION
PC+2 → PC
LOAD ACTIVE?
Y
N
XOP OR BLWP INSTRUCTION?
Y
N
INTERRUPT? (INTREQ ACTIVE)
N
Y
INTERRUPT VALID? (IC0–IC3 ≤ ST12–ST15)
N
Y
GET INTERRUPT LEVEL VECTOR (WP AND PC) STORE PREVIOUS PC, WP, AND ST IN NEW WORKSPACE. SET INTERRUPT MASK (ST12 –ST15) TO LEVEL – 1
IDLE INSTRUCTION?
Y
N

FIGURE 2 – 9900 CPU FLOW CHART

2. 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 in Figure 3.

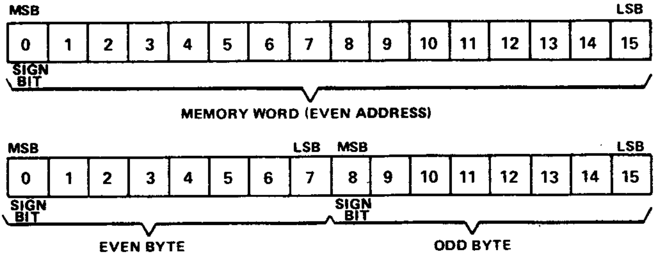


FIGURE 3 – WORD AND BYTE FORMATS

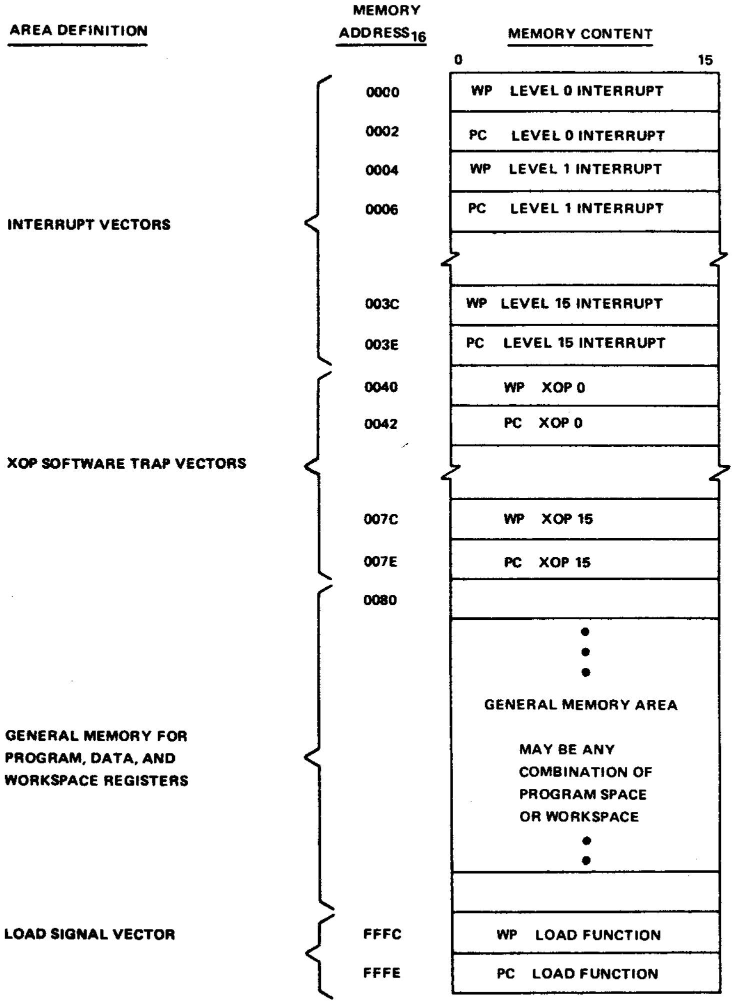
2.1 REGISTERS AND MEMORY

The 9900 employs an advanced memory-to-memory architecture. Blocks of memory designated as workspace replace internal-hardware registers with program-data registers. The 9900 memory map is shown in Figure 4. 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.

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.

A workspace-register file occupies 16 contiguous memory words in the general memory area (see Figure 4). 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 in Figure 5.

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 in the 9900 concept accomplishes a complete context switch withonly three store cycles and three fetch cycles. See Figure 5. 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 9900 that result in a context switch include:



AREA DEFINITION	MEMORY ADDRESS$_{16}$	MEMORY CONTENT (0 … 15)
INTERRUPT VECTORS	0000	WP LEVEL 0 INTERRUPT
	0002	PC LEVEL 0 INTERRUPT
	0004	WP LEVEL 1 INTERRUPT
	0006	PC LEVEL 1 INTERRUPT
	003C	WP LEVEL 15 INTERRUPT
	003E	PC LEVEL 15 INTERRUPT
XOP SOFTWARE TRAP VECTORS	0040	WP XOP 0
	0042	PC XOP 0
	007C	WP XOP 15
	007E	PC XOP 15
GENERAL MEMORY FOR PROGRAM, DATA, AND WORKSPACE REGISTERS	0080	
		• • • GENERAL MEMORY AREA MAY BE ANY COMBINATION OF PROGRAM SPACE OR WORKSPACE • •
LOAD SIGNAL VECTOR	FFFC	WP LOAD FUNCTION
	FFFE	PC LOAD FUNCTION

FIGURE 4 – MEMORY MAP

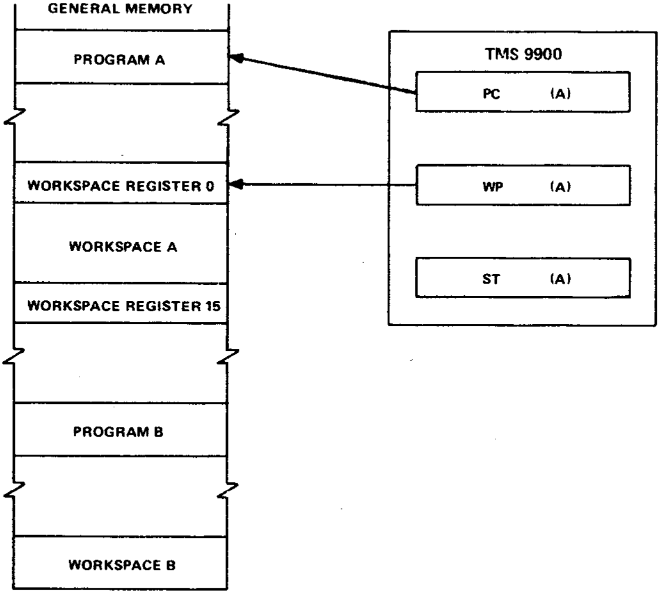


FIGURE 5 – MEMORY-TO-MEMORY WORKSPACE CONCEPT

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 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 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 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 standard TTL components as shown in Figure 6. Note that for eight or less external interrupts a single SN54/74148 is required and for one external interrupt $\overline{\text{INTREQ}}$ is used as the interrupt signal with a hard-wired code IC0 through IC3.

2.3 I/O INTERFACE COMMUNICATIONS-REGISTER-UNIT (CRU)

The SBP 9900 communications-register-unit (CRU) is a versatile, direct command-driven serial I/O interface. The CRU may directly address, in bit-fields of one to sixteen, up to 4096 peripheral input bits and up to 4096 peripheral output bits. The SBP 9900 executes three single-bit and two multiple-bit CRU instructions. The single-bit instructions include TEST BIT (TB), SET BIT TO ONE (SBO), and SET BIT TO ZERO (SBZ); the multiple-bit instructions include LOAD CRU (LDCR) and STORE CRU (STCR).

The SBP 9900 employs three dedicated I/O signals CRUIN, CRUOUT, CRUCLK, and the least significant twelve bits of the address bus to support the CRU interface. CRU interface timing is shown in Section 2.9.

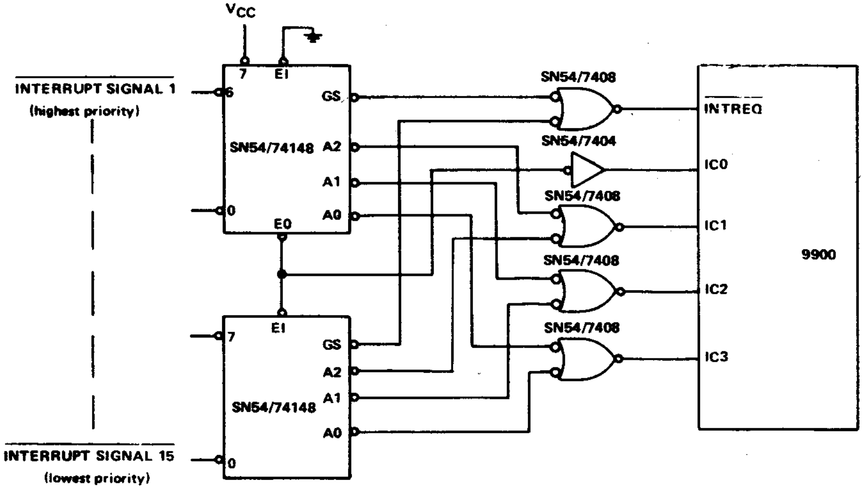


FIGURE 6 – 9900 INTERRUPT INTERFACE

2.4 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 CRU-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 base address from W12 is added to the signed displacement specified in the instruction and the result is loaded onto the address bus. Figure 7 illustrates the development of a single-bit CRU address.

2.5 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 8. 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

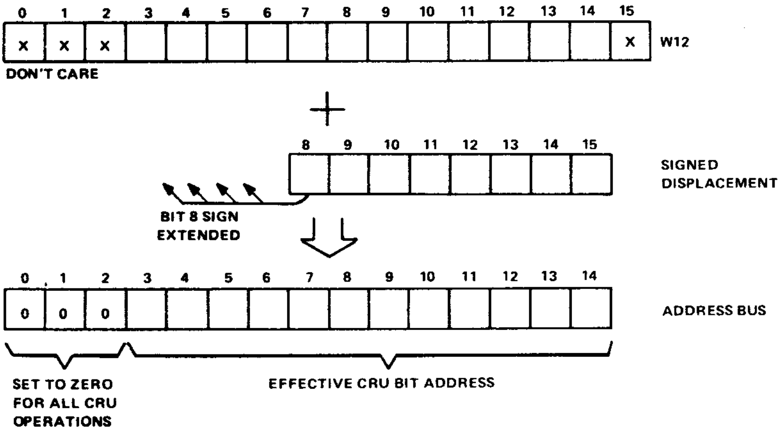


FIGURE 7 – 9900 SINGLE-BIT CRU ADDRESS DEVELOPMENT

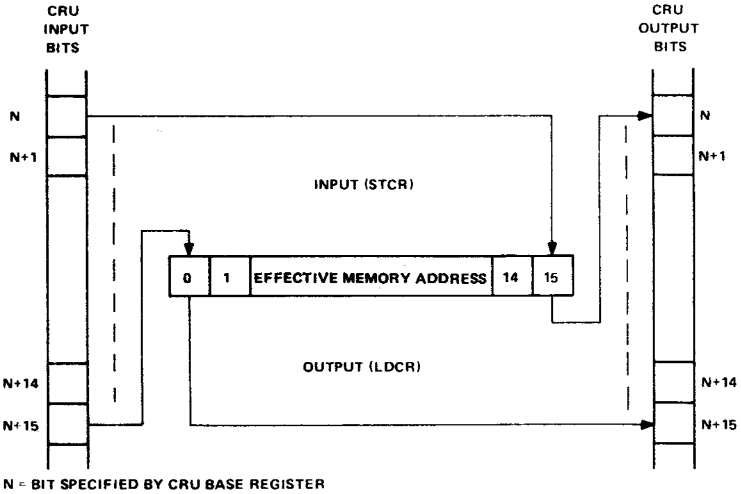


FIGURE 8 – 9900 LDCR/STCR DATA TRANSFERS

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. 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 9 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.

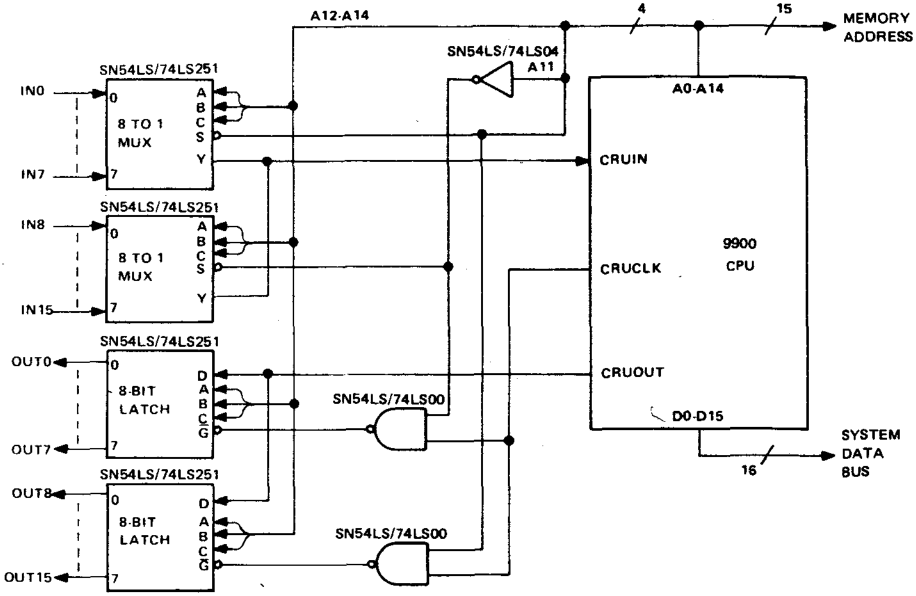


FIGURE 9 – 9900 16-BIT INPUT/OUTPUT INTERFACE

2.6 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 990 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 most-significant 3 bits of the address bus (A0 through A2) along with a CRUCLK pulse. 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 shown in Table 2.

Figure 10 illustrates typical external decode logic to implement these instructions. Note that a signal is generated to inhibit CRU decodes during external instructions.

TABLE 2
EXTERNAL INSTRUCTIONS

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

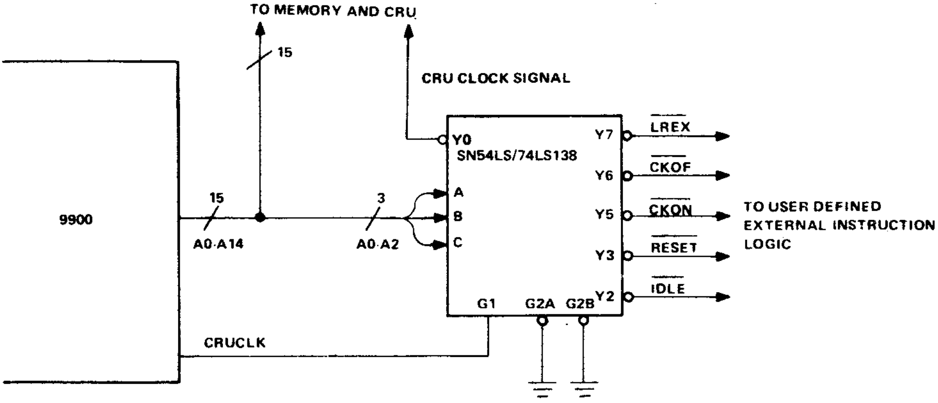


FIGURE 10 – EXTERNAL INSTRUCTION DECODE LOGIC

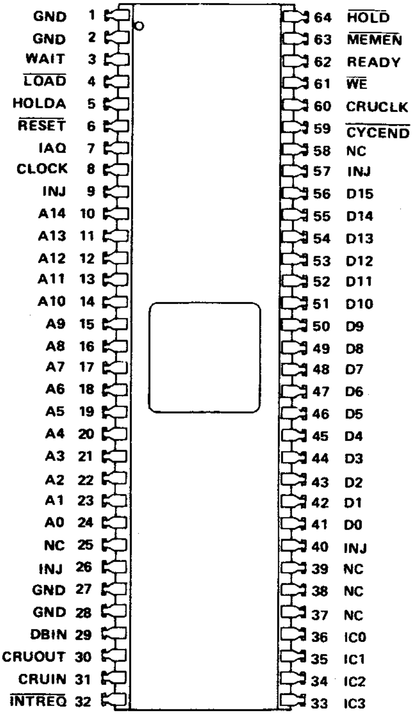
2.7 LOAD FUNCTION

The LOAD signal allows cold-start ROM loaders and front panels to be implemented for the 9900. When active, LOAD causes the 9900 to initiate an interrupt sequence immediately following the instruction being executed. Memory location FFC 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.

2.8 SBP 9900 PIN DESCRIPTION

Table 3 describes the function of each SBP 9900 pin, and Figure 11 illustrates their assigned locations.

FIGURE 11 – SBP 9900 PIN ASSIGNMENTS



NC–No internal connection

TABLE 3
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 reading) the external-memory system when $\overline{\text{MEMEN}}$ 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 9900 has disabled its output buffers to allow the memory to place memory-read data on the data bus during $\overline{\text{MEMEN}}$. 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 9900 data bus is outputting data to be written into memory.

TABLE 3 (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 9900 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 9900 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
$\overline{\mathrm{INTREQ}}$	32	IN	INTERRUPT REQUEST. When active (logic level LOW), $\overline{\mathrm{INTREQ}}$ indicates that an external interrupt is requesting service. If $\overline{\mathrm{INTREQ}}$ is active, the SBP 9900 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 9900 initiates the interrupt sequence. If the comparison fails, the SBP 9900 ignores the interrupt request. In that case, $\overline{\mathrm{INTREQ}}$ should be held active. The SBP 9900 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 identity code, are sampled by the SBP 9900 when $\overline{\mathrm{INTREQ}}$ is active (logic level LOW). When IC0 through IC3 are LLLH, the highest priority *external* interrupt is requesting service; when HHHH, the lowest priority external interrupt is requesting service.
IC0 (LSB)	33	IN	
			MEMORY CONTROL
$\overline{\mathrm{HOLD}}$	64	IN	When active (logic level LOW), $\overline{\mathrm{HOLD}}$ indicates to the SBP 9900 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 9900 enters the hold state after completion of its present memory cycle. The SBP 9900 then allows its address bus, data bus, $\overline{\mathrm{WE}}$, $\overline{\mathrm{MEMEN}}$, DBIN, and HOLDA facilities to be pulled to the logic level HIGH state. When $\overline{\mathrm{HOLD}}$ is deactivated, the SBP 9900 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 9900 is in the hold state and that its address bus, data bus, $\overline{\mathrm{WE}}$, $\overline{\mathrm{MEMEN}}$, 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 9900 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 9900 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 9900 initiated instruction acquisition memory cycle. Consequently, IAQ may be used to facilitate detection of illegal op codes.
$\overline{\mathrm{CYCEND}}$	59	OUT	CYCLE END. When active (logic level LOW), $\overline{\mathrm{CYCEND}}$ indicates that the SBP 9900 will initiate a new microinstruction cycle on the low-to-high transition of the next CLOCK.
$\overline{\mathrm{LOAD}}$	4	IN	When active (logic level LOW), $\overline{\mathrm{LOAD}}$ causes the SBP 9900 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. $\overline{\mathrm{LOAD}}$ will also terminate an idle state. If $\overline{\mathrm{LOAD}}$ is active during the time $\overline{\mathrm{RESET}}$ is active, the $\overline{\mathrm{LOAD}}$ trap will occur after the RESET function is completed. $\overline{\mathrm{LOAD}}$ should remain active for one instruction execution period (IAQ may be

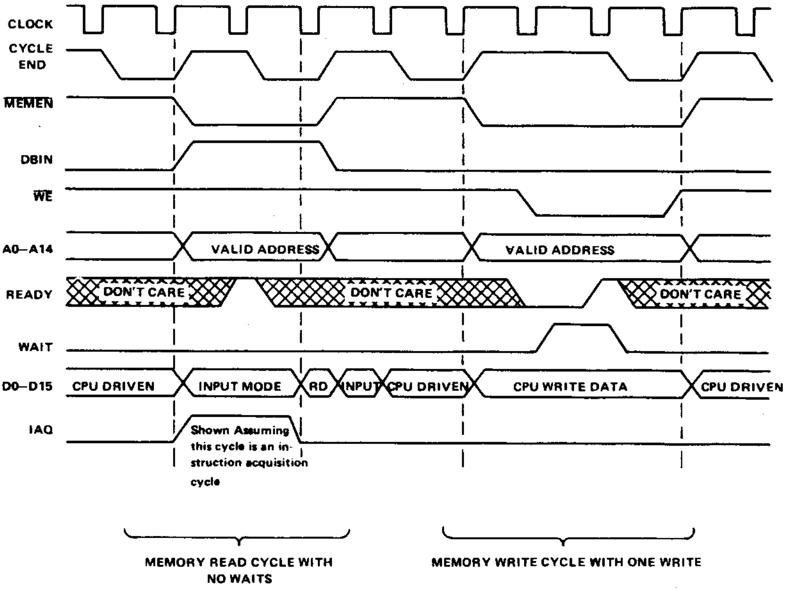
TABLE 3 (CONCLUDED)

SIGNATURE	PIN	I/O	DESCRIPTION
$\overline{\text{LOAD}}$ (Cont.)			used to monitor instruction boundaries). $\overline{\text{LOAD}}$ may be used to implement cold-start ROM 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 9900 to reset itself and inhibit $\overline{\text{WE}}$ and CRUCLK. When $\overline{\text{RESET}}$ is released, the SBP 9900 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.

2.9 SBP 9900 TIMING

2.9.1 SBP 9900 MEMORY

The SBP 9900 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 12. 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).



RD = READ DATA

FIGURE 12 – SBP 9900 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 that 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, $\overline{\text{WE}}$ becomes active (logic level LOW) with the first high-to-low transition of the clock after $\overline{\text{MEMEN}}$ becomes active; DBIN remains inactive. At the end of a memory-write cycle, $\overline{\text{WE}}$ and $\overline{\text{MEMEN}}$ 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 9900. If READY is inactive (logic level LOW) during the first low-to-high transition of the clock after $\overline{\text{MEMEN}}$ becomes active, the SBP 9900 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.

2.9.2 SBP 9900 HOLD

The SBP 9900 hold facilities allow both the '9900 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 ($\overline{\text{HOLD}}$) to the SBP 9900. When the next available non-memory cycle occurs, the SBP 9900 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 9900 by releasing $\overline{\text{HOLD}}$.

When $\overline{\text{HOLD}}$ becomes active (logic level LOW), the SBP 9900 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, $\overline{\text{MEMEN}}$, $\overline{\text{WE}}$, A0 through A14, and D0 through D15. When $\overline{\text{HOLD}}$ becomes inactive, the SBP 9900 exits the hold state and regains memory-bus control. If $\overline{\text{HOLD}}$ becomes active during a CRU operation, the SBP 9900 uses an extra clock cycle after the deactivation of $\overline{\text{HOLD}}$ to reassert the CRU address thereby providing the normal setup time for the CRU-bit transfer.

2.9.3 SBP 9900 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 14. To transfer a data-bit to a peripheral CRU device, the SBP 9900 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 9900 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 9900 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.

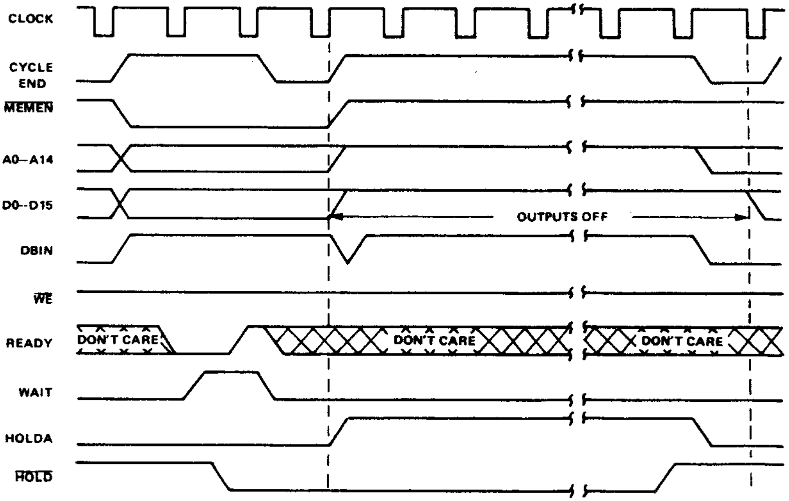


FIGURE 13 – SBP 9900 HOLD TIMING

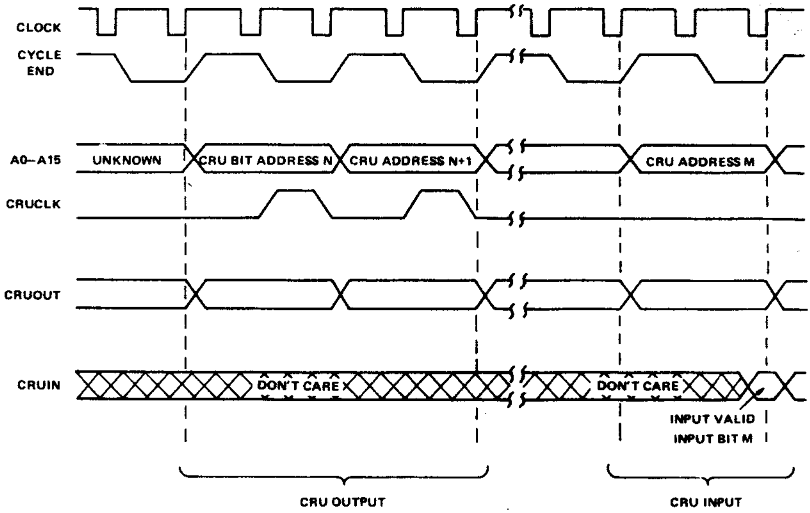


FIGURE 14 – SBP 9900 CRU INTERFACE TIMING

3. 9900 INSTRUCTION SET

3.1 DEFINITION

Each 9900 instruction performs one of the following operations:

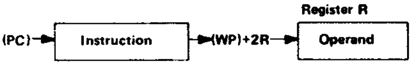
- Arithmetic, logical, comparison, or manipulation operations on data
- Loading or storage of internal registers (program counter, workspace pointer, or status)
- Data transfer between memory and external devices via the CRU
- Control functions.

3.2 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 Section 3.5 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.

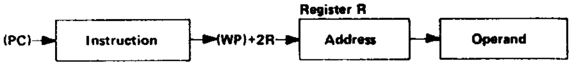
3.2.1 WORKSPACE REGISTER ADDRESSING R

Workspace Register R contains the operand.



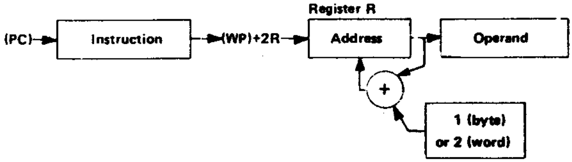
3.2.2 WORKSPACE REGISTER INDIRECT ADDRESSING *R

Workspace Register R contains the address of the operand.



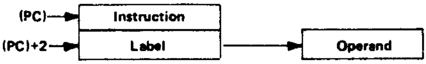
3.2.3 WORKSPACE REGISTER INDIRECT AUTO INCREMENT ADDRESSING *R+

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



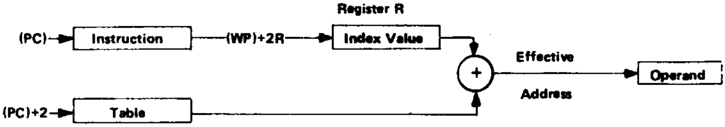
3.2.4 SYMBOLIC (DIRECT) ADDRESSING @ LABEL

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



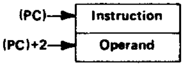
3.2.5 INDEXED ADDRESSING @ TABLE (R)

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



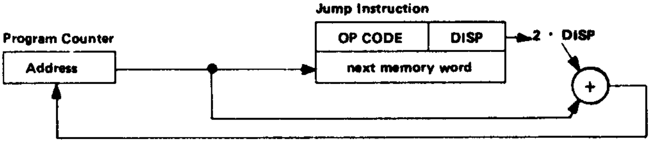
3.2.6 IMMEDIATE ADDRESSING

The word following the instruction contains the operand.



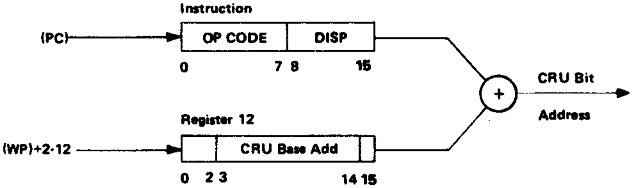
3.2.7 PROGRAM COUNTER RELATIVE ADDRESSING

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.



3.2.8 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.



3.3 TERMS AND DEFINITIONS

The terms used in describing the instructions of the 9900 are defined in Table 4.

TABLE 4
TERM DEFINITIONS

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
$\lvert n \rvert$	Absolute value of n
+	Arithmetic addition
–	Arithmetic subtraction
AND	Logical AND
OR	Logical OR
⊕	Logical exclusive OR
$\overline{n}$	Logical complement of n

3.4 STATUS REGISTER

The status register contains the interrupt mask level and information pertaining to the instruction operation. Table 5 explains the bit indications.

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			

TABLE 5
STATUS REGISTER BIT DEFINITIONS

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 [(DA)-(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
ST2	EQUAL	C, CB	If (SA) = (DA)
		C1	If (W) = IOP
		COC	If (SA) and $\overline{\text{(DA)}}$ = 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

3.5 INSTRUCTIONS

3.5.1 Dual Operand Instructions with Multiple Addressing Modes for Source and Destination Operand

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
General format:	OP CODE			B	T_D		D				T_S		S			

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			B	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
	0	1	2	3				
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)

3.5.2 Dual Operand Instructions with Multiple Addressing Modes for the Source Operand and Workspace Register Addressing for the Destination

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
General format:	OP CODE						D				T_S		S			

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) $\oplus$ (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.

3.5.3 Extended Operation (XOP) Instruction

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
General format:	0	0	1	0	1	1	D				T_S		S			

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:

$(40_{16} + 4D)$ → (WP)
$(42_{16} + 4D)$ → (PC)
SA → (new WR11)
(old WP) → (new WR13)
(old PC) → (new WR14)
(old ST) → (new WR15)

The 9900 does not test interrupt requests ($\overline{\text{INTREQ}}$) upon completion of the XOP instruction.

3.5.4 Single Operand Instructions

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
General format:	OP CODE										T_S		S			

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

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.

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.

3.5.5 CRU Multiple-Bit Instructions

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
General format:	OP CODE						C				T_S		S			

The C field specifies the number of bits to be transferred. If C = 0, 16 bits will be transferred. The CRU base register (WR12, 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 WR12 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	MEANING	RESULT COMPARED TO 0	STATUS BITS AFFECTED	DESCRIPTION
	0 1 2 3 4 5				
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 ≤ C ≤ 8.

3.5.6 CRU Single-Bit Instructions

General format:

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">OP CODE</td><td colspan="8">SIGNED DISPLACEMENT</td></tr>
</table>

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

MNEMONIC	OP CODE	MEANING	STATUS BITS AFFECTED	DESCRIPTION
	0 1 2 3 4 5 6 7			
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.

3.5.7 Jump Instructions

General format:

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">OP CODE</td><td colspan="8">DISPLACEMENT</td></tr>
</table>

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.

MNEMONIC	OP CODE	MEANING	ST CONDITION TO LOAD PC
	0 1 2 3 4 5 6 7		
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

3.5.8 Shift Instructions

General format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP CODE								C				W			

If C = 0, bits 12 through 15 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.

3.5.9 Immediate Register Instructions

General format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP CODE											N	W			
IOP															

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)

3.5.10 Internal Register Load Immediate Instructions

General format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP CODE											N				
IOP															

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

3.5.11 Internal Register Store Instructions

General format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP CODE											N	W			

No ST bits are affected.

MNEMONIC	OP CODE										MEANING	DESCRIPTION
	0	1	2	3	4	5	6	7	8	9	10		
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)

3.5.12 Return Workspace Pointer (RTWP) Instruction

General format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	1	1	1	0	0	N				

The RTWP instruction causes the following transfers to occur:

(WR15) → (ST)
(WR14) → (PC)
(WR13) → (WP)

3.5.13 External Instructions

General format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OP CODE											N				

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										MEANING	STATUS BITS AFFECTED	DESCRIPTION	ADDRESS BUS		
	0	1	2	3	4	5	6	7	8	9	10				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

3.6 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.

3.7 SBP 9900 INSTRUCTION EXECUTION TIMES

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

1) Clock cycle time, t_c

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

3) Number of wait states required per memory access.

The following Table 6 lists the number of clock cycles and memory accesses required to execute each SBP 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:

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

where:

T = total instruction execution time;
t_c = 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 6
INSTRUCTION EXECUTION TIMES

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION†	
			SOURCE	DEST
A	14	4	A	A
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)*	97-124	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	—	—
$\overline{\text{RESET}}$ function	26	5	—	—
$\overline{\text{LOAD}}$ function	22	5	—	—
Interrupt context switch	22	5	—	—

INSTRUCTION	CLOCK CYCLES C	MEMORY ACCESS M	ADDRESS MODIFICATION†	
			SOURCE	DEST
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 **	4	1	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 $t_c = 0.333\,\mu s$ and no wait states are required to access memory. Both operands are addressed in the workspace register mode:

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

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:

$$\begin{aligned} T &= t_c\,(C + W{\cdot}M) \\ C &= 14 + 8 = 22 \\ M &= 4 + 1 = 5 \\ T &= 0.333\,(22 + 2{\cdot}5)\,\mu s = 10.656\,\mu s. \end{aligned}$$

4. 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 15.

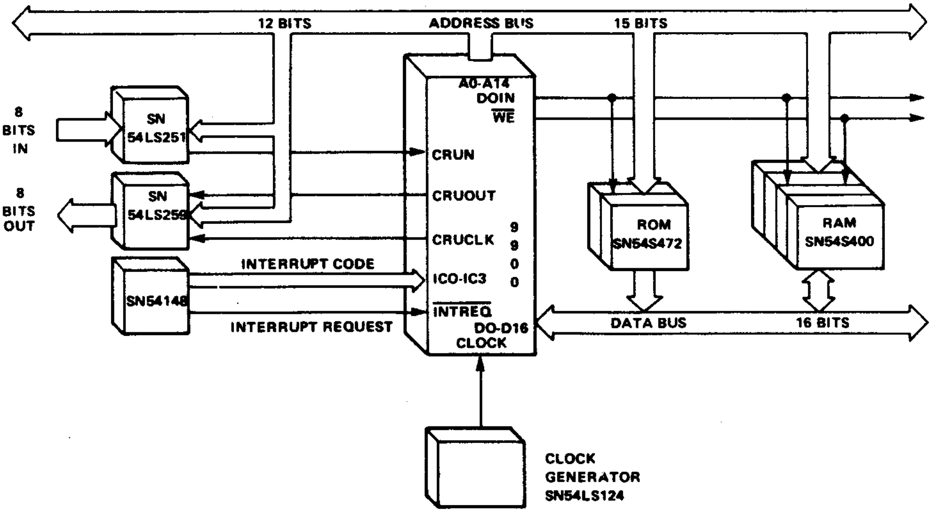


FIGURE 15 – MINIMUM SBP 9900 SYSTEM

4.1 INPUT CIRCUIT

The input circuit used on the SBP 9900 is basically an RTL configuration which has been modified for TTL compatibility as shown in Figure 16A. An input-clamping diode is incorporated to limit negative excursions (ringing) when the SBP 9900 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.

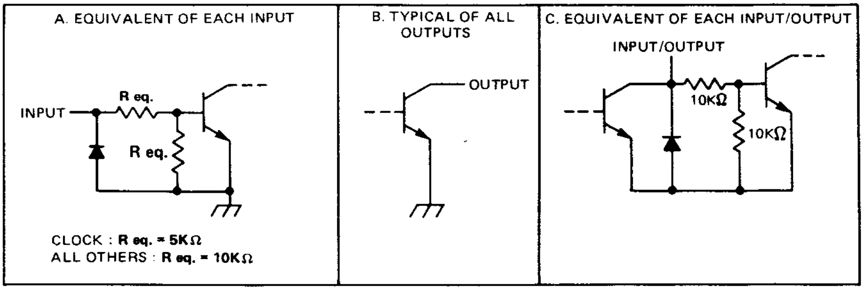

A. EQUIVALENT OF EACH INPUT
INPUT
R eq.
R eq.
CLOCK : R eq. = 5KΩ
ALL OTHERS : R eq. = 10KΩ
B. TYPICAL OF ALL OUTPUTS
OUTPUT
C. EQUIVALENT OF EACH INPUT/OUTPUT
INPUT/OUTPUT
10KΩ
10KΩ

FIGURE 16 – SCHEMATICS OF EQUIVALENT INPUTS, OUTPUTS, INPUTS/OUTPUTS

The input circuit characteristics for input current versus input voltage are shown in Figure 17. 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 9900 is powered down (injector current reduced).

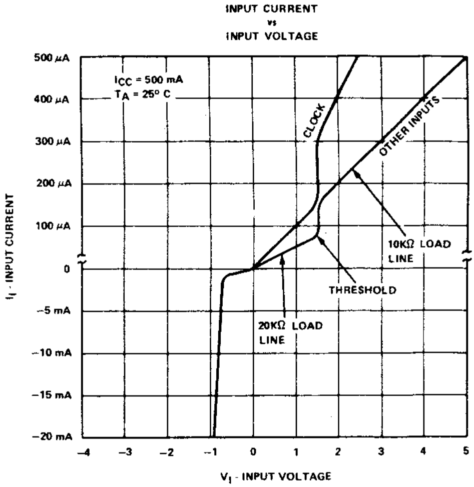


FIGURE 17 – TYPICAL INPUT CHARACTERISTICS

4.1.1 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.

4.1.2 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.

4.2 OUTPUT CIRCUIT

The output circuit selected for the SBP 9900 is an injected open-collector transistor shown in Figure 16B. 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 9900 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 18. At rated injector current, the SBP 9900 output circuit offers a low-level output voltage of typically 220 mV.

The output circuit characteristics for 1) logic-level nigh 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 9900 output restrict both the maximum and minimum value of R_L. Techniques for calculating $R_{L(max)}$ and $R_{L(min)}$ respectively are explained in the SBP 0400A, SBP 0401A data manual in Chapter 2 of this data book.

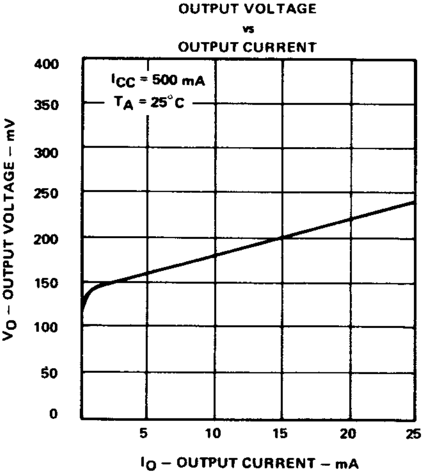
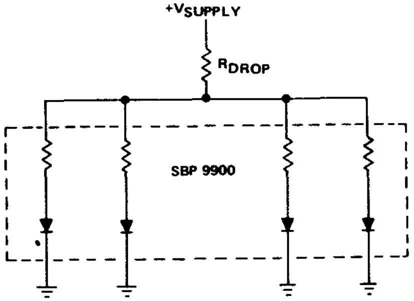


FIGURE 18 – TYPICAL OUTPUT CHARACTERISTICS

5. 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 node 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 19, to select the desired operating current.

An alternate solution utilizes the Texas Instruments TL497 switching-regulator as illustrated in Figure 20.



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 19 – INJECTOR CURRENT CALCULATIONS

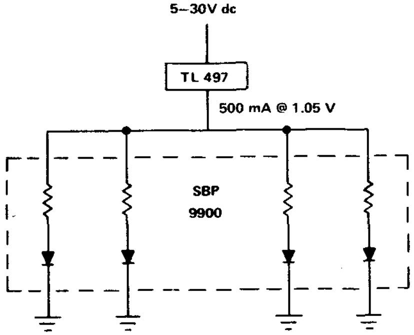


FIGURE 20 – SWITCHING-REGULATOR INJECTOR SOURCE

Operating from a constant current power source, the SBP 9900 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 21.

Figures 22 and 23 show the typical injector node voltages which occur across the temperature and injector current ranges.

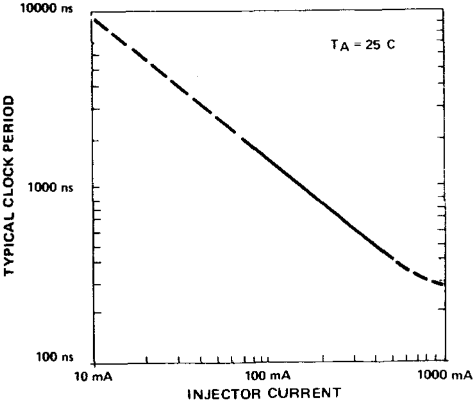


FIGURE 21 – SBP 9900 CLOCK PERIOD VS. INJECTOR CURRENT

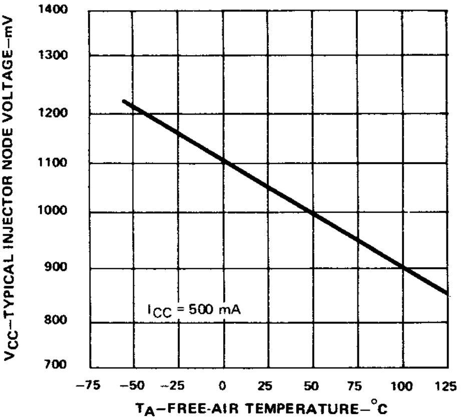


FIGURE 22 – INJECTOR-NODE VOLTAGE VS. FREE-AIR TEMPERATURE

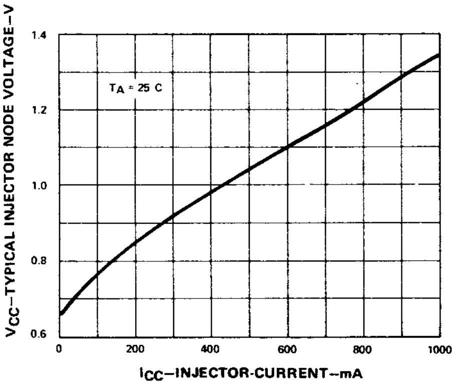


FIGURE 23 – INJECTOR-NODE VOLTAGE VS. INJECTOR CURRENT

6. ELECTRICAL AND MECHANICAL SPECIFICATIONS

6.1 SBP 9900 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)	$\overline{\text{HOLD}}$		160†		ns
	READY		90†		
	D0 – D15		45†		
	CRUIN		25†		
	$\overline{\text{INTREQ}}$		0†		
	IC0 – IC3		0†		
Hold time, t_h (see Figure 24)	$\overline{\text{HOLD}}$		0†		ns
	READY		30†		
	D0 – D15		30†		
	CRUIN		35†		
	$\overline{\text{INTREQ}}$		60†		
	IC0 – IC3		60†		
Operating free-air temperature, T_A		–55		125	°C

† Rising edge of clock pulse is reference.

TENTATIVE DATA SHEET

This document provides tentative information on a new product. Texas Instruments reserves the right to change specifications for this product in any manner without notice

6.2 SBP 9900 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		µA
		All other inputs				240		

†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.

6.3 SBP 9900 SWITCHING CHARACTERISTICS (I_{CC} = 500 mA) SEE FIGURES 24 AND 25.

PARAMETER	FROM	TO	TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
f_{max}	MAXIMUM CLOCK FREQUENCY		C_L = 150 pF, R_L = 280 Ω	2	2.6		MHz
t_{PLH} or t_{PHL}	CLOCK	ADDRESS BUS (A0 – A14)			170		ns
t_{PLH} or t_{PHL}	CLOCK	DATA BUS (D0 – D15)			170		ns
t_{PLH} or t_{PHL}	CLOCK	WRITE ENABLE ($\overline{WE}$)			220		ns
t_{PLH} or t_{PHL}	CLOCK	CYCLE END ($\overline{CYCEND}$)			170		ns
t_{PLH} or t_{PHL}	CLOCK	DATA BUS IN (DBIN)			190		ns
t_{PLH} or t_{PHL}	CLOCK	MEMORY ENABLE ($\overline{MEMEN}$)			155		ns
t_{PLH} or t_{PHL}	CLOCK	CRU CLOCK (CRUCK)			187		ns
t_{PLH} or t_{PHL}	CLOCK	CRU DATA OUT (CRUOUT)			210		ns
t_{PLH} or t_{PHL}	CLOCK	HOLD ACKNOWLEDGE (HLDA)			320		ns
t_{PLH} or t_{PHL}	CLOCK	WAIT			155		ns
t_{PLH} or t_{PHL}	CLOCK	INSTRUCTION ACQUISITION (1AQ)			155		ns

‡All typical values are at 25°C.

TENTATIVE DATA SHEET

This document provides tentative information on a new product. Texas Instruments reserves the right to change specifications for this product in any manner without notice.

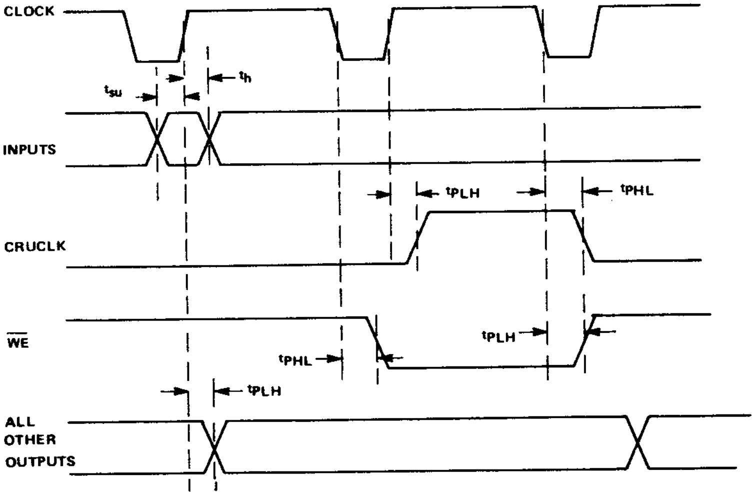


FIGURE 24 – SWITCHING TIMES-VOLTAGE WAVEFORMS

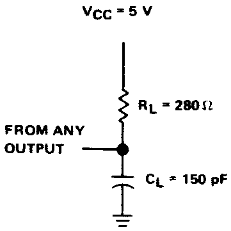


FIGURE 25 – SWITCHING TIMES LOAD CIRCUITS

6.4 CLOCK FREQUENCY VS. TEMPERATURE

Stability of the operational frequency over the full temperature range of −55°C to 125°C is illustrated in Figure 26.

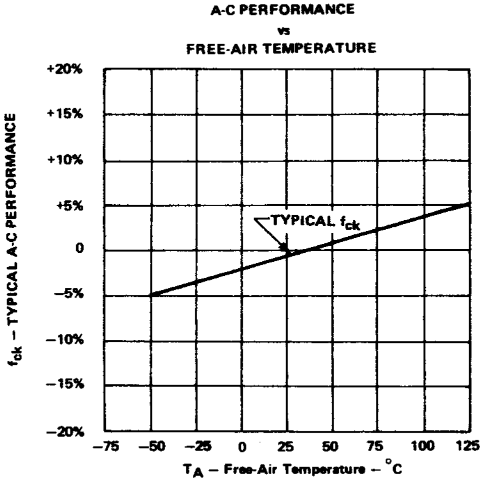
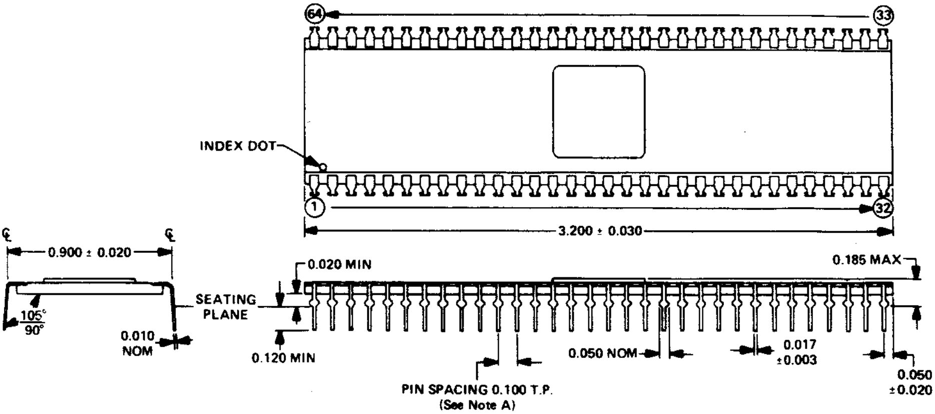


FIGURE 26 – A-C PERFORMANCE VS. TEMPERATURE

7. MECHANICAL DATA



NOTE A: Each pin centerline is located within 0.010 of its true longitudinal position.

8. SBP 9900 PROTOTYPING SYSTEM

8.1 HARDWARE

The TMS 9900 prototyping system enables the user to generate and debug software and to debug I/O controller interfaces. The prototyping system consists of:

- 990/4 computer with TMS 9900 microprocessor
- 1024 bytes of ROM containing the bootstrap loader for loading prototyping system software, the front-panel and maintenance utility, and the CPU self-testing feature
- 16,896 bytes of RAM with provisions for expansion up to 57,334 bytes of RAM
- Programmable-write-protect feature for RAM
- Interface for Texas Instruments Model 733 ASR* Electronic Data Terminal with provisions for up to five additional interface moculdes

* Requires remote device control and 1200 baud EIA interface option on 733 ASR.

- Available with Texas Instruments Model 733 ASR Electronic Data Terminal
- 7-inch-high table-top chassis
- Programmer's front panel with controls for run, halt, single-instruction execute, and entering and displaying memory or register contents
- Power supply with the following voltages:
  5 V dc @ 20 A
  12 V dc @ 2 A
  −12 V dc @ 1 A
  −5 V dc @ 0.1 A
- Complete hardware and software documentation.

8.2 SYSTEM CONSOLE

The system console for the prototyping system is the 733 ASR, which provides keyboard entry, 30-character-per-second thermal printer, and dual cassette drives for program loading and storage.

8.3 SOFTWARE

The following software is provided on cassette for loading into the prototyping system:

- Debug Monitor – Provides full control of the prototyping system during program development and includes single instruction, multiple breakpoints, and entry and display capability for register and memory contents for debugging user software under 733 ASR console control.
- One-Pass Assembler – Converts source code stored on cassette to relocatable object on cassette and generates program listing. (Object is upward compatible with other 990 series assemblers).
- Linking Loader – Allows loading of absolute and relocatable object modules and links object modules as they are loaded.
- Source Editor – Enables user modification of both source and object from cassette with resultant storage on cassette.
- Trace Routine – Allows user to monitor status of computer at completion of each instruction.
- PROM Programming/Documentation Facility – Provides documentation for ROM mask generation, or communicates directly with the optional PROM Programmer Unit.