Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

A P E N D I C E S

" Proyecto y Construcción de un Sistema de Desarrollo Lógico, Una Aplicación de los Microprocesadores "

Tesis Profesional

Que para obtener el título de

Ingeniero Mecánico Electricista

p r e s e n t a

JOSE SALVADOR DE LA MORA REAL

V. II

México, D. F. 1986

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

Appendix A1

Reader's Comments

Your feedback about this document helps us ascertain your needs and fulfill them in the future. Please take the time to fill out this questionaire and return it to us. This information will be helpful to us and, in time, to future users of Zilog products.

Your Name: _____

Company Name: _____

Address: _____

Title of this document: _____

Briefly describe application: _____

_____

_____

Does this publication meet your needs? ☐ Yes ☐ No If not, why not? _____

_____

_____

How are you using this publication?

- ☐ As an introduction to the subject?
- ☐ As a reference manual?
- ☐ As an instructor or student?

How do you find the material?

	Excellent	Good	Poor
Technicality	☐	☐	☐
Organization	☐	☐	☐
Completeness	☐	☐	☐

What would have improved the material? _____

_____

_____

Other comments and suggestions: _____

_____

_____

If you found any mistakes in this document, please let us know what and where they are: _____

_____

_____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL.

FIRST CLASS PERMIT NO. 475 CUPERTINO, CA

POSTAGE WILL BE PAID BY

Zilog
Publications Department
Semiconductor Division
10341 Bubb Road
Cupertino, California 95014

TABLE OF CONTENTS

1.0 INTRODUCTION

The term "microcomputer" has been used to describe virtually every type of small computing device designed within the last few years. This term has been applied to everything from simple "microprogrammed" controllers constructed out of TTL MSI up to low end minicomputers with a portion of the CPU constructed out of TTL LSI "bit slices." However, the major impact of the LSI technology within the last few years has been with MOS LSI. With this technology, it is possible to fabricate complete and very powerful computer systems with only a few MOS LSI components.

The Zilog Z-80 family of components is a significant advancement in the state-of-the art of microcomputers. These components can be configured with any type of standard semiconductor memory to generate computer systems with an extremely wide range of capabilities. For example, as few as two LSI circuits and three standard TTL MSI packages can be combined to form a simple controller. With additional memory and I/O devices a computer can be constructed with capabilities that only a minicomputer could previously deliver. This wide range of computational power allows standard modules to be constructed by a user that can satisfy the requirements of an extremely wide range of applications.

The major reason for MOS LSI domination of the microcomputer market is the low cost of these few LSI components. For example, MOS LSI microcomputers have already replaced TTL logic in such applications as terminal controllers, peripheral device controllers, traffic signal controllers, point of sale terminals, intelligent terminals and test systems. In fact the MOS LSI microcomputer is finding its way into almost every product that now uses electronics and it is even replacing many mechanical systems such as weight scales and automobile controls.

The MOS LSI microcomputer market is already well established and new products using them are being developed at an extraordinary rate. The Zilog Z-80 component set has been designed to fit into this market through the following factors:

1. The Z-80 is fully software compatible with the popular 8080A CPU offered from several sources. Existing designs can be easily converted to include the Z-80 as a superior alternative.
2. The Z-80 component set is superior in both software and hardware capabilities to any other microcomputer system on the market. These capabilities provide the user with significantly lower hardware and software development costs while also allowing him to offer additional features in his system.
3. For increased throughput the Z80A operating at a 4 MHZ clock rate offers the user significant speed advantages over competitive products.
4. A complete product line including full software support with strong emphasis on high level languages and a disk-based development system with advanced real-time debug capabilities is offered to enable the user to easily develop new products.

Microcomputer systems are extremely simple to construct using Z-80 components. Any such system consists of three parts:

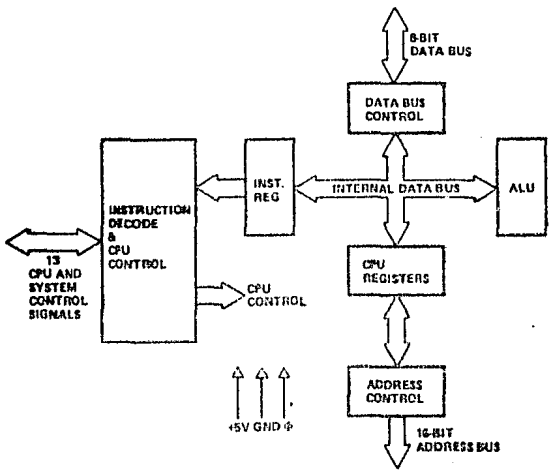
1. CPU (Central Processing Unit)
2. Memory
3. Interface Circuits to peripheral devices

The CPU is the heart of the system. Its function is to obtain instructions from the memory and perform the desired operations. The memory is used to contain instructions and in most cases data that is to be processed. For example, a typical instruction sequence may be to read data from a specific peripheral device, store it in a location in memory, check the parity and write it out to another peripheral device. Note that the Zilog component set includes the CPU and various general purpose I/O device controllers, while a wide range of memory devices may be used from any source. Thus, all required components can be connected together in a very simple manner with virtually no other external logic. The user's effort then becomes primarily one of software development. That is, the user can concentrate on describing his problem and translating it into a series of instructions that can be loaded into the microcomputer memory. Zilog is dedicated to making this step of software generation as simple as possible. A good example of this is our

assembly language in which a simple mnemonic is used to represent every instruction that the CPU can perform. This language is self documenting in such a way that from the mnemonic the user can understand exactly what the instruction is doing without constantly checking back to a complex cross listing.

2.0 Z-80 CPU ARCHITECTURE

A block diagram of the internal architecture of the Z-80 CPU is shown in figure 2.0-1. The diagram shows all of the major elements in the CPU and it should be referred to throughout the following description.



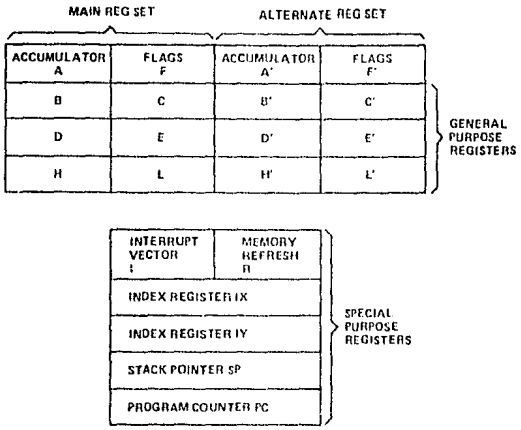
Z-80 CPU BLOCK DIAGRAM
FIGURE 2.0-1

2.1 CPU REGISTERS

The Z-80 CPU contains 208 bits of R/W memory that are accessible to the programmer. Figure 2.0-2 illustrates how this memory is configured into eighteen 8-bit registers and four 16-bit registers. All Z-80 registers are implemented using static RAM. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or in pairs as 16-bit registers. There are also two sets of accumulator and flag registers.

Special Purpose Registers

1. Program Counter (PC). The program counter holds the 16-bit address of the current instruction being fetched from memory. The PC is automatically incremented after its contents have been transferred to the address lines. When a program jump occurs the new value is automatically placed in the PC, overriding the incrementer.
2. Stack Pointer (SP). The stack pointer holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in first-out (LIFO) file. Data can be pushed onto the stack from specific CPU registers or popped off of the stack into specific CPU registers through the execution of PUSH and POP instructions. The data popped from the stack is always the last data pushed onto it. The stack allows simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data manipulation.



Z-80 CPU REGISTER CONFIGURATION
FIGURE 2.0-2

3. **Two Index Registers (IX & IY).** The two independent index registers hold a 16-bit base address that is used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's complement signed integer. This mode of addressing greatly simplifies many types of programs, especially where tables of data are used.

4. **Interrupt Page Address Register (I).** The Z-80 CPU can be operated in a mode where an indirect call to any memory location can be achieved in response to an interrupt. The I Register is used for this purpose to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. This feature allows interrupt routines to be dynamically located anywhere in memory with absolute minimal access time to the routine.

5. **Memory Refresh Register (R).** The Z-80 CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. Seven bits of this 8 bit register are automatically incremented after each instruction fetch. The eighth bit will remain as programmed as the result of an LD R, A instruction. The data in the refresh counter is sent out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer. During refresh, the contents of the I register are placed on the upper 8 bits of the address bus.

Accumulator and Flag Registers

The CPU includes two independent 8-bit accumulators and associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operations while the flag register indicates specific conditions for 8 or 16-bit operations, such as indicating whether or not the result of an operation is equal to zero. The programmer selects the accumulator and flag pair that he wishes to work with with a single exchange instruction so that he may easily work with either pair.

General Purpose Registers

There are two matched sets of general purpose registers, each set containing six 8-bit registers that may be used individually as 8-bit registers or as 16-bit register pairs by the programmer. One set is called BC, DE and HL while the complementary set is called BC', DE' and HL'. At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set. In systems where fast interrupt response is required, one set of general purpose registers and an accumulator/flag register may be reserved for handling this very fast routine. Only a simple exchange commands need be executed to go between the routines. This greatly reduces interrupt service time by eliminating the requirement for saving and retrieving register contents in the external stack during interrupt or subroutine processing. These general purpose registers are used for a wide range of applications by the programmer. They also simplify programming, especially in ROM based systems where little external read/write memory is available.

2.2 ARITHMETIC & LOGIC UNIT (ALU)

The 8-bit arithmetic and logical instructions of the CPU are executed in the ALU. Internally the ALU communicates with the registers and the external data bus on the internal data bus. The type of functions performed by the ALU include:

Add	Left or right shifts or rotates (arithmetic and logical)
Subtract	Increment
Logical AND	Decrement
Logical OR	Set bit
Logical Exclusive OR	Reset bit
Compare	Test bit

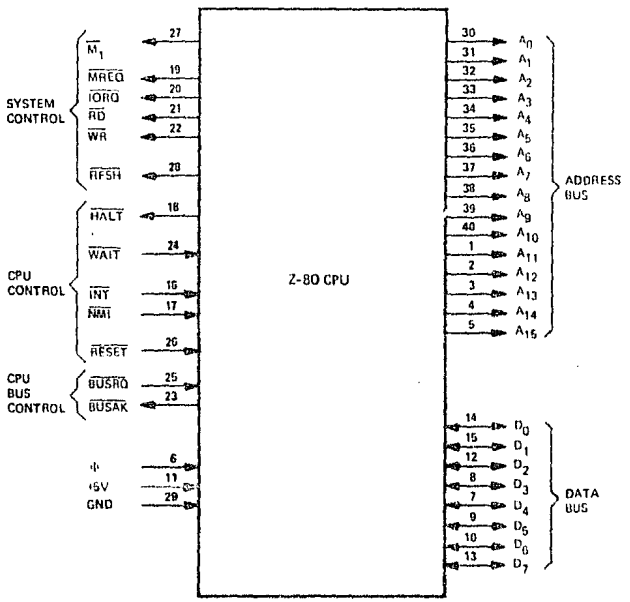
2.3 INSTRUCTION REGISTER AND CPU CONTROL

As each instruction is fetched from memory, it is placed in the instruction register and decoded. The control sections performs this function and then generates and supplies all of the control signals necessary to read or write data from or to the registers, control the ALU and provide all required external control signals.

-BLANK-

3.0 Z-80 CPU PIN DESCRIPTION

The Z-80 CPU is packaged in an industry standard 40 pin Dual In-Line Package. The I/O pins are shown in figure 3.0-1 and the function of each is described below.



Z-80 PIN CONFIGURATION
FIGURE 3.0-1

A_0-A_{15}
(Address Bus)

Tri-state output, active high. A_0-A_{15} constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges. I/O addressing uses the 8 lower address bits to allow the user to directly select up to 256 input or 256 output ports. A_0 is the least significant address bit. During refresh time, the lower 7 bits contain a valid refresh address.

D_0-D_7
(Data Bus)

Tri-state input/output, active high. D_0-D_7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

$\overline{M_1}$
(Machine Cycle one)

Output, active low. $\overline{M_1}$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution. Note that during execution of 2-byte op-codes, $\overline{M1}$ is generated as each op code byte is fetched. These two byte op-codes always begin with CBH, DDH, EDH or FDH. $\overline{M1}$ also occurs with $\overline{IORQ}$ to indicate an interrupt acknowledge cycle.

$\overline{MREQ}$
(Memory Request)

Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{\text{IORQ}}$ (Input/Output Request)	Tri-state output, active low. The $\overline{\text{IORQ}}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{\text{IORQ}}$ signal is also generated with an $\overline{\text{M1}}$ signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. Interrupt Acknowledge operations occur during M_1 time while I/O operations never occur during M_1 time.
$\overline{\text{RD}}$ (Memory Read)	Tri-state output, active low. $\overline{\text{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.
$\overline{\text{WR}}$ (Memory Write)	Tri-state output, active low. $\overline{\text{WR}}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.
$\overline{\text{RFSH}}$ (Refresh)	Output, active low. $\overline{\text{RFSH}}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current $\overline{\text{MREQ}}$ signal should be used to do a refresh read to all dynamic memories.
$\overline{\text{HALT}}$ (Halt state)	Output, active low. $\overline{\text{HALT}}$ indicates that the CPU has executed a HALT software instruction and is awaiting either a non maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.
$\overline{\text{WAIT}}$ (Wait)	Input, active low. $\overline{\text{WAIT}}$ indicates to the Z-80 CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU.
$\overline{\text{INT}}$ (Interrupt Request)	Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled and if the $\overline{\text{BUSRQ}}$ signal is not active. When the CPU accepts the interrupt, an acknowledge signal ($\overline{\text{IORQ}}$ during M_1 time) is sent out at the beginning of the next instruction cycle. The CPU can respond to an interrupt in three different modes that are described in detail in section 5.4 (CPU Control Instructions).
$\overline{\text{NMI}}$ (Non Maskable Interrupt)	Input, negative edge triggered. The non maskable interrupt request line has a higher priority than $\overline{\text{INT}}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{\text{NMI}}$ automatically forces the Z-80 CPU to restart to location 0066_H. The program counter is automatically saved in the external stack so that the user can return to the program that was interrupted. Note that continuous WAIT cycles can prevent the current instruction from ending, and that a $\overline{\text{BUSRQ}}$ will override a $\overline{\text{NMI}}$.

$\overline{\text{RESET}}$	Input, active low. $\overline{\text{RESET}}$ forces the program counter to zero and initializes the CPU. The CPU initialization includes: 1) Disable the interrupt enable flip-flop 2) Set Register I = 00_H 3) Set Register R = 00_H 4) Set Interrupt Mode 0 During reset time, the address bus and data bus go to a high impedance state and all control output signals go to the inactive state.
$\overline{\text{BUSRQ}}$ (Bus Request)	Input, active low. The bus request signal is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these buses. When $\overline{\text{BUSRQ}}$ is activated, the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated.
$\overline{\text{BUSAK}}$ (Bus Acknowledge)	Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.
Φ	Single phase TTL level clock which requires only a 330 ohm pull-up resistor to +5 volts to meet all clock requirements.

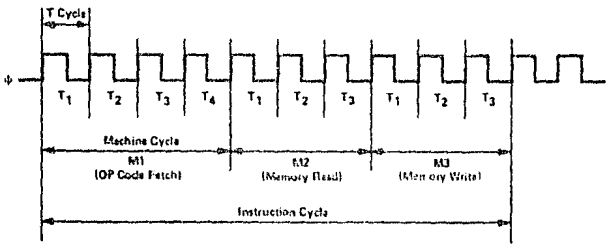
-BLANK-

4.0 CPU TIMING

The Z-80 CPU executes instructions by stepping through a very precise set of a few basic operations. These include:

Memory read or write
I/O device read or write
Interrupt acknowledge

All instructions are merely a series of these basic operations. Each of these basic operations can take from three to six clock periods to complete or they can be lengthened to synchronize the CPU to the speed of external devices. The basic clock periods are referred to as T cycles and the basic operations are referred to as M (for machine) cycles. Figure 4.0-0 illustrates how a typical instruction will be merely a series of specific M and T cycles. Notice that this instruction consists of three machine cycles (M1, M2 and M3). The first machine cycle of any instruction is a fetch cycle which is four, five or six T cycles long (unless lengthened by the wait signal which will be fully described in the next section). The fetch cycle (M1) is used to fetch the OP code of the next instruction to be executed. Subsequent machine cycles move data between the CPU and memory or I/O devices and they may have anywhere from three to five T cycles (again they may be lengthened by wait states to synchronize the external devices to the CPU). The following paragraphs describe the timing which occurs within any of the basic machine cycles. In section 7, the exact timing for each instruction is specified.



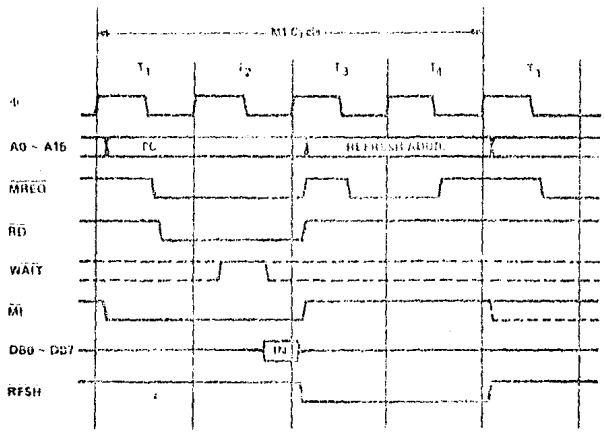
BASIC CPU TIMING EXAMPLE
FIGURE 4.0-0

All CPU timing can be broken down into a few very simple timing diagrams as shown in figure 4.0-1 through 4.0-7. These diagrams show the following basic operations with and without wait states (wait states are added to synchronize the CPU to slow memory or I/O devices).

4.0-1. Instruction OP code fetch (M1 cycle)
4.0-2. Memory data read or write cycles
4.0-3. I/O read or write cycles
4.0-4. Bus Request/Acknowledge Cycle
4.0-5. Interrupt Request/Acknowledge Cycle
4.0-6. Non maskable Interrupt Request/Acknowledge Cycle
4.0-7. Exit from a HALT instruction

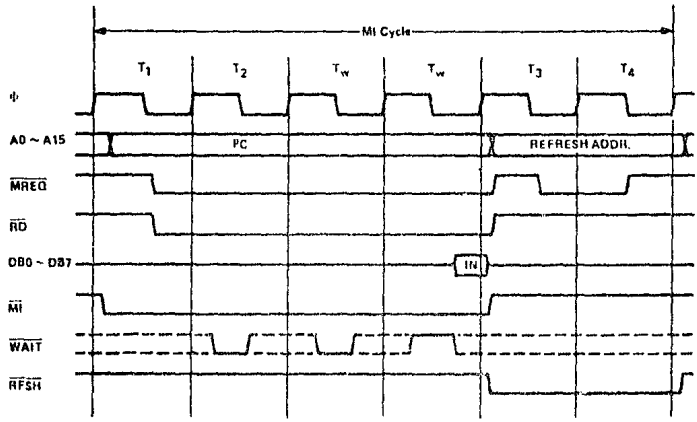
INSTRUCTION FETCH

Figure 4.0-1 shows the timing during an M1 cycle (OP code fetch). Notice that the PC is placed on the address bus at the beginning of the M1 cycle. One half clock time later the $\overline{MREQ}$ signal goes active. At this time the address to the memory has had time to stabilize so that the falling edge of $\overline{MREQ}$ can be used directly as a chip enable clock to dynamic memories. The $\overline{RD}$ line also goes active to indicate that the memory read data should be enabled onto the CPU data bus. The CPU samples the data from the memory on the data bus with the rising edge of the clock of state T3 and this same edge is used by the CPU to turn off the $\overline{RD}$ and $\overline{MRQ}$ signals. Thus the data has already been sampled by the CPU before the $\overline{RD}$ signal becomes inactive. Clock state T3 and T4 of a fetch cycle are used to refresh dynamic memories. (The CPU uses this time to decode and execute the fetched instruction so that no other operation could be performed at this time). During T3 and T4 the lower 7 bits of the address bus contain a memory refresh address and the $\overline{RFSH}$ signal becomes active to indicate that a refresh read of all dynamic memories should be accomplished. Notice that a $\overline{RD}$ signal is not generated during refresh time to prevent data from different memory segments from being gated onto the data bus. The $\overline{MREQ}$ signal during refresh time should be used to perform a refresh read of all memory elements. The refresh signal can not be used by itself since the refresh address is only guaranteed to be stable during $\overline{MREQ}$ time.



INSTRUCTION OP CODE FETCH
FIGURE 4.0-1

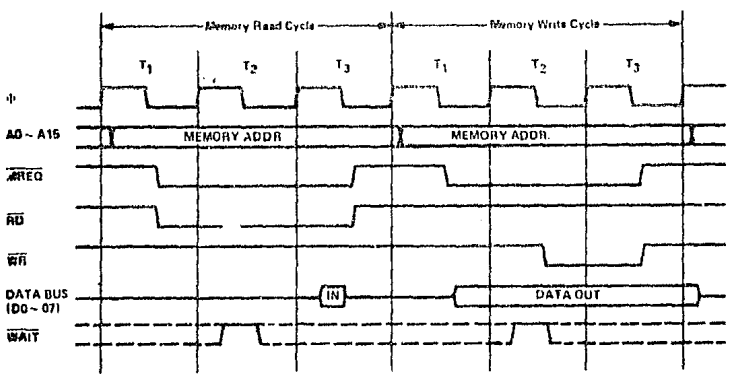
Figure 4.0-1A illustrates how the fetch cycle is delayed if the memory activates the $\overline{WAIT}$ line. During T2 and every subsequent Tw, the CPU samples the $\overline{WAIT}$ line with the falling edge of Φ. If the $\overline{WAIT}$ line is active at this time, another wait state will be entered during the following cycle. Using this technique the read cycle can be lengthened to match the access time of any type of memory device.



INSTRUCTION OP CODE FETCH WITH WAIT STATES
FIGURE 4.0-1A

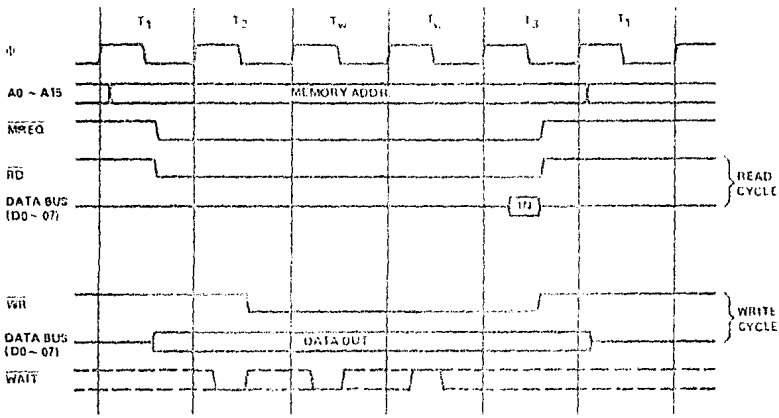
MEMORY READ OR WRITE

Figure 4.0-2 illustrates the timing of memory read or write cycles other than an OP code fetch (M1 cycle). These cycles are generally three clock periods long unless wait states are requested by the memory via the WAIT signal. The MREQ signal and the RD signal are used the same as in the fetch cycle. In the case of a memory write cycle, the MREQ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The WR line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory. Furthermore the WR signal goes inactive one half T state before the address and data bus contents are changed so that the overlap requirements for virtually any type of semiconductor memory type will be met.



MEMORY READ OR WRITE CYCLES
FIGURE 4.0-2

Figure 4.0-2A illustrates how a $\overline{WAIT}$ request signal will lengthen any memory read or write operation. This operation is identical to that previously described for a fetch cycle. Notice in this figure that a separate read and a separate write cycle are shown in the same figure although read and write cycles can never occur simultaneously.



MEMORY READ OR WRITE CYCLES WITH WAIT STATES
FIGURE 4.0-2A

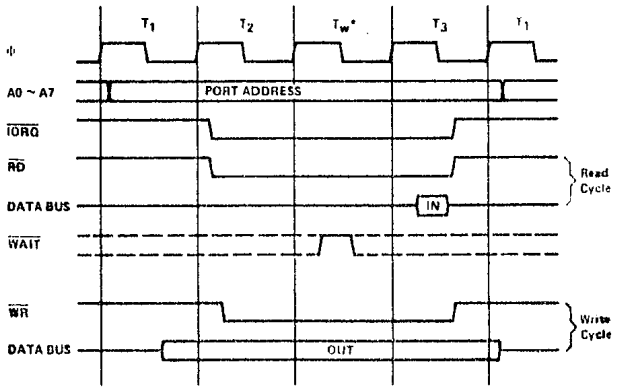
INPUT OR OUTPUT CYCLES

Figure 4.0-3 illustrates an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted. The reason for this is that during I/O operations, the time from when the $\overline{IORQ}$ signal goes active until the CPU must sample the $\overline{WAIT}$ line is very short and without this extra state sufficient time does not exist for an I/O port to decode its address and activate the $\overline{WAIT}$ line if a wait is required. Also, without this wait state it is difficult to design MOS I/O devices that can operate at full CPU speed. During this wait state time the $\overline{WAIT}$ request signal is sampled. During a read I/O operation, the $\overline{RD}$ line is used to enable the addressed port onto the data bus just as in the case of a memory read. For I/O write operations, the $\overline{WR}$ line is used as a clock to the I/O port, again with sufficient overlap timing automatically provided so that the rising edge may be used as a data clock.

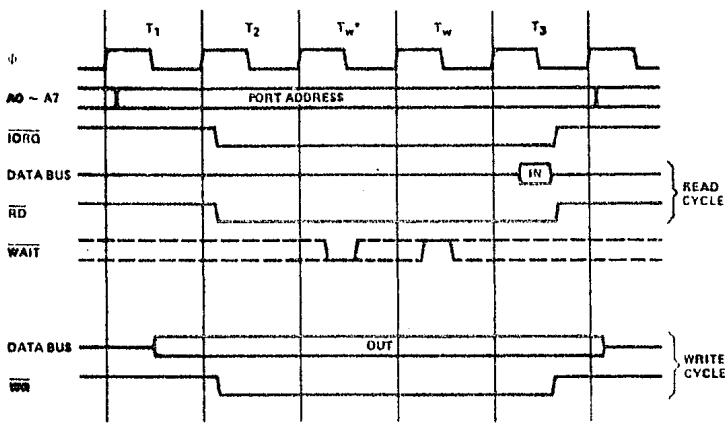
Figure 4.0-3A illustrates how additional wait states may be added with the $\overline{WAIT}$ line. The operation is identical to that previously described.

BUS REQUEST/ACKNOWLEDGE CYCLE

Figure 4.0-4 illustrates the timing for a Bus Request/Acknowledge cycle. The $\overline{BUSRQ}$ signal is sampled by the CPU with the rising edge of the last clock period of any machine cycle. If the $\overline{BUSRQ}$ signal is active, the CPU will set its address, data and tri-state control signals to the high impedance state with the rising edge of the next clock pulse. At that time any external device can control the buses to transfer data between memory and I/O devices. (This is generally known as Direct Memory Access [DMA] using cycle stealing). The maximum time for the CPU to respond to a bus request is the length of a machine cycle and the external controller can maintain control of the bus for as many clock cycles as is desired. Note, however, that if very long DMA cycles are used, and dynamic memories are being used, the external controller must also perform the refresh function. This situation only occurs if very large blocks of data are transferred under DMA control. Also note that during a bus request cycle, the CPU cannot be interrupted by either a $\overline{NMI}$ or an $\overline{INT}$ signal.

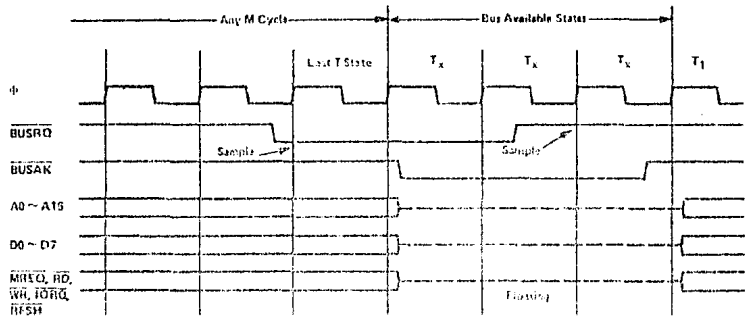


INPUT OR OUTPUT CYCLES
FIGURE 4.0-3



INPUT OR OUTPUT CYCLES WITH WAIT STATES
FIGURE 4.0-3A

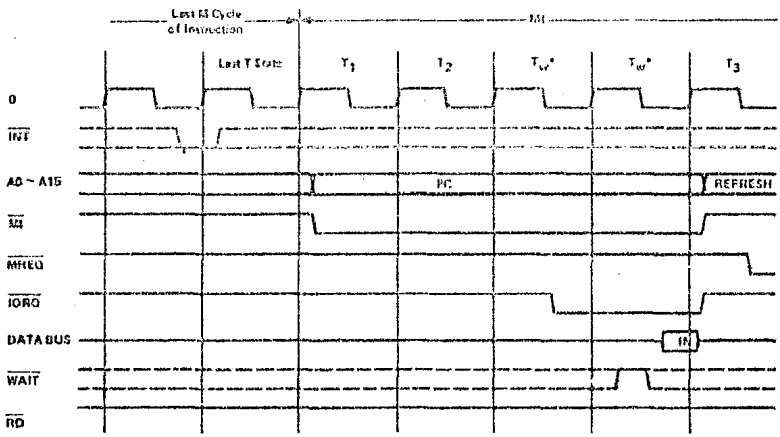
* Automatically inserted WAIT state



BUS REQUEST/ACKNOWLEDGE CYCLE
FIGURE 4.0-4

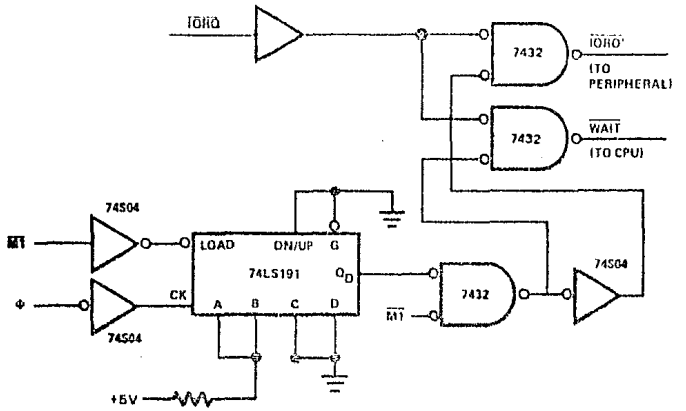
INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

Figure 4.0-5 illustrates the timing associated with an interrupt cycle. The interrupt signal ($\overline{INT}$) is sampled by the CPU with the rising edge of the last clock at the end of any instruction. The signal will not be accepted if the internal CPU software controlled interrupt enable flip-flop is not set or if the $\overline{BUSRQ}$ signal is active. When the signal is accepted a special M1 cycle is generated. During this special M1 cycle the $\overline{IORQ}$ signal becomes active (instead of the normal $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. Notice that two wait states are automatically added to this cycle. These states are added so that a ripple priority interrupt scheme can be easily implemented. The two wait states allow sufficient time for the ripple signals to stabilize and identify which I/O device must insert the response vector. Refer to section 8.0 for details on how the interrupt response vector is utilized by the CPU.

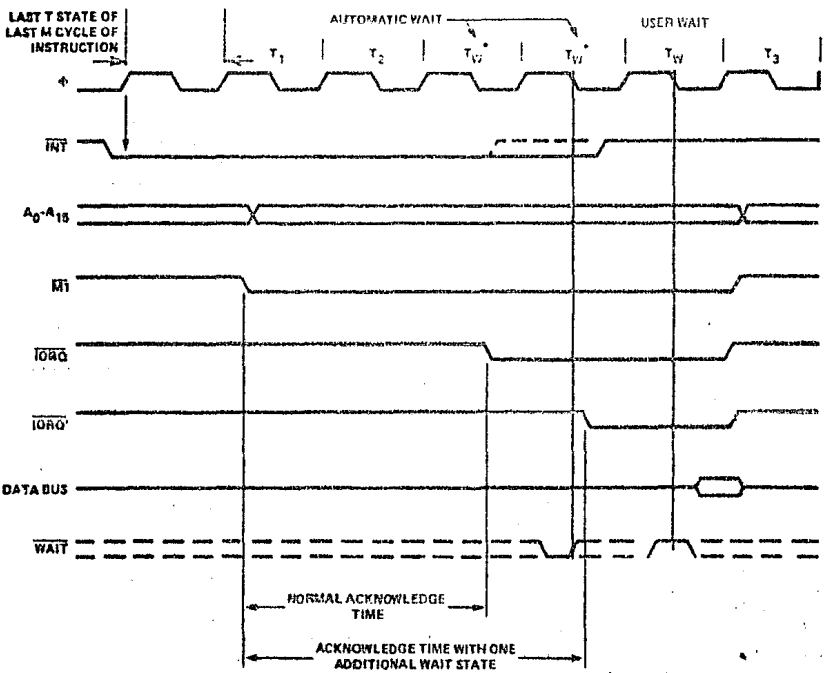


INTERRUPT REQUEST/ACKNOWLEDGE CYCLE
FIGURE 4.0-5

Figures 4.0-5A and 4.0-5B illustrate how a programmable counter can be used to extend interrupt acknowledge time. (Configured as shown to add one wait state)



EXTENDING INTERRUPT ACKNOWLEDGE TIME WITH WAIT STATE
FIGURE 4.0-5A



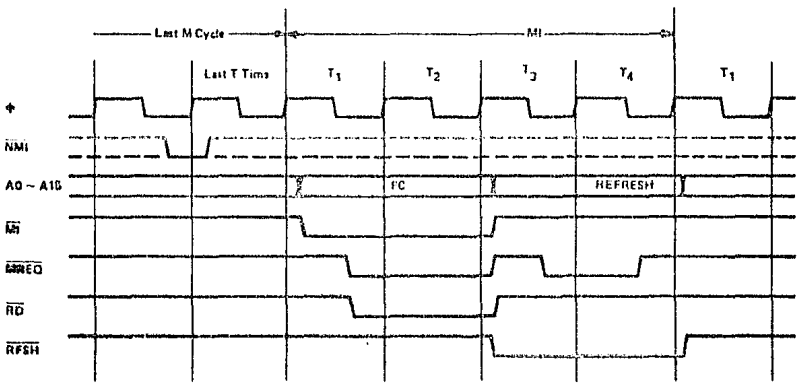
REQUEST/ACKNOWLEDGE CYCLE WITH ONE ADDITIONAL WAIT STATE
FIGURE 4.0-5B

NON MASKABLE INTERRUPT RESPONSE

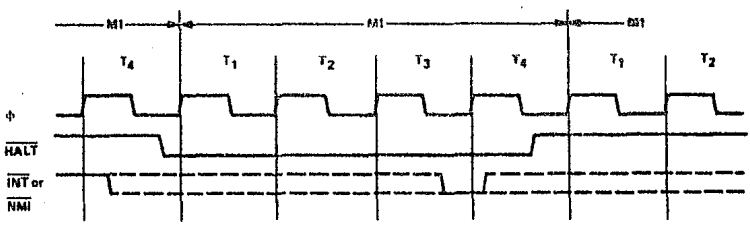
Figure 4.0-6 illustrates the request/acknowledge cycle for the non maskable interrupt. This signal is sampled at the same time as the interrupt line, but this line has priority over the normal interrupt and it can not be disabled under software control. Its usual function is to provide immediate response to important signals such as an impending power failure. The CPU response to a non maskable interrupt is similar to a normal memory read operation. The only difference being that the content of the data bus is ignored while the processor automatically stores the PC in the external stack and jumps to location 0066_H. The service routine for the non maskable interrupt must begin at this location if this interrupt is used.

HALT EXIT

Whenever a software halt instruction is executed the CPU begins executing NOP's until an interrupt is received (either a non maskable or a maskable interrupt while the interrupt flip flop is enabled). The two interrupt lines are sampled with the rising clock edge during each T4 state as shown in figure 4.0-7. If a non maskable interrupt has been received or a maskable interrupt has been received and the interrupt enable flip-flop is set, then the halt state will be exited on the next rising clock edge. The following cycle will then be an interrupt acknowledge cycle corresponding to the type of interrupt that was received. If both are received at this time, then the non maskable one will be acknowledged since it has highest priority. The purpose of executing NOP instructions while in the halt state is to keep the memory refresh signals active. Each cycle in the halt state is a normal M1 (fetch) cycle except that the data received from the memory is ignored and a NOP instruction is forced internally to the CPU. The halt acknowledge signal is active during this time to indicate that the processor is in the halt state.



NON MASKABLE INTERRUPT REQUEST OPERATION
FIGURE 4.0-6



HALT INSTRUCTION IS RECEIVED DURING THIS MEMORY CYCLE

HALT EXIT
FIGURE 4.0-7

5.0 Z-80 CPU INSTRUCTION SET

The Z-80 CPU can execute 158 different instruction types including all 78 of the 8080A CPU. The instructions can be broken down into the following major groups:

- Load and Exchange
- Block Transfer and Search
- Arithmetic and Logical
- Rotate and Shift
- Bit Manipulation (set, reset, test)
- Jump, Call and Return
- Input/Output
- Basic CPU Control

5.1 INTRODUCTION TO INSTRUCTION TYPES

The load instructions move data internally between CPU registers or between CPU registers and external memory. All of these instructions must specify a source location from which the data is to be moved and a destination location. The source location is not altered by a load instruction. Examples of load group instructions include moves between any of the general purpose registers such as move the data to Register B from Register C. This group also includes load immediate to any CPU register or to any external memory location. Other types of load instructions allow transfer between CPU registers and memory locations. The exchange instructions can trade the contents of two registers.

A unique set of block transfer instructions is provided in the Z-80. With a single instruction a block of memory of any size can be moved to any other location in memory. This set of block moves is extremely valuable when large strings of data must be processed. The Z-80 block search instructions are also valuable for this type of processing. With a single instruction, a block of external memory of any desired length can be searched for any 8-bit character. Once the character is found or the end of the block is reached, the instruction automatically terminates. Both the block transfer and the block search instructions can be interrupted during their execution so as to not occupy the CPU for long periods of time.

The arithmetic and logical instructions operate on data stored in the accumulator and other general purpose CPU registers or external memory locations. The results of the operations are placed in the accumulator and the appropriate flags are set according to the result of the operation. An example of an arithmetic operation is adding the accumulator to the contents of an external memory location. The results of the addition are placed in the accumulator. This group also includes 16-bit addition and subtraction between 16-bit CPU registers.

The rotate and shift group allows any register or any memory location to be rotated right or left with or without carry either arithmetic or logical. Also, a digit in the accumulator can be rotated right or left with two digits in any memory location.

The bit manipulation instructions allow any bit in the accumulator, any general purpose register or any external memory location to be set, reset or tested with a single instruction. For example, the most significant bit of register H can be reset. This group is especially useful in control applications and for controlling software flags in general purpose programming.

The jump, call and return instructions are used to transfer between various locations in the user's program. This group uses several different techniques for obtaining the new program counter address from specific external memory locations. A unique type of call is the restart instruction. This instruction actually contains the new address as a part of the 8-bit OP code. This is possible since only 8 separate addresses located in page zero of the external memory may be specified. Program jumps may also be achieved by loading register HL, IX or IY directly into the PC, thus allowing the jump address to be a complex function of the routine being executed.

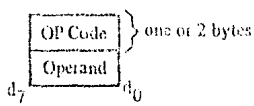
The input/output group of instructions in the Z-80 allow for a wide range of transfers between external memory locations or the general purpose CPU registers, and the external I/O devices. In each case, the port number is provided on the lower 8 bits of the address bus during any I/O transaction. One instruction allows this port number to be specified by the second byte of the instruction while other Z-80 instructions allow it to be specified as the content of the C register. One major advantage of using the C register as a pointer to the I/O device is that it allows different I/O ports to share common software driver routines. This is not possible when the address is part of the OP code if the routines are stored in ROM. Another feature of these input instructions is that they set the flag register automatically so that additional operations are not required to determine the state of the input data (for example its parity). The Z-80 CPU includes single instructions that can move blocks of data (up to 256 bytes) automatically to or from any I/O port directly to any memory location. In conjunction with the dual set of general purpose registers, these instructions provide for fast I/O block transfer rates. The value of this I/O instruction set is demonstrated by the fact that the Z-80 CPU can provide all required floppy disk formatting (i.e., the CPU provides the preamble, address, data and enables the CRC codes) on double density floppy disk drives on an interrupt driven basis.

Finally, the basic CPU control instructions allow various options and modes. This group includes instructions such as setting or resetting the interrupt enable flip flop or setting the mode of interrupt response.

5.2 ADDRESSING MODES

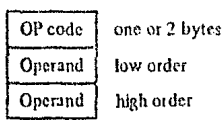
Most of the Z-80 instructions operate on data stored in internal CPU registers, external memory or in the I/O ports. Addressing refers to how the address of this data is generated in each instruction. This section gives a brief summary of the types of addressing used in the Z-80 while subsequent sections detail the type of addressing available for each instruction group.

Immediate. In this mode of addressing the byte following the OP code in memory contains the actual operand.



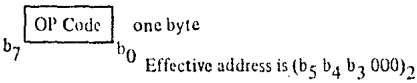
Examples of this type of instruction would be to load the accumulator with a constant, where the constant is the byte immediately following the OP code.

Immediate Extended. This mode is merely an extension of immediate addressing in that the two bytes following the OP codes are the operand.

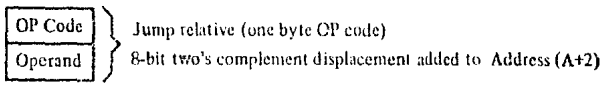


Examples of this type of instruction would be to load the HL register pair (16-bit register) with 16 bits (2 bytes) of data.

Modified Page Zero Addressing. The Z-80 has a special single byte CALL instruction to any of 8 locations in page zero of memory. This instruction (which is referred to as a restart) sets the PC to an effective address in page zero. The value of this instruction is that it allows a single byte to specify a complete 16-bit address where commonly called subroutines are located, thus saving memory space.

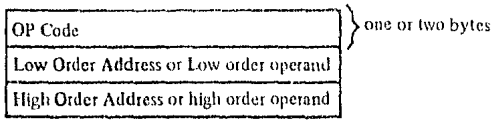


Relative Addressing. Relative addressing uses one byte of data following the OP code to specify a displacement from the existing program to which a program jump can occur. This displacement is a signed two's complement number that is added to the address of the OP code of the following instruction.



The value of relative addressing is that it allows jumps to nearby locations while only requiring two bytes of memory space. For most programs, relative jumps are by far the most prevalent type of jump due to the proximity of related program segments. Thus, these instructions can significantly reduce memory space requirements. The signed displacement can range between +127 and -128 from A + 2. This allows for a total displacement of +129 to -126 from the jump relative OP code address. Another major advantage is that it allows for relocatable code.

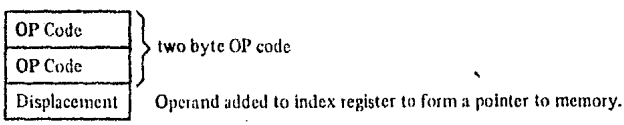
Extended Addressing. Extended Addressing provides for two bytes (16 bits) of address to be included in the instruction. This data can be an address to which a program can jump or it can be an address where an operand is located.



Extended addressing is required for a program to jump from any location in memory to any other location, or load and store data in any memory location.

When extended addressing is used to specify the source or destination address of an operand, the notation (nn) will be used to indicate the content of memory at nn, where nn is the 16-bit address specified in the instruction. This means that the two bytes of address nn are used as a pointer to a memory location. The use of the parentheses always means that the value enclosed within them is used as a pointer to a memory location. For example, (1200) refers to the contents of memory at location 1200.

Indexed Addressing. In this type of addressing, the byte of data following the OP code contains a displacement which is added to one of the two index registers (the OP code specifies which index register is used) to form a pointer to memory. The contents of the index register are not altered by this operation.



An example of an indexed instruction would be to load the contents of the memory location (Index Register + Displacement) into the accumulator. The displacement is a signed two's complement number. Indexed addressing greatly simplifies programs using tables of data since the index register can point to the start of any table. Two index registers are provided since very often operations require two or more tables. Indexed addressing also allows for relocatable code.

The two index registers in the Z-80 are referred to as IX and IY. To indicate indexed addressing the notation:

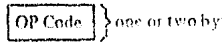
(IX+d) or (IY+d)

is used. Here d is the displacement specified after the OP code. The parentheses indicate that this value is used as a pointer to external memory.

Register Addressing. Many of the Z-80 OP codes contain bits of information that specify which CPU register is to be used for an operation. An example of register addressing would be to load the data in register B into register C.

Implied Addressing. Implied addressing refers to operations where the OP code automatically implies one or more CPU registers as containing the operands. An example is the set of arithmetic operations where the accumulator is always implied to be the destination of the results.

Register Indirect Addressing. This type of addressing specifies a 16-bit CPU register pair (such as HL) to be used as a pointer to any location in memory. This type of instruction is very powerful and it is used in a wide range of applications.



An example of this type of instruction would be to load the accumulator with the data in the memory location pointed to by the HL register contents. Indexed addressing is actually a form of register indirect addressing except that a displacement is added with indexed addressing. Register indirect addressing allows for very powerful but simple to implement memory accesses. The block move and search commands in the Z-80 are extensions of this type of addressing where automatic register incrementing, decrementing and comparing has been added. The notation for indicating register indirect addressing is to put parentheses around the name of the register that is to be used as the pointer. For example, the symbol

(HL)

specifies that the contents of the HL register are to be used as a pointer to a memory location. Often register indirect addressing is used to specify 16-bit operands. In this case, the register contents point to the lower order portion of the operand while the register contents are automatically incremented to obtain the upper portion of the operand.

Bit Addressing. The Z-80 contains a large number of bit set, reset and test instructions. These instructions allow any memory location or CPU register to be specified for a bit operation through one of three previous addressing modes (register, register indirect and indexed) while three bits in the OP code specify which of the eight bits is to be manipulated.

ADDRESSING MODE COMBINATIONS

Many instructions include more than one operand (such as arithmetic instructions or loads). In these cases, two types of addressing may be employed. For example, load can use immediate addressing to specify the source and register indirect or indexed addressing to specify the destination.

5.3 INSTRUCTION OP CODES

This section describes each of the Z-80 instructions and provides tables listing the OP codes for every instruction. In each of these tables the OP codes in shaded areas are identical to those offered in the 8080A CPU. Also shown is the assembly language mnemonic that is used for each instruction. All instruction OP codes are listed in hexadecimal notation. Single byte OP codes require two hex characters while double byte OP codes require four hex characters. The conversion from hex to binary is repeated here for convenience.

Hex		Binary		Decimal	Hex		Binary		Decimal
0	=	0000	=	0	8	=	1000	=	8
1	=	0001	=	1	9	=	1001	=	9
2	=	0010	=	2	A	=	1010	=	10
3	=	0011	=	3	B	=	1011	=	11
4	=	0100	=	4	C	=	1100	=	12
5	=	0101	=	5	D	=	1101	=	13
6	=	0110	=	6	E	=	1110	=	14
7	=	0111	=	7	F	=	1111	=	15

Z-80 instruction mnemonics consist of an OP code and zero, one or two operands. Instructions in which the operand is implied have no operand. Instructions which have only one logical operand or those in which one operand is invariant (such as the Logical OR instruction) are represented by a one operand mnemonic. Instructions which may have two varying operands are represented by two operand mnemonics.

LOAD AND EXCHANGE

Table 5.3-1 defines the OP code for all of the 8-bit load instructions implemented in the Z-80 CPU. Also shown in this table is the type of addressing used for each instruction. The source of the data is found on the top horizontal row while the destination is specified by the left hand column. For example, load register C from register B uses the OP code 48H. In all of the tables the OP code is specified in hexadecimal notation and the 48H (=0100 1000 binary) code is fetched by the CPU from the external memory during M1 time, decoded and then the register transfer is automatically performed by the CPU.

The assembly language mnemonic for this entire group is LD, followed by the destination followed by the source (LD DEST., SOURCE). Note that several combinations of addressing modes are possible. For example, the source may use register addressing and the destination may be register indirect; such as load the memory location pointed to by register HL with the contents of register D. The OP code for this operation would be 72. The mnemonic for this load instruction would be as follows:

LD (HL), D

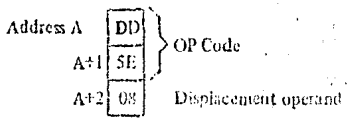
The parentheses around the HL means that the contents of HL are used as a pointer to a memory location. In all Z-80 load instruction mnemonics the destination is always listed first, with the source following. The Z-80 assembly language has been defined for ease of programming. Every instruction is self documenting and programs written in Z-80 language are easy to maintain.

Note in table 5.3-1 that some load OP codes that are available in the Z-80 use two bytes. This is an efficient method of memory utilization since 8, 16, 24 or 32 bit instructions are implemented in the Z-80. Thus often utilized instructions such as arithmetic or logical operations are only 8-bits which results in better memory utilization than is achieved with fixed instruction sizes such as 16-bits.

All load instructions using indexed addressing for either the source or destination location actually use three bytes of memory with the third byte being the displacement d. For example a load register E with the operand pointed to by IX with an offset of +8 would be written:

LD E, (IX + 8)

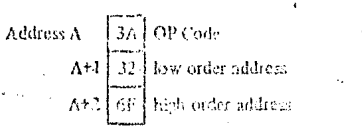
The instruction sequence for this in memory would be:



The two extended addressing instructions are also three byte instructions. For example the instruction to load the accumulator with the operand in memory location 6F32H, would be written:

LD A, (6F32H)

and its instruction sequence would be:

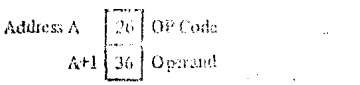


Notice that the low order portion of the address is always the first operand.

The load immediate instructions for the general purpose 8-bit registers are two-byte instructions. The instruction load register H with the value 36H would be written:

LD H, 36H

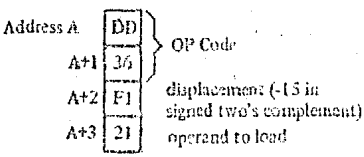
and its sequence would be:



Loading a memory location using indexed addressing for the destination and immediate addressing for the source requires four bytes. For example:

LD (IX + 15), 21H

would appear as:



Notice that with any indexed addressing the displacement always follows directly after the OP code.

Table 5.3-2 specifies the 16-bit load operations. This table is very similar to the previous one. Notice that the extended addressing capability covers all register pairs. Also notice that register indirect operations specifying the stack pointer are the PUSH and POP instructions. The mnemonic for these instructions is "PUSH" and "POP." These differ from other 16-bit loads in that the stack pointer is automatically decremented and incremented as each byte is pushed onto or popped from the stack respectively. For example the instruction:

PUSH AF

is a single byte instruction with the OP code of F5H. When this instruction is executed the following sequence is generated:

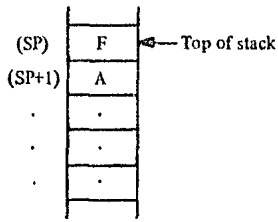
Decrement SP

LD (SP), A

Decrement SP

LD (SP), F

Thus the external stack now appears as follows:



DESTINATION	SOURCE																		
		IMPLIED		REGISTER							REG INDIRECT			INDEXED		EXT ADDR	IMMED
		I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IY+d)	(nn)	n
REGISTER	A	ED 57	LD 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n n	3E n
	B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
	C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
	D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
	E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
	H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
	L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
REG INDIRECT	(HL)			77	70	71	72	73	74	75							36 n
	(BC)			02													
	(DE)			12													
INDEXED	(IX+d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n
	(IY+d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n
EXT. ADDR	(nn)			32 n n													
IMPLIED	I			ED 47													
	R			ED 4F													

8 BIT LOAD GROUP
'LD'
TABLE 5.3–1

The POP instruction is the exact reverse of a PUSH. Notice that all PUSH and POP instructions utilize a 16-bit operand and the high order byte is always pushed first and popped last. That is a:

PUSH BC is PUSH B then C
PUSH DE is PUSH D then E
PUSH HL is PUSH H then L
POP HL is POP L then H

The instruction using extended immediate addressing for the source obviously requires 2 bytes of data following the OP code. For example:

LD DE, 0659H

will be:

Address A	11	OP Code
A+1	59	Low order operand to register E
A+2	06	High order operand to register D

In all extended immediate or extended addressing modes, the low order byte always appears first after the OP code.

Table 5.3-3 lists the 16-bit exchange instructions implemented in the Z-80. OP code 08H allows the programmer to switch between the two pairs of accumulator flag registers while D9H allows the programmer to switch between the duplicate set of six general purpose registers. These OP codes are only one byte in length to absolutely minimize the time necessary to perform the exchange so that the duplicate banks can be used to effect very fast interrupt response times.

BLOCK TRANSFER AND SEARCH

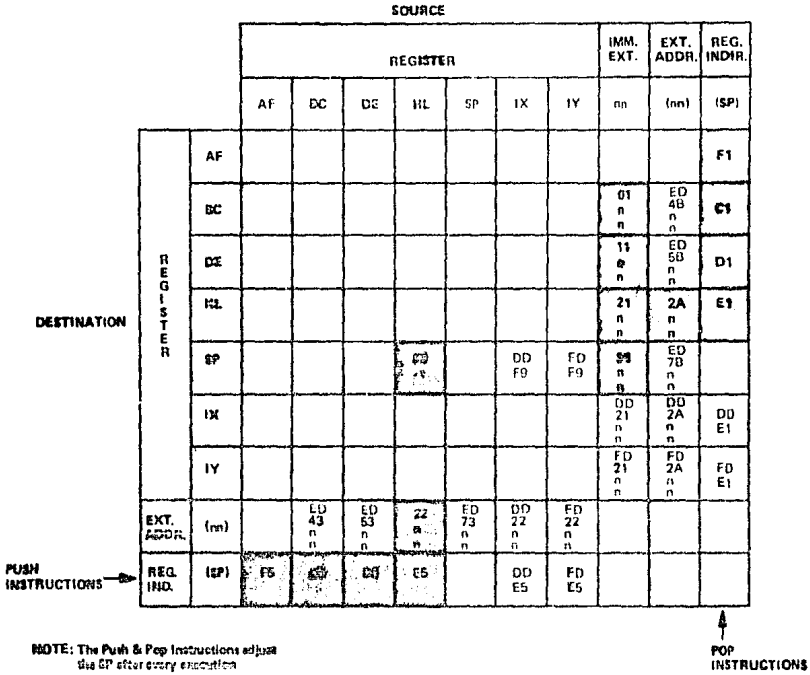
Table 5.3-4 lists the extremely powerful block transfer instructions. All of these instructions operate with three registers.

HL points to the source location.
DE points to the destination location.
BC is a byte counter.

After the programmer has initialized these three registers, any of these four instructions may be used. The LDI (Load and Increment) instruction moves one byte from the location pointed to by HL to the location pointed to by DE. Register pairs HL and DE are then automatically incremented and are ready to point to the following locations. The byte counter (register pair BC) is also decremented at this time. This instruction is valuable when blocks of data must be moved but other types of processing are required between each move. The LDIR (Load, increment and repeat) instruction is an extension of the LDI instruction. The same load and increment operation is repeated until the byte counter reaches the count of zero. Thus, this single instruction can move any block of data from one location to any other.

Note that since 16-bit registers are used, the size of the block can be up to 64K bytes (1K = 1024) long and it can be moved from any location in memory to any other location. Furthermore the blocks can be overlapping since there are absolutely no constraints on the data that is used in the three register pairs.

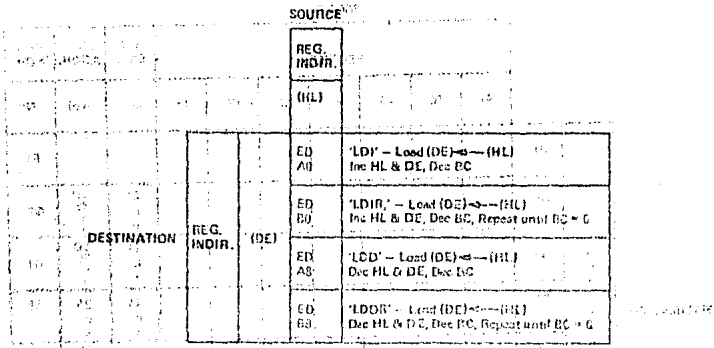
The LDD and LDDR instructions are very similar to the LDI and LDIR. The only difference is that register pairs HL and DE are decremented after every move so that a block transfer starts from the highest address of the designated block rather than the lowest.



16 BIT LOAD GROUP
'LD'
'PUSH' AND 'POP'
TABLE 5.3–2

		IMPLIED ADDRESSING				
		AF	BC, DE & HL	HL	IX	IY
IMPLIED	AF	08				
	BC, DE & HL		D9			
	DE			EB		
REG. INDIR.	(SP)			E3	DD E3	FD E3

EXCHANGES
'EX' AND 'EXX'
TABLE 5.3–3



BLOCK TRANSFER GROUP

TABLE 5.3-4

Table 5.3-5 specifies the OP codes for the four block search instructions. The first, CPI (compare and increment) compares the data in the accumulator with the contents of the memory location pointed to by register HL. The result of the compare is stored in one of the flag bits (see section 6.0 for a detailed explanation of the flag operations) and the HL register pair is then incremented and the byte counter (register pair BC) is decremented.

The instruction CPIR is merely an extension of the CPI instruction in which the compare is repeated until either a match is found or the byte counter (register pair BC) becomes zero. Thus, this single instruction can search the entire memory for any 8-bit character.

The CPD (Compare and Decrement) and CPDR (Compare, Decrement and Repeat) are similar instructions, their only difference being that they decrement HL after every compare so that they search the memory in the opposite direction. (The search is started at the highest location in the memory block).

It should be emphasized again that these block transfer and compare instructions are extremely powerful in string manipulation applications.

ARITHMETIC AND LOGICAL

Table 5.3-6 lists all of the 8-bit arithmetic operations that can be performed with the accumulator, also listed are the increment (INC) and decrement (DEC) instructions. In all of these instructions, except INC and DEC, the specified 8-bit operation is performed between the data in the accumulator and the source data specified in the table. The result of the operation is placed in the accumulator with the exception of compare (CP) that leaves the accumulator unaffected. All of these operations affect the flag register as a result of the specified operation. (Section 6.0 provides all of the details on how the flags are affected by any instruction type). INC and DEC instructions specify a register or a memory location as both source and destination of the result. When the source operand is addressed using the index registers the displacement must follow directly. With immediate addressing the actual operand will follow directly. For example the instruction:

AND 07H

would appear as:

Address A	E6	OP Code
A+1	07	Operand

SEARCH LOCATION

REG. INDIR. (HL)	
ED A1	'CPI' Inc HL, Dec BC
ED B1	'CPIR', Inc HL, Dec BC repeat until BC = 0 or find match
ED A9	'CPD' Dec HL & BC
ED B9	'CPDR' Dec HL & BC Repeat until BC = 0 or find match

HL points to location in memory to be compared with accumulator contents
BC is byte counter

BLOCK SEARCH GROUP
TABLE 5.3-5

Assuming that the accumulator contained the value F3H the result of 03H would be placed in the accumulator:

Acc before operation	1111 0011 = F3H
Operand	0000 0111 = 07H
Result to Acc	0000 0011 = 03H

The Add instruction (ADD) performs a binary add between the data in the source location and the data in the accumulator. The subtract (SUB) does a binary subtraction. When the add with carry is specified (ADC) or the subtract with carry (SBC), then the carry flag is also added or subtracted respectively. The flags and decimal adjust instruction (DAA) in the Z-80 (fully described in section 6.0) allow arithmetic operations for:

multiprecision packed BCD numbers

multiprecision signed or unsigned binary numbers

multiprecision two's complement signed numbers

Other instructions in this group are logical and (AND), logical or (OR), exclusive or (XOR) and compare (CP).

There are five general purpose arithmetic instructions that operate on the accumulator or carry flag. These five are listed in table 5.3-7. The decimal adjust instruction can adjust for subtraction as well as addition, thus making BCD arithmetic operations simple. Note that to allow for this operation the flag N is used. This flag is set if the last arithmetic operation was a subtract. The negate accumulator (NEG) instruction forms the two's complement of the number in the accumulator. Finally notice that a reset carry instruction is not included in the Z-80 since this operation can be easily achieved through other instructions such as a logical AND of the accumulator with itself.

Table 5.3-8 lists all of the 16-bit arithmetic operations between 16-bit registers. There are five groups of instructions including add with carry and subtract with carry. ADC and SBC affect all of the flags. These two groups simplify address calculation operations or other 16-bit arithmetic operations.

SOURCE

	REGISTER ADDRESSING							REG. INDIR.	INDEXED		IMMED.
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
'ADD'	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADD w CARRY 'ADC'	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUBTRACT 'SUB'	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SUB w CARRY 'SBC'	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
'AND'	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
'XOR'	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
'OR'	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
COMPARE 'CP'	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INCREMENT 'INC'	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DECREMENT 'DEC'	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

8 BIT ARITHMETIC AND LOGIC
TABLE 5.3-6

Decimal Adjust Acc, 'DAA'	27
Complement Acc, 'CPL'	2F
Negate Acc, 'NEG' (2's complement)	ED 44
Complement Carry Flag, 'CCF'	3F
Set Carry Flag, 'SCF'	37

GENERAL PURPOSE AF OPERATIONS
TABLE 5.3-7

SOURCE

	REGISTER ADDRESSING							REG. INDIR.	INDEXED		IMMED
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
'ADD'	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADD w CARRY 'ADC'	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUBTRACT 'SUB'	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SUB w CARRY 'SBC'	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
'AND'	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
'XOR'	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
'OR'	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
COMPARE 'CP'	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INCREMENT 'INC'	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DECREMENT 'DEC'	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

8 BIT ARITHMETIC AND LOGIC
TABLE 5.3-6

Decimal Adjust Acc, 'DAA'	27
Complement Acc, 'CPL'	2F
Negate Acc, 'NEG' (2's complement)	ED 44
Complement Carry Flag, 'CCF'	3F
Set Carry Flag, 'SCF'	37

GENERAL PURPOSE AF OPERATIONS
TABLE 5.3-7

DESTINATION			SOURCE BC	DE	HL	SP	IX	IY
	'ADD'	HL	09	19	29	39		
		IX	DD 09	DD 19		DD 39	DD 29	
		IY	FD 09	FD 19		FD 39		FD 29
	ADD WITH CARRY AND SET FLAGS 'ADC'	HL	ED 4A	ED 5A	ED 6A	ED 7A		
	SUB WITH CARRY AND SET FLAGS 'SBC'	HL	ED 42	ED 52	ED 62	ED 72		
	INCREMENT 'INC'		03	13	23	33	DD 23	FD 23
	DECREMENT 'DEC'		0B	1B	2B	3B	DD 2B	FD 2B

16 BIT ARITHMETIC
TABLE 5.3–8

ROTATE AND SHIFT

A major capability of the Z-80 is its ability to rotate or shift data in the accumulator, any general purpose register, or any memory location. All of the rotate and shift OP codes are shown in table 5.3-9. Also included in the Z-80 are arithmetic and logical shift operations. These operations are useful in an extremely wide range of applications including integer multiplication and division. Two BCD digit rotate instructions (RRD and RLD) allow a digit in the accumulator to be rotated with the two digits in a memory location pointed to by register pair HL. (See figure 5.3-9). These instructions allow for efficient BCD arithmetic.

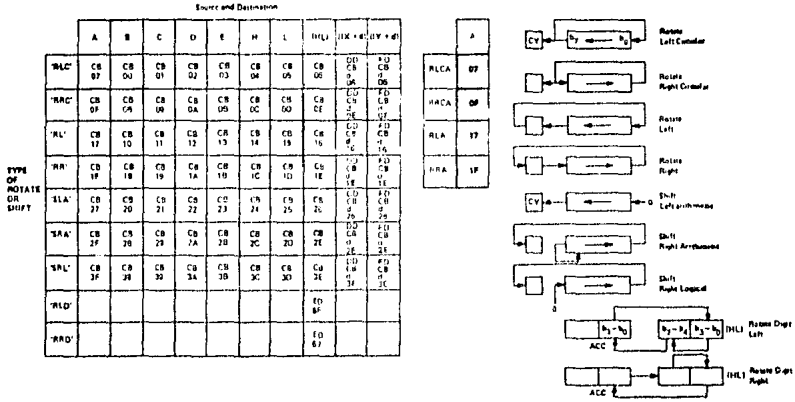
BIT MANIPULATION

The ability to set, reset and test individual bits in a register or memory location is needed in almost every program. These bits may be flags in a general purpose software routine, indications of external control conditions or data packed into memory locations to make memory utilization more efficient.

The Z-80 has the ability to set, reset or test any bit in the accumulator, any general purpose register or any memory location with a single instruction. Table 5.3-10 lists the 240 instructions that are available for this purpose. Register addressing can specify the accumulator or any general purpose register on which the operation is to be performed. Register indirect and indexed addressing are available to operate on external memory locations. Bit test operations set the zero flag (Z) if the tested bit is a zero. (Refer to section 6.0 for further explanation of flag operation).

JUMP, CALL AND RETURN

Figure 5.3-11 lists all of the jump, call and return instructions implemented in the Z-80 CPU. A jump is a branch in a program where the program counter is loaded with the 16-bit value as specified by one of the three available addressing modes (Immediate Extended, Relative or Register Indirect). Notice that the jump group has several different conditions that can be specified to be met before the jump will be made. If these conditions are not met, the program merely continues with the next sequential instruction. The conditions are all dependent on the data in the flag register. (Refer to section 6.0 for details on the flag register). The immediate extended addressing is used to jump to any location in the memory. This instruction requires three bytes (two to specify the 16-bit address) with the low order address byte first followed by the high order address byte.



ROTATES AND SHIFTS
TABLE 5.3-9

For example an unconditional Jump to memory location 3E32H would be:

Address A	C3	OP Code
A+1	32	Low order address
A+2	3E	High order address

The relative jump instruction uses only two bytes, the second byte is a signed two's complement displacement from the existing PC. This displacement can be in the range of +129 to -126 and is measured from the address of the instruction OP code.

Three types of register indirect jumps are also included. These instructions are implemented by loading the register pair HL or one of the index registers IX or IY directly into the PC. This capability allows for program jumps to be a function of previous calculations.

A call is a special form of a jump where the address of the byte following the call instruction is pushed onto the stack before the jump is made. A return instruction is the reverse of a call because the data on the top of the stack is popped directly into the PC to form a jump address. The call and return instructions allow for simple subroutine and interrupt handling. Two special return instructions have been included in the Z-80 family of components. The return from interrupt instruction (RETI) and the return from non maskable interrupt (RETN) are treated in the CPU as an unconditional return identical to the OP code C9H. The difference is that (RETI) can be used at the end of an interrupt routine and all Z-80 peripheral chips will recognize the execution of this instruction for proper control of nested priority interrupt handling. This instruction coupled with the Z-80 peripheral devices implementation simplifies the normal return from nested interrupt. Without this feature the following software sequence would be necessary to inform the interrupting device that the interrupt routine is completed:

		REGISTER ADDRESSING							REG. INDIR.	INDEXED	
	BIT	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
TEST 'BIT'	0	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 46	CB 45	DD CB d 46	FD CB d 46
	1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
	2	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
	3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
	4	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
	5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
	6	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
	7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
RESET BIT 'RES'	0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
	1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
	2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
	4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
	6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
SET BIT 'SET'	0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
	2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
	4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
	6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

BIT MANIPULATION GROUP
TABLE 5.3–10

Disable Interrupt	-- prevent interrupt before routine is exited.
LD A, n OUT n, A	-- notify peripheral that service routine is complete
Enable Interrupt	
Return	

This seven byte sequence can be replaced with the one byte EI instruction and the two byte RETI instruction in the Z80. This is important since interrupt service time often must be minimized.

To facilitate program loop control the instruction DJNZ e can be used advantageously. This two byte, relative jump instruction decrements the B register and the jump occurs if the B register has not been decremented to zero. The relative displacement is expressed as a signed two's complement number. A simple example of its use might be:

Address	Instruction	Comments
N, N + 1	LD B, 7	; set B register to count of 7
N + 2 to N + 9	(Perform a sequence of instructions)	; loop to be performed 7 times
N + 10, N + 11	DJNZ -8	; to jump from N + 12 to N + 2
N + 12	(Next Instruction)	

			CONDITION									
			UN-COND	CARRY	NON CARRY	ZERO	NON ZERO	PARITY EVEN	PARITY ODD	SIGN NEG	SIGN POS	REG B≠0
JUMP 'JP'	IMMED. EXT.	nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n	
JUMP 'JR'	RELATIVE	PC+e	18 e-2	38 e-2	30 e-2	28 e-2	20 e-2					
JUMP 'JP'	REG. INDIR.	(HL)	E9									
JUMP 'JP'		(IX)	DD E9									
JUMP 'JP'		(IY)	FD E9									
'CALL'	IMMED. EXT.	nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
DECREMENT B, JUMP IF NON ZERO 'DJNZ'	RELATIVE	PC+e										10 e-2
RETURN 'RET'	REGISTER INDIR.	(SP) (SP+1)	C9	D8	D0	C8	C0	E8	E0	F8	F0	
RETURN FROM INT 'RETI'	REG. INDIR.	(SP) (SP+1)	ED 4D									
RETURN FROM NON MASKABLE INT 'RETN'	REG. INDIR.	(SP) (SP+1)	ED 45									

NOTE—CERTAIN FLAGS HAVE MORE THAN ONE PURPOSE. REFER TO SECTION 6.0 FOR DETAILS

JUMP, CALL and RETURN GROUP
TABLE 5.3–11

Table 5.3-12 lists the eight OP codes for the restart instruction. This instruction is a single byte call to any of the eight addresses listed. The simple mnemonic for these eight calls is also shown. The value of this instruction is that frequently used routines can be called with this instruction to minimize memory usage.

		OP CODE	
CALL ADDRESS	0000_H	C7	'RST 0'
	0008_H	CF	'RST 8'
	0010_H	D7	'RST 16'
	0018_H	DF	'RST 24'
	0020_H	E7	'RST 32'
	0028_H	EF	'RST 40'
	0030_H	F7	'RST 48'
	0038_H	FF	'RST 56'

RESTART GROUP
TABLE 5.3–12

INPUT/OUTPUT

The Z-80 has an extensive set of Input and Output instructions as shown in table 5.3-13 and table 5.3-14. The addressing of the input or output device can be either absolute or register indirect, using the C register. Notice that in the register indirect addressing mode data can be transferred between the I/O devices and any of the internal registers. In addition eight block transfer instructions have been implemented. These instructions are similar to the memory block transfers except that they use register pair HL for a pointer to the memory source (output commands) or destination (input commands) while register B is used as a byte counter. Register C holds the address of the port for which the input or output command is desired. Since register B is eight bits in length, the I/O block transfer command handles up to 256 bytes.

In the instructions IN A, n and OUT n, A the I/O device address n appears in the lower half of the address bus (A_0-A_7) while the accumulator content is transferred in the upper half of the address bus. In all register indirect input output instructions, including block I/O transfers the content of register C is transferred to the lower half of the address bus (device address) while the content of register B is transferred to the upper half of the address bus.

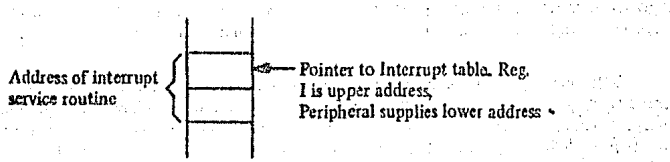
INPUT DESTINATION			SOURCE PORT ADDRESS: NAMED (n)	SOURCE PORT ADDRESS: REG. INDIR. (C)
INPUT 'IN'	REG ADDRESS	A	DB n	ED 78
		B		ED 40
		C		ED 48
		D		ED 50
		E		ED 58
		H		ED 60
		L		ED 68
'INI' – INPUT & Inc HL, Dec B	REG. INDIR	(HL)		ED A2
'INIR' – INP, Inc HL, Dec B, REPEAT IF B≠0				ED B2
'IND' – INPUT & Dec HL, Dec B				ED AA
'INDR' – INPUT, Dec HL, Dec B, REPEAT IF B≠0				ED BA

BLOCK INPUT COMMANDS

INPUT GROUP
TABLE 5.3-13

CPU CONTROL GROUP

The final table, table 5.3-15 illustrates the six general purpose CPU control instructions. The NOP is a do-nothing instruction. The HALT instruction suspends CPU operation until a subsequent interrupt is received, while the DI and EI are used to lock out and enable interrupts. The three interrupt mode commands set the CPU into any of the three available interrupt response modes as follows. If mode zero is set the interrupting device can insert any instruction on the data bus and allow the CPU to execute it. Mode 1 is a simplified mode where the CPU automatically executes a restart (RST) to location 0038H so that no external hardware is required. (The old PC content is pushed onto the stack). Mode 2 is the most powerful in that it allows for an indirect call to any location in memory. With this mode the CPU forms a 16-bit memory address where the upper 8-bits are the content of register I and the lower 8-bits are supplied by the interrupting device. This address points to the first of two sequential bytes in a table where the address of the service routine is located. The CPU automatically obtains the starting address and performs a CALL to this address.



			SOURCE							
			REGISTER							REG. IND.
			A	B	C	D	E	H	L	(HL)
'OUT'	IMMED.	(n)	D3 n							
	REG. IND.	(C)	ED 79	ED 41	ED 49	ED 51	ED 59	ED 61	ED 69	
'OUTI' – OUTPUT Inc HL, Dec b	REG. IND.	(C)								ED A3
'OTIR' – OUTPUT, Inc HL, Dec B, REPEAT IF B≠0	REG. IND.	(C)								ED B3
'OUTD' – OUTPUT Dec HL & B	REG. IND.	(C)								ED AB
'OTDR' – OUTPUT, Dec HL & B, REPEAT IF B≠0	REG. IND.	(C)								ED BB

BLOCK OUTPUT COMMANDS

PORT DESTINATION ADDRESS

OUTPUT GROUP
TABLE 5.3–14

'NOP'	00
'HALT'	76
DISABLE INT '(DI)'	F3
ENABLE INT '(EI)'	FB
SET INT MODE 0 'IM0'	ED 46
SET INT MODE 1 'IM1'	ED 56
SET INT MODE 2 'IM2'	ED 5E

8080A MODE

CALL TO LOCATION 0038_H

INDIRECT CALL USING REGISTER I AND 8 BITS FROM INTERRUPTING DEVICE AS A POINTER.

MISCELLANEOUS CPU CONTROL
TABLE 5.3–15

-BLANK-

6.0 FLAGS

Each of the two Z-80 CPU Flag registers contains six bits of information which are set or reset by various CPU operations. Four of these bits are testable; that is, they are used as conditions for jump, call or return instructions. For example a jump may be desired only if a specific bit in the flag register is set. The four testable flag bits are:

1) Carry Flag (C) – This flag is the carry from the highest order bit of the accumulator. For example, the carry flag will be set during an add instruction where a carry from the highest bit of the accumulator is generated. This flag is also set if a borrow is generated during a subtraction instruction. The shift and rotate instructions also affect this bit.

2) Zero Flag (Z) – This flag is set if the result of the operation loaded a zero into the accumulator. Otherwise it is reset.

3) Sign Flag (S) – This flag is intended to be used with signed numbers and it is set if the result of the operation was negative. Since bit 7 (MSB) represents the sign of the number (A negative number has a 1 in bit 7), this flag stores the state of bit 7 in the accumulator.

4) Parity/Overflow Flag (P/V) – This dual purpose flag indicates the parity of the result in the accumulator when logical operations are performed (such as AND A, B) and it represents overflow when signed two's complement arithmetic operations are performed. The Z-80 overflow flag indicates that the two's complement number in the accumulator is in error since it has exceeded the maximum possible (+127) or is less than the minimum possible (-128) number than can be represented in two's complement notation. For example consider adding:

```
+120 =    0111 1000
+105 =    0110 1001
-----------------------
  C  =  0  1110 0001  = -95 (wrong) Overflow has occured
```

Here the result is incorrect. Overflow has occurred and yet there is no carry to indicate an error. For this case the overflow flag would be set. Also consider the addition of two negative numbers:

```
 -5 =    1111 1011
-16 =    1111 0000
-----------------------
  C  =  1  1110 1011  = -21 correct
```

Notice that the answer is correct but the carry is set so that this flag can not be used as an overflow indicator. In this case the overflow would not be set.

For logical operations (AND, OR, XOR) this flag is set if the parity of the result is even and it is reset if it is odd.

There are also two non-testable bits in the flag register. Both of these are used for BCD arithmetic. They are:

1) Half carry (H) – This is the BCD carry or borrow result from the least significant four bits of operation. When using the DAA (Decimal Adjust Instruction) this flag is used to correct the result of a previous packed decimal add or subtract.

2) Subtract Flag (N) – Since the algorithm for correcting BCD operations is different for addition or subtraction, this flag is used to specify what type of instruction was executed last so that the DAA operation will be correct for either addition or subtraction.

The Flag register can be accessed by the programmer and its format is as follows:

S	Z	X	H	X	P/V	N	C

X means flag is indeterminate.

Table 6.0-1 lists how each flag bit is affected by various CPU instructions. In this table a '•' indicates that the instruction does not change the flag, an 'X' means that the flag goes to an indeterminate state, a '0' means that it is reset, a '1' means that it is set and the symbol '↕' indicates that it is set or reset according to the previous discussion. Note that any instruction not appearing in this table does not affect any of the flags.

Table 6.0-1 includes a few special cases that must be described for clarity. Notice that the block search instruction sets the Z flag if the last compare operation indicated a match between the source and the accumulator data. Also, the parity flag is set if the byte counter (register pair BC) is not equal to zero. This same use of the parity flag is made with the block move instructions. Another special case is during block input or output instructions, here the Z flag is used to indicate the state of register B which is used as a byte counter. Notice that when the I/O block transfer is complete, the zero flag will be reset to a zero (i.e. B=0) while in the case of a block move command the parity flag is reset when the operation is complete. A final case is when the refresh or I register is loaded into the accumulator, the interrupt enable flip flop is loaded into the parity flag so that the complete state of the CPU can be saved at any time.

Instruction	C	Z	P/V	S	N	H	Comments
ADD A, s; ADC A,s	↕	↕	V	↕	0	↕	8-bit add or add with carry
SUB s; SBC A, s; CP s; NEG	↕	↕	V	↕	1	↕	8-bit subtract, subtract with carry, compare and negate accumulator
AND s	0	↕	P	↕	0	1	Logical operations
OR s; XOR s	0	↕	P	↕	0	0	And set's different flags
INC s	•	↕	V	↕	0	↕	8-bit increment
DEC m	•	↕	V	↕	1	↕	8-bit decrement
ADD DD, ss	↕	•	•	•	0	X	16-bit add
ADC HL, ss	↕	↕	V	↕	0	X	16-bit add with carry
SBC HL, ss	↕	↕	V	↕	1	X	16-bit subtract with carry
RLA; RLCA, RRA, RRCA	↕	•	•	•	0	0	Rotate accumulator
RL m; RLC m; RR m; RRC m SLA m; SRA m; SRL m	↕	↕	P	↕	0	0	Rotate and shift location m
RLD, RRD	•	↕	P	↕	0	0	Rotate digit left and right
DAA	↕	↕	P	↕	•	↕	Decimal adjust accumulator
CPL	•	•	•	•	1	1	Complement accumulator
SCF	1	•	•	•	0	0	Set carry
CCF	↕	•	•	•	0	X	Complement carry
IN r, (C)	•	↕	P	↕	0	0	Input register indirect
INI; IND; OUTI; OUTD	•	↕	X	X	1	X	Block input and output
INIR; INDR; OTIR; OTDR	•	1	X	X	1	X	Z = 0 if B ≠ 0 otherwise Z = 1
LDI, LDD	•	X	↕	X	0	0	Block transfer instructions
LDIR, LDDR	•	X	0	X	0	0	P/V = 1 if BC ≠ 0, otherwise P/V = 0
CPI, CPIR, CPD, CPDR	•	↕	↕	↕	1	X	Block search instructions Z = 1 if A = (HL), otherwise Z = 0 P/V = 1 if BC ≠ 0, otherwise P/V = 0
LD A, I; LD A, R	•	↕	IFF	↕	0	0	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag
BIT b, s	•	↕	X	X	0	1	The state of bit b of location s is copied into the Z flag
NEG	↕	↕	V	↕	1	↕	Negate accumulator

The following notation is used in this table:

Symbol	Operation
C	Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
Z	Zero flag. Z=1 if the result of the operation is zero.
S	Sign flag. S=1 if the MSB of the result is one.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
H	Half-carry flag. H=1 if the add or subtract operation produced a carry into or borrow from into bit 4 of the accumulator.
N	Add/Subtract flag. N=1 if the previous operation was a subtract.
	H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
↕	The flag is affected according to the result of the operation.
•	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is a "don't care."
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one of the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8-bit value in range <0, 255>
nn	16-bit value in range <0, 65535>
m	Any 8-bit location for all the addressing modes allowed for the particular instruction.

SUMMARY OF FLAG OPERATION
TABLE 6.0-1

-BLANK-

7.0 SUMMARY OF OP CODES AND EXECUTION TIMES

The following section gives a summary of the Z-80 instructions set. The instructions are logically arranged into groups as shown on tables 7.0-1 through 7.0-11. Each table shows the assembly language mnemonic OP code, the actual OP code, the symbolic operation, the content of the flag register following the execution of each instruction, the number of bytes required for each instruction as well as the number of memory cycles and the total number of T states (external clock periods) required for the fetching and execution of each instruction. Care has been taken to make each table self-explanatory without requiring any cross reference with the test or other tables.

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	OP-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T Cycles	Comments	
LD r, r'	r ← r'	•	•	•	•	•	•	01 r r'	1	1	4	r, r'	Reg.
LD r, n	r ← n	•	•	•	•	•	•	00 r 110	2	2	7	000	B
								← n →				001	C
LD r, (HL)	r ← (HL)	•	•	•	•	•	•	01 r 110	1	2	7	010	D
LD r, (IX+d)	r ← (IX+d)	•	•	•	•	•	•	11 011 101	3	5	19	011	E
								01 r 110				100	H
								← d →				101	L
LD r, (IY+d)	r ← (IY+d)	•	•	•	•	•	•	11 111 101	3	5	19	111	A
								01 r 110					
								← d →					
LD (HL), r	(HL) ← r	•	•	•	•	•	•	01 110 r	1	2	7		
LD (IX+d), r	(IX+d) ← r	•	•	•	•	•	•	11 011 101	3	5	19		
								01 110 r					
								← d →					
LD (IY+d), r	(IY+d) ← r	•	•	•	•	•	•	11 111 101	3	5	19		
								01 110 r					
								← d →					
LD (HL), n	(HL) ← n	•	•	•	•	•	•	00 110 110	2	3	10		
								← n →					
LD (IX+d), n	(IX+d) ← n	•	•	•	•	•	•	11 011 101	4	5	19		
								00 110 110					
								← d →					
								← n →					
LD (IY+d), n	(IY+d) ← n	•	•	•	•	•	•	11 111 101	4	5	19		
								00 110 110					
								← d →					
								← n →					
LD A, (BC)	A ← (BC)	•	•	•	•	•	•	00 001 010	1	2	7		
LD A, (DE)	A ← (DE)	•	•	•	•	•	•	00 011 010	1	2	7		
LD A, (nn)	A ← (nn)	•	•	•	•	•	•	00 111 010	3	4	13		
								← n →					
								← n →					
LD (BC), A	(BC) ← A	•	•	•	•	•	•	00 000 010	1	2	7		
LD (DE), A	(DE) ← A	•	•	•	•	•	•	00 010 010	1	2	7		
LD (nn), A	(nn) ← A	•	•	•	•	•	•	00 110 010	3	4	13		
								← n →					
								← n →					
LD A, I	A ← I	•	‡	IFF	‡	0	0	11 101 101	2	2	9		
								01 010 111					
LD A, R	A ← R	•	‡	IFF	‡	0	0	11 101 101	2	2	9		
								01 011 111					
LD I, A	I ← A	•	•	•	•	•	•	11 101 101	2	2	9		
								01 000 111					
LD R, A	R ← A	•	•	•	•	•	•	11 101 101	2	2	9		
								01 001 111					

Notes: r, r' means any of the registers A, B, C, D, E, H, L

IFF the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,

‡ = flag is affected according to the result of the operation.

8-BIT LOAD GROUP
TABLE 7.0-1

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	•	•	•	•	00 dd0 001 ← n → ← n →	3	3	10	dd Pair 00 BC 01 DE
LD IX, nn	IX ← nn	•	•	•	•	•	•	11 011 101 00 100 001 ← n → ← n →	4	4	14	10 HL 11 SP
LD IY, nn	IY ← nn	•	•	•	•	•	•	11 111 101 00 100 001 ← n → ← n →	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	•	•	•	•	00 101 010 ← n → ← n →	3	5	16	
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	•	•	•	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	4	6	20	
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	•	•	•	•	•	•	11 011 101 00 101 010 ← n → ← n →	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	•	•	•	•	•	•	11 111 101 00 101 010 ← n → ← n →	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	•	•	•	•	00 100 010 ← n → ← n →	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	•	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	4	6	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	•	•	•	•	11 011 101 00 100 010 ← n → ← n →	4	6	20	
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	•	•	•	•	11 111 101 00 100 010 ← n → ← n →	4	6	20	
LD SP, HL	SP ← HL	•	•	•	•	•	•	11 111 001	1	1	6	
LD SP, IX	SP ← IX	•	•	•	•	•	•	11 011 101 11 111 001	2	2	10	
LD SP, IY	SP ← IY	•	•	•	•	•	•	11 111 101 11 111 001	2	2	10	
PUSH qq	(SP-2) ← qq_L (SP-1) ← qq_H	•	•	•	•	•	•	11 qq0 101	1	3	11	qq Pair 00 BC 01 DE
PUSH IX	(SP-2) ← IX_L (SP-1) ← IX_H	•	•	•	•	•	•	11 011 101 11 100 101	2	4	15	10 HL 11 AF
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H	•	•	•	•	•	•	11 111 101 11 100 101	2	4	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP)	•	•	•	•	•	•	11 qq0 001	1	3	10	
POP IX	IX_H ← (SP+1) IX_L ← (SP)	•	•	•	•	•	•	11 011 101 11 100 001	2	4	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP)	•	•	•	•	•	•	11 111 101 11 100 001	2	4	14	

Notes: dd is any of the register pairs BC, DE, HL, SP
qq is any of the register pairs AF, BC, DE, HL
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively.
E.g. BC_L = C, AF_H = A

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ flag is affected according to the result of the operation.

16-BIT LOAD GROUP
TABLE 7.0-2

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	DE ↔ HL	•	•	•	•	•	•	11 101 011	1	1	4	
EX AF, AF'	AF ↔ AF'	•	•	•	•	•	•	00 001 000	1	1	4	
EXX	(BC DE HL) ↔ (BC' DE' HL')	•	•	•	•	•	•	11 011 001	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	•	•	•	•	11 100 011	1	5	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	•	•	•	•	•	•	11 011 101 11 100 011	2	6	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	•	•	•	•	•	•	11 111 101 11 100 011	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	① 1	•	0	0	11 101 101 10 100 000	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC = 0	•	•	① 0	•	0	0	11 101 101 10 110 000	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	① 1	•	0	0	11 101 101 10 101 000	2	4	16	
LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC = 0	•	•	① 0	•	0	0	11 101 101 10 111 000	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A - (HL) HL ← HL+1 BC ← BC-1	•	② 1	① 1	1	1	1	11 101 101 10 100 001	2	4	16	
CPIR	A - (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	•	② 1	① 1	1	1	1	11 101 101 10 110 001	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A - (HL) HL ← HL-1 BC ← BC-1	•	② 1	① 1	1	1	1	11 101 101 10 101 001	2	4	16	
CPDR	A - (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	•	② 1	① 1	1	1	1	11 101 101 10 111 001	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)

Notes: ① P/V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1
② Z flag is 1 if A = (HL), otherwise Z = 0.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

TABLE 7.0-3

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A, r	A ← A + r	↕	↕	V	↕	0	↕	10 [000] r	1	1	4	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, n	A ← A + n	↕	↕	V	↕	0	↕	11 [000] 110 ← n →	2	2	7	
ADD A, (HL)	A ← A + (HL)	↕	↕	V	↕	0	↕	10 [000] 110	1	2	7	
ADD A, (IX+d)	A ← A + (IX+d)	↕	↕	V	↕	0	↕	11 011 101 10 [000] 110 ← d →	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	↕	↕	V	↕	0	↕	11 111 101 10 [000] 110 ← d →	3	5	19	
ADC A, s	A ← A + s + CY	↕	↕	V	↕	0	↕	[001]				s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction
SUB s	A ← A − s	↕	↕	V	↕	1	↕	[010]				
SBC A, s	A ← A − s − CY	↕	↕	V	↕	1	↕	[011]				
AND s	A ← A ∧ s	0	↕	P	↕	0	1	[100]				
OR s	A ← A ∨ s	0	↕	P	↕	0	0	[110]				The indicated bits replace the 000 in the ADD set above.
XOR s	A ← A ⊕ s	0	↕	P	↕	0	0	[101]				
CP s	A − s	↕	↕	V	↕	1	↕	[111]				
INC r	r ← r + 1	•	↕	V	↕	0	↕	00 r [100]	1	1	4	
INC (HL)	(HL) ← (HL)+1	•	↕	V	↕	0	↕	00 110 [100]	1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	•	↕	V	↕	0	↕	11 011 101 00 110 [100] ← d →	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d) + 1	•	↕	V	↕	0	↕	11 111 101 00 110 [100] ← d →	3	6	23	
DEC m	m ← m−1	•	↕	V	↕	1	↕	[101]				m is any of r, (HL), (IX+d), (IY+d) as shown for INC. Same format and states as INC. Replace 100 with 101 in OP code.

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity, V = 1 means overflow, V = 0 means not overflow, P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

8-BIT ARITHMETIC AND LOGICAL GROUP
TABLE 7.0-4

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76 543 210				
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands	‡	‡	P	‡	•	‡	00 100 111	1	1	4	Decimal adjust accumulator
CPL	$A \leftarrow \bar{A}$	•	•	•	•	1	1	00 101 111	1	1	4	Complement accumulator (one's complement)
NEG	$A \leftarrow 0 - A$	‡	‡	V	‡	1	‡	11 101 101 01 000 100	2	2	8	Negate acc. (two's complement)
CCF	$CY \leftarrow \overline{CY}$	‡	•	•	•	0	X	00 111 111	1	1	4	Complement carry flag
SCF	CY ← 1	1	•	•	•	0	0	00 110 111	1	1	4	Set carry flag
NOP	No operation	•	•	•	•	•	•	00 000 000	1	1	4	
HALT	CPU halted	•	•	•	•	•	•	01 110 110	1	1	4	
DI	IFF ← 0	•	•	•	•	•	•	11 110 011	1	1	4	
EI	IFF ← 1	•	•	•	•	•	•	11 111 011	1	1	4	
IM 0	Set interrupt mode 0	•	•	•	•	•	•	11 101 101 01 000 110	2	2	8	
IM 1	Set interrupt mode 1	•	•	•	•	•	•	11 101 101 01 010 110	2	2	8	
IM2	Set interrupt mode 2	•	•	•	•	•	•	11 101 101 01 011 110	2	2	8	

Notes: IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ = flag is affected according to the result of the operation.

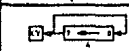
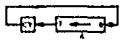
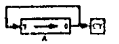
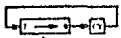
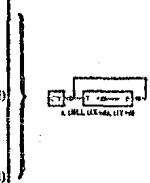
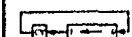
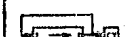
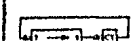
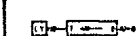
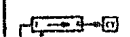
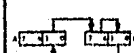
GENERAL PURPOSE ARITHMETIC AND CPU CONTROL GROUPS
TABLE 7.0-5

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments	
		C	Z	P/V	S	N	H	76 543 210					
ADD HL, ss	HL ← HL+ss	↕	•	•	•	0	X	00 ss1 001	1	3	11	ss	Reg.
												00	BC
ADC HL, ss	HL←HL+ss+CY	↕	↕	V	↕	0	X	11 101 101	2	4	15	01	DE
								01 ss1 010				10	HL
SBC HL, ss	HL←HL-ss-CY	↕	↕	V	↕	1	X	11 101 101	2	4	15	11	SP
								01 ss0 010					
ADD IX, pp	IX ← IX + pp	↕	•	•	•	0	X	11 011 101	2	4	15	pp	Reg.
								00 pp1 001				00	BC
												01	DE
												10	IX
												11	SP
ADD IY, rr	IY←IY+rr	↕	•	•	•	0	X	11 111 101	2	4	15	rr	Reg.
								00 rr1 001				00	BC
												01	DE
												10	IY
												11	SP
INC ss	ss ← ss + 1	•	•	•	•	•	•	00 ss0 011	1	1	6		
INC IX	IX ← IX + 1	•	•	•	•	•	•	11 011 101	2	2	10		
								00 100 011					
INC IY	IY ← IY + 1	•	•	•	•	•	•	11 111 101	2	2	10		
								00 100 011					
DEC ss	ss ← ss - 1	•	•	•	•	•	•	00 ss1 011	1	1	6		
DEC IX	IX ← IX - 1	•	•	•	•	•	•	11 011 101	2	2	10		
								00 101 011					
DEC IY	IY ← IY - 1	•	•	•	•	•	•	11 111 101	2	2	10		
								00 101 011					

Notes: ss is any of the register pairs BC, DE, HL, SP
pp is any of the register pairs BC, DE, IX, SP
rr is any of the register pairs BC, DE, IY, SP.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

16-BIT ARITHMETIC GROUP
TABLE 7.0-6

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		‡	•	•	•	0	0	00 000 111	1	1	4	Rotate left circular accumulator
RLA		‡	•	•	•	0	0	00 010 111	1	1	4	Rotate left accumulator
RRCA		‡	•	•	•	0	0	00 001 111	1	1	4	Rotate right circular accumulator
RRA		‡	•	•	•	0	0	00 011 111	1	1	4	Rotate right accumulator
RLC r		‡	‡	P	‡	0	0	11 001 011 00 000 r	2	2	8	Rotate left circular register r
RLC (HL)		‡	‡	P	‡	0	0	11 001 011 00 000 110	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)		‡	‡	P	‡	0	0	11 011 101 11 001 011 ← d → 00 000 110	4	6	23	
RLC (IY+d)		‡	‡	P	‡	0	0	11 111 101 11 001 011 ← d → 00 000 110	4	6	23	
RL m		‡	‡	P	‡	0	0	010				Instruction format and states are as shown for RLC,m. To form new OP-code replace 000 of RLC,m with shown code
RRC m		‡	‡	P	‡	0	0	001				
RR m		‡	‡	P	‡	0	0	011				
SLA m		‡	‡	P	‡	0	0	100				
SRA m		‡	‡	P	‡	0	0	101				
SRL m		‡	‡	P	‡	0	0	111				
RLD		•	‡	P	‡	0	0	11 101 101 01 101 111	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected
RRD		•	‡	P	‡	0	0	11 101 101 01 100 111	2	5	18	

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ‡ = flag is affected according to the result of the operation.

ROTATE AND SHIFT GROUP
TABLE 7.0-7

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b, r	$Z \leftarrow \overline{r_b}$	•	‡	X	X	0	1	11 001 011 01 b r	2	2	8	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	•	‡	X	X	0	1	11 001 011 01 b 110	2	3	12	b: 000 Bit 0, 001 1, 010 2, 011 3, 100 4, 101 5, 110 6, 111 7
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)_b}$	•	‡	X	X	0	1	11 011 101 11 001 011 ← d → 01 b 110	4	5	20	
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)_b}$	•	‡	X	X	0	1	11 111 101 11 001 011 ← d → 01 b 110	4	5	20	
SET b, r	$r_b \leftarrow 1$	•	•	•	•	•	•	11 001 011 [11] b r	2	2	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	11 001 011 [11] b 110	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	11 011 101 11 001 011 ← d → [11] b 110	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	11 111 101 11 001 011 ← d → [11] b 110	4	6	23	
RES b, m	$s_b \leftarrow 0$ m ≡ r, (HL), (IX+d), (IY+d)							[10]				To form new OP-code replace [11] of SET b,m with [10]. Flags and time states for SET instruction

Notes: The notation s_b indicates bit b (0 to 7) of location s.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ‡ = flag is affected according to the result of the operation.

BIT SET, RESET AND TEST GROUP
TABLE 7.0-8

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
JP nn	PC ← nn	•	•	•	•	•	•	11 000 011 ← n → ← n →	3	3	10	
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	•	•	•	•	11 cc 010 ← n → ← n →	3	3	10	cc: 000 NZ non zero; 001 Z zero; 010 NC non carry; 011 C carry; 100 PO parity odd; 101 PE parity even; 110 P sign positive; 111 M sign negative
JR e	PC ← PC + e	•	•	•	•	0	0	00 011 000 ← e-2 →	2	3	12	
JR C, e	If C = 0, continue	•	•	•	•	0	•	00 111 000 ← e-2 →	2	2	7	If condition not met
	If C = 1, PC ← PC+e								2	3	12	If condition is met
JR NC, e	If C = 1, continue	•	•	•	•	0	•	00 110 000 ← e-2 →	2	2	7	If condition not met
	If C = 0, PC ← PC + e								2	3	12	If condition is met
JR Z, e	If Z = 0 continue	•	•	•	•	0	•	00 101 000 ← e-2 →	2	2	7	If condition not met
	If Z = 1, PC ← PC + e								2	3	12	If condition is met
JR NZ, e	If Z = 1, continue	•	•	•	•	0	•	00 100 000 ← e-2 →	2	2	7	If condition not met
	If Z = 0, PC ← PC + e								2	3	12	If condition met
JP (HL)	PC ← HL	•	•	•	•	•	•	11 101 001	1	1	4	
JP (IX)	PC ← IX	•	•	•	•	•	•	11 011 101 11 101 001	2	2	8	
JP (IY)	PC ← IY	•	•	•	•	•	•	11 111 101 11 101 001	2	2	8	
DJNZ, e	B ← B-1 If B = 0, continue	•	•	•	•	•	•	00 010 000 ← e-2 →	2	2	8	If B = 0
	If B ≠ 0, PC ← PC + e								2	3	13	If B ≠ 0

Notes: e represents the extension in the relative addressing mode.

e is a signed two's complement number in the range <-126, 129>

e-2 in the op-code provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

JUMP GROUP
TABLE 7.0-9

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALL nn	$(SP-1)\leftarrow PC_H$ $(SP-2)\leftarrow PC_L$ $PC\leftarrow nn$	•	•	•	•	•	•	11 001 101 ← n → ← n →	3	5	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	•	•	•	•	11 cc 100 ← n → ← n →	3 3	3 5	10 17	If cc is false If cc is true
RET	$PC_L\leftarrow(SP)$ $PC_H\leftarrow(SP+1)$	•	•	•	•	•	•	11 001 001	1	3	10	
RET cc	If condition cc is false continue, otherwise same as RET	•	•	•	•	•	•	11 cc 000	1 1	1 3	5 11	If cc is false If cc is true
RETI	Return from interrupt	•	•	•	•	•	•	11 101 101 01 001 101	2	4	14	
RETN	Return from non maskable interrupt	•	•	•	•	•	•	11 101 101 01 000 101	2	4	14	
RST p	$(SP-1)\leftarrow PC_H$ $(SP-2)\leftarrow PC_L$ $PC_H\leftarrow 0$ $PC_L\leftarrow P$	•	•	•	•	•	•	11 t 111	1	3	11	

cc	Condition	
000	NZ	non zero
001	Z	zero
010	NC	non carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign positive
111	M	sign negative

t	P
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown
t = flag is affected according to the result of the operation.

CALL AND RETURN GROUP
TABLE 7.0-10

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	•	•	•	•	•	•	11 011 011 ← n →	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	•	↕	P	↕	0	?	11 101 101 01 r 000	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	(HL) ← (C) B ← B - 1 HL ← HL + 1	X	↕ ①	X	X	1	X	11 101 101 10 100 010	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	(HL) ← (C) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	1	X	11 101 101 10 110 010	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	(HL) ← (C) B ← B - 1 HL ← HL - 1	X	↕ ①	X	X	1	X	11 101 101 10 101 010	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	(HL) ← (C) B ← B - 1 HL ← HL -1 Repeat until B = 0	X	1	X	X	1	X	11 101 101 10 111 010	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT (n), A	(n) ← A	•	•	•	•	•	•	11 010 011 ← n →	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
OUT (C), r	(C) ← r	•	•	•	•	•	•	11 101 101 01 r 001	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	(C) ← (HL) B ← B - 1 HL ← HL + 1	X	↕ ①	X	X	1	X	11 101 101 10 100 011	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	(C) ← (HL) B ← B - 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	1	X	11 101 101 10 110 011	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	(C) ← (HL) B ← B - 1 HL ← HL - 1	X	↕ ①	X	X	1	X	11 101 101 10 101 011	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	(C) ← (HL) B ← B - 1 HL ← HL -1 Repeat until B = 0	X	1	X	X	1	X	11 101 101 10 111 011	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

Notes: ① If the result of B - 1 is zero the Z flag is set, otherwise it is reset.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

INPUT AND OUTPUT GROUP
TABLE 7.0-11

8.0 INTERRUPT RESPONSE

The purpose of an interrupt is to allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start a peripheral service routine. Usually this service routine is involved with the exchange of data, or status and control information, between the CPU and the peripheral. Once the service routine is completed, the CPU returns to the operation from which it was interrupted.

INTERRUPT ENABLE – DISABLE

The Z80 CPU has two interrupt inputs, a software maskable interrupt and a non maskable interrupt. The non maskable interrupt (NMI) can *not* be disabled by the programmer and it will be accepted whenever a peripheral device requests it. This interrupt is generally reserved for very important functions that must be serviced whenever they occur, such as an impending power failure. The maskable interrupt (INT) can be selectively enabled or disabled by the programmer. This allows the programmer to disable the interrupt during periods where his program has timing constraints that do not allow it to be interrupted. In the Z80 CPU there is an enable flip flop (called IFF) that is set or reset by the programmer using the Enable Interrupt (EI) and Disable Interrupt (DI) instructions. When the IFF is reset, an interrupt can not be accepted by the CPU.

Actually, for purposes that will be subsequently explained, there are two enable flip flops, called IFF_1 and IFF_2.

Actually disables interrupts from being accepted.

Temporary storage location for IFF_1.

The state of IFF_1 is used to actually inhibit interrupts while IFF_2 is used as a temporary storage location for IFF_1. The purpose of storing the IFF_1 will be subsequently explained.

A reset to the CPU will force both IFF_1 and IFF_2 to the reset state so that interrupts are disabled. They can then be enabled by an EI instruction at any time by the programmer. When an EI instruction is executed, any pending interrupt request will not be accepted until after the instruction following EI has been executed. This single instruction delay is necessary for cases when the following instruction is a return instruction and interrupts must not be allowed until the return has been completed. The EI instruction sets both IFF_1 and IFF_2 to the enable state. When an interrupt is accepted by the CPU, both IFF_1 and IFF_2 are automatically reset, inhibiting further interrupts until the programmer wishes to issue a new EI instruction. Note that for all of the previous cases, IFF_1 and IFF_2 are always equal.

The purpose of IFF_2 is to save the status of IFF_1 when a non maskable interrupt occurs. When a non maskable interrupt is accepted, IFF_1 is reset to prevent further interrupts until reenabled by the programmer. Thus, after a non maskable interrupt has been accepted, maskable interrupts are disabled but the previous state of IFF_1 has been saved so that the complete state of the CPU just prior to the non maskable interrupt can be restored at any time. When a Load Register A with Register I (LD A, I) instruction or a Load Register A with Register R (LD A, R) instruction is executed, the state of IFF_2 is copied into the parity flag where it can be tested or stored.

A second method of restoring the status of IFF_1 is thru the execution of a Return From Non Maskable Interrupt (RETN) instruction. Since this instruction indicates that the non maskable interrupt service routine is complete, the contents of IFF_2 are now copied back into IFF_1, so that the status of IFF_1 just prior to the acceptance of the non maskable interrupt will be restored automatically.

Figure 8.0-1 is a summary of the effect of different instructions on the two enable flip flops.

Action	IFF_1	IFF_2	
CPU Reset	0	0	
DI	0	0	
EI	1	1	
LD A, I	•	•	$IFF_2 \rightarrow$Parity flag
LD A, R	•	•	$IFF_2 \rightarrow$Parity flag
Accept NMI	0	•	
RETN	IFF_2	•	$IFF_2 \rightarrow IFF_1$

"•" indicates no change

FIGURE 8.0-1
INTERRUPT ENABLE/DISABLE FLIP FLOPS

CPU RESPONSE

Non Maskable

A nonmaskable interrupt will be accepted at all times by the CPU. When this occurs, the CPU ignores the next instruction that it fetches and instead does a restart to location 0066H. Thus, it behaves exactly as if it had received a restart instruction but, it is to a location that is not one of the 8 software restart locations. A restart is merely a call to a specific address in page 0 of memory.

Maskable

The CPU can be programmed to respond to the maskable interrupt in any one of three possible modes.

Mode 0

This mode is identical to the 8080A interrupt response mode. With this mode, the interrupting device can place any instruction on the data bus and the CPU will execute it. Thus, the interrupting device provides the next instruction to be executed instead of the memory. Often this will be a restart instruction since the interrupting device only need supply a single byte instruction. Alternatively, any other instruction such as a 3 byte call to any location in memory could be executed.

The number of clock cycles necessary to execute this instruction is 2 more than the normal number for the instruction. This occurs since the CPU automatically adds 2 wait states to an interrupt response cycle to allow sufficient time to implement an external daisy chain for priority control. Section 5.0 illustrates the detailed timing for an interrupt response. After the application of RESET the CPU will automatically enter interrupt Mode 0.

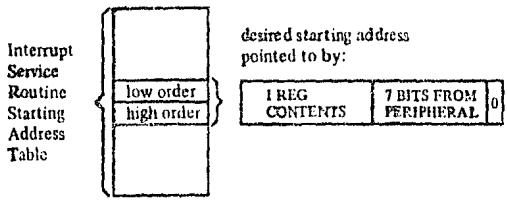
Mode 1

When this mode has been selected by the programmer, the CPU will respond to an interrupt by executing a restart to location 0038H. Thus the response is identical to that for a non maskable interrupt except that the call location is 0038H instead of 0066H. Another difference is that the number of cycles required to complete the restart instruction is 2 more than normal due to the two added wait states.

Mode 2

This mode is the most powerful interrupt response mode. With a single 8 bit byte from the user an indirect call can be made to any memory location.

With this mode the programmer maintains a table of 16 bit starting addresses for every interrupt service routine. This table may be located anywhere in memory. When an interrupt is accepted, a 16 bit pointer must be formed to obtain the desired interrupt service routine starting address from the table. The upper 8 bits of this pointer is formed from the contents of the I register. The I register must have been previously loaded with the desired value by the programmer, i.e. LD I, A. Note that a CPU reset clears the I register so that it is initialized to zero. The lower eight bits of the pointer must be supplied by the interrupting device. Actually, only 7 bits are required from the interrupting device as the least significant bit must be a zero. This is required since the pointer is used to get two adjacent bytes to form a complete 16 bit service routine starting address and the addresses must always start in even locations.



The first byte in the table is the least significant (low order) portion of the address. The programmer must obviously fill this table in with the desired addresses before any interrupts are to be accepted.

Note that this table can be changed at any time by the programmer (if it is stored in Read/Write Memory) to allow different peripherals to be serviced by different service routines.

Once the interrupting devices supplies the lower portion of the pointer, the CPU automatically pushes the program counter onto the stack, obtains the starting address from the table and does a jump to this address. This mode of response requires 19 clock periods to complete (7 to fetch the lower 8 bits from the interrupting device, 6 to save the program counter, and 6 to obtain the jump address.)

Note that the Z80 peripheral devices all include a daisy chain priority interrupt structure that automatically supplies the programmed vector to the CPU during interrupt acknowledge. Refer to the Z80-PIO, Z80-SIO and Z80-CTC manuals for details.

-BLANK-

9.0 HARDWARE IMPLEMENTATION EXAMPLES

This chapter is intended to serve as a basic introduction to implementing systems with the Z80-CPU.

MINIMUM SYSTEM

Figure 9.0-1 is a diagram of a very simple Z-80 system. Any Z-80 system must include the following five elements:

1) Five volt power supply
2) Oscillator
3) Memory devices
4) I/O circuits
5) CPU

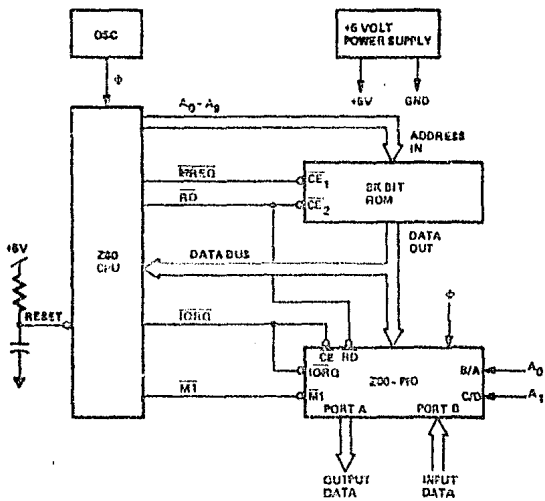


FIGURE 9.0-1
MINIMUM Z80 COMPUTER SYSTEM

Since the Z80-CPU only requires a single 5 volt supply, most small systems can be implemented using only this single supply.

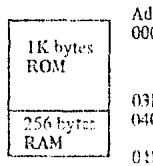
The oscillator can be very simple since the only requirement is that it be a 5 volt square wave. For systems not running at full speed, a simple RC oscillator can be used. When the CPU is operated near the highest possible frequency, a crystal oscillator is generally required because the system timing will not tolerate the drift or jitter that an RC network will generate. A crystal oscillator can be made from inverters and a few discrete components or monolithic circuits are widely available.

The external memory can be any mixture of standard RAM, ROM, or PROM. In this simple example we have shown a single 8K bit ROM (1K bytes) being utilized as the entire memory system. For this example we have assumed that the Z-80 internal register configuration contains sufficient Read/Write storage so that external RAM memory is not required.

Every computer system requires I/O circuits to allow it to interface to the "real world." In this simple example it is assumed that the output is an 8 bit control vector and the input is an 8 bit status word. The input data could be gated onto the data bus using any standard tri-state driver while the output data could be latched with any type of standard TTL latch. For this example we have used a Z80-PIO for the I/O circuit. This single circuit attaches to the data bus as shown and provides the required 16 bits of TTL compatible I/O. (Refer to the Z80-PIO manual for details on the operation of this circuit.) Notice in this example that with only three LSI circuits, a simple oscillator and a single 5 volt power supply, a powerful computer has been implemented.

ADDING RAM

Most computer systems require some amount of external Read/Write memory for data storage and to implement a "stack." Figure 9.0-2 illustrates how 256 bytes of static memory can be added to the previous example. In this example the memory space is assumed to be organized as follows:



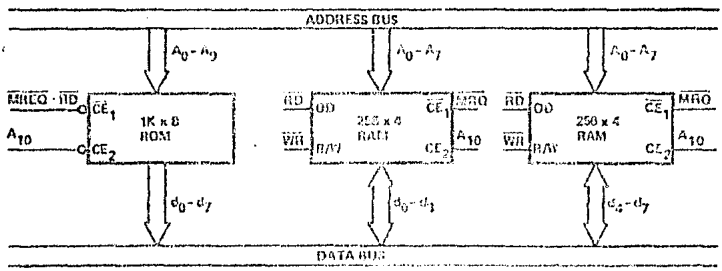


FIGURE 9.0-2
ROM & RAM IMPLEMENTATION EXAMPLE

In this diagram the address space is described in hexidecimal notation. For this example, address bit A_{10} separates the ROM space from the RAM space so that it can be used for the chip select function. For larger amounts of external ROM or RAM, a simple TTL decoder will be required to form the chip selects.

MEMORY SPEED CONTROL

For many applications, it may be desirable to use slow memories to reduce costs. The $\overline{\text{WAIT}}$ line on the CPU allows the Z-80 to operate with any speed memory. By referring back to section 4 you will notice that the memory access time requirements are most severe during the M1 cycle instruction fetch. All other memory accesses have an additional one half of a clock cycle to be completed. For this reason it may be desirable in some applications to add one wait state to the M1 cycle so that slower memories can be used. Figure 9.0-3 is an example of a simple circuit that will accomplish this task. This circuit can be changed to add a single wait state to any memory access as shown in Figure 9.0-4.

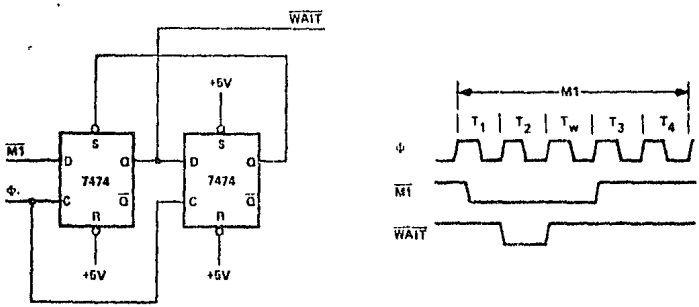


FIGURE 9.0-3
ADDING ONE WAIT STATE TO AN M1 CYCLE

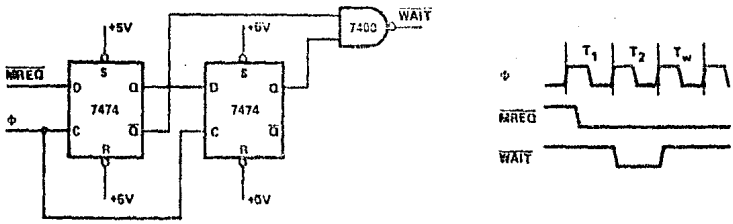


FIGURE 9.0-4
ADDING ONE WAIT STATE TO ANY MEMORY CYCLE

INTERFACING DYNAMIC MEMORIES

This section is intended only to serve as a brief introduction to interfacing dynamic memories. Each individual dynamic RAM has varying specifications that will require minor modifications to the description given here and no attempt will be made in this document to give details for any particular RAM. Separate application notes showing how the Z80-CPU can be interfaced to most popular dynamic RAM's are available from Zilog.

Figure 9.0-5 illustrates the logic necessary to interface 8K bytes of dynamic RAM using 18 pin 4K dynamic memories. This figure assumes that the RAM's are the only memory in the system so that A_{12} is used to select between the two pages of memory. During refresh time, all memories in the system must be read. The CPU provides the proper refresh address on lines A_0 through A_6. To add additional memory to the system it is necessary to only replace the two gates that operate on A_{12} with a decoder that operates on all required address bits. For larger systems, buffering for the address and data bus is also generally required.

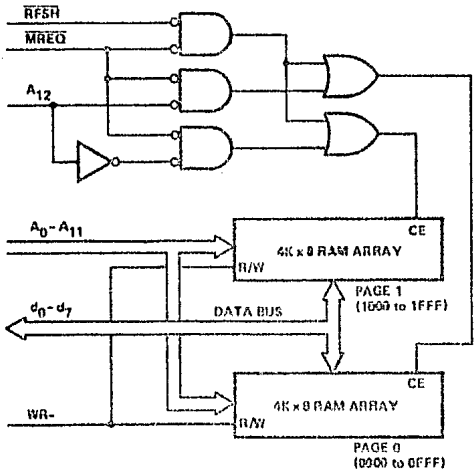


FIGURE 9.0-5
INTERFACING DYNAMIC RAMS

10.0 SOFTWARE IMPLEMENTATION EXAMPLES

10.1 METHODS OF SOFTWARE IMPLEMENTATION

Several different approaches are possible in developing software for the Z-80 (Figure 10.1). First of all, Assembly Language or PL/Z may be used as the source language. These languages may then be translated into machine language on a commercial time sharing facility using a cross-assembler or cross-compiler or, in the case of assembly language, the translation can be accomplished on a Z-80 Development System using a resident assembler. Finally, the resulting machine code can be debugged either on a time-sharing facility using a Z-80 simulator or on a Z-80 Development System which uses a Z80-CPU directly.

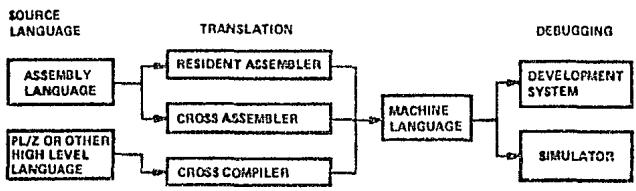


FIGURE 10.1

In selecting a source language, the primary factors to be considered are clarity and ease of programming vs. code efficiency. A high-level language such as PL/Z with its machine independent constructs is typically better for formulating and maintaining algorithms, but the resulting machine code is usually somewhat less efficient than what can be written directly in assembly language. These tradeoffs can often be balanced by combining PL/Z and assembly language routines, identifying those portions of a task which must be optimized and writing them as assembly language subroutines.

Deciding whether to use a resident or cross assembler is a matter of availability and short-term vs. long-term expense. While the initial expenditure for a development system is higher than that for a time-sharing terminal, the cost of an individual assembly using a resident assembler is negligible while the same operation on a time-sharing system is relatively expensive and in a short time this cost can equal the total cost of a development system.

Debugging on a development system vs. a simulator is also a matter of availability and expense combined with operational fidelity and flexibility. As with the assembly process, debugging is less expensive on a development system than on a simulator available through time-sharing. In addition, the fidelity of the operating environment is preserved through real-time execution on a Z80-CPU and by connecting the I/O and memory components which will actually be used in the production system. The only advantage to the use of a simulator is the range of criteria which may be selected for such debugging procedures as tracing and setting breakpoints. This flexibility exists because a software simulation can achieve any degree of complexity in its interpretation of machine instructions while development system procedures have hardware limitations such as the capacity of the real-time storage module, the number of breakpoint registers and the pin configuration of the CPU. Despite such hardware limitations, debugging on a development system is typically more productive than on a simulator because of the direct interaction that is possible between the programmer and the authentic execution of his program.

10.2 SOFTWARE FEATURES OFFERED BY THE Z80-CPU

The Z-80 instruction set provides the user with a large and flexible repetoire of operations with which to formulate control of the Z80-CPU.

The primary, auxiliary and index registers can be used to hold the arguments of arithmetic and logical operations, or to form memory addresses, or as fast-access storage for frequently used data.

Information can be moved directly from register to register; from memory to memory; from memory to registers; or from registers to memory. In addition, register contents and register/memory contents can be exchanged without using temporary storage. In particular, the contents of primary and auxilary registers can be completely exchanged by executing only two instructions, EX and EXX. This register exchange procedure can be used to separate the set of working registers between different logical procedures or to expand the set of available registers in a single procedure.

Storage and retrieval of data between pairs of registers and memory can be controlled on a last-in first-out basis through PUSH and POP instructions which utilize a special stack pointer register, SP. This stack register is available both to manipulate data and to automatically store and retrieve addresses for subroutine linkage. When a subroutine is called, for example, the address following the CALL instruction is placed on the top of the push-down stack pointed to by SP. When a subroutine returns to the calling routine, the address on the top of the stack is used to set the program counter for the address of the next instruction. The stack pointer is adjusted automatically to reflect the current "top" stack position during PUSH, POP, CALL and RET instructions. This stack mechanism allows pushdown data stacks and subroutine calls to be nested to any practical depth because the stack area can potentially be as large as memory space.

The sequence of instruction execution can be controlled by six different flags (carry, zero, sign, parity/overflow, add-subtract, half-carry) which reflect the results of arithmetic, logical, shift and compare instructions. After the execution of an instruction which sets a flag, that flag can be used to control a conditional jump or return instruction. These instructions provide logical control following the manipulation of single bit, eight-bit byte (or) sixteen-bit data quantities.

A full set of logical operations, including AND, OR, XOR (exclusive - OR), CPL (NOR) and NEG (two's complement) are available for Boolean operations between the accumulator and 1) all other eight-bit registers, 2) memory locations or 3) immediate operands.

In addition, a full set of arithmetic and logical shifts in both directions are available which operate on the contents of all eight-bit primary registers or directly on any memory location. The carry flag can be included or simply set by these shift instructions to provide both the testing of shift results and to link register/register or register/memory shift operations.

10.3 EXAMPLES OF USE OF SPECIAL Z80 INSTRUCTIONS

A. Let us assume that a string of data in memory starting at location "DATA" is to be moved into another area of memory starting at location "BUFFER" and that the string length is 737 bytes. This operation can be accomplished as follows:

```
LD      HL , DATA       ;START ADDRESS OF DATA STRING
LD      DE , BUFFER     ;START ADDRESS OF TARGET BUFFER
LD      BC , 737        ;LENGTH OF DATA STRING
LDIR                    ;MOVE STRING – TRANSFER MEMORY POINTED TO
                        ;BY HL INTO MEMORY LOCATION POINTED TO BY DE
                        ;INCREMENT HL AND DE, DECREMENT BC
                        ;PROCESS UNTIL BC = 0.
```

11 bytes are required for this operation and each byte of data is moved in 21 clock cycles.

B. Let's assume that a string in memory starting at location "DATA" is to be moved into another area of memory starting at location "BUFFER" until an ASCII $ character (used as string delimiter) is found. Let's also assume that the maximum string length is 132 characters. The operation can be performed as follows:

```
        LD      HL , DATA       ;STARTING ADDRESS OF DATA STRING
        LD      DE , BUFFER     ;STARTING ADDRESS OF TARGET BUFFER
        LD      BC , 132        ;MAXIMUM STRING LENGTH
        LD      A ,'$'          ;STRING DELIMITER CODE
LOOP:CP         (HL)            ;COMPARE MEMORY CONTENTS WITH DELIMITER
        JR      Z , END – $     ;GO TO END IF CHARACTERS EQUAL
        LDI                     ;MOVE CHARACTER (HL) to (DE)
                                ;INCREMENT HL AND DE, DECREMENT BC
        JP      PE , LOOP       ;GO TO "LOOP" IF MORE CHARACTERS
END:                            ;OTHERWISE, FALL THROUGH
                                ;NOTE: P/V FLAG IS USED
                                ;TO INDICATE THAT REGISTER BC WAS
                                ; DECREMENTED TO ZERO.
```

19 bytes are required for this operation.

C. Let us assume that a 16-digit decimal number represented in packed BCD format (two BCD digits/ byte) has to be shifted as shown in the Figure 10.2 in order to mechanize BCD multiplication or division. The operation can be accomplished as follows:

```
        LD      HL , DATA       ; ADDRESS OF FIRST BYTE
        LD      B , COUNT       ; SHIFT COUNT
        XOR     A               ; CLEAR ACCUMULATOR
ROTAT: RLD                      ; ROTATE LEFT LOW ORDER DIGIT IN ACC
                                ; WITH DIGITS IN (HL)
        INC     HL              ; ADVANCE MEMORY POINTER
        DJNZ    ROTAT – $       ; DECREMENT B AND GO TO ROTAT IF
                                ; B IS NOT ZERO, OTHERWISE FALL THROUGH
```

11 bytes are required for this operation.

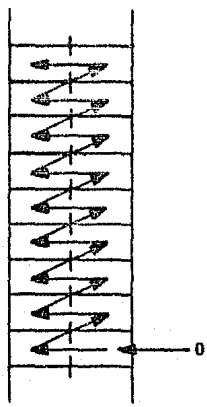


FIGURE 10.2

D. Let us assume that one number is to be subtracted from another and a) that they are both in packed BCD format, b) that they are of equal but varying length, and c) that the result is to be stored in the location of the minuend. The operation can be accomplished as follows:

```
        LD      HL , ARG1       ; ADDRESS OF MINUEND
        LD      DE , ARG2       ; ADDRESS OF SUBTRAHEND
        LD      B , LENGTH      ; LENGTH OF TWO ARGUMENTS
        AND     A               ; CLEAR CARRY FLAG
SUBDEC: LD      A , (DE)        ; SUBTRAHEND TO ACC
        SBC     A , (HL)        ; SUBTRACT (HL) FROM ACC
        DAA                     ; ADJUST RESULT TO DECIMAL CODED VALUE
        LD      (HL) , A        ; STORE RESULT
        INC     HL              ; ADVANCE MEMORY POINTERS
        INC     DE
        DJNZ    SUBDEC – $      ; DECREMENT B AND GO TO "SUBDEC" IF B
                                ; NOT ZERO, OTHERWISE FALL THROUGH
```

17 bytes are required for this operation.

10.4 EXAMPLES OF PROGRAMMING TASKS

A. The following program sorts an array of numbers each in the range (0,255) into ascending order using a standard exchange sorting algorithm.

```
LOC  OBJ CODE   STMT  SOURCE STATEMENT

                   1  ;  *** STANDARD EXCHANGE (BUBBLE) SORT ROUTINE ***
                   2  ;
                   3  ;  AT ENTRY:  HL CONTAINS ADDRESS OF DATA
                   4  ;              C CONTAINS NUMBER OF ELEMENTS TO BE SORTED
                   5  ;              (1<C<256)
                   6  ;
                   7  ;  AT EXIT: DATA SORTED IN ASCENDING ORDER
                   8  ;
                   9  ;  USE OF REGISTERS
                  10  ;
                  11  ;  REGISTER   CONTENTS
                  12  ;
                  13  ;  A          TEMPORARY STORAGE FOR CALCULATIONS
                  14  ;  B          COUNTER FOR DATA ARRAY
                  15  ;  C          LENGTH OF DATA ARRAY
                  16  ;  D          FIRST ELEMENT IN COMPARISON
                  17  ;  E          SECOND ELEMENT IN COMPARISON
                  18  ;  H          FLAG TO INDICATE EXCHANGE
                  19  ;  L          UNUSED
                  20  ;  IX         POINTER INTO DATA ARRAY
                  21  ;  IY         UNUSED
                  22  ;
0000  222600      23  SORT:  LD     (DATA), HL     ; SAVE DATA ADDRESS
0003  CB84        24  LOOP:  RES    FLAG, H        ; INITIALIZE EXCHANGE FLAG
0005  41          25         LD     B, C           ; INITIALIZE LENGTH COUNTER
0006  05          26         DEC    B              ; ADJUST FOR TESTING
0007  DD2A2600    27         LD     IX, (DATA)     ; INITIALIZE ARRAY POINTER
000B  DD7E00      28  NEXT:  LD     A, (IX)        ; FIRST ELEMENT IN COMPARISON
000E  57          29         LD     D, A           ; TEMPORARY STORAGE FOR ELEMENT
000F  DD5E01      30         LD     E, (IX+1)      ; SECOND ELEMENT IN COMPARISON
0012  93          31         SUB    E              ; COMPARISON FIRST TO SECOND
0013  3008        32         JR     NC, NOEX-$     ; IF FIRST > SECOND, NO JUMP
0015  DD7300      33         LD     (IX), E        ; EXCHANGE ARRAY ELEMENTS
0018  DD7201      34         LD     (IX+1), D
001B  CBC4        35         SET    FLAG, H        ; RECORD EXCHANGE OCCURRED
001D  DD23        36  NOEX:  INC    IX             ; POINT TO NEXT DATA ELEMENT
001F  10EA        37         DJNZ   NEXT-$         ; COUNT NUMBER OF COMPARISONS
                  38                               ; REPEAT IF MORE DATA PAIRS
0021  CB44        39         BIT    FLAG, H        ; DETERMINE IF EXCHANGE OCCURRED
0023  20DE        40         JR     NZ, LOOP-$     ; CONTINUE IF DATA UNSORTED
0025  C9          41         RET                   ; OTHERWISE, EXIT
                  42  ;
0026              43  FLAG:  EQU    0              ; DESIGNATION OF FLAG BIT
0026              44  DATA:  DEFS   2              ; STORAGE FOR DATA ADDRESS
                  45         END
```

B. The following program multiplies two unsigned 16 bit integers and leaves the result in the HL register pair.

```
01/22/76     11:32:36                MULTIPLY LISTING                    PAGE 1
LOC    OBJ CODE    STMT    SOURCE STATEMENT

0000                 1    MULT:;   UNSIGNED SIXTEEN BIT INTEGER MULTIPLY.
                     2    ;        ON ENTRANCE:  MULTIPLIER IN DE.
                     3    ;                      MULTIPLICAND IN HL.
                     4    ;
                     5    ;        ON EXIT:  RESULT IN HL.
                     6    ;
                     7    ;        REGISTER USES:
                     8    ;
                     9    ;
                    10    ;        H     HIGH ORDER PARTIAL RESULT
                    11    ;        L     LOW ORDER PARTIAL RESULT
                    12    ;        D     HIGH ORDER MULTIPLICAND
                    13    ;        E     LOW ORDER MULTIPLICAND
                    14    ;        B     COUNTER FOR NUMBER OF SHIFTS
                    15    ;        C     HIGH ORDER BITS OF MULTIPLIER
                    16    ;        A     LOW ORDER BITS OF MULTIPLIER
                    17    ;
0000   0610        18              LD    B, 16;          NUMBER OF BITS- INITIALIZE
0002   4A          19              LD    C, D;           MOVE MULTIPLIER
0003   7B          20              LD    A, E;
0004   EB          21              EX    DE, HL;         MOVE MULTIPLICAND
0005   210000      22              LD    HL, 0;          CLEAR PARTIAL RESULT
0008   CB39        23    MLOOP:    SRL   C;              SHIFT MULTIPLIER RIGHT
000A   1F          24              RRA                   LEAST SIGNIFICANT BIT IS
                    25    ;                             IN CARRY.
000B   3001        26              JR    NC, NOADD-$;    IF NO CARRY, SKIP THE ADD.
000D   19          27              ADD   HL, DE;         ELSE ADD MULTIPLICAND TO
                    28    ;                             PARTIAL RESULT.
000E   EB          29    NOADD:    EX    DE, HL;         SHIFT MULTIPLICAND LEFT
000F   29          30              ADD   HL, HL;         BY MULTIPLYING IT BY TWO.
0010   EB          31              EX    DE, HL;
0011   10F5        32              DJNZ  MLOOP-$;        REPEAT UNTIL NO MORE BITS.
0013   C9          33              RET;
                    34              END;
```

Absolute Maximum Ratings

Temperature Under Bias	Specified operating range
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

*Comment

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note: For Z80B CPU, all AC and DC characteristics remain the same for the military grade parts except I_{CC}.

I_{CC} = 200mA

Z80-CPU D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ± 5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} -.6		V_{CC}+.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL}=1.8mA
V_{OH}	Output High Voltage	2.4			V	I_{OH}=-250μA
I_{CC}	Power Supply Current			150	mA	
I_{LI}	Input Leakage Current			±10	μA	V_{IN}=0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			±10	μA	V_{OUT}=2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT}=0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	$0 \leq V_{IN} \leq V_{CC}$

Capacitance

T_A = 25°C, f = 1 MHz,
unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80-CPU Ordering Information

C = Ceramic
P = Plastic
S = Standard 5V ±5% 0° to 70°C
E = Extended 5V ±5% -40° to 85°C
M = Military 5V ±10% -55° to 125°C

Z80A-CPU D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ± 5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} -.6		V_{CC}+.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL}=1.8mA
V_{OH}	Output High Voltage	2.4			V	I_{OH}=-250μA
I_{CC}	Power Supply Current		90	200	mA	
I_{LI}	Input Leakage Current			±10	μA	V_{IN}=0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			±10	μA	V_{OUT}=2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT}=0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	$0 \leq V_{IN} \leq V_{CC}$

Capacitance

T_A = 25°C, f = 1 MHz,
unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80A-CPU Ordering Information

C = Ceramic
P = Plastic
S = Standard 5V ±5% 0° to 70°C

T_A = 0°C to 70°C, Vcc = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition	
Φ	t_c	Clock Period	4	1121	μsec		[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_f + t_r$
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	190	[1]	nsec		
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	190	2000	nsec		
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec		
A_{0-15}	$t_{D(AD)}$	Address Output Delay		145	nsec	C_L = 50pF	
	$t_{F(AD)}$	Delay to Float		110	nsec		
	t_{acm}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	111		nsec		[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$
	t_{aci}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	727		nsec		[2] $t_{aci} = t_c - 80$
	t_{ca}	Address Stable from $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ or $\overline{MREQ}$	131		nsec		
	t_{caf}	Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float	141		nsec		
D_{0-7}	$t_{D(D)}$	Data Output Delay		230	nsec	C_L = 50pF	[3] $t_{cd} = t_{w(\Phi L)} + t_f - 40$
	$t_{F(D)}$	Delay to Float During Write Cycle		90	nsec		
	$t_{sd(D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	50		nsec		[4] $t_{cdf} = t_{w(\Phi L)} + t_f - 60$
	$t_{s(D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	60		nsec		[5] $t_{dcm} = t_c - 210$
	t_{dcm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	191		nsec		
	t_{dci}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	161		nsec		[6] $t_{dzf} = t_{w(\Phi L)} + t_f - 210$
	t_{cdf}	Data Stable From $\overline{WR}$	191				
	t_H	Any Hold Time for Setup Time	0		nsec		[7] $t_{cdf} = t_{w(\Phi L)} + t_f - 80$
$\overline{MREQ}$	$t_{DL(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	C_L = 50pF	
	$t_{DH(MR)}$	$\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High		100	nsec		
	$t_{DH(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High		100	nsec		
	$t_{w(MRL)}$	Pulse Width, $\overline{MREQ}$ Low	191		nsec		[8] $t_{w(MRL)} = t_c - 40$
	$t_{w(MRH)}$	Pulse Width, $\overline{MREQ}$ High	191		nsec		[9] $t_{w(MRH)} = t_{w(\Phi H)} + t_f - 30$
$\overline{IORQ}$	$t_{DL(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low		90	nsec	C_L = 50pF	
	$t_{DL(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ Low		110	nsec		
	$t_{DH(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High		100	nsec		
	$t_{DH(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High		110	nsec		
$\overline{RD}$	$t_{DL(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ Low		100	nsec	C_L = 50pF	
	$t_{DL(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low		130	nsec		
	$t_{DH(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High		100	nsec		
	$t_{DH(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High		110	nsec		
$\overline{WR}$	$t_{DL(WR)}$	$\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low		80	nsec	C_L = 50pF	
	$t_{DL(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low		90	nsec		
	$t_{DH(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ High		100	nsec		
	$t_{w(WRL)}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec		[10] $t_{w(WRL)} = t_c - 40$
$\overline{M1}$	$t_{DL(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ Low		130	nsec	C_L = 50pF	
	$t_{DH(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ High		130	nsec		
$\overline{RFSH}$	$t_{DL(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ Low		180	nsec	C_L = 50pF	
	$t_{DH(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ High		150	nsec		
$\overline{WAIT}$	$t_{s(WT)}$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec		
$\overline{HALT}$	$t_{D(HT)}$	$\overline{HALT}$ Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF	
$\overline{INT}$	$t_{s(IT)}$	$\overline{INT}$ Setup Time to Rising Edge of Clock	80		nsec		
$\overline{NMI}$	$t_{w(NRL)}$	Pulse Width, $\overline{NMI}$ Low	80		nsec		
$\overline{BUSRQ}$	$t_{s(BQ)}$	$\overline{BUSRQ}$ Setup Time to Rising Edge of Clock	80		nsec		
$\overline{BUSAK}$	$t_{DL(BA)}$	$\overline{BUSAK}$ Delay From Rising Edge of Clock, $\overline{BUSAK}$ Low		120	nsec	C_L = 50pF	
	$t_{DH(BA)}$	$\overline{BUSAK}$ Delay From Falling Edge of Clock, $\overline{BUSAK}$ High		110	nsec		
$\overline{RESET}$	$t_{s(RS)}$	$\overline{RESET}$ Setup Time to Rising Edge of Clock	90		nsec		
	$t_{F(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		100	nsec		
	t_{ml}	$\overline{M1}$ Stable Prior to $\overline{IORQ}$ (Interrupt Ack.)	[11]		nsec		[11] $t_{ml} = 3t_c + t_{w(\Phi H)} + t_f - 80$

NOTES:

A. Data should be enabled onto the CPU data bus when $\overline{RD}$ is active. During interrupt acknowledge data should be enabled when $\overline{M1}$ and $\overline{IORQ}$ are both active.

B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.

C. The $\overline{RESET}$ signal must be active for a minimum of 3 clock cycles.

D. Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V ±5%
Add 10nsec delay for each 50pf increase in load up to a maximum of 200pf for the data bus & 100pf for address & control lines

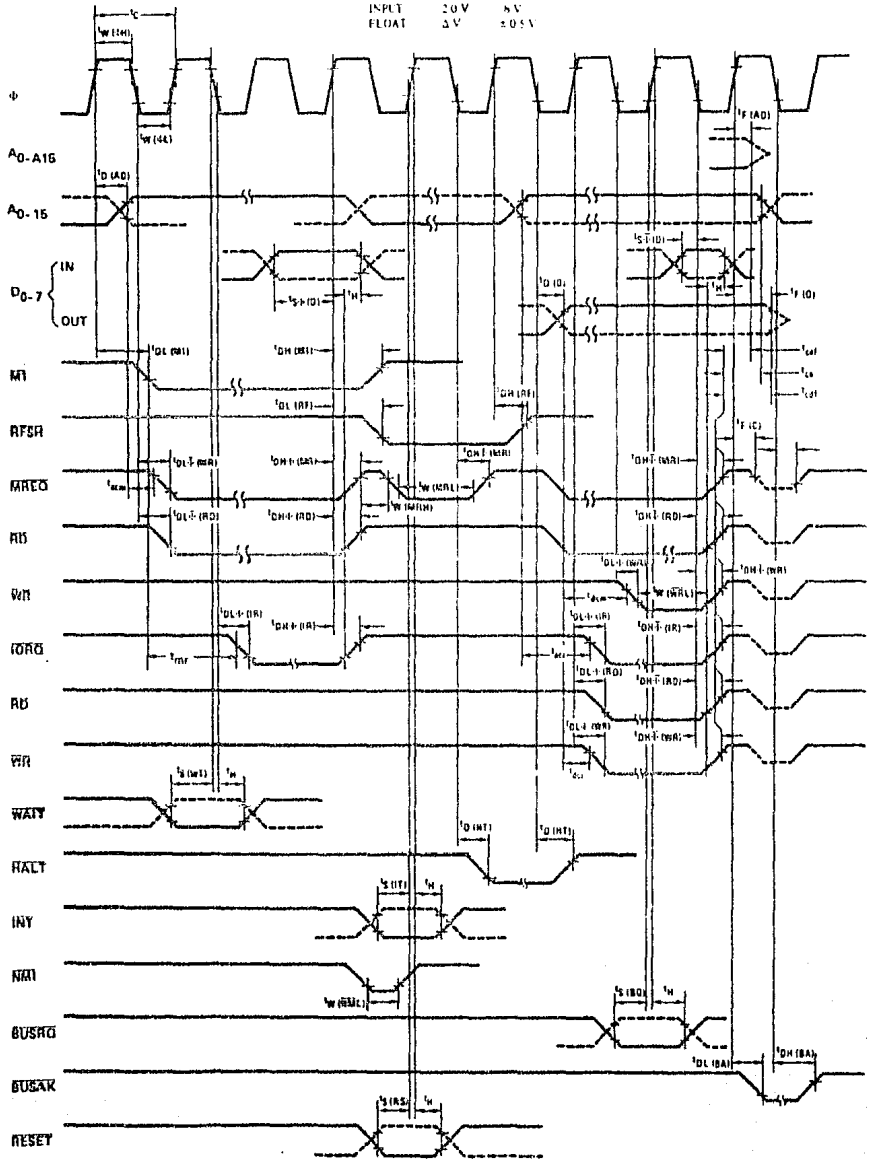
E. Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 μsec maximum.

Load circuit for Output

A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	V_{CC} - .6V	.45V
OUTPUT	2.0V	.8V
INPUT	2.0V	.8V
FLOAT	.5V	±0.5V



T_A = 0°C to 70°C, V_{CC} = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition	
Φ	t_c	Clock Period	25	[12]	nsec		[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	110	[11]	nsec		
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	110	2000	nsec		
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec		
A_{0-15}	$t_{D(AD)}$	Address Output Delay		110	nsec	C_L = 50pF	
	$t_{F(AD)}$	Delay to Float		90	nsec		
	t_{acm}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	[10]		nsec		
	t_{aci}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[5]		nsec		[1] $t_{acm} = t_{w(\Phi H)} + t_f - 65$
	t_{ca}	Address Stable from $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ or $\overline{MREQ}$	[11]		nsec		[2] $t_{aci} = t_c - 70$
	t_{caf}	Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float	[4]		nsec		
D_{0-7}	$t_{D(D)}$	Data Output Delay		150	nsec	C_L = 50pF	[3] $t_{cd} = t_{w(\Phi L)} + t_r - 50$
	$t_{F(D)}$	Delay to Float During Write Cycle		90	nsec		[4] $t_{caf} = t_{w(\Phi L)} + t_r - 45$
	$t_{sD(D)}$	Data Setup Time to Falling Edge of Clock During M1 Cycle	[3]		nsec		[5] $t_{dcm} = t_c - 170$
	$t_{sD(D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	[3]		nsec		[6] $t_{dci} = t_{w(\Phi L)} + t_r - 170$
	t_{dm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[6]		nsec		[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 70$
	t_{di}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec		
	t_{df}	Data Stable From $\overline{WR}$	[7]		nsec		
	t_H	Any Hold Time for Setup Time		0	nsec		
$t_{0\Phi TO}$	$t_{D(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low		85	nsec	C_L = 50pF	[8] $t_{w(\overline{MRH})} = t_c - 30$
	$t_{D(MR)}$	$\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High		85	nsec		
	$t_{F(MR)}$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High		85	nsec		[9] $t_{w(\overline{MRL})} = t_{w(\Phi H)} + t_{w(\Phi L)} + t_f - 20$
	$t_{w(\overline{MRL})}$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec		
	$t_{w(\overline{MRH})}$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec		
$\overline{IORQ}$	$t_{D(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low		75	nsec	C_L = 50pF	
	$t_{D(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ Low		85	nsec		
	$t_{F(IR)}$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High		85	nsec		
	$t_{DF(IR)}$	$\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High		85	nsec		
$\overline{RD}$	$t_{D(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ Low		85	nsec	C_L = 50pF	
	$t_{D(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low		95	nsec		
	$t_{F(RD)}$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High		85	nsec		
	$t_{DF(RD)}$	$\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High		85	nsec		
$\overline{WR}$	$t_{D(WR)}$	$\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low		65	nsec	C_L = 50pF	
	$t_{D(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low		95	nsec		
	$t_{F(WR)}$	$\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ High		85	nsec		
	$t_{w(\overline{WRL})}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec		[10] $t_{w(\overline{WRL})} = t_c - 30$
$\overline{M1}$	$t_{D(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock $\overline{M1}$ Low		100	nsec	C_L = 50pF	
	$t_{D(M1)}$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ High		100	nsec		
$\overline{RFSH}$	$t_{DL(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ Low		120	nsec	C_L = 50pF	
	$t_{DH(RF)}$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ High		120	nsec		
$\overline{WAIT}$	$t_s(WT)$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec		
$\overline{HALT}$	$t_{D(HT)}$	$\overline{HALT}$ Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF	
$\overline{INT}$	$t_s(IT)$	$\overline{INT}$ Setup Time to Rising Edge of Clock	80		nsec		
$\overline{NMI}$	$t_w(\overline{NML})$	Pulse Width, $\overline{NMI}$ Low	80		nsec		
$\overline{BUSRQ}$	$t_s(BQ)$	$\overline{BUSRQ}$ Setup Time to Rising Edge of Clock	50		nsec		
$\overline{BUSAK}$	$t_{DL(BA)}$	$\overline{BUSAK}$ Delay From Rising Edge of Clock, $\overline{BUSAK}$ Low		100	nsec	C_L = 50pF	
	$t_{DH(BA)}$	$\overline{BUSAK}$ Delay From Falling Edge of Clock, $\overline{BUSAK}$ High		100	nsec		
$\overline{RESET}$	$t_s(RS)$	$\overline{RESET}$ Setup Time to Rising Edge of Clock	60		nsec		
	$t_{F(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		80	nsec		
	t_{ml}	M1 Stable Prior to IORQ (Interrupt Ack.)	[11]		nsec		[11] $t_{ml} = 2t_c + t_{w(\Phi H)} + t_f - 65$

NOTES:

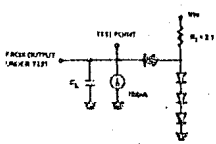
A. Data should be enabled onto the CPU data bus when $\overline{RD}$ is active. During interrupt acknowledge data should be enabled when $\overline{M1}$ and $\overline{IORQ}$ are both active.

B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.

C. The $\overline{RESET}$ signal must be active for a minimum of 3 clock cycles.

D. Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V ±5%
Add 10nsec delay for each 50pf increase in load up to maximum of 200pf for data bus and 100pf for address & control lines.

E. Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 μsec maximum



Load circuit for Output

ADC HL, ss	Add with Carry Reg. pair ss to HL
ADC A, s	Add with carry operand s to Acc.
ADD A, n	Add value n to Acc.
ADD A, r	Add Reg. r to Acc.
ADD A, (HL)	Add location (HL) to Acc.
ADD A, (IX+d)	Add location (IX+d) to Acc.
ADD A, (IY+d)	Add location (IY+d) to Acc.
ADD HL, ss	Add Reg. pair ss to HL
ADD IX, pp	Add Reg. pair pp to IX
ADD IY, rr	Add Reg. pair rr to IY
AND s	Logical 'AND' of operand s and Acc.
BIT b, (HL)	Test BIT b of location (HL)
BIT b, (IX+d)	Test BIT b of location (IX+d)
BIT b, (IY+d)	Test BIT b of location (IY+d)
BIT b, r	Test BIT b of Reg. r
CALL cc, nn	Call subroutine at location nn if condition cc if true
CALL nn	Unconditional call subroutine at location nn
CCF	Complement carry flag
CP s	Compare operand s with Acc.
CPD	Compare location (HL) and Acc. decrement HL and BC
CPDR	Compare location (HL) and Acc. decrement HL and BC, repeat until BC=0
CPI	Compare location (HL) and Acc. increment HL and decrement BC
CPIR	Compare location (HL) and Acc. increment HL, decrement BC repeat until BC=0
CPL	Complement Acc. (1's comp)
DAA	Decimal adjust Acc.
DEC m	Decrement operand m
DEC IX	Decrement IX
DEC IY	Decrement IY
DEC ss	Decrement Reg. pair ss
DI	Disable interrupts
DJNZ e	Decrement B and Jump relative if B≠0
EI	Enable interrupts
EX (SP), HL	Exchange the location (SP) and HL
EX (SP), IX	Exchange the location (SP) and IX
EX (SP), IY	Exchange the location (SP) and IY
EX AF, AF'	Exchange the contents of AF and AF'
EX DE, HL	Exchange the contents of DE and HL
EXX	Exchange the contents of BC, DE, HL with contents of BC', DE', HL' respectively
HALT	HALT (wait for interrupt or reset)
IM 0	Set interrupt mode 0
IM 1	Set interrupt mode 1
IM 2	Set interrupt mode 2
IN A, (n)	Load the Acc. with input from device n
IN r, (C)	Load the Reg. r with input from device (C)
INC (HL)	Increment location (HL)
INC IX	Increment IX
INC (IX+d)	Increment location (IX+d)
INC IY	Increment IY
INC (IY+d)	Increment location (IY+d)
INC r	Increment Reg. r
INC ss	Increment Reg. pair ss
IND	Load location (HL) with input from port (C), decrement HL and B
INDR	Load location (HL) with input from port (C), decrement HL and decrement B, repeat until B=0
INI	Load location (HL) with input from port (C); and increment HL and decrement B

INIR	Load location (HL) with input from port (C), increment HL and decrement B, repeat until B=0
JP (HL)	Unconditional Jump to (HL)
JP (IX)	Unconditional Jump to (IX)
JP (IY)	Unconditonal Jump to (IY)
JP cc, nn	Jump to location nn if condition cc is true
JP nn	Unconditional jump to location nn
JP C, e	Jump relative to PC+e if carry=1
JR e	Unconditional Jump relative to PC+e
JP NC, e	Jump relative to PC+e if carry=0
JR NZ, e	Jump relative to PC+e if non zero (Z=0)
JR Z, e	Jump relative to PC+e if zero (Z=1)
LD A, (BC)	Load Acc. with location (BC)
LD A, (DE)	Load Acc. with location (DE)
LD A, I	Load Acc. with I
LD A, (nn)	Load Acc. with location nn
LD A, R	Load Acc. with Reg. R
LD (BC), A	Load location (BC) with Acc.
LD (DE), A	Load location (DE) with Acc.
LD (HL), n	Load location (HL) with value n
LD dd, nn	Load Reg. pair dd with value nn
LD HL, (nn)	Load HL with location (nn)
LD (HL), r	Load location (HL) with Reg. r
LD I, A	Load I with Acc.
LF IX, nn	Load IX with value nn
LD IX, (nn)	Load IX with location (nn)
LD (IX+d), n	Load location (IX+d) with value n
LD (IX+d), r	Load location (IX+d) with Reg. r
LD IY, nn	Load IY with value nn
LD IY, (nn)	Load IY with location (nn)
LD (IY+d), n	Load location (IY+d) with value n
LD (IY+d), r	Load location (IY+d) with Reg. r
LD (nn), A	Load location (nn) with Acc.
LD (nn), dd	Load location (nn) with Reg. pair dd
LD (nn), HL	Load location (nn) with HL
LD (nn), IX	Load location (nn) with IX
LD (nn), IY	Load location (nn) with IY
LD R, A	Load R with Acc.
LD r, (HL)	Load Reg. r with location (HL)
LD r, (IX+d)	Load Reg. r with location (IX+d)
LD r, (IY+d)	Load Reg. r with location (IY+d)
LD r, n	Load Reg. r with value n
LD r, r'	Load Reg. r with Reg. r'
LD SP, HL	Load SP with HL
LD SP, IX	Load SP with IX
LD SP, IY	Load SP with IY
LDD	Load location (DE) with location (HL), decrement DE, HL and BC
LDDR	Load location (DE) with location (HL), decrement DE, HL and BC; repeat until BC=0
LDI	Load location (DE) with location (HL), increment DE, HL, decrement BC
LDIR	Load location (DE) with location (HL), increment DE, HL, decrement BC and repeat until BC=0
NEG	Negate Acc. (2's complement)
NOP	No operation
OR s	Logical 'OR' or operand s and Acc.
OTDR	Load output port (C) with location (HL) decrement HL and B, repeat, until B=0
OTIR	Load output port (C) with location (HL), increment HL, decrement B, repeat until B=0
OUT (C), r	Load output port (C) with Reg. r
OUT (n), A	Load output port (n) with Acc.
OUTD	Load output port (C) with location (HL), decrement HL and B
OUTI	Load output port (C) with location (HL), increment HL and decrement B

POP IX	Load IX with top of stack
POP IY	Load IY with top of stack
POP qq	Load Reg. pair qq with top of stack
PUSH IX	Load IX onto stack
PUSH IY	Load IY onto stack
PUSH qq	Load Reg. pair qq onto stack
RES b, m	Reset Bit b of operand m
RET	Return from subroutine
RET cc	Return from subroutine if condition cc is true
RETI	Return from interrupt
RETN	Return from non maskable interrupt
RL m	Rotate left through carry operand m
RLA	Rotate left Acc. through carry
RLC (HL)	Rotate location (HL) left circular
RLC (IX+d)	Rotate location (IX+d) left circular
RLC (IY+d)	Rotate location (IY+d) left circular
RLC r	Rotate Reg. r left circular
RLCA	Rotate left circular Acc.
RLD	Rotate digit left and right between Acc. and location (HL)
RR m	Rotate right through carry operand m
RRA	Rotate right Acc. through carry
RRC m	Rotate operand m right circular
RRCA	Rotate right circular Acc.
RRD	Rotate digit right and left between Acc. and location (HL)
RST p	Restart to location p
SBC A, s	Subtract operand s from Acc. with carry
SBC HL, ss	Subtract Reg. pair ss from HL with carry
SCF	Set carry flag (C=1)
SET b, (HL)	Set Bit b of location (HL)
SET b, (IX+d)	Set Bit b of location (IX+d)
SET b, (IY+d)	Set Bit b of location (IY+d)
SET b, r	Set Bit b of Reg. r
SLA m	Shift operand m left arithmetic
SRA m	Shift operand m right arithmetic
SRL m	Shift operand m right logical
SUB s	Subtract operand s from Acc.
XOR s	Exclusive 'OR' operand s and Acc.

TELEVIDEO®
MODEL 910 *PLUS* TERMINAL OPERATOR'S MANUAL

TeleVideo Document No. B300021-001

February 1982

Disclaimer

TeleVideo Systems, Inc. makes no representations or warranties with respect to this manual. Further, TeleVideo Systems, Inc. reserves the right to make changes in the specifications of the product described within this manual at any time without notice and without obligation of TeleVideo Systems, Inc. to notify any person of such revision or changes.

"Warning: This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual may cause interference to radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference, in which case the user of his own expenses will be required to correct the interference."

TeleVideo® is a registered trademark of TeleVideo Systems, Inc.

TELEVIDEO SYSTEMS, INC.
1170 Morse Avenue
Sunnyvale, California 94086
408/745-7760

LIMITED WARRANTY AND EXTENDED WARRANTY

STATEMENT OF LIMITED WARRANTY

TeleVideo Systems, Inc. ("TeleVideo") warrants to Distributor that its products, except software, will be free from defects in materials and workmanship for 180 days after shipment by TeleVideo. TeleVideo will repair or replace, at its option, any equipment or parts that TeleVideo determines were defective when shipped to Distributor. Distributor may pass along to its initial end-user customer ("Customer") the balance of its limited warranty coverage on the terms and conditions stated herein, up to a maximum of 90 days beginning on the date of shipment to Customer. This warranty may not be further assigned.

If service is required under this warranty, Distributor or Customer must: (1) notify TeleVideo in writing or by telephone of the defect; (2) obtain a TeleVideo Return Material Authorization ("RMA") number which must be shown on the outside of all shipping containers and in all correspondence; and (3) send the unit to TeleVideo freight prepaid unless otherwise instructed by TeleVideo. In making repairs, TeleVideo will replace parts as needed on an exchange basis. After repair, TeleVideo will prepay return freight charges.

LIMITED WARRANTY EXCLUSIONS

This limited warranty does not cover losses or damage which occur:

1. In shipment to or from Distributor or Customer.
2. Due to neglect, misuse or any cause other than ordinary use in commercial or industrial applications.
3. Due to adjustment, repair or modifications by other than TeleVideo-authorized personnel.
4. Due to improper environment, excessive or inadequate heating or air conditioning, and electrical power failures, surges or other irregularities.
5. Due to other causes beyond the control of TeleVideo, including but not limited to natural causes or disasters, war, insurrection, civil disturbances, labor disputes, requirements of domestic or foreign governments, and shortage or allocation of materials, utilities or other resources.

Any statements made about TeleVideo's products by salesmen, dealers, distributors or agents, unless in writing signed by an officer of TeleVideo, do not constitute warranties, shall not be relied upon by Distributor or Customer and are not part of the contract of sale.

All statements, technical information and recommendations in this document and in any manuals or related documents are believed to be reliable, but the accuracy or completeness thereof is not guaranteed.

TELEVIDEO DOES NOT WARRANT THAT ITS PRODUCTS ARE MERCHANTABLE OR FIT FOR ANY PARTICULAR PURPOSE. THIS LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS, IMPLIED OR STATUTORY. TELEVIDEO MAKES NO WARRANTY WHATSOEVER CONCERNING ANY SOFTWARE PRODUCTS, WHICH ARE SOLD "AS IS" AND "WITH ALL FAULTS."

TELEVIDEO'S LIABILITY, WHETHER BASED ON CONTRACT, TORT, WARRANTY, STRICT LIABILITY OR ANY OTHER THEORY, SHALL NOT EXCEED THE PRICE OF THE INDIVIDUAL UNIT WHOSE DEFECT OR DAMAGE IS THE BASIS OF THE CLAIM. IN NO EVENT SHALL TELEVIDEO BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF USE OF FACILITIES OR EQUIPMENT, OR OTHER INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES. THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE REMEDY PROVIDED TO DISTRIBUTOR AND TO CUSTOMER.

EXTENDED WARRANTY

TeleVideo offers an Extended Warranty Contract on the same terms and conditions as the original Limited Warranty. To take advantage of this Extended Warranty, you must sign the Extended Warranty Contract and return it, together with full payment, to TeleVideo prior to the end of your normal warranty period. The extended warranty lasts for one year; the cost is $75.00.

To renew the extended warranty for another year, the same procedure must be followed.

Shipping charges are *not* included in the Extended Warranty. This is the only expense you incur.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This manual will explain how to install, operate, program, and troubleshoot your new terminal. The manual has been designed to help you use the terminal easily regardless of your previous experience with terminals

1.1 TERMINAL OVERVIEW

The Model 910 *PLUS* CRT terminal is a modular-design unit. Its nonglare green screen with high resolution characters reduces operator fatigue. Characters can be green on black or black on green.

The terminal includes many deluxe features. During installation you can change the terminal to one of four language character sets (English, Spanish, German, or French). Fifteen baud rates are available to fit your system requirements. An RS232C printer port allows you to connect an auxiliary printer of your choice. An optional current loop interface can be added, allowing the terminal to be installed up to 1,000 feet from your computer system.

You can select video attributes, transmission modes, and cursor appearance. Additional commands control protected fields, editing modes, monitor mode, handshaking protocol, and extension or copy print. Using a special "FUNCT" key plus an additional character allows you to quickly transmit a preprogrammed command sequence. Transmission can be conversational or block, editing can be local or duplex.

1.2 HOW TO USE THIS MANUAL

As you progress through the manual, you will find the following chapters:

2. INSTALLATION

Setting up your site for the terminal, the power requirements, unpacking and checking the terminal, setting switches to take advantage of the options available, configuring the terminal for your computer system and printer.

3. OPERATION

Turning on the terminal, a description of the keyboard and functions of the keys, using tabs, editing, sending data to the computer and the printer.

4. PROGRAMMING

Controlling the terminal through commands from your computer system: programming special functions, setting visual attributes, monitoring the program, loading and reading the cursor position, adding custom RAM and ROM, disabling the keyboard and printer.

5. TROUBLESHOOTING AND SERVICE

Periodic cleaning and inspection of the terminal, troubleshooting simple problems (using a table of symptoms, possible causes, and solutions), using self-test, service under warranty.

GLOSSARY

Explanation of terms commonly used in this manual.

APPENDICES

Specifications and reference tables.

INDEX

References to main subsections by subject.

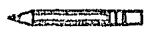
OPERATOR'S QUICK REFERENCE GUIDE

Lists all control and escape commands.

Each section of the manual is numbered. To find a topic later, look in the index and find the appropriate section.

As you read the manual, you will notice some special symbols at the left margin of the text. These symbols call your attention to information of special importance. The symbols used are:

General note giving information to every operator.



Programming note giving information of special significance to the programmer.

Warning giving information concerning the safety of the operator or possible loss of data. **When you see this note, STOP and read the note before proceeding!**

1.3 PROTECT YOURSELF!

When you install or test the terminal, observe standard safety precautions (as you would with any electrical or electronic equipment). Only qualified service personnel should open the terminal housing. Disconnect all power before performing any inspection or maintenance.

Beyond the normal precautions, you should be aware of two additional conditions:

1. If the CRT tube breaks, always wear heavy rubber gloves or use tongs to pick up the broken CRT fragments since the coating on the inside of the tube is poisonous.

2. Even after the power is turned off, charges are retained by the CRT and capacitors. Always discharge them to ground before touching them. Never reach into the terminal enclosure unless someone capable of giving aid is present.

2. INSTALLATION

2.1 INTRODUCTION

This chapter will tell you how to unpack and check your terminal for damage, check power and site requirements, and set the power and interface configurations. A brief checklist at the end will make sure you did not skip any part of the installation process.

Once your terminal is installed, you will be ready to operate the terminal. You will probably not need to refer to this chapter again *unless* you move the terminal, reship it, or use it with another computer system.

As you start the installation, you will want to have some information about your computer system and its configuration requirements.

2.2 UNPACKING AND INSPECTING THE TERMINAL

2.2.1 Shipping Damage Inspection

After the terminal is delivered to you, inspect the shipping container as well as the terminal (inside and out) for damage before taking it to your installation site. You should inspect the container for obvious damage before accepting delivery of the terminal. If damage is found, note it on the waybill and require the delivery agent to sign the waybill. Notify the transfer company immediately and submit a damage report to the carrier, your dealer, and to TeleVideo. If no exterior damage is found, unpack the terminal and inspect it for hidden damage.

2.2.2 Unpacking the Terminal

Carefully unpack the terminal from the shipping container. Avoid using sharp instruments to open the container. Save the packing container and material for possible use in reshipping the terminal.

2.2.3 Inspecting the Terminal

After you unpack the terminal, inspect it thoroughly for hidden damage and loose components or fittings. The inspection checklist is as follows:

1. Remove the terminal cover by removing the screws underneath the front bottom of the keyboard. Lift up the cover carefully.

The terminal will now be top heavy and will have a tendency to fall over backwards. Be sure there is sufficient table room.

2. Inspect the keyboard and display cabinet interior for shipping damage.

3. Examine cable harnesses for stress, loose or broken wires, or broken cable ties.

4. Examine all internally mounted components for loose or missing hardware.

5. Tighten all loose hardware.

6. Clean loose debris from the cabinet interior.

7. Replace the cover. Do not overtighten the screws.

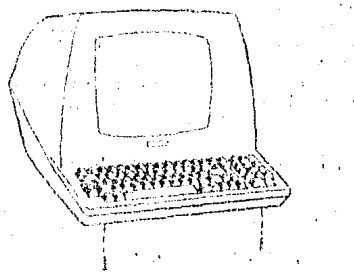


Figure 2-1 Location of Screws in the Terminal Cover

2.2.4 Reporting Damage

If hidden damage is found, immediately notify the transfer company of the damage. Save all packing materials for the transfer company's inspection, file a damage report with the carrier, and notify your dealer and TeleVideo of the damage. Since terms of sale for the terminal are FOB TeleVideo, Sunnyvale, California, TeleVideo is not responsible for any damage which occurred during shipment and will not repair this damage under warranty. All repairs for shipping damage are billable. Prompt notification of damage will ensure claim validity and expedite payment for necessary repairs by the transfer company or its insurance agent.

2.2.5 Reshipping the Terminal

Should you need to reship the terminal, follow these procedures:

1. Remove the two screws on the bottom front of the terminal and lift off the cover.
2. Check the integrity of the cabling and security of internal mounting hardware.
3. Replace cover, being careful not to overtighten the screws.
4. Repack the terminal in the original TeleVideo shipping container or other suitable materials.

2.3 PREPARING THE SITE

Before you proceed with the actual installation, make sure you are ready with the proper power and a large enough table.

2.3.1 Power Requirements

- 115 VAC 60 Hertz at 0.5 amp
  OR
  230 VAC 50 Hertz at 0.25 amp
- 55 watts
- NEMA standard 5-15R, 3-prong receptacle (US only)

2.3.2 Physical Requirements

- Flat, level area
- Surface dimensions: 13¼ inches (33.66 cm) high
  16⅛ inches (40.96 cm) wide
  20⅛ inches (50.96 cm) deep
- Recommended ventilation clearance is 4 inches (10.2 cm) on all sides. Refer to Figure 2-2.

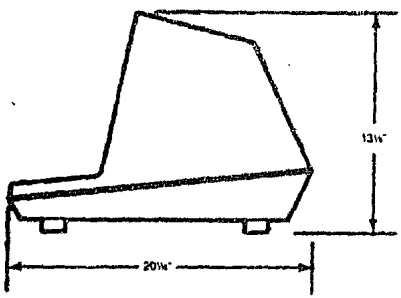


Figure 2-2 Dimensions

2.4 INSTALLATION

The actual installation and set-up consists of only three steps:

1. Configuring the terminal for either 115 or 230 VAC operation.
2. Connecting the terminal to the computer or a modem (and to a printer, if used).
3. Configuring the terminal by setting switches and installing jumper options.

2.4.1 Power Configuration

Depending on your location, the terminal can be configured to operate with either 115 VAC (United States) or 230 VAC (international).

115 VAC Configuration—Keep the three-prong plug which is provided with the terminal and make sure your outlet is grounded. If an adapter is used, ground with a pigtail.

230 VAC Configuration—If you are located outside the United States and use 230 VAC power, cut off the US-style three-prong plug provided and install a connector compatible with your local power receptacles. The power cord wires are color-coded as follows:

- Green Earth ground
- Black Primary power (hot)
- White Primary power return (neutral)

Set the power select switch (located underneath the terminal) to either 115 or 230 V (Fig. 2-3). You will set Hertz to match your power frequency when you set switch S2.

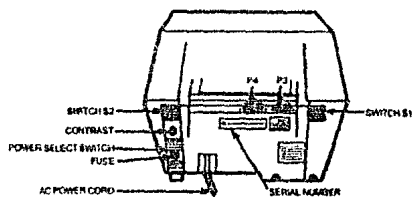


Figure 2-3 Rear Panel

2.4.2 Connecting the Terminal to a Computer System or Modem

You can connect the terminal directly to your computer system or use a modem. Table 2-1 points out pin connections which may be used.

The interface connection to the computer system (main) port is P3, located on the rear of the terminal. The connector configuration of P3 is given in Table 2–1.

Table 2–1
P3 (Computer Interface)
Pin Connections

Pin No.	Signal Name[1]
1	Frame Ground
2	Transmit Data Output
3	Receive Data Input
4	Request To Send Output
5	Clear To Send Input
6	Data Set Ready Input (opt.)
7	Signal Ground
8	Carrier Detect Input
12	Current Loop +, Receive
13	Current Loop –, Transmit
20	Data Terminal Ready Output
24	Current Loop –, Receive
25	Current Loop +, Transmit

Notes
1. Reference EIA Standard RS232 for Signal Definitions

2.4.3 Connecting the Terminal to a Printer

Your terminal can be connected to an auxiliary serial printer to make a permanent hard copy of data displayed on the screen. The terminal's serial printer interface allows the terminal to be used with most RS232-compatible serial printers currently available on the market, including both character-by-character and buffered printers. The serial printer interface is a 25-pin connector, P4, located on the rear of the terminal. Table 2–2 defines the serial printer interface pin connections.

Table 2–2
P4 (Serial Printer Interface)
Pin Connections

Pin No.	Signal Name
1	Protect Ground
3	Transmit Data
6	Data Set Ready
7	Signal Ground
20	Data Terminal Ready

2.4.4 Configuring the Terminal for the Computer and Printer

Two switches (located on the rear of the terminal and shown in Fig. 2–2) allow you to configure the terminal to operate according to the requirements of your computer system and printer. This section describes these switch settings.

The optional conditions controlled by these switches are:

Baud Rates

You can select any of 15 baud rates according to the requirements of your computer system.

Character Sets

You can select English, French, German, or Spanish character sets.

Hertz

You can set the Hertz switch to match your powerline frequency.

Parity, Stop Bits, and Word Structure

You can set the parity, number of stop bits, and number of bits in the word structure to match the requirements of your computer system.

Signals

You can connect/disconnect Data Set Ready, Data Carrier Detect, and Data Terminal Ready.

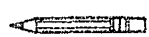
Transmission

You can select half or full duplex (conversational mode) or block mode.

Set your printer's baud rate to match the computer's baud rate. (This rate is also used for switch S1 dipswitches 1 through 4 as described in Table 2–3b.)

Whenever you change any switches, press BREAK twice while holding down the SHIFT key. This allows the software to scan all new switch positions.

2.4.4.1 Character Sets—You can select any of four possible character sets. The standard set is English. To select another character set, refer to Table 2–5.



Character sets are resident in the character generator. You must reprogram the terminal system ROM for the particular keyboard layout desired.

Table 2–3a
External Switch Settings

Switch	Dipswitch	Position: Open (Up)	Position: Closed (Down)	Function
S1	1,2,3,4			Computer baud rate; see Table 2–3b
	5	x		Seven-bit word structure
			x	Eight-bit word structure
	6	x		Send parity
			x	No parity
	7	x		Even parity
			x	Odd parity
	8	x		Two stop bits
			x	One stop bit
	9	x		Autowrap on
			x	Autowrap off
	10	x		Performs CR/LF upon receipt of CR
			x	Performs CR upon receipt of CR
S2	1	x		Block
			x	Conversational
	2	x		Half duplex
			x	Full duplex
	3	x		50 Hertz[1]
			x	60 Hertz[1]
	4	x		Local edit
			x	Duplex edit
	5	x		Underline cursor
			x	Block cursor
	6	x		Cursor down key as in 912/920
			x	Cursor down key as in 925/950
	7	x		Green on black
			x	Black on green
	8	x		Data Set Ready disconnected
			x	Data Set Ready connected
	9	x		Data Carrier Detect disconnected
			x	Data Carrier Detect connected
	10	x		Data Terminal Ready disconnected
			x	Data Terminal Ready connected

NOTES
[1]Set to match powerline frequency to avoid screen flicker.

Table 2–3b
Switch Settings for Computer Band Rates

Switch	Position				Band Rate Setting
S1	1	2	3	4	
	D	D	D	D	9600
	D	D	D	U	50
	D	D	U	D	75
	D	D	U	U	110
	D	U	D	D	135
	D	U	D	U	150
	D	U	U	D	300
	D	U	U	U	600
	U	D	D	D	1200
	U	D	D	U	1800
	U	D	U	D	2400
	U	D	U	U	3600
	U	U	D	D	4800
	U	U	D	U	7200
	U	U	U	D	9600
	U	U	U	U	19200

Legend: U = Up
D = Down

Table 2–4
Switch Settings of S1 for Common Word Structures
(Data Bits, Stop Bits, and Parity)

S	Position 6	7	8	Data Bits	Parity	Stop Bits
U	D	X	D	7	None	1
U	D	X	U	7	None	2
U	U	D	D	7	Odd	1
U	U	D	U	7	Odd	2
U	U	U	D	7	Even	1
U	U	U	U	7	Even	2
D	D	X	D	8	None	1
D	D	X	U	8	None	2
D	U	D	D	8	Odd	1
D	U	U	D	8	Even	1

Legend: U = Up
D = Down
X = Either up or down

If word structure, parity, or stop bits are set incorrectly, the terminal will only display @ signs when it is turned on.

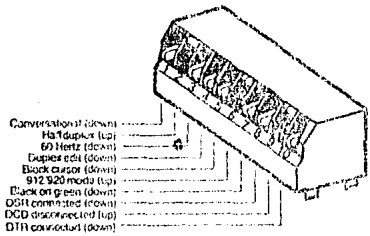


Figure 2–4 Switch Setting Example

Table 2–5
Character Set Jumper Options

French	Cut trace between E4 and E5. Ensure that E6 and E7 are connected.
German	Cut trace between E6 and E7. Ensure that E4 and E5 are connected.
Spanish	Cut trace between E6 and E7 and E4 and E5.

2.4.4.2 Video Display— You can set the terminal display to be green on black (normal) or black on green (reverse) and cause the cursor to be an underline or a block, displayed as steady or blinking. (See Table 2–3a.)

2.4.4.3 Composite Video Jumper Option— To drive a monitor in addition to or other than the terminal monitor, modify the logic board (Fig. 2–5) by adding an Amphenol BNC connector (Part 227169–5) to the rear of the terminal case. (See Fig. 2–6 for recommended placement.)

Connect the center lead of the BNC connector to P2 pin 6 and the BNC ground lead to P2 pin 3. Cut the trace between E10 and E11. Install a jumper between E12 and E13.

The monitor should not be more than 10 feet from the terminal.

2.4.4.4 Current Loop Option—Installing an optional current-loop board enables you to operate the terminal up to 1,000 feet from your computer system.

Before you install the optional current loop board, inspect it for possible shipping damage (i.e., bent pins, cracked board, etc.).

Make cuts and jumpers on the current loop board (shown in Figure 2–7) according to the desired configuration. (Possible configurations are described in Table 2–6)

Remove the terminal cover by removing the two screws in the bottom of the case. (Figure 2–1 shows the location of these screws.)

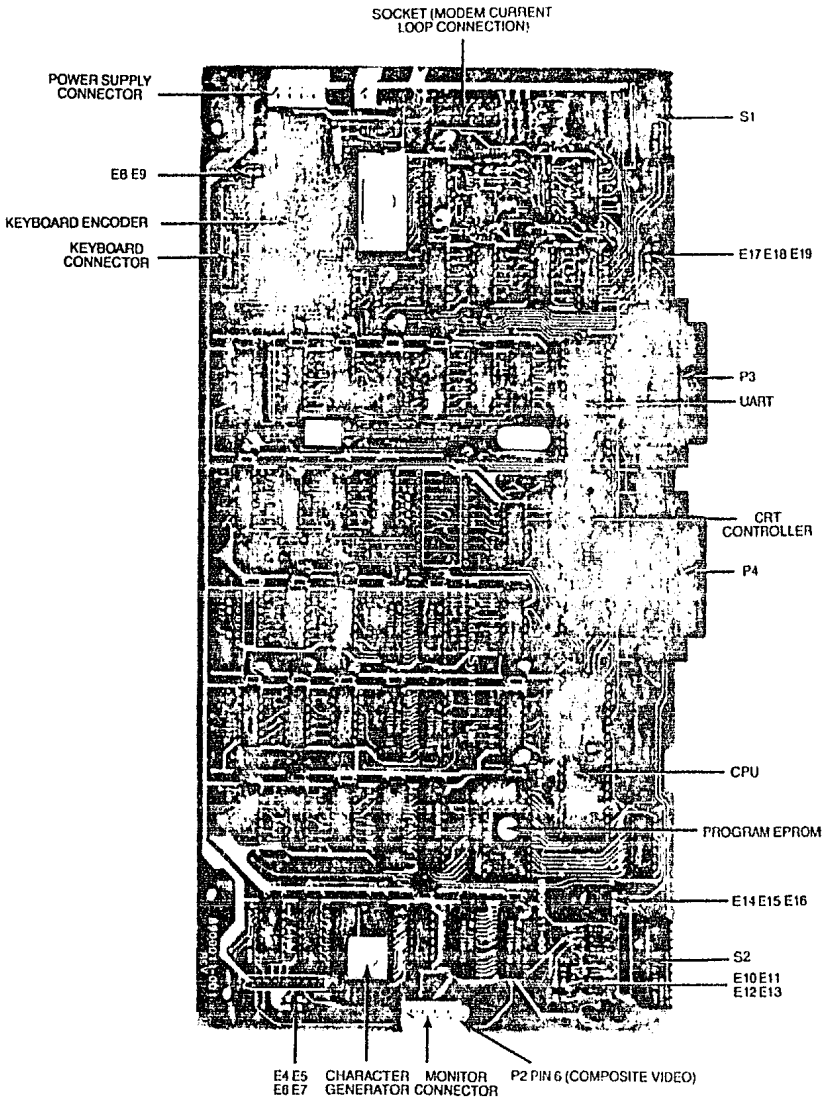


Figure 2–5 Logic Board

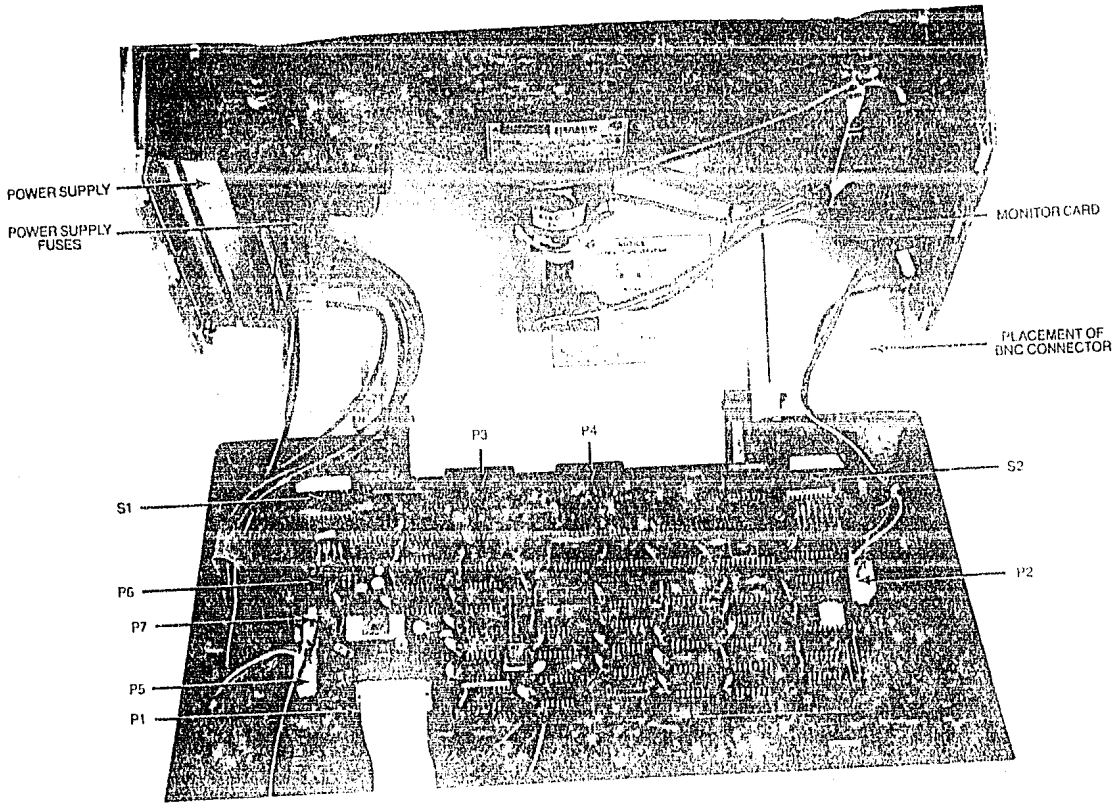


Figure 2-6 Interior of Terminal

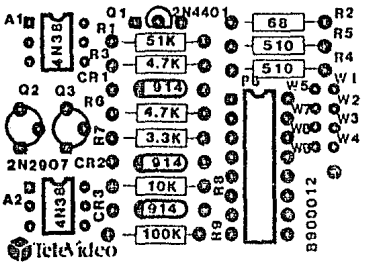


Figure 2–7 Current Loop Board

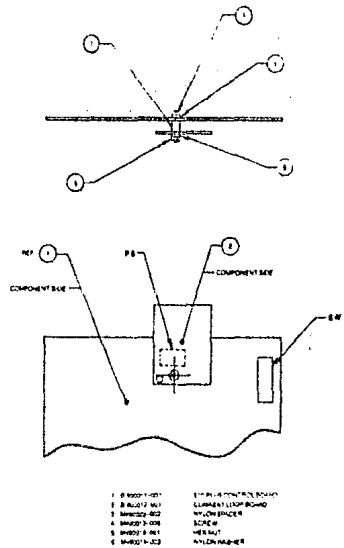


Figure 2–8 Current Loop Board in Relation to Logic Board

Table 2–6
Possible Current Loop Configurations

Mode[1]	Transmit/ Receive	Cut	Jumper	P3 Pin No.
Full Duplex				
Active	Transmit	W2 to W3	W1 to W3 W3 to W4	Pin 25 – Pin 13 +
	Receive	W6 to W7	W5 to W6 W7 to W8	Pin 12 – Pin 24 +
Passive	Transmit	None	None	Pin 25 + Pin 13 –
	Receive	None	None	Pin 12 + Pin 24 –
Half Duplex				
Active	Transmit and Receive	W3 to W4	W1 to W2 P3–13 to P3–12	Pin 25 – Pin 24 +
Passive	Transmit and Receive	None	P3–13 to P3–12	Pin 25 + Pin 24 –

Notes
[1]Active = terminal supplies the current source; passive = computer supplies the current source.

Insert the current loop board into the 16-pin socket located on the terminal control board. Refer to Figure 2–3 for socket position; Figure 2–8 shows where to insert the screw, spacer, washer, and nut.

Replace the terminal cover using the screws you removed. Be careful not to overtighten the screws.

Connect your computer to the terminal, using a cable with pins as shown in the column labeled **P3 Pin No.** in Table 2–6.

2.4.4.5 Additional Modifications—Table 2–7 describes other jumper options which will change the terminal's interfaces.

Table 2–7
RS232C Terminal Interface Jumper Options

1. **Standard Set Up (no modifications to printed circuit board)**
   a. Data Carrier Detect (DCD), P3 pin 8, is used to monitor status of an external modem.
   b. Data Terminal Ready (DTR) output is sent to the computer when DTR from printer port is received.

2. **Jumper Options**
   a. Data Set Ready (DSR), P3 pin 6, can be used to monitor the external modem rather than DCD.
      To install: Cut the trace between E14 and E15. Add a jumper between E15 and E16.
   b. Use Request to Send (RTS) to send DTR to computer rather than DTR from printer.
      To install: Cut the trace between E18 and E19. Add a jumper between E17 and E19.
      (Refer to Figure 2–5)

2.5 INSTALLATION CHECKLIST

Before you proceed to the next chapter and turn on the terminal, check to be sure you installed the terminal correctly.

1. Did you install the correct power plug for your wall outlet?
2. Did you set the power selector switch to match your power requirements?
3. Is the main interface cable to the computer system properly wired and plugged in?
4. If you are using a printer, did you plug in the printer interface connector?
5. Did you set the switches for the correct
   - Baud rate (both for terminal and printer)?
   - Stop bits?
   - Word structure?
   - Parity?
6. Did you set switches for
   - 50 or 60 Hertz (to match your powerline/frequency requirements)?
   - Full or half duplex?

If the answers are YES, then you are ready to proceed with actually using the terminal.

Enter here the serial number, date received, and switch settings. This will expedite any technical conversations about your terminal.

Serial Number_____

Date Received_____

Switch Settings Used:
(Enter U or D for Up or Down)

	U/D		U/D
S1	1 __	S2	1 __
	2 __		2 __
	3 __		3 __
	4 __		4 __
	5 __		5 __
	6 __		6 __
	7 __		7 __
	8 __		8 __
	9 __		9 __
	10 __		10 __

3. OPERATION

3.1 INTRODUCTION

This chapter will lead you step-by-step through the operation of the terminal. Even if you have never used a computer terminal before, you will be able to use the terminal easily if you read this chapter carefully. If you are a programmer, you will want to continue on to Chapter 4, which covers additional information for programming a computer to interface with your terminal.

In this chapter you will learn about:

- Turning on and adjusting the terminal's display screen
- Using the various keys on the keyboard
- Directing data to the computer system and the printer through send commands
- Setting tabs
- Communicating with your computer system

3.2 TURNING ON THE TERMINAL

Turn on the terminal as follows:

1. Make sure the ON/OFF switch at the back of the terminal (Figure 2-3) is OFF.
2. Plug the terminal cord into a grounded outlet (115 VAC in United States).
3. Push the end of the rocker power switch marked with a white dot. The terminal should beep within one second, indicating that power is on and the CPU has initialized the terminal. After another 10 to 15 seconds, the cursor should appear in the upper left corner of the screen (home).
4. If the cursor does not appear at the home position, press the home key on the keyboard. If the cursor still does not appear, check the contrast control at the rear of the terminal (Figure 2-3).
5. Adjust the contrast control for the desired screen intensity.
6. Follow the sign-on protocol required by your computer system.
7. Refer to Chapter 5 if the installation does not proceed smoothly.

3.3 KEYBOARD CONTROLS

In addition to standard alphanumeric typewriter keys, your terminal has several keys which perform special operations. These special keys can be used in conjunction with your computer to allow:

- Modifying action on other keys
- Editing
- Entering preprogrammed data

Each key on the keyboard is actually a switch. Sometimes two keys can be used with any alpha or numeric keys to provide a totally different message to the computer. When used together, these keys control the generation of data sent to the computer system and the receipt and printing of information.

3.3.1 Keyboard Layout

Figure 3-1 illustrates the keyboard layout. The character keys highlighted in Figure 3-1a include all alphabetic characters (a through z), numbers (0 through 9), punctuation marks, and mathematical symbols.

3.3.2 Key Functions

Table 3-1 summarizes the function of the special keys which are highlighted in Figure 3-1b. Many of these keys are also listed in the Operator's Quick Reference Guide on the inside back cover.

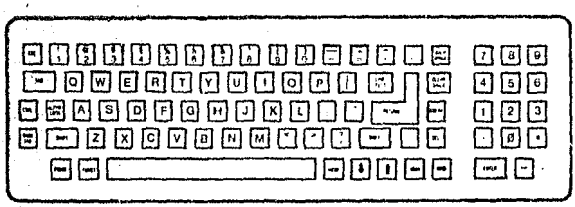


Figure 3-1a Keyboard Layout

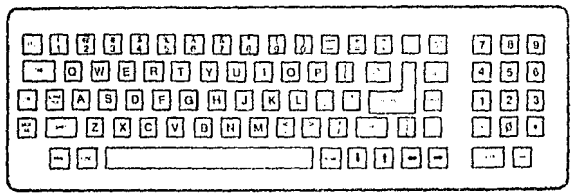


Figure 3–1b Keyboard Layout

Table 3–1
Function of Keys

Key Name	Transmitted? (Yes/No)	Repeat Action? (Yes/No)	Description
Space Bar	Y	Y	Causes a blank space to appear on the display and transmits an ASCII space code (20 Hex).
SHIFT	N	N	Selects upper character inscribed on a key, changes operation of most special keys, and capitalizes alpha characters.
ALPHA LOCK	N	N	Locks the SHIFT keys so that all alpha keys transmit codes for upper-case characters. The key is pressed to lock and pressed again to release.
TAB	Y	Y	(CTRL/I)—TAB moves the cursor forward to typewriter tabs (Protect mode off) or to the start of the next unprotected field (Protect mode on).
BACK TAB	Y	Y	(ESC I)—Moves cursor backward to typewriter tabs (Protect mode off) or to the previous start of an unprotected field (Protect mode on).
CTRL (Control)	N	N	Generates normally-nondisplayed ASCII control codes when used in conjunction with another key. The CTRL key combinations are used for special action by the terminal and/or the application program in the computer. ☼ The CTRL key is always used simultaneously with the other character in the command; i.e., the CTRL key is pressed first and held down while the other key is pressed. (It is similar in action to the SHIFT key.) **The commands which require simultaneous depression of two keys are indicated by a slash separating the two key names.**
ESC (Escape)	Y	N	The ESC key sends an ASCII code for escape to the display processor. The key is generally used to momentarily leave (escape) an application program in order to use a special feature or function. Another function of the ESC key causes the next control character entered to be displayed on the screen. This facilitates putting control characters on the screen without going into monitor mode.

Table 3–1
Function of Keys

Key Name	Transmitted? (Yes/No)	Repeat Action? (Yes/No)	Description
			
			The ESC key is used in conjunction with one alphanumeric character in the command sequence; i.e., **the ESC key is pressed and released before the second key is pressed.**
			Although escape sequences appear here with a space before the alphanumeric character, this space is *not* to be entered as part of the sequence. It is included only for the sake of clarity.
RETURN/ ENTER	Y	N	(CTRL/M)—The RETURN and ENTER keys perform the same function. They send the ASCII code for a carriage return (CR) to the display or computer. The communication mode used causes the terminal to transmit a CR (or CR/LF) to the computer and/or the cursor to be moved to the first unprotected position.
			If the entire current line is protected, the code moves the cursor to the next unprotected position on the page.
			
			The terminal features an auto wraparound function which eliminates the need to manually enter a carriage return and a linefeed at the end of each 80-character line.
HOME	Y/N	N	(CTRL/ ^)—Moves cursor to first unprotected character position on the page.
LINEFEED	Y	Y	(CTRL/J)—The LINEFEED key sends an ASCII code (OAH) for a linefeed (LF) to the computer. The code causes the terminal to transmit an LF code to the computer and the cursor to be moved down one line on the screen in half duplex, or echoed by the computer in full duplex.
BACKSPACE ←	Y/N	Y	(CTRL/H)—Moves cursor one character to the left.
↑	Y/N	Y	(CTRL/K)—Moves cursor up one line.
↓	Y/N	Y	(CTRL/J or CTRL/V)—Moves cursor down one line. If the cursor is on the bottom line of the screen, the code has no effect. The code transmitted is determined by setting of switch S2.
→	Y/N	Y	(CTRL/L)—Moves cursor one character to the right.
DEL (Delete)	Y	Y	The DEL key sends an ASCII DEL character to the computer. This is usually interpreted by the computer as a character erase code.
BREAK	Y	N	Transmits a 250-millisecond break pulse to the computer.
Clear Space	Y/N	Y	Replaces all unprotected characters on the page with spaces. When pressed the same time as SHIFT (ESC.*), it clears the entire page to nulls and turns off protect and half intensity modes.
PRINT	N	N	The PRINT key toggles the extension print mode on or off.
"FUNCT"	Y	N	The FUNCT key transmits a user-selected character bracketed by CTRL/A (SOH) and CR.

3.3.3 Cursor Control

The lighted rectangular block which appears on the screen indicates the entry spot for the following characters to be typed. It is called a "cursor." During typing, the cursor moves from left to right. As it reaches the end of a line, it "wraps around" to the beginning of the next line. If you place the cursor over a character which you have already typed, the character within the cursor will be changed into a reverse image within the cursor. (If the characters have been green on a black background, the cursor will appear as a green rectangle around a black character.)

The movement of the cursor is easy to control. To move the cursor, press one of the cursor control keys marked with an arrow. The cursor will move in the direction of the arrow until you release the key. To return the cursor quickly to the top left position on the screen, press HOME. The cursor will now be in column one, line one.

If you are in local edit mode, cursor movement will not be transmitted to the computer.

The cursor display may appear any one of five ways. See 4.11.

3.4 BASIC OPERATIONS

This section describes various options available to you as you use the terminal:

- Editing data
- Tab controls
- Communicating with your computer system
- Printing

3.4.1 Tab Controls

You can set regular typewriter-style tabs or (if you are using protect mode as described in 4.6) field tabs. Refer to 4.7 for complete instructions on setting, using, and clearing tabs.

3.4.2 Editing

Should you need to change text on the screen, you can delete a line (either partially or completely) or the display (either partially or completely). This will give you space to enter the correct text. Deletions will start with the column position of the cursor. The terminal can also modify screen data using character insert/delete and line insert/delete. These both start at the character position also. Commands for editing are described in 4.8.

You can select one of three transmission modes by switch settings or using escape sequences. The three modes available are block, half duplex (conversation), and full duplex (conversation). The communications flow caused by these modes are illustrated in Figure 3-2.

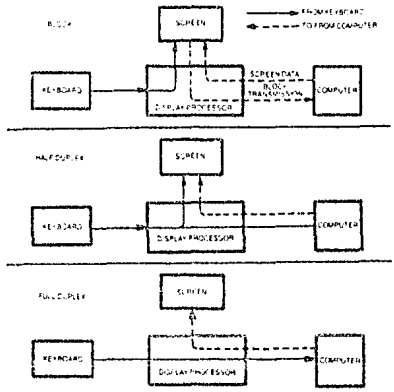


Figure 3-2 Communication Modes

3.4.2.1 Block Mode—Operating in the block mode generally consists of entering or changing text locally. In this mode, the terminal sends the results to the screen. When you are satisfied with the results of the data entry or change, you can enter an escape sequence to send the data to the computer. Block mode allows you to make all corrections before transmission.

To enter block mode, enter

ESC B

If switch S2 is set for block mode, the terminal will revert to conversation mode when an ESC C is received or entered.

3.4.2.2 Conversation Mode—In this mode, two-way transmission occurs continuously between the screen and the computer. You can either send the information to the computer *at the same time it is displayed on the screen*, or you can send it to the computer and the computer will echo back the information on the screen. (The time needed to echo back the information is so short it will *seem* to happen simultaneously.) Regardless of when you send data, the terminal can always receive information from the computer. When the information is displayed simultaneously with the transmission, it is called "half duplex." When the information is sent first to the computer and echoed back to the terminal, it is "full duplex." Refer to Figure 3-2 for a diagram of the information flow.

To enter conversation mode, enter

ESC C

The terminal is conversational in either half or full duplex modes.

Half Duplex Mode

The half duplex mode sends keyboard entries to the screen and to the computer at the same time.

Full Duplex Mode

The full duplex mode sends keyboard entries to the computer only. If the computer is programmed to act upon a code received from a keyboard entry, it may echo the result back to the terminal. For example, if the "A" is pressed on the keyboard, the computer will probably send the "A" back to the screen.

3.4.3 Sending Data to the Printer

When the printer is printing on a continuous basis, it is an extension of the line from the computer to the terminal—this mode of printing is thus called **extension** or **copy all.**

To start extension printing, either press the PRINT key or enter

ESC *α*

To stop printing, press PRINT again or enter

ESC A

You can also send information from the computer to the printer **without** displaying it on the screen. This is called **transparent** mode.

Section 4.15 describes commands used for transparent print.

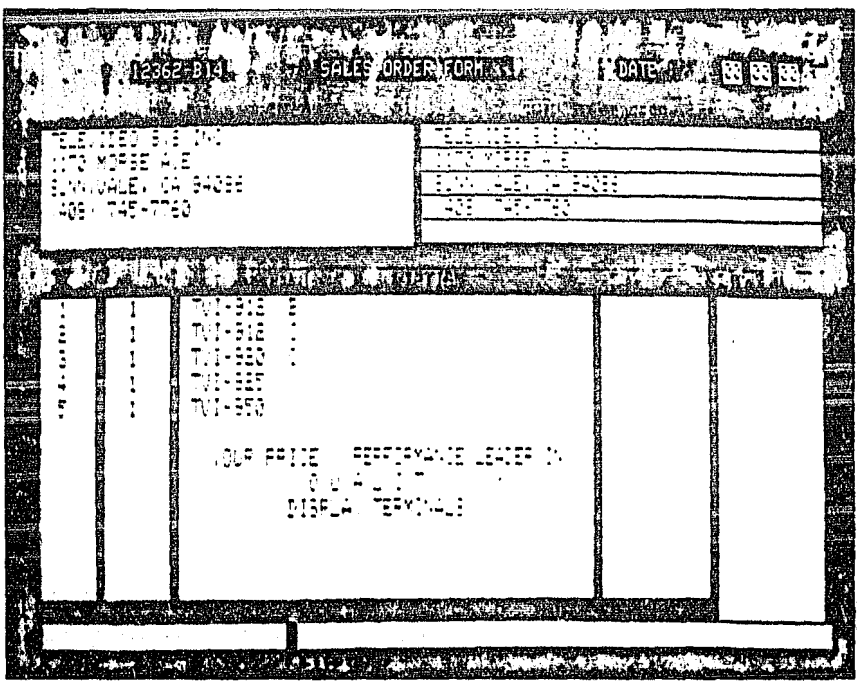


Figure 3–3 Screen Display

4. PROGRAMMING

4.1 INTRODUCTION

Your computer program can completely control your terminal by transferring the appropriate ASCII codes. This chapter tells you how to translate keyboard functions into remote control functions.

Unless otherwise specified in the text, all control code sequences are transmitted to the terminal to elicit the response associated with the code.

4.2 MONITORING CONTROL COMMANDS

You can monitor control commands in several ways:

- Activate the monitor mode without transmitting the monitor mode code itself to the computer
- Transmit the monitor mode code to the computer

To enable monitor mode without transmitting that code to the computer, enter

CTRL/1

To terminate this mode, enter

CTRL/2

To enable monitor mode via the computer, enter

ESC U

This must be echoed by the computer or monitor mode will not be activated.

To terminate the display of the control commands, enter either

ESC u or ESC X

Figure 4–1 illustrates the monitor mode codes.

4.3 FUNCT KEY

Using the FUNCT (function) key in combination with any key enables you to quickly transmit a three-character sequence of commands.

To enter a function command, press the FUNCT key and *at the same time* press a key. The first code which is transmitted will always be

SOH (Control A)

The second code will be the ASCII code of the depressed key. The third code will always be a CR (Table 2–3).

Program your computer's input/output string routine to catch the entire string and then process it (unless you are using an interrupt-driven computer, in which case you do not need to worry about data being lost).

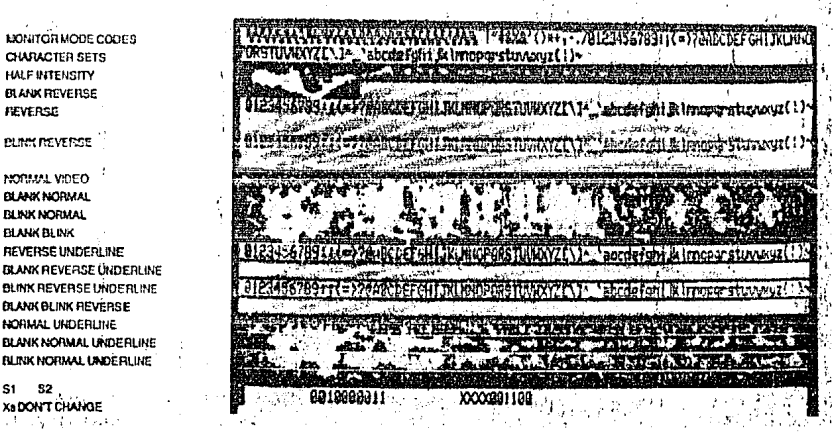


Figure 4–1 Video Attributes

4.4 ADDRESSING AND READING THE CURSOR

The computer can tell the terminal where to position the cursor with a four-character escape sequence. This is called **addressing** or **loading** the cursor.

4.4.1 Addressing the Cursor

To address the cursor, enter

ESC =

Then enter two more characters to represent the absolute row or line and column where the cursor will rest. Using Table 4-1, find the ASCII code representing the desired row. Note that the line number can not be greater than 24. Enter the appropriate ASCII code. Next find the ASCII code corresponding to the desired column position (1 through 80 possible) and enter that code. For example, if you want to program the cursor to go to Row 9, Column 50, enter

ESC = (Q

4.4.2 Reading the Cursor

The computer can also **read** the cursor's row and column position. To read the cursor's position, enter

ESC ?

Following the cursor coordinates (row and column), the terminal will transmit a CR.

4.5 VISUAL ATTRIBUTES

You can define the appearance of each line on the screen (a whole line or only part of a line). Each line must be defined separately (except half intensity). Several attributes can be used on each line (i.e., blinking set at the beginning followed by underlining set later in the line).

Reverse Video Changes background of screen on that line to the reverse of that which appears on power ON. If screen is normally black with green characters, this line will now be green with black characters.[1]

Table 4-1
Cursor Coordinates

CURSOR POSITIONING

POSITION R or C[1]	ASCII CODE Transmitted	POSITION C	ASCII CODE Transmitted	POSITION C	ASCII CODE Transmitted
1	Space	33	@	65	`
2	!	34	A	66	a
3	"	35	B	67	b
4	#	36	C	68	c
5	$	37	D	69	d
6	%	38	E	70	e
7	&	39	F	71	f
8	'	40	G	72	g
9	(	41	H	73	h
10	)	42	I	74	i
11	*	43	J	75	j
12	+	44	K	76	k
13	,	45	L	77	l
14	-	46	M	78	m
15	.	47	N	79	n
16	/	48	O	80	o
17	0	49	P	81	p
18	1	50	Q	82	q
19	2	51	R	83	r
20	3	52	S	84	s
21	4	53	T	85	t
22	5	54	U	86	u
23	6	55	V	87	v
24	7	56	W	88	w
25	8	57	X	89	x
26	9	58	Y	90	y
27	:	59	Z	91	z
28	;	60	[	92	{
29	<	61	\	93	|
30	=	62	]	94	}
31	>	63	^	95	~
32	?	64	_	96	DEL/RUB

Notes
1. Value of R can't be greater than 24.

Half Intensity Changes intensity to half of normal on a character-by-character basis.

Half intensity differs from other visual attributes in that once it is set, it affects all characters entered (regardless of cursor position) until it is turned off.

Underline Creates a solid line below all characters on the line (including the line created by the underscore key).[1]

Blink Causes all characters on the line to blink.[1]

Blank: All data entered on the line will be invisible to you but will print out and be transmitted to the computer. (A typical application might be payroll information.)[1]

Note:
1. Attribute starts with cursor position and continues until another attribute or end of line is encountered.

Figure 4-1 illustrates the visual attributes.

4.5.1 Setting

To set a visual attribute, place the cursor one position *before* you want the attribute to start. Attributes occupy a character position. If you want the whole line changed, place the cursor at column one before entering the attribute command (ESC Gn).

Table 4-2
Escape Sequences for Visual Attributes

Description	Escape Sequence
Normal video (green on black)	ESC G0
Blank (invisible normal video)	ESC G1
Blink	ESC G2
Blank (invisible blink)	ESC G3
Reverse video (black on green)	ESC G4
Blank (invisible reverse video)	ESC G5
Reverse and blink	ESC G6
Blank (invisible reverse blink)	ESC G7
Underline	ESC G8
Blank (invisible underline)	ESC G9
Underline and blink	ESC G:
Blank (invisible blink underline)	ESC G;
Underline and reverse	ESC G <
Blank (invisible underline reverse)	ESC G =
Underline and reverse and blink	ESC G >
Blank (invisible underline reverse blink)	ESC G?

4.6 PROTECT MODE

4.6.1 Application

Using protect mode during the creation of a page allows you to protect designated areas of the page from future change by the operator and control the transmission of those areas.

Using protect mode is actually a two-step process: input and protection.

A typical application would be the creation of a form, leaving blank spaces for later entry or variable information. Were the form headings not protected by protect mode, they would be vulnerable to change or accidental deletion as the form was being filled in.

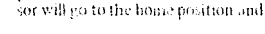
4.6.2 Effect

Protected areas appear on the screen at half the regular intensity. The cursor is not able to enter a field which has been protected, but will instead advance across that area to the first unprotected field when the operator enters → or ←. Linefeed, ↑, or ↓ may, however, move the cursor to the protected field. The screen does not scroll up in protect mode. If the whole screen is protected, the cursor will go to the home position and will not move.

Protect mode affects cursor action during tabulating, editing, sending, and printing.

4.6.3 Procedure

4.6.3.1 *Input* – Individual areas (fields) which will be given blanket protection from later change are created using protected writing mode.



Information must be input using this procedure if it is to be protected later.

To start protected writing, position the cursor where the first protected character is to be located.

Enter

ESC)

This turns on protected writing mode (also called **half intensity**). Until the mode is reset, each character entered is displayed at half intensity.

Enter the information for that area of the screen.

Proofread the entry and change if necessary.

End data entry in that area by entering

ESC (

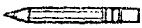
This turns off the protected writing mode (half intensity).

Move the cursor to the next area to be protected and repeat.

4.6.3.2 **Protection**—When all areas to be protected have been entered correctly, the whole screen is ready to be protected from change (protect mode on). Once this protection is given, the cursor will not be able to enter those areas unless the protection is removed.

To start protect mode, enter

ESC &



The position of the cursor is irrelevant during this escape sequence.

Protect mode protects all visual attribute codes within the defined protected area from overwriting or erasure. All data within protected areas is also protected.

To remove protect mode (protect off), enter

ESC '

4.7 TAB PROGRAMMING

As briefly described in Chapter 3, the cursor may be moved on the screen to preset typewriter-style tabs or, if protect mode is on, to field tabs. This section describes how to set, use, and clear both types of tabs.

4.7.1 Setting a Tab

To set a tab, move the cursor to the column position where you want a tab. Enter

ESC 1

Be sure you enter a numeral one, not a lower case **L**.

When protect mode is on, this ESC 1 code generates a vertical column of half-intensity spaces from the cursor position down to the first write-protected character or to the end of the page, whichever is first.

When protect mode is off, the code sets a typewriter-style column tab.

4.7.2 Using Tabs

4.7.2.1 **Typewriter Tabs (Protected and Unprotected)**—When the protect mode is off, CTRL/I causes the cursor to advance to the next typewriter-style tab set. If no tabs are set, the code has no effect and the cursor will not move.

When the protect mode is on, the cursor is moved to the first unprotected character following the next protected field.

4.7.2.2 **Field Tabs (Protected Only)**—With protect mode on, ESC i causes the cursor to move to the first unprotected character following the next protected field.

With protect mode off, this code has no effect.

4.7.2.3 **Back Tab**—When protect mode is off, ESC I causes the cursor to go back to the previous tab position set. If no tabs are set or if the cursor is on the first tab position, this code moves the cursor to the first column on the line.

If protect mode is on, ESC I moves the cursor back to the start of the first preceding unprotected field. If no preceding positions exist, the cursor will not move.

If the cursor is at the first unprotected position on the page, the code has no effect. If no protected fields exist, home position is considered the start of an unprotected field.

4.7.3 Clearing Tabs

4.7.3.1 **Typewriter Tabs**—You can clear a typewriter tab by putting the cursor on the tab position you wish to clear and entering

ESC 2

This code has no effect when protect mode is on.

4.7.3.2 **All Tabs**—To clear all tabs, enter

ESC 3

The position of the cursor when this code is entered is not important.

4.8 EDITING CONTROLS

The editing control sequences and a description of their functions follow:

Use of the editing commands may result in the loss of data. Read the following explanations of the editing control functions carefully.

4.8.1 Edit Modes

The edit modes which are described in this section can be selected either with the switches on the rear of the terminal or with control codes.

There are two edit modes available: local edit and duplex edit.

4.8.1.1 **Local Edit Mode**—Operating in local edit mode enables you to change the text using the edit keys (CLEARSPACE, BACKSPACE, ↑, ↓, →, ←, TAB, HOME, and BACK TAB) without transmitting these keys or any changes caused by these keys to the computer. To enter local edit mode, enter

ESC k

All other keys will operate normally while local edit is on.

4.8.1.2 Duplex Edit Mode—To set the edit keys described in 4.8.1.1 to operate in the mode set for the alphanumeric keys, enter

ESC l (lower case "L")

For example, if the terminal is set for half-duplex operation, both the alphanumeric and edit keys will operate in half duplex mode.

4.8.2 Cursor Control

The cursor control key operation is described in 3.3.3. Escape and control sequences may be sent from the computer to perform the various cursor functions.

4.8.2.1 Cursor Control Codes—The cursor control codes and a description of their functions are described below.

Cursor Up. CTRL/K moves the cursor up one line.

Cursor Down. Depending on the switch settings, CTRL/V or CTRL/J moves the cursor down one line. If the cursor is on the bottom line of the screen and switch 2 dipswitch 6 is down (925/950 mode), the code has no effect.

Cursor Left. CTRL/H is the same as BACKSPACE; it moves the cursor left to the next unprotected position on the page. If the cursor is currently in the first column of the line, it will move to the last column of the preceding line. If the cursor is currently at the first unprotected position on the screen, the code has no effect.

Cursor Right. CTRL/L moves the cursor right one column. If the cursor is at column 80, it moves the cursor to the first column of the next line. With protect mode off, it causes a scroll if the cursor is at column 80 of the last line. With protect mode on and the cursor at the last unprotected position on the page, the cursor will move to the first unprotected position.

Carriage Return. CTRL/M moves the cursor left to column 1 of the current line. If protect mode is on, it moves the cursor to the first unprotected position of the next unprotected field.

Cursor HOME. CTRL/^ moves the cursor to the first unprotected character on the page.

New Line. CTRL/_ (underline) causes the terminal to perform a CR and a LF.

4.8.2.2 Linefeed—With protect mode off, CTRL/J or linefeed (LF) advances the cursor to the next line on the page. If the cursor is at the bottom of the screen, a LF causes a new line of data to appear at the bottom of the screen and results in the loss of the top line of data on the page (shifts the cursor down). The new line contains spaces.

If the cursor is at the bottom of the screen with protect mode on, LF moves the cursor to the top of the screen at the current column position. If that position is protected, it then moves the cursor to the next unprotected position.

4.8.3 Editing Commands

4.8.3.1 Character Insert—ESC Q causes the character at the cursor to move right one column and enters a space character at the cursor position. The character at column 80 is lost. If protect mode is on, this control will insert from the cursor position to the end of the line or to the first protected field.

4.8.3.2 Character Delete—ESC W deletes the character at the cursor position and moves all following characters left one position. At the end of the delete function, a space character is written into the last position on the line. If protect mode is on, character delete operates only from the cursor position to the end of the unprotected field or line.

4.8.3.3 Line Insert—With protect mode off, ESC E inserts a line consisting of spaces at the cursor position. This causes the cursor to move to the start of the new line and all following lines to move down one line, resulting in the loss of the last line on the screen. If protect mode is on, a line insert command has no effect.

4.8.3.4 Line Delete—When protect mode is off, ESC R deletes the line at the cursor position and all following lines move up one line. The cursor will move to column 1 of the line and spaces will be loaded into the last line of the screen. When protect mode is on, this code has no effect.

4.8.3.5 Erase to End of Line—ESC T erases all unprotected characters from the cursor to the end of the line (or field, if protect mode is on) and replaces them with spaces. If half intensity is on, half-intensity spaces will replace the erased characters.

4.8.3.6 Erase to End of Line with Nulls—ESC t erases all characters from the cursor position to the end of the line or the end of an unprotected field and replaces them with null characters.

4.8.3.7 Erase to End of Screen—ESC Y replaces unprotected characters from the cursor position to the end of the screen with spaces. If half intensity is on, erased characters will be replaced with half-intensity spaces.

4.8.3.8 Erase to End of Screen with Nulls—ESC y erases all unprotected characters from the cursor position to the end of the screen and replaces them with null characters.

4.9 CLEAR FUNCTION

The clear function is used in one of four ways to clear data from screen memory and/or computer memory.

4.9.1 Clear Unprotected to Nulls

ESC : clears all unprotected data on the screen to the null character.

4.9.2 Clear Unprotected to Spaces

ESC ; or ESC + or CTRL/Z clears all unprotected data on the screen to spaces. If half intensity is on, the screen will be cleared to half-intensity spaces.

4.9.3 Clear Screen to Half-Intensity Spaces

ESC , clears all unprotected data on the screen to half-intensity spaces.

4.9.4 Clear All Data to Nulls

ESC * clears all data on the screen to nulls and resets the half intensity and protect modes.

4.10 DISABLING AND ENABLING THE KEYBOARD

You can disable all keyboard functions by remote commands from the computer. Once the keyboard is disabled, it can *only* be enabled once again by another remote command from the computer.

If your computer system echoes all codes, the keyboard may be accidentally disabled.

To disable the keyboard remotely, enter

ESC #

While the keyboard is disabled, all keys are disabled except FUNCT, PRINT, BREAK, CTRL/1, and CTRL/2.

To subsequently enable the keyboard, the terminal must receive an ESC" or you must press BREAK twice while holding down the SHIFT key to reset the terminal completely.

4.11 CURSOR ATTRIBUTES

The cursor display may appear any one of five ways. To set the cursor display, enter the control code for the desired attribute. Type the code in the exact sequence shown below:

Attribute	Code
Not displayed	ESC.0
Blinking block	ESC.1
Steady block	ESC.2
Blinking underline	ESC.3
Steady underline	ESC.4

4.12 WORD STRUCTURE, PARITY SETTINGS, AND STOP BITS

Each computer system has its own method for checking the transmission of characters from the terminal to verify receipt. In Chapter 2 you were shown how to set the switches in the terminal to match the requirements of your computer system. Since these settings may be of importance in your programming, they are discussed in more detail here.

The first bit of the transmission is always used as a start bit to tell the computer that a character will be transmitted. (This is *not* part of the character code.) This start bit is always a one. A one may also be referred to as *true* or *mark* or *high*. A zero bit can also be called a *false*, *space*, or *low*.

Following the start bit, the terminal will now send either a 7- or 8-bit character code. These are *data bits*.

To verify correct receipt of the character code, computers may now require that the next bit received serve as a check on the transmission. This is called *parity*. Several methods are used, varying from system to system. The methods used are listed in Table 4–3.

Following any parity bit required, the terminal will also send (as set by the switch settings) either one or two stop bits to signal the end of the character code transmission. Stop bits are always ones.

Figure 4-2 shows the structure of a serial data word.

Table 4–3
Switch Settings for Parity and Data Bits

Switch S1 Dipswitch	Position Up	Position Down	Parity	Description
7		x	ODD	Requires that the total number of valid data bits be odd.
	x		EVEN	Terminal will add a one as necessary to make the total valid data bits sent even.
8	x[1]		One (or MARK or TRUE)	Requires that a one be sent in the parity position.
5		x[1]	ZERO (or SPACE or FALSE)	Requires that a zero be sent in the parity position.
6	x		SEND	Allows an odd or even parity to be sent.
		x	NONE (or NO)	Does not require a parity bit to be sent.
5	x			Causes 7 data bits to be sent.

NOTES

1. Selecting 2 stop bits on the terminal results in ONE parity.
2. Selecting 8 data bits on the terminal results in ZERO parity.

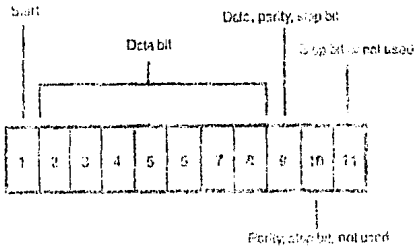


Figure 4-2 Bit Structure of a Serial Data Word

4.13 SEND FUNCTION

Once you have entered and edited data or text, you can transmit it to the computer by entering an escape sequence to send specific data.

4.13.1 Send Line Unprotected

ESC 4 sends all unprotected data on a line from column 1 through the cursor position. This consists of sends an ESC code (1C Hex) as a field delimiter in place of each protected field and an end-of-text character at the end of the send transmission.

4.13.2 Send Screen Unprotected

ESC 5 sends all unprotected data on the screen from home through the cursor position. It sends an ESC code (1C Hex) as a field delimiter in place of each protected field. The code also sends a line delimiter at the end of a line and an end-of-text character at the end of the send transmission.

4.13.3 Send Line All

ESC 6 sends all data from the first column through the cursor position. It also sends ESC) at the beginning of each protected field and ESC (at the end of each protected field. If the character at the cursor position is protected, the terminal sends ESC (to the computer. The code sends an end-of-text character at the end of the send transmission.

If the data to be sent includes attribute characters, these will be sent also [the terminal will automatically include the suitable escape sequences (ESC Gn)].

4.13.4 Send Screen All

ESC 7 sends all data on the screen from home through the cursor position. It also sends ESC) at the start of each protected field and ESC (at the end of each protected field. If the character at the cursor position is protected, the terminal sends an ESC (to the computer. This code also sends a line delimiter at the end of each line and the end-of-text character at the end of the send transmission.

4.13.5 Send Unprotected Message

ESC S sends all unprotected data bracketed by the start of text (STX) and end of text (ETX) codes displayed on a screen. After the data is sent, the terminal positions the cursor at the ETX code. If the screen contains an STX code, transmission begins from the home position. If the screen contains no ETX code, the terminal sends to the end of the screen and positions the cursor at home after the data is sent. If the screen contains neither an STX nor an ETX code, the entire screen will be sent. The code sends an ESC code (1C Hex) as a field delimiter in place of protected fields. It also sends line delimiters at the end of each line and an end-of-text delimiter at the end of the send transmission.

4.13.6 Send Entire Message

ESC X is similar in effect to an ESC S except that protected fields delimited by start-protected field (ESC)) and end-protected field (ESC ( ) are also transmitted.

4.14 TERMINATION CHARACTER SELECTION

4.14.1 Page Terminator

At the completion of each send sequence a CR is sent to the computer. This termination character may be changed to any ASCII code by entering:

ESC x4NN

where NN = any two ASCII characters. For NN, two characters must be entered. Use a NULL (CTRL/@) or other code.

For example, to change the termination character to ETX, enter:

CTRL/C (ETX) CTRL/@ (NULL)

4.14.2 Line Terminator

At the end of each line, a US (1FH) is transmitted. To change the line termination character, enter

ESC x1NN

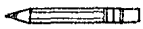
where NN = any two ASCII characters.

4.15 PRINT FUNCTION PROGRAMMING

The terminal's printer port may be set to pass data received from the computer through to the printer. If the printer can not accept any more data during a print operation, the printer may signal the terminal to stop sending data by sending a Printer Busy signal (P4 pin 20 low). This sends an X-off character (DC3) or passes the DTR signal to the computer (if the DTR switch is selected). The printer may then request more data by sending a Printer Ready signal (P4 pin 20 high). This sends an X-on character (DC1) or passes the DTR signal to the computer.

As discussed in Chapter 3, there are two methods of print commands available: transparent and extension.

The protocol described above functions in either transparent or extension mode.



Ports P3 and P4 must both be set for the same baud rate.

4.15.1 Transparent Print

ESC ` allows all subsequent data received by the terminal (including control and escape characters) to be passed through to the printer. No screen updating occurs while this mode is active.

To stop transparent printing and return to extension printing, enter

ESC @

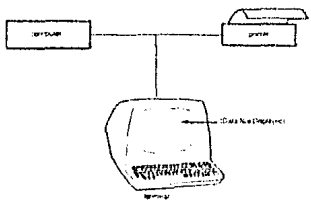
To stop transparent printing but allow screen updating to continue, enter either

ESC a or ESC A

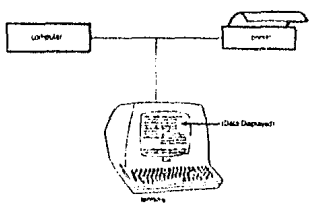
4.15.2 Extension (Copy) Print

ESC @ causes all subsequent data received by the terminal to be sent to the screen and passed to the printer.

ESC A turns off the extension mode. Screen updating continues normally.



a. Transparent Mode



b. Extension Mode

Figure 4-3 Print Modes

4.16 X-ON/X-OFF CONTROL

The terminal automatically transmits X-off to the computer (requesting it to stop sending data) when the 256-byte receive buffer is almost full (less than 16 characters).

This buffer is for data being sent to the terminal by the computer. It is not a buffer for data being sent to the printer.

When the data in the buffer has been sent to the screen, the terminal automatically transmits X-on to the computer, indicating that the computer may resume sending data to the terminal.

To turn this feature off (i.e., enable Data Terminal Ready control), enter

CTRL/N

To reenable this feature (i.e., disable Data Terminal Ready control), enter

CTRL/O

At power on, X-on/X-off is enabled.

4.17 DATA TERMINAL READY CONTROL

If you have disabled the X-on/X-off feature described above, the Data Terminal Ready feature is enabled (i.e., the DTR line is high). In that case, when the 256-byte receive buffer in the terminal has received 240 bytes from the computer, the DTR line will go low until the buffer is 20 percent empty again.

You can turn the DTR feature on or off by changing switch S2 dipswitch 10 on the rear of the terminal. (Up disconnects DTR; down connects it.)

4.18 CUSTOM EPROM APPLICATIONS

You can replace the 2532 EPROM (supplied with the terminal) with your own special 2532 EPROM or (if more space is needed) your own 2564 EPROM. The 2564 EPROM will provide an additional 4K EPROM, giving you a total of 8K EPROM space for special application programs.

4.19 BELL

You can cause a short loud bell to sound by entering

CTR/G

5. TROUBLESHOOTING AND SERVICE

5.1 CARE

Periodic cleaning and inspection will prolong the useful life of your terminal.

5.1.1 Cleaning

To clean the terminal exterior:

1. Vacuum the keyboard every three months with a soft brush attachment (or use a small soft brush).
2. Clean the housing with a soft, lint-free cloth and a commercial detergent every three months.

DO NOT use solvent-based or abrasive cleaners.

5.1.2 Inspection

Description	Frequency
1. Inspect the terminal cabinet for cracks or breaks.	1/Yr.
2. Check each key for free movement.	1/Yr.
3. Check the cable connector (at the rear of the terminal cabinet) for damage.	1/Yr.

5.2 TROUBLESHOOTING

Your computer terminal is just one of several components in the computer system. A failure anywhere else in the system can cause the improper operation of the terminal. The computer system, memory systems, cables, modems, and operational procedures should be checked if there has been a malfunction. Table 5–1 will be helpful in determining the cause of a problem. If this table does not help locate the cause of the problem, run the self test or call a qualified service technician for assistance.

5.2.1 Testing the Terminal (Self Test)

You can test the terminal yourself to verify proper operation of the video display circuitry, the transmit and receive portion of the RS232C interface, and the control processor. The test will display all displayable characters, and all 16 video attributes—in both half and full duplex.

To start the test, enter

ESC V

Switch S1 dipswitch 9 must be UP or the self test will fail.

The display screen should now look like that in Figure 4–1. Look at the display carefully to verify that all characters appear, all video attributes appear correctly, and all half intensity characters are shown. Each character should be formed properly and you should not be able to see any extra dots (and no dots should be missing).

Check the switch settings on the terminal against those on the display (see Fig. 4–1). The display will show the dipswitches as a 1 (up) or a 0 (down).

To stop the test, press BREAK twice while holding down the SHIFT key (to reset the terminal).

Should your display not appear as pictured in Figure 4–1, call qualified service technician.

Table 5–1
Troubleshooting Terminal Problems

Symptom	Possible Cause	Solution
Terminal dead (no beep; no cursor)	No AC power	Plug in power cord. Turn on power switch. Check 115/230 power switch setting.
Terminal dead; cursor may appear	Loose or defective line or power supply fuses	Turn terminal power off and change fuses.
Terminal will not go on line	System is not "up"	Check status of system.
	Loose, unconnected, or damaged cables	Attach all cables and check for cable damage.
		Check main port (P3) interface cable pins:

Table 5–1
Troubleshooting Terminal Problems

Symptom	Possible Cause	Solution
		• 5, 6, and 8 must be driven by +12 VDC or not connected at all for normal operation. • 1 and 7 must be grounded. • 3 must be connected to the host transmitter. • 2 must be connected to the host receiver.
	Modem not turned on, defective, or phone handset on modem upside down	Turn on modem. Attach different modem. Check phone handset position.
Cursor will not appear	Defective contrast pot Contrast set too light	Refer to technical representative for adjustment of contrast settings.
System does not respond while on line	Incorrect parity switch setting, word structure, stop bits	Set parity switch to match system.
Terminal is not responding to settings	Terminal not powered down after being reconfigured; software has not scanned new settings	Power down terminal and turn back on.
Terminal "locked up"	System is not responding; communication link broken	Set to half duplex and try to type. If terminal will type, check cables, modem, phone lines, and computer system. Set to full duplex and perform self test.
	Terminal incorrectly set for on line and full duplex	Set to half duplex.
Terminal locked up	Keyboard disabled from computer	Enter ESC "
	Switches set incorrectly	Review Chapter 2 switch settings carefully and check all switch settings.
Terminal prints correct data only part of the time	Parity settings incorrect	Check parity settings with system requirements.
	Stop bits or word structure wrong	Change switch settings.
Display is wavy	Hertz setting incorrect; does not match local power frequency	Change switch setting.
Printer does not print what is typed	Correct print mode selected?	Refer to 3.4 and 4.15.
	Cable connector pins connected incorrectly	Refer to 2.4. Check printer port (P4) interface cable connector pins: • 20 must be driven by +12 VDC or

Table 5–1
Troubleshooting Terminal Problems

Symptom	Possible Cause	Solution
		not connected at all for normal operation • 3 must be connected to printer data input Check other printer port device requirements.
Escape and control codes to not function as specified	The escape and/or control codes being used are not correct	Check model number of terminal and code table for correct model of terminal being used. Makes sure upper and lower case codes are used. Is a numeral one required instead of lowercase "L"?
	Keyboard locked in SHIFT position (AUTO LOCK on)	Put in lower case. Connect P3–2 to P3–3 and try in full duplex. Disconnect computer system.
Terminal prints "garbage"	Improper baud rate setting	Set correct baud rate.
	Improper handshaking protocol	Check handshaking protocol requirements of system with terminal protocol.
	Defective modem	Replace modem.
	Noisy telephone lines	Check phone lines. Install dedicated phone lines.
Erroneous data sent to computer Scrambled output Terminal loses memory	Static electricity	1. Check operating environment for static problems. 2. Install antistatic floor mat. 3. Spray carpeting with antistatic spray. 4. Increase humidity.
	AC outlet not wired properly	Check for proper wiring and grounding.
Terminal does not print what is typed while on line	Duplex switch incorrectly set	Set duplex switch to match host system.
Terminal only prints @ characters	Word length switch set incorrectly	Set word length switch to match computer system.
	Parity switch set incorrectly	Set parity switch to match computer system.
	Stop bits set incorrectly	Set stop bit switch to match computer system.

5.3 REPAIR

Operator repair is limited to changing the line fuse and the two internal power supply fuses.

5.3.1 Changing the Line Fuse

To change the line fuse, proceed as follows:

To avoid electrical shock, disconnect the terminal power cord before changing the line fuse.

1. Disconnect the terminal power cord from primary power.
2. Remove the fuse holder (see Figure 2–3) by unscrewing it counterclockwise.
3. Remove the blown fuse and replace it with a 3AG, 1 amp "slo blo" 125 VAC or 0.5 amp, 250V fuse for 220 VAC applications instantaneous (fast blow) fuse.
4. Install the fuse in the reverse order of Steps 1 through 3.

5.3.2 Changing the Power Supply Fuses

The terminal power supply fuses are installed in fuse clips on the power supply assembly inside the terminal (see Figure 2–6). To replace either of these fuses, proceed as follows:

Hazardous voltages are exposed in the cabinet. Turn off the power switch and disconnect power *before* opening the terminal cabinet.

1. Disconnect the terminal power cord from primary power.
2. Turn the terminal upside down and set it on a soft surface to prevent marring the cabinet. Remove the two Phillips screws that hold the cabinet cover on the terminal.
3. Turn the terminal right side up and lift off the cabinet cover.

Make sure there is adequate table space for the open terminal. It is top heavy and could fall over.

4. Remove the blown fuse from its fuse clip (see Figure 2–6).
5. Replace the blown fuse with a 3AG, 3 amp, 125 VAC fuse.
6. Reinstall the terminal cover and secure it with the two screws. (Do not overtighten screws!)

5.4 TECHNICAL ASSISTANCE

The Service Department is open from 7:00 a.m. until 5:00 p.m., Pacific Standard Time, Monday through Friday (except holidays). Be specific when describing the problem and failure history. If the line is busy and your problem can wait, leave a message with the TeleVideo operator and your call will be returned at our first opportunity.

APPENDIX A
SPECIFICATIONS

Monitor

Size: 12 inches measured diagonally
Phosphor: P31 green nonglare read-out

Displayed Character Set

128 displayable characters
(96 character ASCII upper/lower case alphabet with true descenders plus 32 control characters)
24 lines
80 characters per line
1920 characters per screen
Security (blank) fields
Reverse video
Underlined fields
Half intensity

Character Sets

English, French, German, Spanish

Character Font

7X8 dot matrix
8X10 resolution

Cursor Control

↑, ↓, ←, →, Home, Tab, Back Tab, Return, Line Feed, Backspace

Editing

Line insert/delete
Character insert/delete

Repeat

20-cps auto-repeat

Parity

Even, Odd, Send, Mark, Space or No Parity

Transmission

Conversation mode: Full or half duplex (keyboard selectable)
Block mode

Word Structure

7 or 8 data bits
10 or 11 bit word

Video Attributes

Blinking fields

Baud Rates

15 baud rates:

50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19,200

Interfaces

Standard RS232C point-to-point (50 ft. max.)
20ma current loop (optional) (1000 ft. max.)
RS232C printer port (unidirectional)

Auxiliary Port

Printer RS232C, transparent, screen copy

Communication Protocol

X-ON/X-OFF, DTR

Dimensions

Height:	13¼"	(33.66 cm)
Width:	16¼"	(40.96 cm)
Depth:	20⅛"	(50.96 cm)

Ventilation Requirements

Minimum 4" (10.2 cm)

Weight

30 lbs. (13.95 kg)

Operating Environment

Ambient temperature range:
0°C to 50°C (32°F to 122°F)

Maximum relative humidity (noncondensing): 95%
(Nonoperating: no restrictions)

Power Requirements

115 VAC at 0.5 amp
230 VAC at 0.25 amp
50/60 Hz, 65W

APPENDIX B
ASCII CODE CHART

b7 b6 b5 / b4 b3 b2 b1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS←	CAN	(	8	H	X	h	x
1 0 0 1	9	SKIP HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT↑	ESC	+	;	K	[	k	{
1 1 0 0	12	FF→	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	HOME RS	.	>	N	^	n	~
1 1 1 1	15	SI	NEW LINE US	/	?	O	_	o	DEL RUB

GLOSSARY

ASCII

The acronym for American Standard Code for Information Interchange. This is a standardized code for the transmission of data within the United States. It is composed of 128 characters (upper and lower case letters, numerals, punctuation marks, symbols, and control characters) in a 7-bit binary format.

Asynchronous Communication

A method of communication where the time synchronization of the transmission of data between the sending and receiving stations is set by start and stop bits and the baud rate.

Baud

The rate of transmission of data. One baud equals one binary bit per second.

Bit

An abbreviation for binary digit. A bit is the smallest unit of data. ASCII codes are composed of seven bits.

BREAK

To break or interrupt communications. When the BREAK switch on the terminal is toggled, a 250-millisecond tone is sent to the computer to immediately halt communications.

Buffer

An electronic device within the terminal that allows for the temporary storage of incoming data should the transmission rate of the incoming data be faster than the terminal's printing speed.

Bug

An error in a computer program or in the operation of the computer.

Byte

A coded group of binary bits which represents a character (letter, numeral, symbol, command, etc.).

Code

A method of representing data by groups of binary digits.

Command

A code that will cause the terminal or computer to perform an electronic or mechanical action.

Computer

An electronic system which, in accordance with its programming, will store the process information and perform high-speed mathematical or logical operations.

Control Codes

Special nonprinting codes which cause the terminal or computer to perform specific electronic or mechanical actions (such as setting tabs, etc.).

CPU

Central Processing Unit. The "brains" of a computer or computer terminal; that section where the logic and control functions are performed.

Default

Condition which exists from POWER ON or RESET if no instructions to the contrary are given to the terminal.

DEL

The ASCII DELETE code used in some instances to delete transmitted characters or to exit modes of operation.

Digit

One of the numerals in a number system.

Digital

Information in the form of individual parts—bits or digits.

EOT

An ASCII code that means "end of transmission" (EOT); used in the EOT/ACK handshaking protocol. The computer sends an EOT at the end of each transmission to the terminal. When the terminal is ready to receive more data, it transmits an acknowledge (ACK) back to the computer.

ESC

An ASCII code meaning "escape" which is used to control various electronic and mechanical functions of the terminal.

Full Duplex

In full duplex communication, the terminal can transmit and receive *simultaneously*. The transmitted data is not printed locally unless it is "echoed back" by the computer.

Half Duplex

In half duplex communication, the terminal transmits and receives data in separate, *consecutive operations*. Transmitted data is printed locally.

Handshaking

A communications protocol which is necessarily used when the transmitting speed of the computer is faster than the printing speed of the terminal. It consists of a set of commands, recognized by both stations, which control the flow of the data transmission from the computer.

Host

The computer system.

Interface

A communications channel which is typically used for external devices.

Main

The computer system.

Memory

That part of a computer system or terminal where information is stored.

Microprocessor

An electronic circuit on the surface of a small silicon chip which can be programmed to perform a wide variety of functions within the computer system or terminal.

Modem

An electronic device which converts (modulates) the serial communications between the computer and terminal into audible tones which can be transmitted over telephone lines. All received data is reconverted (demodulated) from the audible tones into serial information.

NUL

An ASCII code ("nothing") used as a fill character in some communications formats.

Parity

A method of checking for errors in data communications. An extra bit (either a "1" or "0"), called the parity bit, is added to the end of each ASCII character to make the final count of "1" bits in the character an even or odd number, according to a prearranged format. Some systems always use even parity, some always use odd parity, and some do not check for parity. Both terminal and system *must* be set for the same parity.

Protocol

All of the conventions which must be observed in order for the computer and terminal to communicate with each other.

Serial Communication

The standard method of ASCII character transmission where bits are sent, one at a time, in sequence. Each 7-bit ASCII character is preceded by a start bit (see Asynchronous Communication) and ended with a parity bit and stop bit.

Toggle

Activation or deactivation of function or mode key (either a receive key, command sequence, or manual keystroke.)

Wraparound

Movement of the cursor as it reaches the right edge of screen, disappears, and "wraps around" to the beginning of the next line.

X-ON/X-OFF

A handshaking protocol. When the terminal's buffer is nearly full, it transmits an X-OFF to the computer to stop transmission; when the buffer is almost empty, an X-ON is transmitted to the host to resume transmission.

INDEX

OPERATOR'S QUICK REFERENCE GUIDE

Function	Command
CURSOR	
Home	CTRL/^
New line	CTRL/_
Carriage return	CTRL/M
Linefeed/cursor down	CTRL/J
Cursor up	CTRL/K
Backspace/cursor left	CTRL/H
Cursor right	CTRL/L
Cursor off	ESC.0
TAB	
Set column tab	ESC1
Typewriter tab	CTRL/I
Field tab	ESCi
Back tab	ESCI
Clear typewriter tab	ESC2
Clear all tabs	ESC3
EDIT	
Local edit	ESCk
Duplex edit	ESCl
Character insert	ESCQ
Character delete	ESCW
Line insert	ESCE
Line delete	ESCR
Erase line to spaces	ESCT
Erase line to nulls	ESCt
Erase screen to spaces	ESCY
Erase screen to nulls	ESCy
Clear screen to spaces	ESC+
	CTRL/Z
	ESC;
Clear screen to nulls	ESC*
Clear unprotected to nulls	ESC:
Clear screen to half-intensity spaces	ESC,
PROGRAM	
Address cursor (row, column)	ESC=
Read cursor (row, column)	ESC?
Enable keyboard	ESC"
Disable keyboard	ESC#
Extension print on	ESC@
Stop printing	ESCA
Transparent print on	ESC`
Transparent print off (update on)	ESCa
Bell	CTRL/G
VIDEO	
Normal video (green on black)	ESCG0
Blank (invisible normal)	ESCG1
Blink	ESCG2
Blank (invisible blink)	ESCG3
Reverse video (black on green)	ESCG4
Blank (invisible reverse)	ESCG5
VIDEO continued	
Reverse blink	ESCG6
Blank (invisible reverse blink)	ESCG7
Underline	ESCG8
Blank (invisible underline)	ESCG9
Underline blink	ESCG:
Blank (invisible underline blink)	ESCG;
Reverse blink underline	ESCG<
Blank (invisible reverse underline)	ESCG=
Reverse blink underline	ESCG>
Blank (invisible reverse blink underline)	ESCG?
Half intensity (protected writing) on	ESC)
Half intensity (protected writing) off	ESC(
Protect on	ESC&
Protect off	ESC'
Cursor visible/invisible	ESC.
Blinking block cursor	ESC.1
Steady block cursor	ESC.2
Blinking underline	ESC.3
Steady underline	ESC.4
MONITOR	
Monitor mode on (not transmitted)	CTRL/1
Monitor mode off (not transmitted)	CTRL/2
Monitor mode on (transmitted)	ESCU
Monitor mode off (transmitted)	ESCu
	ESCX
SELF TEST	
Start self test	ESCV
SEND	
Send line unprotected	ESC4
Send screen unprotected	ESC5
Send line all	ESC6
Send entire screen	ESC7
Send unprotected message	ESCS
Send entire message	ESCs
Select screen terminator	ESCx4NN
Select line terminator	ESCx1NN
Data Terminal Ready on	CTRL/N
Data Terminal Ready off	CTRL/O

National Semiconductor

Voltage Regulators

LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

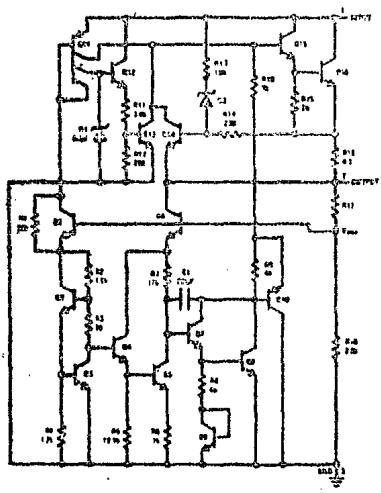
Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

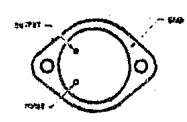
Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

Schematic and Connection Diagrams

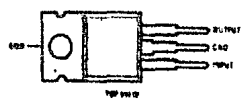


Metal Can Package
TO-3 (K)
Aluminum



Order Numbers
LM7805CK
LM7812CK
LM7815CK
See Package KC02A

Plastic Package
TO-220 (T)



Order Numbers
LM7805CT
LM7812CT
LM7815CT
See Package T03B

Absolute Maximum Ratings

Input Voltage (V_O = 5V, 12V and 15V)	35V
Internal Power Dissipation (Note 1)	Internally Limited
Operating Temperature Range (T_A)	0°C to +70°C
Maximum Junction Temperature	
(K Package)	150°C
(T Package)	125°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	
TO-3 Package K	300°C
TO-220 Package T	230°C

Electrical Characteristics LM78XXC (Note 2) 0°C ≤ T_J ≤ 125°C unless otherwise noted.

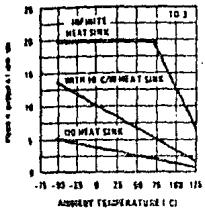
OUTPUT VOLTAGE				5V			12V			15V			
INPUT VOLTAGE (unless otherwise noted)				10V			19V			23V			Units
PARAMETER		CONDITIONS		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V_O	Output Voltage	T_J = 25°C, 5 mA ≤ I_O ≤ 1A		4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		P_D ≤ 15W, 5 mA ≤ I_O ≤ 1A		4.75		5.25	11.4		12.6	14.25		15.75	V
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}		(7 ≤ V_{IN} ≤ 20)			(14.5 ≤ V_{IN} ≤ 27)			(17.5 ≤ V_{IN} ≤ 30)			V
ΔV_O	Line Regulation	I_O = 500 mA	T_A = 25°C		3	50		4	120		4	150	mV
			ΔV_{IN}	(7 ≤ V_{IN} ≤ 25)			(14.5 ≤ V_{IN} ≤ 30)			(17.5 ≤ V_{IN} ≤ 30)			V
			0°C ≤ T_J ≤ +125°C			50			120			150	mV
			ΔV_{IN}	(8 ≤ V_{IN} ≤ 25)			(15 ≤ V_{IN} ≤ 27)			(18.5 ≤ V_{IN} ≤ 29)			V
		I_O ≤ 1A	T_J = 25°C			50			120			150	mV
			ΔV_{IN}	(7 ≤ V_{IN} ≤ 25)			(14.6 ≤ V_{IN} ≤ 27)			(17.7 ≤ V_{IN} ≤ 30)			V
			0° ≤ T_J ≤ +125°C			25			60			75	mV
			ΔV_{IN}	(8 ≤ V_{IN} ≤ 12)			(15 ≤ V_{IN} ≤ 22)			(20 ≤ V_{IN} ≤ 26)			V
ΔV_O	Load Regulation	T_J = 25°C	5 mA ≤ I_O ≤ 1.5A		10	50		12	120		12	150	mV
			250 mA ≤ I_O ≤ 750 mA			25			60			75	mV
		5 mA ≤ I_O ≤ 1A, 0°C ≤ T_J ≤ +125°C				50			120			150	mV
I_Q	Quiescent Current	I_O ≤ 1A	T_J = 25°C			8			8			8	mA
			0°C ≤ T_J ≤ +125°C			8.5			8.5			8.5	mA
ΔI_Q	Quiescent Current Change	5 mA ≤ I_O ≤ 1A				0.5			0.5			0.5	mA
		T_J = 25°C, I_O ≤ 1A				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}		(7.5 ≤ V_{IN} ≤ 20)			(14.6 ≤ V_{IN} ≤ 27)			(17.5 ≤ V_{IN} ≤ 30)			V
		I_O ≤ 500 mA, 0°C ≤ T_J ≤ +125°C				1.0			1.0			1.0	mA
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}		(7 ≤ V_{IN} ≤ 25)			(14.6 ≤ V_{IN} ≤ 30)			(17.5 ≤ V_{IN} ≤ 30)			V
V_N	Output Noise Voltage	T_A = 25°C, 10 Hz ≤ f ≤ 100 kHz			40			75			90		μV
$\frac{\Delta V_{IN}}{\Delta V_{OUT}}$	Ripple Rejection	f = 120 Hz	I_O ≤ 1A, T_J = 25°C or I_O ≤ 500 mA, 0°C ≤ T_J ≤ 125°C	62	80		55	72		54	70		dB
				62			55			54			dB
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}		(8 ≤ V_{IN} ≤ 18)			(15 ≤ V_{IN} ≤ 25)			(18.5 ≤ V_{IN} ≤ 28.5)			V
R_O	Dropout Voltage	T_J = 25°C, I_{OUT} = 1A			2.0			2.0			2.0		V
	Output Resistance	f = 1 kHz			8			18			19		mΩ
	Short Circuit Current	T_J = 25°C			2.1			1.5			1.2		A
	Peak Output Current	T_J = 25°C			2.4			2.4			2.4		A
	Average TC of V_{OUT}	0°C ≤ T_J ≤ +125°C, I_O = 5 mA			0.6			1.8			1.8		mV/°C
V_{IN}	Input Voltage Required to Maintain Line Regulation	T_J = 25°C, I_O ≤ 1A		7.5			14.6			17.7			V

NOTE 1: Thermal resistance of the TO-3 package (K, KC) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

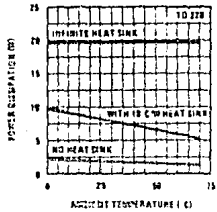
NOTE 2: All characteristics are measured with capacitor across the input of 0.22 μF, and a capacitor across the output of 0.1 μF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_W ≤ 10 ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

ypical Performance Characteristics

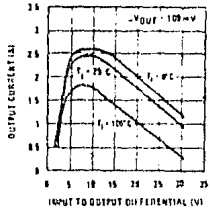
Maximum Average Power Dissipation



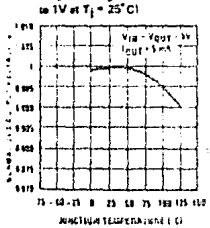
Maximum Average Power Dissipation



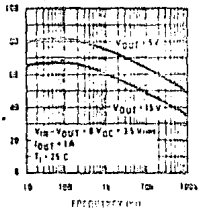
Peak Output Current



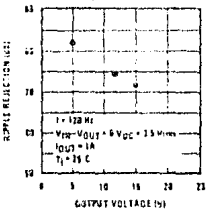
Output Voltage (Normalized to 1V at T_J = 25°C)



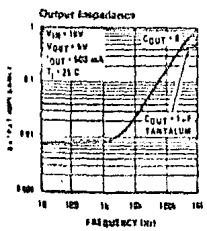
Ripple Rejection



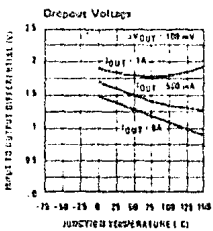
Ripple Rejection



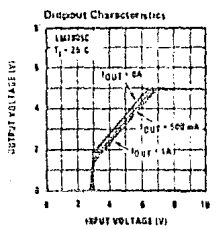
Output Impedance



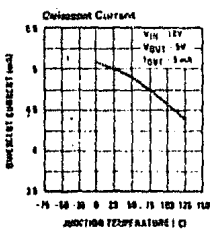
Dropout Voltage



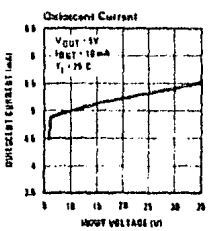
Dropout Characteristics



Quiescent Current



Quiescent Current



National Semiconductor

Transmission Line Drivers/Receivers

DS1488 Quad Line Driver

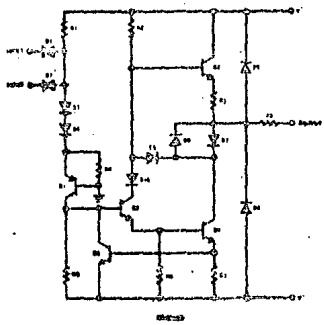
General Description

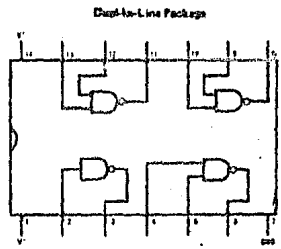
The DS1488 is a quad line driver which converts standard DTL/TTL input logic levels through one stage of inversion to output levels which meet EIA Standard No. RS-232C and CCITT Recommendation V. 24.

Features

- Current limited output ±10 mA typ
- Power-off source impedance 300Ω min
- Simple slew rate control with external capacitor
- Flexible operating supply range
- Inputs are DTL/TTL compatible

Schematic and Connection Diagrams



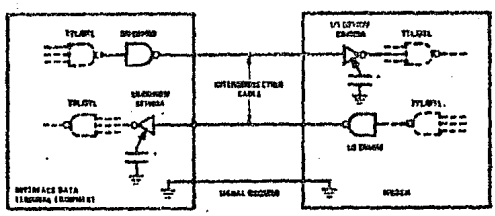


Dual-In-Line Package

Order Number DS1488J or DS1488N
See NS Package J14A or N14A

Typical Applications

RS232C Data Transmission



Absolute Maximum Ratings (Note 1)

Supply Voltage	
V^+	+15V
V^-	−15V
Input Voltage (V_{IN})	−15V ≤ V_{IN} ≤ 7.0V
Output Voltage	±15V
Operating Temperature Range	0°C to +75°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec)	300°C

Electrical Characteristics (Notes 2, 3 and 4)

	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
I_{IL}	Logical "0" Input Current	V_{IN} = 0V			1.0	−1.3	mA
I_{IH}	Logical "1" Input Current	V_{IN} = +5.0V			0.005	10.0	μA
V_{OH}	High Level Output Voltage	R_L = 3.0 kΩ,	V^+ = 9.0V, V^- = −9.0V	6.0	7.0		V
		V_{IN} = 0.8V	V^+ = 13.2V, V^- = −13.2V	9.0	10.5		V
V_{OL}	Low Level Output Voltage	R_L = 3.0 kΩ,	V^+ = 9.0V, V^- = −9.0V		−6.0	−6.0	V
		V_{IN} = 1.9V	V^+ = 13.2V, V^- = −13.2V		−10.5	−9.0	V
I_{OS}	High Level Output Short-Circuit Current	V_{OUT} = 0V, V_{IN} = 0.8V		−6.0	−10.0	−12.0	mA
I_{OS}	Low Level Output Short-Circuit Current	V_{OUT} = 0V, V_{IN} = 1.9V		6.0	10.0	12.0	mA
R_{OUT}	Output Resistance	V^+ = V^- = 0V, V_{OUT} = ±2V		300			Ω
I_{CC}	Positive Supply Current (Output Open)	V_{IN} = 1.9V	V^+ = 9.0V, V^- = −9.0V		15.0	20.0	mA
			V^+ = 12V, V^- = −12V		19.0	25.0	mA
			V^+ = 15V, V^- = −15V		25.0	34.0	mA
		V_{IN} = 0.8V	V^+ = 9.0V, V^- = −9.0V		4.5	6.0	mA
			V^+ = 12V, V^- = −12V		5.5	7.0	mA
			V^+ = 15V, V^- = −15V		8.0	12.0	mA
I_{CC}	Negative Supply Current (Output Open)	V_{IN} = 1.9V	V^+ = 9.0V, V^- = −9.0V		−13.0	−17.0	mA
			V^+ = 12V, V^- = −12V		−18.0	−23.0	mA
			V^+ = 15V, V^- = −15V		−25.0	−34.0	mA
		V_{IN} = 0.8V	V^+ = 9.0V, V^- = −9.0V		−0.001	−0.015	mA
			V^+ = 12V, V^- = −12V		−0.001	−0.015	mA
			V^+ = 15V, V^- = −15V		−0.01	−2.5	mA
P_d	Power Dissipation	V^+ = 9.0V, V^- = −9.0V			252	333	mW
		V^+ = 12V, V^- = −12V			444	576	mW

Switching Characteristics (V_{CC} = 9V, V_{EE} = −9V, T_A = 25°C)

	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
t_{pd1}	Propagation Delay to a Logical "1"	R_L = 3.0 kΩ, C_L = 15 pF, T_A = 25°C		230	350	ns
t_{pd0}	Propagation Delay to a Logical "0"	R_L = 3.0 kΩ, C_L = 15 pF, T_A = 25°C		70	175	ns
t_r	Rise Time	R_L = 3.0 kΩ, C_L = 15 pF, T_A = 25°C		75	100	ns
t_f	Fall Time	R_L = 3.0 kΩ, C_L = 15 pF, T_A = 25°C		40	75	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified min/max limits apply across the 0°C to +75°C temperature range for the DS1488.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltages referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

Applications

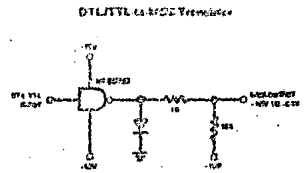
By connecting a capacitor to each driver output the slew rate can be controlled utilizing the output current limiting characteristics of the DS1488. For a set slew rate the appropriate capacitor value may be calculated using the following relationship

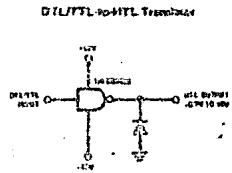
$$C = I_{SC}(\Delta T/\Delta V)$$

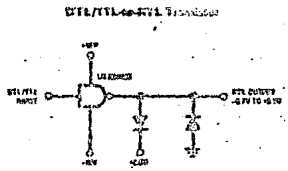
where C is the required capacitor, I_{SC} is the short circuit current value, and $\Delta V/\Delta T$ is the slew rate.

RS232C specifies that the output slew rate must not exceed 30V per microsecond. Using the worst case output short circuit current of 12 mA in the above equation, calculations result in a required capacitor of 400 pF connected to each output.

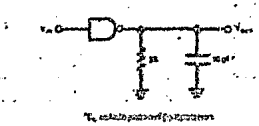
Typical Applications (Continued)



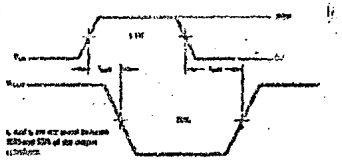




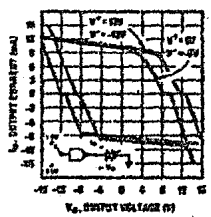
AC Load Circuit



Switching Time Waveforms



Typical Performance Characteristics



National Semiconductor

Transmission Line Drivers/Receivers

DS1489/DS1489A Quad Line Receiver

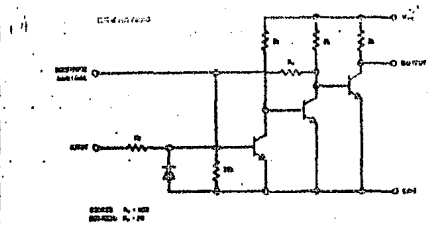
General Description

The DS1489/DS1489A are quad line receivers designed to interface data terminal equipment with data communications equipment. They are constructed on a single monolithic silicon chip. These devices satisfy the specifications of EIA standard No. RS232C. The DS1489/DS1489A meet and exceed the specifications of MC1489/MC1489A and are pin-for-pin replacements. The DS1489/DS1489A are available in 14-lead ceramic dual-in-line package.

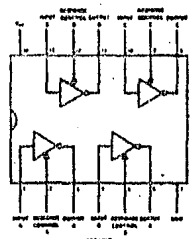
Features

- Four totally separate receivers per package
- Programmable threshold
- Built-in input threshold hysteresis
- "Fail safe" operating mode
- Inputs withstand ±30V

Schematic and Connection Diagrams

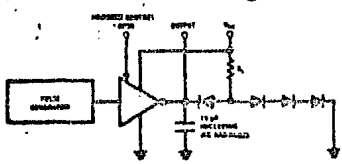


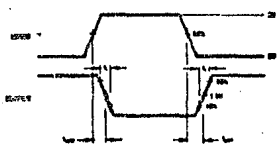
Dual-In-Line Package



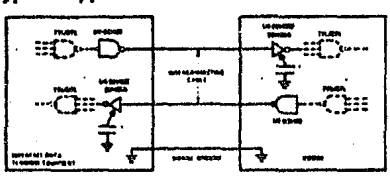
Order Number DS1489J, DS1489AJ, DS1489N or DS1489AN
See NS Package J14A or N14A

AC Test Circuit and Voltage Waveforms

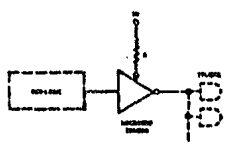




Typical Applications



RS232C Data Transmission



MOS to TTL/DTL Translator

Absolute Maximum Ratings (Note 1)

The following apply for T_A 25°C unless otherwise specified.

Power Supply Voltage	10V
Input Voltage Range	±30V
Output Load Current	20 mA
Power Dissipation (Note 2)	1W
Operating Temperature Range	0°C to +75°C
Storage Temperature Range	65°C to +150°C

Electrical Characteristics (Notes 2, 3 and 4)

DS1489/DS1489A: The following apply for V_{CC} = 5.0V ±1%, 0°C ≤ T_A ≤ +75°C unless otherwise specified.

	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
V_{TH}	Input High Threshold Voltage	T_A = 25°C, V_{OUT} ≤ 0.45V, I_{OUT} = 10 mA	DS1489	1.0		1.5	V
			DS1489A	1.75		2.25	V
V_{TL}	Input Low Threshold Voltage	T_A = 25°C, V_{OUT} ≥ 2.5V, I_{OUT} = −0.5 mA		0.75		1.25	V
I_{IN}	Input Current	V_{IN} = +25V		+3.6	+5.6	+8.3	mA
		V_{IN} = −25V		−3.6	−5.6	−8.3	mA
		V_{IN} = +3V		+0.43	+0.53		mA
		V_{IN} = −3V		−0.43	−0.53		mA
V_{OH}	Output High Voltage	I_{OUT} = −0.5 mA	V_{IN} = 0.75V	2.6	3.8	5.0	V
			Input = Open	2.6	3.8	5.0	V
V_{OL}	Output Low Voltage	V_{IN} = 3.0V, I_{OUT} = 10 mA			0.33	0.45	V
I_{SC}	Output Short Circuit Current	V_{IN} = 0.75V			3.0		mA
I_{CC}	Supply Current	V_{IN} = 5.0V			14	26	mA
P_d	Power Dissipation	V_{IN} = 5.0V			70	130	mW

Switching Characteristics (V_{CC} = 5V, T_A = 25°C)

	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
t_{pd1}	Input to Output "High" Propagation Delay	R_L = 3.9k, *(Figure 1)* (see Test Circuit)		28	85	ns
t_{pd0}	Input to Output "Low" Propagation Delay	R_L = 390Ω, *(Figure 1)* (see Test Circuit)		20	50	ns
t_r	Output Rise Time	R_L = 3.9k, *(Figure 1)* (see Test Circuit)		110	175	ns
t_f	Output Fall Time	R_L = 390Ω, *(Figure 1)* (see Test Circuit)		9	20	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified min/max limits apply across the 0°C to +75°C temperature range for the DS1489 and DS1489A.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltages referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

Note 4: These specifications apply for response control pin = open.

54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

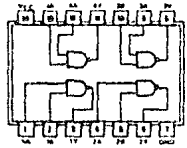
PIN ASSIGNMENTS (TOP VIEWS)

QUADRUPLE 2-INPUT
POSITIVE-NAND GATES

00

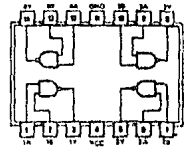
positive logic:
$Y = \overline{AB}$

See page 6-2



SN5400 (J)
SN54H00 (J)
SN54L00 (J)
SN54LS00 (J, W)
SN54S00 (J, W)

SN7400 (J, N)
SN74H00 (J, N)
SN74L00 (J, N)
SN74LS00 (J, N)
SN74S00 (J, N)



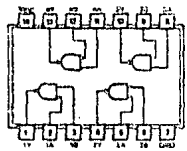
SN5400 (W)
SN54H00 (W)
SN54L00 (T)

QUADRUPLE 2-INPUT
POSITIVE-NAND GATES
WITH OPEN-COLLECTOR OUTPUTS

01

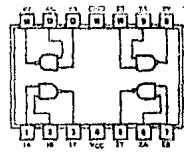
positive logic:
$Y = \overline{AB}$

See page 6-3

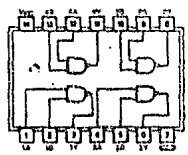


SN5401 (J)
SN54LS01 (J, W)

SN7401 (J, N)
SN74LS01 (J, N)



SN5401 (W)
SN54H01 (W)
SN54L01 (T)



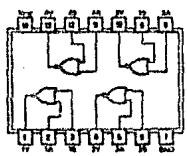
SN54S01 (J)
SN74S01 (J, N)

QUADRUPLE 2-INPUT
POSITIVE-NOR GATES

02

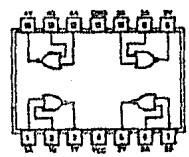
positive logic:
$Y = \overline{A+B}$

See page 6-4



SN5402 (J)
SN54L02 (J)
SN54LS02 (J, W)
SN54S02 (J, W)

SN7402 (J, N)
SN74L02 (J, N)
SN74LS02 (J, N)
SN74S02 (J, N)



SN5402 (W)
SN54L02 (T)

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

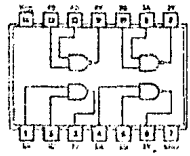
PIN ASSIGNMENTS (TOP VIEWS)

QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

03

positive logic:
$Y = \overline{AB}$

See page 6-8



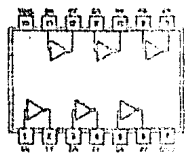
SN5403 (J) SN7403 (J, N)
SN54LS03 (J) SN74LS03 (J, N)
SN54S03 (J, W) SN74LS03 (J, N)
SN74S03 (J, N) SN74S03 (J, N)

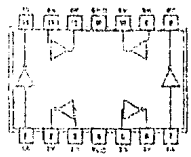
HEX INVERTERS

04

positive logic:
$Y = \overline{A}$

See page 6-2





SN5404 (J) SN7404 (J, N)
SN54H04 (J) SN74H04 (J, N)
SN54LS04 (J) SN74LS04 (J, N)
SN54S04 (J, W) SN74S04 (J, N)
SN54S14 (J, W) SN74S14 (J, N)

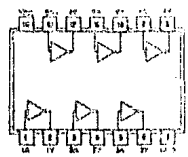
SN7404 (N)
SN54H04 (W)
SN74LS14 (N)

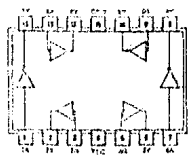
HEX INVERTERS WITH OPEN-COLLECTOR OUTPUTS

05

positive logic:
$Y = \overline{A}$

See page 6-5





SN5405 (J) SN7405 (J, N)
SN54H05 (J) SN74H05 (J, N)
SN54LS05 (J, W) SN74LS05 (J, N)
SN54S05 (J, W) SN74S05 (J, N)

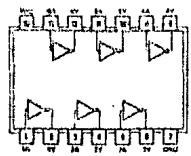
SN54H05 (W)
SN74LS05 (N)

HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS

06

positive logic:
$Y = \overline{A}$

See page 6-14



SN5406 (J, W) SN7406 (J, N)

recommended operating conditions

	54 FAMILY / 74 FAMILY	SERIES 54 / SERIES 74 '00, '04, '10, '20, '30			SERIES 54H / SERIES 74H 'H00, 'H04, 'H10, 'H20, 'H30			SERIES 54L / SERIES 74L 'L00, 'L04, 'L10, 'L20, 'L30			SERIES 54LS / SERIES 74LS 'LS00, 'LS04, 'LS10, 'LS20, 'LS30			SERIES 54S / SERIES 74S 'S00, 'S04, 'S10, 'S20, 'S30, 'S133			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
	74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}	54 Family			-400			-500			-100			-400			-1000	μA
	74 Family			-400			-500			-200			-400			-1000	
Low-level output current, I_{OL}	54 Family			16			20			3			4			20	mA
	74 Family			16			20			3.6			8			20	
Operating free-air temperature, T_A	54 Family	-55		125	-55		125	-55		125	-55		125	-55		125	°C
	74 Family	0		70	0		70	0		70	0		70	0		70	

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

PARAMETER	TEST FIGURE	TEST CONDITIONS†			SERIES 54 / SERIES 74 '00, '04, '10, '20, '30			SERIES 54H / SERIES 74H 'H00, 'H04, 'H10, 'H20, 'H30			SERIES 54L / SERIES 74L 'L00, 'L04, 'L10, 'L20, 'L30			SERIES 54LS / SERIES 74LS 'LS00, 'LS04, 'LS10, 'LS20, 'LS30			SERIES 54S / SERIES 74S 'S00, 'S04, 'S10, 'S20, 'S30, 'S133			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH} High-level input voltage	1, 2				2			2			2			2			2			V
V_{IL} Low-level input voltage	1, 2			54 Family			0.8			0.8			0.7			0.7			0.8	V
				74 Family			0.8			0.8			0.7			0.8			0.8	
V_{IK} Input clamp voltage	3	V_{CC} = MIN, I_I = I_{IK}				-1.5			-1.5						-1.5			-1.2	V
V_{OH} High-level output voltage	1	V_{CC} = MIN, V_{IL} = V_{IL} max, I_{OH} = MAX		54 Family	2.4	3.4		2.4	3.5		2.4	3.3		2.5	3.4		2.5	3.4		V
				74 Family	2.4	3.4		2.4	3.5		2.4	3.2		2.7	3.4		2.7	3.4		
V_{OL} Low-level output voltage	2	V_{CC} = MIN, V_{IH} = 2 V	I_{OL} = MAX	54 Family		0.2	0.4		0.2	0.4		0.15	0.3		0.25	0.4			0.5	V
				74 Family		0.2	0.4		0.2	0.4		0.2	0.4		0.25	0.5			0.5	
			I_{OL} = 4 mA	Series 74LS												0.4				
I_I Input current at maximum input voltage	4	V_{CC} = MAX		V_I = 5.5 V			1			1			0.1						1	mA
				V_I = 7 V												0.1				
I_{IH} High-level input current	4	V_{CC} = MAX		V_{IH} = 2.4 V			40			50			10							μA
				V_{IH} = 2.7 V												20			50	
I_{IL} Low-level input current	5	V_{CC} = MAX		V_{IL} = 0.3 V									-0.18							mA
				V_{IL} = 0.4 V			-1.6			-2						-0.4				
				V_{IL} = 0.5 V															-2	
I_{OS} Short-circuit output current§	6	V_{CC} = MAX		54 Family	-20		-55	-40		-100	-3		-15	-20		-100	-40		-100	mA
				74 Family	-18		-55	-40		-100	-3		-15	-20		-100	-40		-100	
I_{CC} Supply current	7	V_{CC} = MAX			See table on next page															μA

†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.
§I_I = −12 mA for SN54'/SN74', −8 mA for SN54H'/SN74H', and −18 mA for SN54LS'/SN74LS' and SN54S'/SN74S'.
¶Not more than one output should be shorted at a time, and for SN54H'/SN74H', SN54LS'/SN74LS', and SN54S'/SN74S', duration of short-circuit should not exceed 1 second.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

supply current†

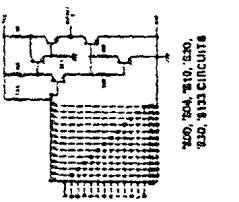
switching characteristics at V_{CC} = 5 V, T_A = 25°C

TYPE	I_{CCH} (mA) Total with outputs high		I_{CCL} (mA) Total with outputs low		I_{CC} (mA) Average per gate (50% duty cycle)	TYPE	TEST CONDITIONS‡	tPLH (ns) Propagation delay time, low-to-high-level output			tPHL (ns) Propagation delay time, high-to-low-level output		
	TYP	MAX	TYP	MAX	TYP			MIN	TYP	MAX	MIN	TYP	MAX
'00	4	8	12	22	2	'20, '10	C_L = 15 pF, R_L = 400 Ω		11	22		7	15
'04	6	12	18	33	3	'24, '30			12	22		8	15
'10	3	6	9	16.5	2	'33			13	22		8	15
'20	2	4	6	11	2	'00	C_L = 15 pF, R_L = 400 Ω		5.5	10		6.2	10
'30	1	2	3	6	2	'04			6	10		8.5	10
'00	10	16.5	25	45	4.5	'10			5.5	10		6.2	10
'04	15	25	35	60	4.5	'20			5.5	10		6.2	10
'10	7.5	12.5	19.5	30	4.5	'30			6	10		7	10
'20	5	8.4	13	20	4.5	'33			6.15	10		8.9	12
'30	2.5	4.2	8.5	10	4.5	'00, '01, '10, '20	C_L = 50 pF, R_L = 400 Ω		23	50		31	60
'03	0.44	0.8	1.10	2.00	0.70	'33			23	50		10	100
'04	0.63	1.2	1.74	3.00	0.70	'03, '51							
'10	0.33	0.6	0.87	1.50	0.50	'30	C_L = 15 pF, R_L = 2 kΩ		9	15		10	15
'20	0.22	0.4	0.65	1.00	0.50	'51, '53							
SN54L30	0.11	0.33	0.79	0.91	0.50	'00, '04	C_L = 15 pF, R_L = 2 kΩ		8	18		13	20
SN74L30	0.11	0.2	0.22	0.51	0.50	'10, '20	C_L = 50 pF, R_L = 2 kΩ		3	4.5		3	5
'00	0.5	1.0	2.4	4.4	0.4	'30, '33	C_L = 15 pF, R_L = 2 kΩ		4.5			5	
'04	1.2	2.4	3.6	6.6	0.4		C_L = 50 pF, R_L = 2 kΩ		4	6		4.5	7
'10	0.6	1.2	1.8	3.3	0.4	'51, '53	C_L = 50 pF, R_L = 2 kΩ		5.5			6.5	
'20	0.4	0.8	1.2	2.2	0.4								
'30	0.33	0.5	0.8	1.1	0.48								
'53	2	4	8	12	2.4								
'00	10	16	25	33	3.15								
'04	15	24	33	54	3.15								
'10	7.5	12	16	27	3.15								
'20	5	8	10	18	3.15								
'30	2.5	4	8.5	10	4.25								
'133	2	3	8	10	4.25								

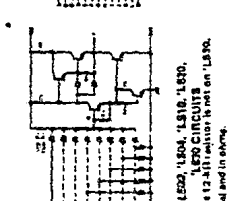
†Maximum values of I_{CC} are over the recommended operating ranges of V_{CC} and T_A; typical values are at V_{CC} = 5 V, T_A = 25°C.

‡Load circuits and voltage waveforms are shown on Pages 3-10 and 3-11.

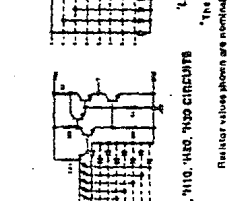
schematics (each gate)



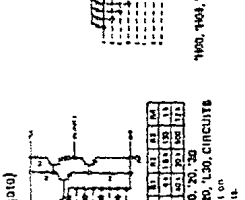
'L00, 'L04, 'L10, 'L20, 'L30, CIRCUITS
Input clamp diodes not on SN54L/SN74L' circuits.



'L02, 'L04, 'L10, 'L20, 'L30 CIRCUITS



'L22, 'L24, 'L10, 'L30,
'L20 CIRCUITS
*The 12-kΩ resistor is not on 'L30.



'L00, 'L04, 'L10, 'L30,
'L20, 'L133 CIRCUITS

Resistor values shown are nominal and in ohms.

recommended operating conditions

	54 FAMILY / 74 FAMILY	SERIES 54 / SERIES 74 '02			SERIES 74 '25, '27			SERIES 54L / SERIES 74L L02			SERIES 54LS / SERIES 74LS 'LS02, 'LS27			SERIES 54S / SERIES 74S 'S02, 'S260			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
	74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}	54 Family			−400			−800			−100			−400			−1000	μA
	74 Family			−400			−800			−200			−400			−1000	
Low-level output current, I_{OL}	54 Family			16			16			2			4			20	mA
	74 Family			16			16			3.6			8			20	
Operating free-air temperature, T_A	54 Family	−55		125	−55		125	−65		125	−55		125	−55		125	°C
	74 Family	0		70	0		70	0		70	0		70	0		70	

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

PARAMETER			TEST FIGURE	TEST CONDITIONS†		SERIES 54 / SERIES 74 '02, '25, '27			SERIES 54L / SERIES 74L L02			SERIES 54LS / SERIES 74LS 'LS02, 'LS27			SERIES 54S / SERIES 74S 'S02, 'S260			UNIT
						MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage		1, 2			2			2			2			2			V
V_{IL}	Low-level input voltage		1, 2		54 Family			0.8			0.7			0.7			0.8	V
					74 Family			0.8			0.7			0.8			0.8	
V_{IK}	Input clamp voltage		3	V_{CC} = MIN, I_I = −12 mA				−1.5						−1.5			−1.2	V
V_{OH}	High-level output voltage		1	V_{CC} = MIN, V_{IL} = V_{IL} max, I_{OH} = MAX	54 Family	2.4	3.4		2.4	3.3		2.5	3.4		2.4	3.4		V
					74 Family	2.4	3.4		2.4	3.2		2.7	3.4		2.7	3.4		
V_{OL}	Low-level output voltage		2	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = MAX	54 Family		0.2	0.4		0.15	0.3		0.25	0.4			0.5	V
					74 Family		0.2	0.4		0.2	0.4		0.35	0.5			0.5	
					Series 74LS								0.25	0.4				
I_I	Input current at maximum input voltage		4	V_{CC} = MAX	V_I = 5.5 V			1			0.1						1	mA
					V_I = 7 V									0.1				
I_{IH}	High-level input current	Data inputs	4	V_{CC} = MAX	V_{IH} = 2.4 V			40			10							μA
		Strobe of '25						160										
		All inputs			V_{IH} = 2.7 V									20			80	
I_{IL}	Low-level input current	All inputs	5	V_{CC} = MAX	V_{IL} = 0.3 V						−0.18							mA
		Data inputs			V_{IL} = 0.4 V			−1.6						−0.4				
		Strobe of '25						−6.4										
		All inputs			V_{IL} = 0.5 V												−2	
I_{OS}	Short-circuit output current§		6	V_{CC} = MAX	54 Family	−20		−55	−3		−15	−20		−100	−40		−100	mA
					74 Family	−18		−55	−3		−15	−20		−100	−40		−100	
I_{CC}	Supply current		7	V_{CC} = MAX														mA

See table on next page

†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.

§I_I = −12 mA for SN54'/SN74' and −18 mA for SN54LS'/SN74LS' and SN54S'/SN74S'.

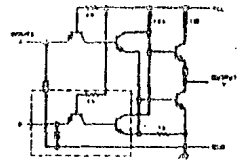
¶Not more than one output should be shorted at a time, and for SN54LS'/SN74LS' and SN54S'/SN74S', duration of output short-circuit should not exceed one second.

supply current†

TYPE	I_{CCH} (mA) Total with outputs high		I_{CCL} (mA) Total with outputs low		I_{CC} (mA) Average per gate (50% duty cycle)
	TYP	MAX	TYP	MAX	TYP
'02	8	16	14	27	2.75
'26	8	16	10	19	2.25
'27	10	16	16	26	4.34
'L02	0.8	1.6	1.4	2.6	0.275
'LS02	1.6	3.2	2.8	5.4	0.55
'LS27	2.0	4	3.4	6.8	0.8
'S02	17	29	26	45	5.39
'S260	17	29	23	45	10.76

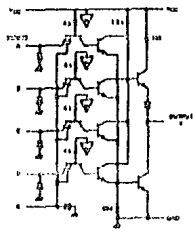
†Maximum values of I_{CC} are over the recommended operating ranges of V_{CC} and T_A; typical values are at V_{CC} = 5 V, T_A = 25°C

schematics (each gate)



The portion of the schematic within the dashed lines is repeated for the C input of the '27.

'02, '27 CIRCUITS



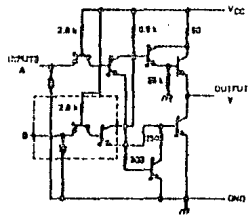
'26 CIRCUITS

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

TYPE	TEST CONDITIONS‡	t_{PLH} (ns) Propagation delay time, low-to-high-level output			t_{PHL} (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'02	C_L = 15 pF, R_L = 400 Ω		12	18		8	15
'26			13	22		8	15
'27			10	15		7	11
'L02	C_L = 50 pF, R_L = 4 kΩ		31	60		35	60
'LS02, 'LS27	C_L = 15 pF, R_L = 2 kΩ		10	15		10	15
'S02	C_L = 15 pF, R_L = 280 Ω		3.5	5.5		3.5	5.5
	C_L = 50 pF, R_L = 280 Ω		5			5	
'S260	C_L = 15 pF, R_L = 280 Ω		4	5.5		4	6

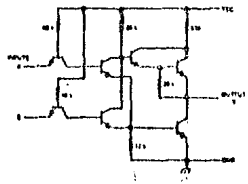
‡Load circuit and voltage waveforms are shown on pages 3-10 and 3-11.

Resistor values are nominal and in ohms



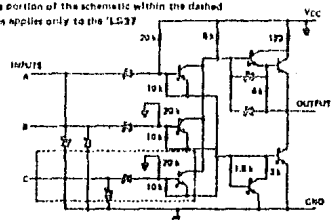
The portion of the schematic within the dashed lines is repeated for each additional input of the 'S260, and the 0.5-kΩ resistor is changed to 0.6 kΩ.

'S02, 'S260 CIRCUITS



'L02 CIRCUITS

The portion of the schematic within the dashed lines applies only to the 'LS27



'L02, 'LS27 CIRCUITS

54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

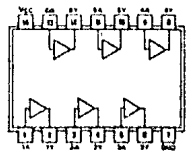
PIN ASSIGNMENTS (TOP VIEWS)

HEX BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

07

positive logic:
Y = A

See page 6-24



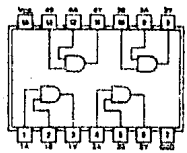
SN5407 (J, W) SN7407 (J, N)

QUADRUPLE 2-INPUT
POSITIVE-AND GATES

08

positive logic:
Y = AB

See page 6-12



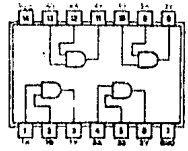
SN5408 (J, W) SN7408 (J, N)
SN54LS08 (J, W) SN74LS08 (J, N)
SN54S08 (J, W) SN74S08 (J, N)

QUADRUPLE 2-INPUT
POSITIVE-AND GATES
WITH OPEN-COLLECTOR OUTPUTS

09

positive logic:
Y = AB

See page 6-12



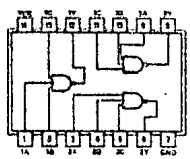
SN5409 (J, W) SN7409 (J, N)
SN54LS09 (J, W) SN74LS09 (J, N)
SN54S09 (J, W) SN74S09 (J, N)

TRIPLE 3-INPUT
POSITIVE-NAND GATES

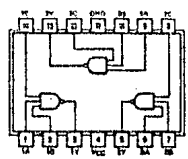
10

positive logic:
$Y = \overline{ABC}$

See page 6-2



SN5410 (J) SN7410 (J, N)
SN54H10 (J) SN74H10 (J, N)
SN54L10 (J) SN74L10 (J, N)
SN54LS10 (J, W) SN74LS10 (J, N)
SN54S10 (J, W) SN74S10 (J, N)



SN5410 (W)
SN54H10 (W)
SN54L10 (T)

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

recommended operating conditions

		54 FAMILY 74 FAMILY	SERIES 54 SERIES 74 '00,			SERIES 54H SERIES 74H 'H11, 'H21			SERIES 54LS SERIES 74LS 'LS08, 'LS11, 'LS21			SERIES 54S SERIES 74S 'S08, 'S11			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply Voltage, V_{CC}		54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
		74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}					−400			−500			−400			−1000	μA
Low-level output current, I_{OL}		54 Family			16			20			4			20	mA
		74 Family			16			20			8			20	
Operating free-air temperature, T_A		54 Family	−55		125	−55		125	−55		125	−55		125	°C
		74 Family	0		70	0		70	0		70	0		70	

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

PARAMETER		TEST FIGURE	TEST CONDITIONS†			SERIES 54 SERIES 74 '00			SERIES 54H SERIES 74H 'H11, 'H21			SERIES 54LS SERIES 74LS 'LS08, 'LS11, 'LS21			SERIES 54S SERIES 74S 'S08, 'S11			UNIT
						MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage	1, 2				2			2			2			2			V
V_{IL}	Low-level input voltage	1, 2			54 Family			0.8			0.8			0.7			0.8	V
					74 Family			0.8			0.8			0.8			0.8	
V_{IK}	Input clamp voltage	3	V_{CC} = MIN, I_I = I_{IK}					−1.5			−1.5			−1.5			−1.2	V
V_{OH}	High-level output voltage	1	V_{CC} = MIN, V_{IH} = 2 V, I_{OH} = MAX		54 Family	2.4	3.4		2.4	3.4		2.5	3.4		2.5	3.4		V
					74 Family	2.4	3.4		2.4	3.4		2.7	3.4		2.7	3.4		
V_{OL}	Low-level output voltage	2	V_{CC} = MIN, V_{IH} = 2 V	I_{OL} = MAX	54 Family		0.2	0.4		0.15	0.3		0.25	0.4			0.5	V
					74 Family		0.2	0.4		0.2	0.4		0.35	0.5			0.5	
				I_{OL} = 4 mA	Series 74LS								0.25	0.4				
I_I	Input current at maximum input voltage	4	V_{CC} = MAX		V_I = 5.5 V			1			0.1						1	mA
					V_I = 7 V									0.1				
I_{IH}	High-level input current	4	V_{CC} = MAX		V_{IH} = 2.4 V			40			50							μA
					V_{IH} = 2.7 V									20			50	
I_{IL}	Low-level input current	5	V_{CC} = MAX		V_{IL} = 0.4 V			−1.6			−2			−0.4				mA
					V_{IL} = 0.5 V												−2	
I_{OS}	Short-circuit output current§	6	V_{CC} = MAX		54 Family	−20		−55	−40		−100	−20		−100	−40		−100	mA
					74 Family	−18		−55	−40		−100	−20		−100	−40		−100	
I_{CC}	Supply current	7	V_{CC} = MAX			See table on next page												mA

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

‡All typical values are at V_{CC} = 25°C.

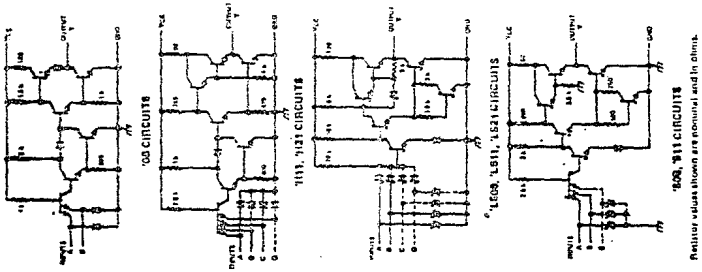
§I_I = −12 mA for SN54'/SN74', −8 mA for SN54H'/SN74H', and −18 mA for SN54LS'/SN74LS' and SN54S'/SN74S'.

¶Not more than one output should be shorted at a time, and for SN54H'/SN74H', SN54LS'/SN74LS' and SN54S'/SN74S', duration of output short circuit should not exceed one second.

16

POSITIVE-AND GATES WITH TOTEM-POLE OUTPUTS

schematics (each gate)



supply current†

TYPES	I_{CCH} (mA) Total with outputs high		I_{CCL} (mA) Total with outputs low		I_{CC} (mA) Average per gate (50% duty cycle)
	TYP	MAX	TYP	MAX	TYP
'08	11	21	20	33	3.09
'H11	18	30	30	48	6
'H21	12	20	25	32	8
'L03	2.4	4.8	4.4	6.6	0.65
'L11	1.8	3.0	3.3	5.5	0.65
'L21	1.2	2.4	2.2	4.4	0.65
'S03	19	31	32	51	6.25
'S11	13.5	24	24	42	6.25

†Measurement values of I_{CCH} are lower than recommended operating range of V_{CC} and T_A; typical values at V_{CC} = 5 V, T_A = 25°C.

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

TYPE	TEST CONDITIONS‡	t_{PLH} (ns) Propagation delay time, low-to-high-level output		t_{PHL} (ns) Propagation delay time, high-to-low-level output	
		MIN	TYP	MAX	TYP	MAX
'08	C_L = 15 pF, R_L = 400 Ω		17.5	27	12	19
'H11, 'H21	C_L = 25 pF, R_L = 280 Ω		7.0	12	6.0	12
'L08, 'L11, 'L21	C_L = 15 pF, R_L = 2 kΩ		8	15	10	20
'S08, 'S11	C_L = 15 pF, R_L = 280 Ω		4.5	7	6	7.5
	C_L = 50 pF, R_L = 280 Ω		8		7.5	

‡Load circuit and voltage waveforms are shown on pages 3-10 and 3-11.

recommended operating conditions

		54 FAMILY / 74 FAMILY	SERIES 54* / SERIES 74*												UNIT
			'08, '07			'10, '17			'20			'33, '38			
			MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	54 Family		4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
	74 Family		4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output voltage, V_{OH}					30			15			15			5.5	V
Low-level output current, I_{OL}	54 Family				30			30			10			48	mA
	74 Family				40			40			16			48	
Operating free-air temperature, T_A	54 Family		−55		125	−55		125	−55		125	−55		125	°C
	74 Family		0		70	0		70	0		70	0		70	

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

PARAMETER		TEST FIGURE	TEST CONDITIONS†		SERIES 54* / SERIES 74*								UNIT
					'08, '07		'10, '17		'20		'33, '38		
					MIN	TYP MAX	MIN	TYP MAX	MIN	TYP MAX	MIN	TYP MAX	
V_{IH}	High-level input voltage	1, 2			2		2		2		2		V
V_{IL}	Low-level input voltage	1, 2				0.8		0.8		0.8		0.8	V
V_{IK}	Input clamp voltage	3	V_{CC} = MIN, I_I = −12 mA			−1.5		−1.5		−1.5		−1.5	V
I_{OH}	High-level output current	1	V_{CC} = MIN, V_I = ‡	V_{OH} = 12 V		250				50			μA
				V_{OH} = MAX				250		1000		250	
V_{OL}	Low-level output voltage	2	V_{CC} = MIN, V_I = ‡	I_{OL} = 10 mA		0.4		0.4		0.4			V
				I_{OL} = MAX		0.7		0.7		0.4		0.4	
I_I	Input current at maximum input voltage	4	V_{CC} = MAX, V_I = 5.5 V			1		1		1		1	mA
I_{IH}	High-level input current	4	V_{CC} = MAX, V_{IH} = 2.4 V			40		40		40		40	μA
I_{IL}	Low-level input current	5	V_{CC} = MAX, V_{IL} = 0.4 V			−1.6		−1.6		−1.6		−1.6	mA
I_{CC}	Supply current	7	V_{CC} = MAX		See table on next page								mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ The input voltage is V_{IH} = 2 V or V_{IL} = V_{IL} max, as appropriate. See tables with test figures 1 and 2.

supply current†

TYPE	I_{CCH} (mA) Total with outputs high		I_{CCL} (mA) Total with outputs low		I_{CC} (mA) Average per gate 50% duty cycle
	TYP	MAX	TYP	MAX	TYP
'06, '16	30	42	32	61	5.17
'07, '17	29	41	21	30	4.17
'26	4	8	12	22	2.00
'33	12	21	33	57	5.83
'38	6	8.5	34	54	4.58

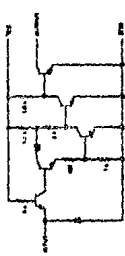
†Maximum values of I_{CC} shown are over the recommended operating ranges of V_{CC} and T_A. Typical values are at V_{CC} = 5 V, T_A = 25°C.

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

TYPE	TEST CONDITIONS‡	t_{PLH} (ns) Propagation delay time, low-to-high-level output		t_{PHL} (ns) Propagation delay time, high-to-low-level output	
		TYP	MAX	TYP	MAX
'06, '16	C_L = 15 pF, R_L = 110 Ω	10	15	15	23
'07, '17		6	10	20	30
'26	C_L = 15 pF, R_L = 1 kΩ	16	24	11	17
'33	C_L = 50 pF, R_L = 133 Ω	10	15	12	18
	C_L = 150 pF, R_L = 133 Ω	10	22	16	24
'38	C_L = 15 pF, R_L = 133 Ω	10	22	11	18

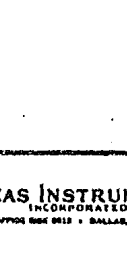
‡Load circuit and voltage waveforms are shown on page 3-10.

schematics (each gate)

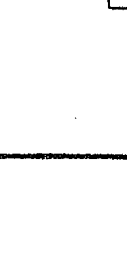


'06, '16 CIRCUITS

'07, '17 CIRCUITS



'26, '33 CIRCUITS



'38 CIRCUITS

CIRCUITS	R1	R2	R3
'26	4 kΩ	1.6 kΩ	1 kΩ
'38	4 kΩ	200 Ω	400 Ω

'26, '38 CIRCUITS

54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

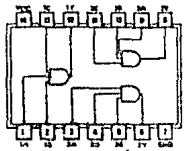
PIN ASSIGNMENTS (TOP VIEWS)

TRIPLE 3-INPUT POSITIVE-AND GATES

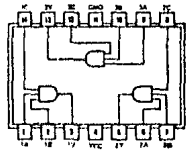
11

positive logic
$Y = ABC$

See page 6-13



SN54H11 (J) SN74H11 (J, N)
SN54LS11 (J, W) SN74LS11 (J, N)
SN54S11 (J, W) SN74S11 (J, N)



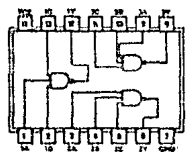
SN54S11 (W)

TRIPLE 3-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

12

positive logic
$Y = \overline{ABC}$

See page 6-4



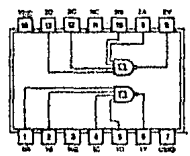
SN5412 (J, W) SN7412 (J, N)
SN54LS12 (J, W) SN74LS12 (J, N)

DUAL 4-INPUT POSITIVE-NAND SCHMITT TRIGGERS

13

positive logic
$Y = \overline{ABCD}$

See page 6-14



SN5413 (J, W) SN7413 (J, N)
SN54LS13 (J, W) SN74LS13 (J, N)

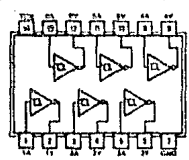
NC—No internal connection

HEX SCHMITT-TRIGGER INVERTERS

14

positive logic
$Y = \overline{A}$

See page 6-14



SN5414 (J, W) SN7414 (J, N)
SN54LS14 (J, W) SN74LS14 (J, N)

SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

recommended operating conditions

		SN74AS1* SN74AS1Y	SERIES 54 SERIES 74			'14, '132			SERIES 54S SERIES 74S			SERIES 54LS SERIES 74LS '5132, 'LS14, 'LS132			SERIES 54S SERIES 74S '5132			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
		74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}					-1.6			-1.6			-1.6			-0.4			-1.6	mA
Low-level output current, I_{OL}		54 Family			16			16			16			4			20	mA
		74 Family			16			16			16			8			20	
Operating free-air temperature, T_A		54 Family	-55		125	-55		125	-55		125	-55		125	-55		125	°C
		74 Family	0		70	0		70	0		70	0		70	0		70	

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

PARAMETER		TEST FIGURE	† TEST CONDITIONS‡			SERIES 54 SERIES 74						SERIES 54S SERIES 74S						SERIES 54LS SERIES 74LS '5132, 'LS14, 'LS132						SERIES 54S SERIES 74S '5132						UNIT
						'13			'14, '132			'13			'14, '132				'LS13, 'LS14, 'LS132						'5132						
						MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX		MIN	TYP‡	MAX				MIN	TYP‡	MAX				
V_{T+}	Positive-going threshold voltage	6	V_{CC} = 5 V			1.5	1.7	2	1.5	1.7	2	1.4	1.6	1.9	1.4	1.6	1.9		1.4	1.6	1.9				1.4	1.7	1.9				V
V_{T-}	Negative-going threshold voltage	6	V_{CC} = 5 V			0.8	0.9	1.1	0.8	0.9	1.1	0.5	0.8	1	0.5	0.8	1		0.3	0.5	0.7				0.3	0.7	1.2				V
V_{H}	Hysteresis ($V_{T+} - V_{T-}$)	6	V_{CC} = 5 V			0.4	0.8		0.4	0.8		0.4	0.8		0.4	0.8			0.2	0.55				0.2	0.55					V
V_{IK}	Input clamp voltage	3	V_{CC} = MIN,	I_I = -18 mA			-1.5			-1.5			-1.5			-1.5				-1.5					-1.2					V
V_{OH}	High-level output voltage	9	V_{CC} = MIN,	I_{OH} = MAX,	54 Family	2.4	3.4		2.4	3.4		2.5	3.4		2.5	3.4			2.5	3.4				2.5	3.4					V
			V_I = V_{T-}, min		74 Family	2.4	3.4		2.4	3.4		2.7	3.4		2.7	3.4			2.7	3.4				2.7	3.4					
V_{OL}	Low-level output voltage	9	V_{CC} = MIN,	I_{OL} = MAX,	54 Family		0.2	0.4		0.2	0.4		0.25	0.4		0.25	0.4			0.25	0.4					0.5				V
			V_I = V_{T+}, min		74 Family		0.2	0.4		0.2	0.4		0.25	0.5		0.25	0.5			0.25	0.5					0.5				
				I_{OL} = 4 mA	Series 74LS																									
I_{T+}	Input current at positive-going threshold	5	V_{CC} = 5 V,	V_I = V_{T+}			-0.85			-0.85			-0.43			-0.14				-0.14					-0.9					mA
I_{T-}	Input current at negative-going threshold	8	V_{CC} = 5 V,	V_I = V_{T-}			-0.25			-0.25			-0.18			-0.18				-0.18					-1.1					mA
I_I	Maximum input current	6	V_{CC} = MAX		V_I = 5.5 V			1			1																			mA
					V_I = 7 V									1			0.1													
I_{IH}	High-level input current	4	V_{CC} = MAX		V_I = 2.4 V			40			40																			μA
					V_I = 2.7 V									20			20													
I_{IL}	Low-level input current	8	V_{CC} = MAX		V_I = 0.4 V		-1	-1.6		-1.6	-1.2			-0.4			-0.4									-2				mA
					V_I = 0.5 V																									
I_{OS}	Short-circuit output current§	8	V_{CC} = MAX			-18		-55	-18		-55	-20		-100	-20		-100	-40			-100	-40			-100					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.
§ I_{T+} = -12 mA for SN54/SN74' and -18 mA for 'LS13, 'LS14, 'LS132, and '5132.
¶ Not more than one output should be shorted at a time, and for SN54LS/SN74LS' and '5132, duration of output short circuit should not exceed one second.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

supply current†

TYPE	I_{CCH} (mA) Total with outputs high		I_{CCL} (mA) Total with outputs low		I_{CC} (mA) Average per gate (50% duty cycle)
	TYP	MAX	TYP	MAX	TYP
'13	14	23	20	32	8.1
'14	22	38	39	60	5.1
'132	15	24	28	40	5.1
'LS13	2.9	5	4.1	7	1.75
'LS14	8.6	16	12	21	1.72
'LS132	5.9	11	8.2	14	1.75
'S132	28	44	44	63	9

† Maximum values of I_{CC} are over the recommended operating ranges of V_{CC} and T_A; typical values are at V_{CC} = 5 V, T_A = 25°C.

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

TYPE	TEST CONDITIONS	t_{PLH} (ns) Propagation delay time, low-to-high-level output			t_{PHL} (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'13	C_L = 15 pF, R_L = 400 Ω		18	27		15	22
'14, '132			15	22		15	22
'LS13			15	22		18	27
'LS14	C_L = 15 pF, R_L = 2 kΩ		15	22		15	22
'LS132			15	22		16	22
'S132	C_L = 15 pF, R_L = 280 Ω		7	10.5		5.5	13

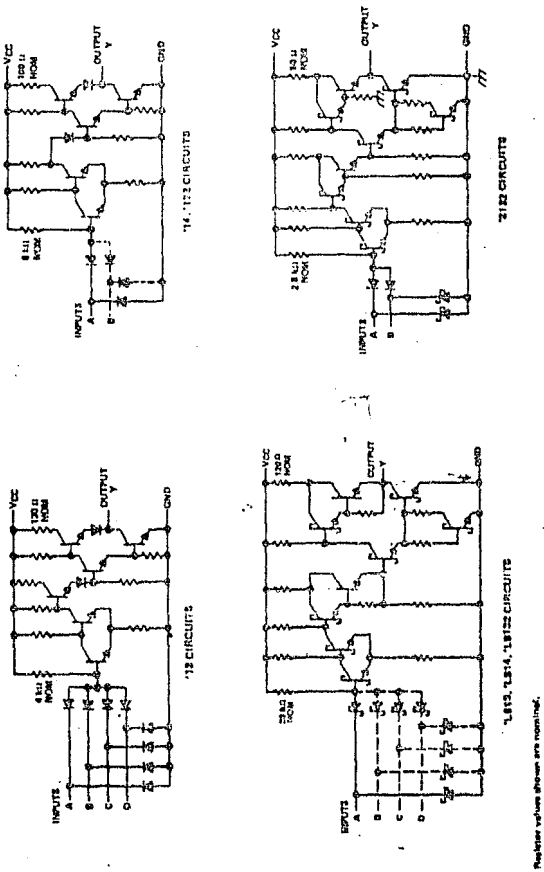
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT VOLTAGE WAVEFORMS

NOTES: A. All diodes are 1N916 or 1N3064.
B. C_L includes probe and jig capacitance.
C. Generator characteristics and reference voltages are:

	Generator Characteristics				Reference Voltages		
	Z_{out}	PRR	t_r	t_f	$V_{in\,ref(H)}$	$V_{in\,ref(L)}$	$V_{O\,ref}$
'H54'/'H74'	50 Ω	1 MHz	10 ns	10 ns	1.7 V	0.9 V	1.5 V
'LS54'/'LS74'	50 Ω	1 MHz	15 ns	6 ns	1.6 V	0.8 V	1.3 V
'S132	50 Ω	1 MHz	2.5 ns	2.5 ns	1.5 V	1.2 V	1.5 V

SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS



schematics (each gate)

Resistor values shown are nominal.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

TYPICAL CHARACTERISTICS OF '13, '14, AND '132 CIRCUITS†

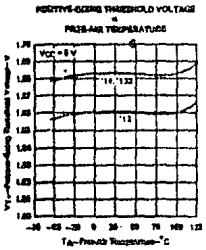


FIGURE 1

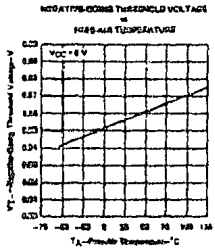


FIGURE 2

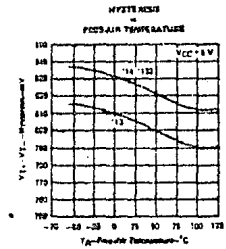


FIGURE 3

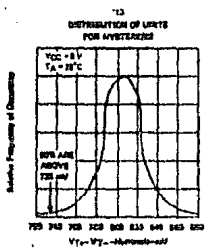


FIGURE 4

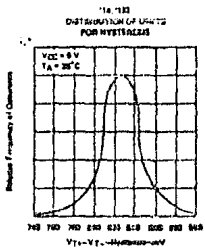


FIGURE 5

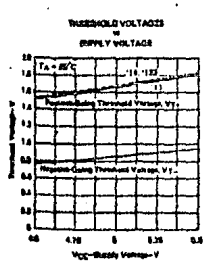


FIGURE 6

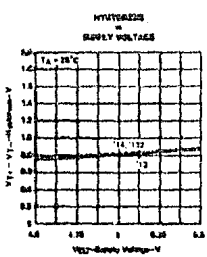


FIGURE 7

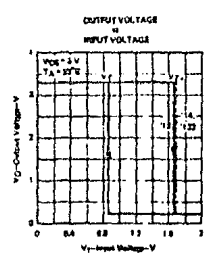


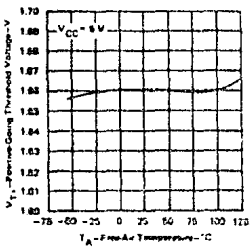
FIGURE 8

†Data for temperatures below 0°C and 70°C and supply voltages below 4.75V and above 5.25 V are applicable for SN5413, SN5414, and SN54132 only.

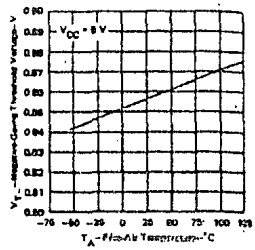
SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

TYPICAL CHARACTERISTICS OF 'LS13, 'LS14, AND 'LS132 CIRCUITS†

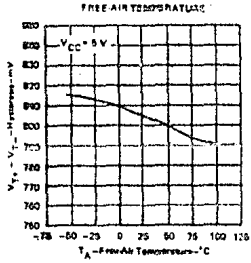
POSITIVE-GOING THRESHOLD VOLTAGE vs FREE-AIR TEMPERATURE



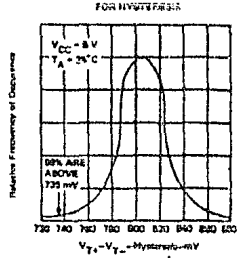
NEGATIVE-GOING THRESHOLD VOLTAGE vs FREE-AIR TEMPERATURE



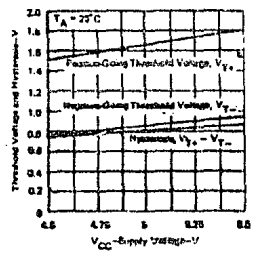
HYSTERESIS vs FREE-AIR TEMPERATURE



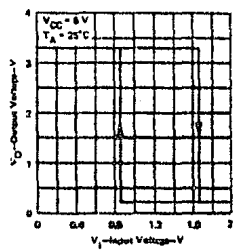
DISTRIBUTION OF UNITS FOR HYSTERESIS



THRESHOLD VOLTAGES AND HYSTERESIS vs SUPPLY VOLTAGE



OUTPUT VOLTAGE vs INPUT VOLTAGE

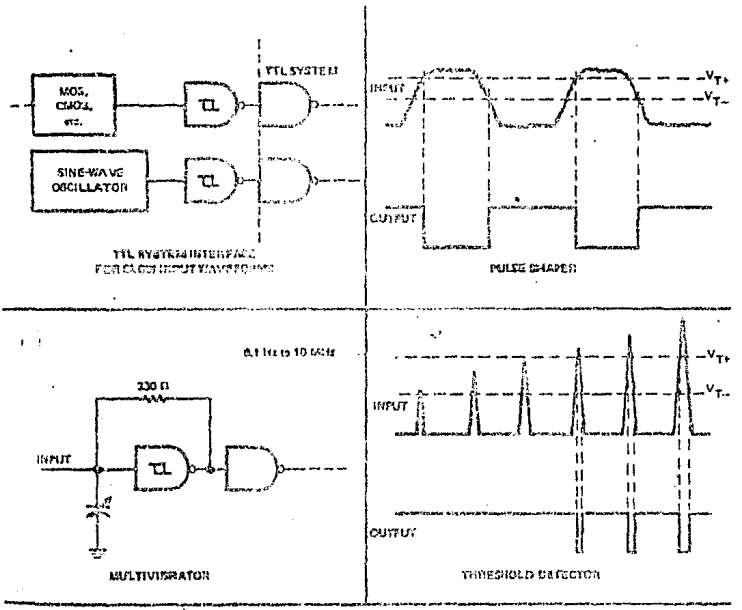


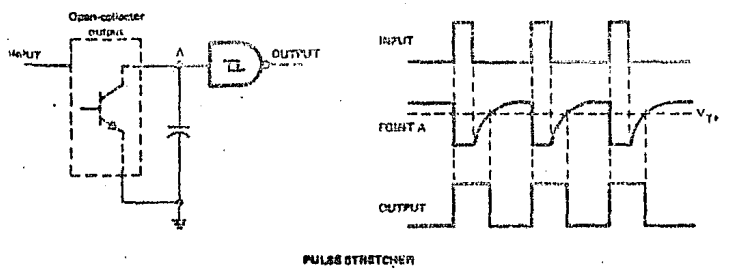
† Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 V are applicable for SN54LS13, SN54LS14, and SN54LS132 only.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

TYPICAL APPLICATION DATA





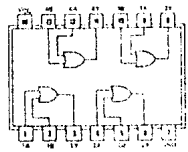
PIN ASSIGNMENTS (TOP VIEWS)

QUADRUPLE 2-INPUT
POSITIVE-OR GATES

32

positive logic:
$Y = A + B$

See page 6-23



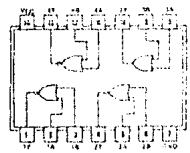
SN5432 (J, W) SN7432 (J, N)
SN54LS32 (J, W) SN74LS32 (J, N)
SN54S32 (J, W) SN74S32 (J, N)

QUADRUPLE 2-INPUT
POSITIVE-NOR BUFFERS
WITH OPEN-COLLECTOR OUTPUTS

33

positive logic:
$Y = \overline{A+B}$

See pages 6-24 and 6-28



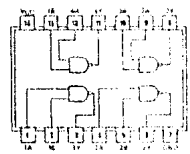
SN5433 (J, W) SN7433 (J, N)
SN54LS33 (J, W) SN74LS33 (J, N)

QUADRUPLE 2-INPUT
POSITIVE-NAND BUFFERS

37

positive logic:
$Y = \overline{AB}$

See page 6-29



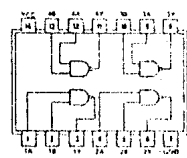
SN5437 (J, W) SN7437 (J, N)
SN54LS37 (J, W) SN74LS37 (J, N)
SN54S37 (J, W) SN74S37 (J, N)

QUADRUPLE 2-INPUT
POSITIVE-NAND BUFFERS
WITH OPEN-COLLECTOR OUTPUTS

38

positive logic:
$Y = \overline{AB}$

See pages 6-24 and 6-28



SN5438 (J, W) SN7438 (J, N)
SN54LS38 (J, W) SN74LS38 (J, N)
SN54S38 (J, W) SN74S38 (J, N)

recommended operating conditions

		54 FAMILY 74 FAMILY	SERIES 54 SERIES 74 '32			SERIES 54LS SERIES 74LS 'LS32			SERIES 54S SERIES 74S 'S32			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.25	V
		74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}					-800			-400			-1000	μA
Low-level output current, I_{OL}		54 Family			16			4			20	mA
		74 Family			16			8			20	
Operating free-air temperature, T_A		54 Family	-55		125	-55		125	-55		125	°C
		74 Family	0		70	0		70	0		70	

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

PARAMETER			TEST FIGURE	TEST CONDITIONS†			SERIES 54 SERIES 74 '32			SERIES 54LS SERIES 74LS 'LS32			SERIES 54S SERIES 74S 'S32			UNIT
							MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage		1, 2				2			2			2			V
V_{IL}	Low-level input voltage		1, 2			54 Family			0.8			0.7			0.8	V
						74 Family			0.8			0.8			0.8	
V_{IK}	Input clamp voltage		3	V_{CC} = MIN,	I_I = I_{IK}				-1.5			-1.5			-1.2	V
V_{OH}	High-level output voltage		1	V_{CC} = MIN, I_{OH} = MAX	V_{IH} = 2 V,	54 Family	2.4	3.4		2.5	3.4		2.5	3.4		V
						74 Family	2.4	3.4		2.7	3.4		2.7	3.4		
V_{OL}	Low-level output voltage		2	V_{CC} = MIN, V_{IL} = V_{IL} max	I_{OL} = MAX	54 Family		0.2	0.4		0.25	0.4			0.5	V
						74 Family		0.2	0.4		0.33	0.5			0.5	
					I_{OL} = 4 mA	Series 74LS					0.25	0.4				
I_I	Input current at maximum input voltage		4	V_{CC} = MAX		V_I = 5.5 V			1						1	mA
						V_I = 7 V						0.1				
I_{IH}	High-level input current		4	V_{CC} = MAX		V_{IH} = 2.4 V			40							μA
						V_{IH} = 2.7 V						20			50	
I_{IL}	Low-level input current		5	V_{CC} = MAX		V_{IL} = 0.4 V			-1.6			-0.4				mA
						V_{IL} = 0.5 V									-2	
I_{OS}	Short-circuit output current§		6	V_{CC} = MAX		54 Family	-20		-55	-20		-100	-40		-100	mA
						74 Family	-18		-55	-20		-100	-40		-100	
I_{CC}	Supply current	Total, outputs high	7	V_{CC} = MAX				15	22		3.1	5.2		18	32	mA
		Total, outputs low						22	33		4.9	9.9		38	62	
		Average per gate		V_{CC} = 5 V,	50% duty cycle			4.75			1.0			7		

†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.
‖I_I = −12 mA (for SN54'/SN74' and −18 mA for SN54LS'/SN74LS' and SN54S'/SN74S').
§Not more than one output should be shorted at a time, and for SN54LS'/SN74LS' and SN54S'/SN74S', duration of the short-circuit should be less than one second.

schematics (each gate)

Resistor values shown are nominal and in ohms.

switching characteristics at VCC = 5 V, TA = 25°C

TYPE	TEST CONDITIONS*	t_{PLH} (ns) Propagation delay time, low-to-high-level output			t_{PHL} (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'32	C_L = 15 pF, R_L = 400 Ω		10	15		14	22
'LS32	C_L = 15 pF, R_L = 2 kΩ		14	22		14	22
	C_L = 15 pF, R_L = 280 Ω		4	7		4	7
'S32	C_L = 50 pF, R_L = 280 Ω		5			5	

*Load circuit and voltage waveforms are shown on pages 3-10 and 3-11.

PIN ASSIGNMENTS (TOP VIEWS)

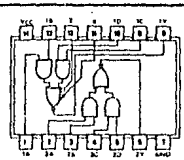
DUAL 2-WIDE 2-INPUT AND-OR-INVERT GATES (ONE GATE EXPANDABLE)

50

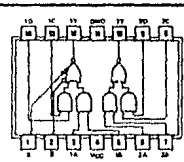
positive logic:
$Y = \overline{AB+CD+X}$

'50: X = output of SN5460/SN7460
'H50: X = output of SN54H60/SN74H60 or SN54H62/SN74H62

See page 6-33



SN5450 (J) SN7450 (J, N)
SN54H50 (J) SN74H50 (J, N)

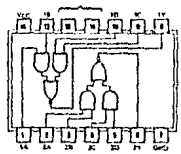


SN5450 (W)
SN54H50 (W)

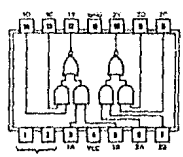
AND-OR-INVERT GATES

51

'51, 'H51, 'S51
DUAL 2-WIDE 2-INPUT
positive logic:
$Y = \overline{AB+CD}$



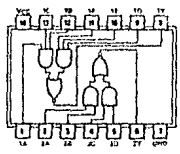
SN5451 (J) SN7451 (J, N)
SN54H51 (J) SN74H51 (J, N)
SN54S51 (J, W) SN74S51 (J, N)



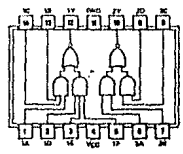
SN5451 (W)
SN54H51 (W)

'LS1, 'LS51
2-WIDE 3-INPUT,
2-WIDE 2-INPUT
positive logic:
$1Y = \overline{(1A \cdot 1B \cdot 1C)+(1D \cdot 1E \cdot 1F)}$
$2Y = \overline{(2A \cdot 2B)+(2C \cdot 2D)}$

See page 6-30



SN54LS51 (J) SN74LS51 (J, N)
SN54LS51 (J, W) SN74LS51 (J, W)



SN54LS51 (T)

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

recommended operating conditions

	54 FAMILY 74 FAMILY	SERIES 54 SERIES 74 '21, '54			SERIES 54H SERIES 74H 'H51, 'H54			SERIES 54L SERIES 74L 'L51, 'L54, 'L55			SERIES 54LS SERIES 74LS 'LS51, 'LS54, 'LS55			SERIES 54S SERIES 74S 'S51, 'S54			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
	74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}	54 Family			−400			−500			−100			−400			−1000	mA
	74 Family			−400			−500			−200			−400			−1000	
Low-level output current, I_{OL}	54 Family			16			20			3			4			20	mA
	74 Family			16			20			3.6			8			20	
Operating free-air temperature, T_A	54 Family	−55		125	−55		125	−55		125	−55		125	−55		125	°C
	74 Family	0		70	0		70	0		70	0		70	0		70	

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

PARAMETER	TEST FIGURE	TEST CONDITIONS†		SERIES 54 SERIES 74 '21, '54			SERIES 54H SERIES 74H 'H51, 'H54			SERIES 54L SERIES 74L 'L51, 'L54, 'L55			SERIES 54LS SERIES 74LS 'LS51, 'LS54 'LS55			SERIES 54S SERIES 74S 'S51, 'S54			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH} High-level input voltage	1, 2			2			2			2			2			2			V
V_{IL} Low-level input voltage	1, 2		54 Family			0.8			0.8			0.7			0.7			0.8	V
			74 Family			0.8			0.8			0.7			0.8			0.8	
V_{IK} Input clamp voltage	3	V_{CC} = MIN, I_I = I_{IK}				−1.5			−1.5			−1.5			−1.5			−1.2	V
V_{OH} High-level output voltage	1	V_{CC} = MIN, V_{IL} = V_{IL} max, I_{OH} = MAX	74 Family	2.4	3.4		2.4	3.4		2.4	3.3		2.5	3.4		2.5	3.4		V
			54 Family	2.4	3.4		2.4	3.4		2.4	3.2		2.7	3.4		2.7	3.4		
V_{OL} Low-level output voltage	2	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = MAX	54 Family		0.2	0.4		0.2	0.4		0.15	0.2		0.25	0.4			0.5	V
			74 Family		0.2	0.4		0.2	0.4		0.2	0.4		0.35	0.5			0.5	
		I_{OL} = 4 mA	Series 54LS											0.25	0.4				
I_I Input current at maximum input voltage	4	V_{CC} = MAX	V_I = 5.5 V			1			1			0.1			0.1			1	mA
			V_I = 7 V												0.1				
I_{IH} High-level input current	4	V_{CC} = MAX	V_{IH} = 2.4 V			40			50			10			20			50	µA
			V_{IH} = 2.7 V												20				
I_{IL} Low-level input current	5	V_{CC} = MAX	V_{IL} = 0.3 V									−0.18							mA
			V_{IL} = 0.4 V			−1.6			−2						−0.4				
			V_{IL} = 0.5 V															−2	
I_{OS} Short-circuit output current‖	6	V_{CC} = MAX	54 Family	−20		−55	−40		−100	−3		−15	−20		−100	−40		−100	mA
			74 Family	−18		−55	−40		−100	−3		−15	−20		−100	−40		−100	
I_{CC} Supply current	7	V_{CC} = MAX		See table on next page															mA

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

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

§ I_I = −12 mA for SN54'/SN74', −8 mA for SN54H'/SN74H', and −18 mA for SN54LS'/SN74LS' and SN54S'/SN74S'.

¶ Not more than one output should be shorted at a time, and for SN54LS'/SN74LS', SN54H'/SN74H', and SN54S'/SN74S', duration of the short-circuit should not exceed one second.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

supply currents†

TYPE	I_{CCH} (mA) Total with outputs high		I_{CCL} (mA) Total with outputs low		I_{CC} (mA) Average per AOI gate (50% duty cycle)
	TYP	MAX	TYP	MAX	TYP
'51	4	8	7.4	14	2.25
'54	4	8	8.1	15	4.55
'H51	8.2	12.8	15.2	24	5.65
'H54	7.1	11	9.4	14	9.25
'LS51	0.44	0.8	0.75	1.3	0.70
'LS4	0.29	0.8	0.60	0.99	0.60
'LS5	0.22	0.4	0.38	0.55	0.30
'LS51	0.8	1.6	1.4	2.6	0.65
'LS54	0.8	1.6	1.0	2	0.9
'LS55	0.4	0.8	0.7	1.3	0.65
'551	8.2	17.8	13.6	22	5.45
'554	7	12.5	8.5	15	3.75

† Maximum value of I_{CC} are over the recommended operating range of V_{CC} and T_A; typical values are at V_{CC} = 5 V, T_A = 25°C.

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

TYPE	TEST CONDITIONS*	t_{PLH} (ns) Propagation delay time, low-to-high-level output			t_{PHL} (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'51, '54	C_L = 15 pF, R_L = 400 Ω		13	22		8	15
'H51	C_L = 25 pF, R_L = 280 Ω		6.8	11		6.2	11
'H54	C_L = 25 pF, R_L = 280 Ω		7	11		6.2	11
'LS51, 'LS4, 'LS5	C_L = 50 pF, R_L = 4 kΩ		60	90		35	60
'LS51, 'LS55	C_L = 15 pF, R_L = 2 kΩ		12	20		12.5	20
'LS54	C_L = 15 pF, R_L = 2 kΩ		12	20		12.5	20
'551, '554	C_L = 15 pF, R_L = 280 Ω		3.5	6.5		3.6	6.5
	C_L = 25 pF, R_L = 280 Ω			6		6.6	

* Load circuit and voltage waveforms are shown on page 3-10 and 3-11.

schematics (each gate)

The portion of the circuit within the dashed lines is repeated (with as many emitter or input diodes as applicable) for each additional AND section.

Resistor values shown are nominal and in ohms.

'51, 'H51 CIRCUITS

'LS51, 'LS54, 'LS55 CIRCUITS

'551, '554 CIRCUITS

'51, 'H54, 'LS4 CIRCUITS

'H51, 'H54 CIRCUITS

PIN ASSIGNMENTS (TOP VIEWS)

AND-GATED J-K MASTER-SLAVE FLIP-FLOPS WITH PRESET AND CLEAR

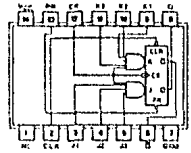
72

FUNCTION TABLE

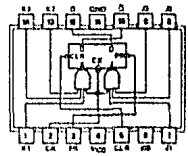
INPUTS					OUTPUTS	
PRESET	CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H*	H*
H	H	⎍	L	L	Q_0	$\bar{Q}_0$
H	H	⎍	H	L	H	L
H	H	⎍	L	H	L	H
H	H	⎍	H	H	TOGGLE	

positive logic: J = J1·J2·J3; K1·K2·K3

See pages 6-49, 6-50, and 6-54



SN5472 (J) SN7472 (J, N) SN54H72 (W)
SN54H72 (J) SN74H72 (J, N) SN54S72 (W)
SN54L72 (J) SN74L72 (J, N) SN54L72 (W)



NC—No internal connection

DUAL J-K FLIP-FLOPS WITH CLEAR

73

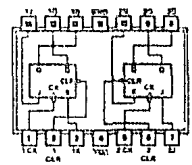
'73, 'H73, 'L73 FUNCTION TABLE

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	⎍	L	L	Q_0	$\bar{Q}_0$
H	⎍	H	L	H	L
H	⎍	L	H	L	H
H	⎍	H	H	TOGGLE	

'LS73A FUNCTION TABLE

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	↓	L	L	Q_0	$\bar{Q}_0$
H	↓	H	L	H	L
H	↓	L	H	L	H
H	↓	H	H	TOGGLE	
H	H	X	X	Q_0	$\bar{Q}_0$

See pages 6-49, 6-50, 6-54, and 6-55



SN5473 (J, W) SN7473 (J, N)
SN54H73 (J, W) SN74H73 (J, N)
SN54L73 (J, T) SN74L73 (J, N)
SN54LS73A (J, W) SN74LS73A (J, N)

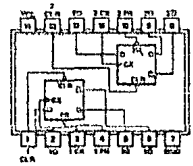
DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

74

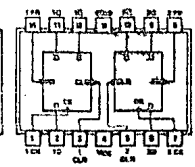
FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\bar{Q}_0$

See pages 6-49, 6-50, 6-54, and 6-55



SN5474 (J) SN7474 (J, N) SN54S74 (W)
SN54H74 (J) SN74H74 (J, N) SN54H76 (W)
SN54L74 (J) SN74L74 (J, N) SN54L76 (W)
SN54LS74A (J, W) SN74LS74A (J, N)
SN54S74 (J, W) SN74S74 (J, N)



See explanation of function tables on page 3-8.

* This configuration is nonstable; that is, it will not persist when preset and clear inputs return to their inactive (high) level.

recommended operating conditions

		SERIES 54LS/74LS	'LS73A, 'LS73A, 'LS112A			'L74A			'LS76A, 'LS112A			LS78A, LS114A			LS109A			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		Series 54LS	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
		Series 74LS	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}					−400			−400			−400			−400			−400	μA
Low-level output current, I_{OL}		Series 54LS			4			4			4			4			4	mA
		Series 74LS			8			8			8			8			8	
Clock frequency, f_{clock}			0		30	0		25	0		30	0		30	0		25	MHz
Pulse width, t_w	Clock high		20			35			20			20			25			ns
	Preset or clear low		25			25			25			25			25			
Setup time, t_{su}	High-level data		20↓			25↓			20↓			20↓			20↓			ns
	Low-level data		20↑			25↑			20↑			20↑			25↑			
Hold time, t_h			0↓			5↑			0↓			0↓			5↑			ns
Operating free-air temperature, T_A		Series 54LS	−55		125	−55		125	−55		125	−55		125	−55		125	°C
		Series 74LS	0		70	0		70	0		70	0		70	0		70	

↑↓ The arrow indicates the edge of the clock pulse used for reference: ↑ for the rising edge, ↓ for the falling edge.

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

PARAMETER			TEST CONDITIONS†		'LS73A, 'LS83A, 'LS112A			'L74A			'LS76A, 'LS112A			'LS78A, LS114A			'LS109A			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage				2			2			2			2			2			V
V_{IL}	Low-level input voltage	Series 54LS					0.7			0.7			0.7			0.7			0.7	V
		Series 74LS					0.8			0.8			0.8			0.8			0.8	
V_{IK}	Input clamp voltage		V_{CC} = MIN,	I_I = −18 mA			−1.5			−1.5			−1.5			−1.5			−1.5	V
V_{OH}	High-level output voltage	Series 54LS	V_{CC} = MIN,	V_{IH} = 2 V,	2.5	3.4		2.5	3.4		2.5	3.4		2.5	3.4		2.5	3.4		V
		Series 74LS	V_{IL} = V_{IL} max,	I_{OH} = −400 μA	2.7	3.4		2.7	3.4		2.7	3.4		2.7	3.4		2.7	3.4		
V_{OL}	Low-level output voltage	Series 54LS	V_{CC} = MIN,	I_{OL} = MAX		0.25	0.4		0.25	0.4		0.25	0.4		0.25	0.4		0.25	0.4	V
		Series 74LS	V_{IL} = V_{IL} max,			0.35	0.5		0.35	0.5		0.35	0.5		0.35	0.5		0.35	0.5	
		Series 74LS	V_{IH} = 2 V	I_{OL} = 4 mA		0.25	0.4		0.25	0.4		0.25	0.4		0.25	0.4		0.25	0.4	
I_I	Input current at maximum input voltage	D, J, K, or R	V_{CC} = MAX,	V_I = 7 V			0.1			0.1			0.1			0.1			0.1	mA
		Clear					0.3			0.7			0.3			0.6			0.3	
		Preset					0.3			0.2			0.3			0.3			0.2	
		Clock					0.4			0.1			0.4			0.2			0.1	
I_{IH}	High-level input current	D, J, K, or R	V_{CC} = MAX,	V_I = 2.7 V			20			20			20			20			20	μA
		Clear					60			40			60			120			40	
		Preset					60			40			60			60			40	
		Clock					60			20			60			100			20	
I_{IL}	Low-level input current	D, J, K, or R	V_{CC} = MAX,	V_I = 0.4 V			−0.4			−0.4			−0.4			−0.4			−0.4	mA
		Clear					−0.8			−0.8			−0.8			−1.6			−0.8	
		Preset					−0.8			−0.8			−0.8			−0.8			−0.8	
		Clock					−0.8			−0.4			−0.8			−1.6			−0.4	
I_{OS}	Short-circuit output current§	Series 54LS	V_{CC} = MAX		−20		−100	−20		−100	−20		−100	−20		−100	−20		−100	mA
		Series 74LS			−20		−100	−20		−100	−20		−100	−20		−100	−20		−100	
I_{CC}	Supply current (Total)		V_{CC} = MAX,	See Note 1		6	9		8	12		6	9		8	12		6	9	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 of short-circuit should not exceed one second.
NOTE 1: With all outputs open, I_{CC} is measured with the Q and $\overline{Q}$ outputs high in turn. At the time of measurement, the clock input is grounded.

TEXAS INSTRUMENTS
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

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

PARAMETER†	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'LS73A, 'LS76A, 'LS78A, 'LS107A, 'LS112A, 'LS113A, 'LS114A			'LS74A, 'LS109A			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}	Clock		C_L = 15 pF,	30	45		25	33		MHz
t_{PLH}	Clear, preset, or	Q or $\overline{Q}$	R_L = 2 kΩ,		15	20		13	25	ns
t_{PHL}	clock (as appropriate)		See Note 2		15	20		15	40	ns

f_{max} = maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 2: Load circuit and voltage waveforms are shown on page 3-11.

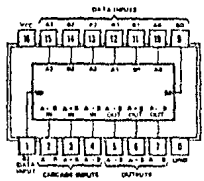
functional block diagrams and schematics of inputs and outputs

schematics of 'LS74A and 'LS109A

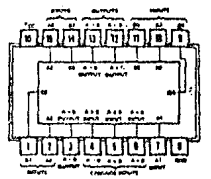
PIN ASSIGNMENTS (TOP VIEWS)

4-BIT MAGNITUDE COMPARATORS

85



SN5485 (J, W) SN7485 (J, N)
SN54LS85 (J, W) SN74LS85 (J, N)
SN54S85 (J, W) SN74S85 (J, N)

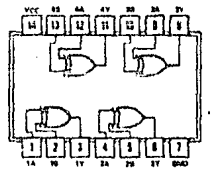


SN54L85 (J) SN74L85 (J, N)

See page 7-57

QUADRUPLE 2-INPUT EXCLUSIVE-OR GATES

86 $Y = A \oplus B = \overline{A}B + A\overline{B}$



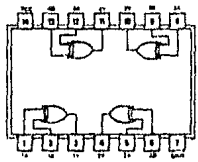
SN5486 (J, W) SN7486 (J, N)
SN54LS86 (J, W) SN74LS86 (J, N)
SN54S86 (J, W) SN74S86 (J, N)

FUNCTION TABLE

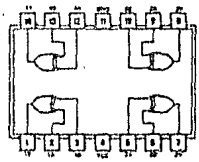
INPUTS		OUTPUT
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = high level, L = low level

See page 7-65



SN54L86 (J) SN74L86 (J, N)



SN54LS86 (T)

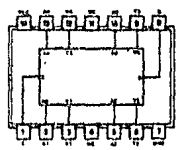
4-BIT TRUE/COMPLEMENT, ZERO/ONE ELEMENTS

87

FUNCTION TABLE

CONTROL INPUTS		OUTPUTS			
B	C	Y1	Y2	Y3	Y4
L	L	$\overline{A1}$	$\overline{A2}$	$\overline{A3}$	$\overline{A4}$
L	H	A1	A2	A3	A4
H	L	H	H	H	H
H	H	L	L	L	L

H = high level, L = low level
A1, A2, A3, A4 = the level of the respective A input



SN54H87 (J, W) SN74H87 (J, N)
NC—No internal connection

See page 7-70

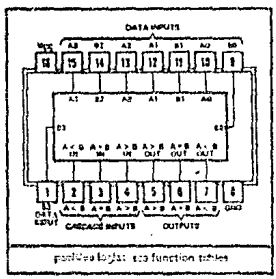
TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

TYPES SN5485, SN54L85, SN54LS85, SN54S85, SN7485, SN74L85, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS

BULLETIN NO. DL-S 7611810, MARCH 1974 - REVISED OCTOBER 1976

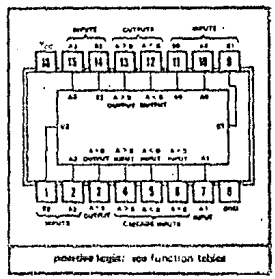
TYPE	TYPICAL POWER DISSIPATION	TYPICAL DELAY (4-BIT WORDS)
'85	275 mW	23 ns
'L85	20 mW	90 ns
'LS85	52 mW	24 ns
'S85	305 mW	11 ns

SN5485, SN54L85, SN54S85 . . . J OR W PACKAGE
SN7485, SN74L85, SN74S85 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function tables

SN54LS85 . . . J PACKAGE
SN74LS85 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function tables

description

These four-bit magnitude comparators perform comparison of straight binary and straight BCD (8-4-2-1) codes. Three fully decoded decisions about two 4-bit words (A, B) are made and are externally available at three outputs. These devices are fully expandable to any number of bits without external gates. Words of greater length may be compared by connecting comparators in cascade. The A > B, A < B, and A = B outputs of a stage handling less significant bits are connected to the corresponding A > B, A < B, and A = B inputs of the next stage handling more-significant bits. The stage handling the least-significant bits must have a high-level voltage applied to the A = B input and in addition for the 'L85, low-level voltages applied to the A > B and A < B inputs. The cascading paths of the '85, 'LS85, and 'S85 are implemented with only a two-gate-level delay to reduce overall comparison times for long words. An alternate method of cascading which further reduces the comparison time is shown in the typical application data.

FUNCTION TABLES

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H

'85, 'LS85, 'S85

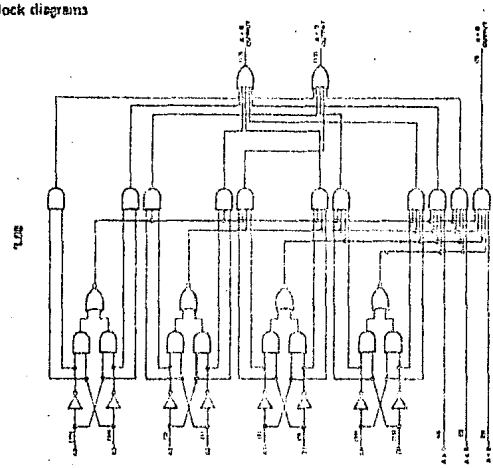
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

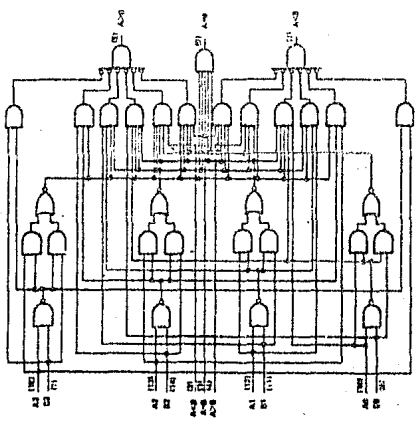
'L85

A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	H	L	H	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	H	H	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	H	H	H	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	H	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	L	L	L

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

functional block diagrams

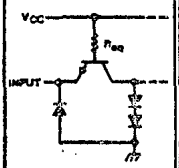




TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

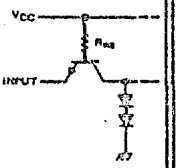
schematics of inputs and outputs

EQUIVALENT OF EACH INPUT FOR '85



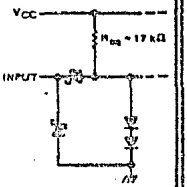
A = B, Any A or B:
R_{eq} = 1.67 kΩ NOM
A > B, A < B:
R_{eq} = 4 kΩ NOM

EQUIVALENT OF EACH INPUT FOR 'L85

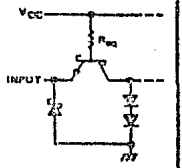


Any A or B:
R_{eq} = 16.7 kΩ NOM
A = B, A > B, A < B:
R_{eq} = 40 kΩ NOM

EQUIVALENT OF EACH INPUT FOR 'LS85

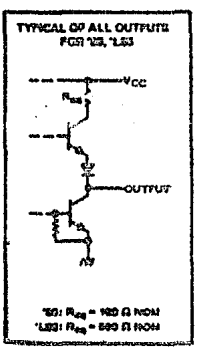


EQUIVALENT OF EACH INPUT FOR 'S85



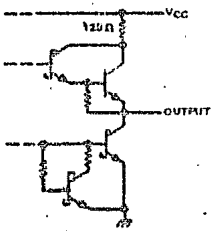
A = B, Any A or B:
R_{eq} = 633 Ω NOM
A > B, A < B:
R_{eq} = 2.8 kΩ NOM

TYPICAL OF ALL OUTPUTS FOR '85, 'L85

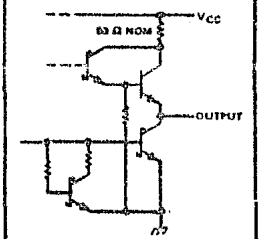


'85: R_{eq} = 100 Ω NOM
'L85: R_{eq} = 500 Ω NOM

TYPICAL OF ALL OUTPUTS FOR 'LS85



TYPICAL OF ALL OUTPUTS FOR 'S85



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

	SN54* SN54S*	SN54L*	SN54LS*	SN74* SN74S*	SN74L*	SN74LS*	UNIT
Supply voltage, V_{CC} (see Note 1)	7	8	7	7	8	7	V
Input voltage (see Note 2)	5.5	5.5	7	5.5	5.5	7	V
Interemitter voltage (see Note 3)	5.5			5.5			V
Operating free-air temperature range	−55 to 125			0 to 70			°C
Storage temperature range	−65 to 150			−65 to 150			°C

NOTES: 1. Voltage values, except interemitter voltage, are with respect to network ground terminal.
2. Input voltages for 'L85 must be zero or positive with respect to network ground terminal.
3. This is the voltage between two emitters of a multiple-emitter input transistor. This rating applies to each A input in conjunction with its respective B input of the '85 and 'L85.

recommended operating conditions

	SN5485			SN7485			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}			16			16	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†			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		V_{CC} = MIN,	I_I = −12 mA				−1.5	V
V_{OH} High-level output voltage		V_{CC} = MIN, V_{IL} = 0.8 V,	V_{IH} = 2 V, I_{OH} = −400 μA		2.4	3.4		V
V_{OL} Low-level output voltage		V_{CC} = MIN, V_{IL} = 0.8 V,	V_{IH} = 2 V, I_{OL} = 16 mA			0.2	0.4	V
I_I Input current at maximum input voltage		V_{CC} = MAX,	V_I = 5.5 V				1	mA
I_{IH} High-level input current	A < B, A > B inputs	V_{CC} = MAX,	V_I = 2.4 V				40	μA
	all other inputs						120	
I_{IL} Low-level input current	A < B, A > B inputs	V_{CC} = MAX,	V_I = 0.4 V				−1.6	mA
	all other inputs						−4.8	
I_{OS} Short-circuit output current§		V_{CC} = MAX,	V_O = 0	SN5485	−20		−55	mA
				SN7485	−18		−55	
I_{CC} Supply current		V_{CC} = MAX,	See Note 4			66	83	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.
NOTE 4: I_{CC} is measured with outputs open, A = B grounded, and all other inputs at 4.5 V.

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

PARAMETER¶	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B data input	A < B, A > B	1	C_L = 15 pF, R_L = 400 Ω, See Note 5		7		ns
			2			13		
			3			13	20	
		A = B	4			23	35	
t_{PHL}	Any A or B data input	A < B, A > B	1			11		ns
			2			16		
			3			20	30	
		A = B	4			20	30	
t_{PLH}	A < B or A = B	A > B	1			7	11	ns
t_{PHL}	A < B or A = B	A > B	1			11	17	ns
t_{PLH}	A = B	A = B	2			13	20	ns
t_{PHL}	A = B	A = B	2			11	17	ns
t_{PLH}	A > B or A = B	A < B	1			7	11	ns
t_{PHL}	A > B or A = B	A < B	1			11	17	ns

¶t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 5: Load circuit and voltage waveforms are shown on page 3-15.

recommended operating conditions

	SN54L85			SN74L85			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}			−100			−200	μA
Low-level output current, I_{OL}			2			3.6	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

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

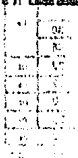
PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH} High-level input voltage				2			V
V_{IL} Low-level input voltage						0.7	V
V_{OH} High-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.7 V, I_{OH} = MAX	SN54L85	2.4	3.3		V
			SN74L85	2.4	3.3		
V_{OL} Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.7 V, I_{OL} = MAX	SN54L85		0.15	0.3	V
			SN74L85		0.2	0.4	
I_I Input current at maximum input voltage	A < B, A > B, or A = B	V_{CC} = MAX, V_I = 5.5 V				100	μA
	A or B inputs					200	
I_{IH} High-level input current	A < B, A > B, or A = B	V_{CC} = MAX, V_I = 2.4 V				10	μA
	A or B inputs					30	
I_{IL} Low-level input current	A < B, A > B, or A = B	V_{CC} = MAX, V_I = 0.3 V				−0.18	mA
	A or B inputs					−0.54	
I_{OS} Short-circuit output current§		V_{CC} = MAX		−3		−15	mA
I_{CC} Supply current		V_{CC} = MAX, See Note 6	Condition A		4.0	7.7	mA
			Condition B		3.2	7.2	

† 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.
NOTE 6: With all outputs open, I_{CC} is measured for Condition A with all inputs at 4.5 V, and for Condition B with all inputs grounded.

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

PARAMETER†	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B	Any	C_L = 50 pF, R_L = 4 kΩ, See Note 7		90	150	ns
t_{PHL}					75	150	
t_{PLH}	A > B, A < B, or A = B	Any			75	150	ns
t_{PHL}					93	100	

† t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 7: Load circuit and voltage waveforms are shown on page 3-11.



recommended operating conditions

	SN54LS85			SN74LS85			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
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†		SN54LS85			SN74LS85			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} = −400 μA	2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	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	A < B, A > B inputs	V_{CC} = MAX,	V_I = 7 V			0.1			0.1	mA
		all other inputs					0.3			0.3	
I_{IH}	High-level input current	A < B, A > B inputs	V_{CC} = MAX,	V_I = 2.7 V			20			20	μA
		all other inputs					60			60	
I_{IL}	Low-level input current	A < B, A > B inputs	V_{CC} = MAX,	V_I = 0.4 V			−0.4			−0.4	mA
		all other inputs					−1.2			−1.2	
I_{OS}	Short-circuit output current§		V_{CC} = MAX		−20		−100	−20		−100	mA
I_{CC}	Supply current		V_{CC} = MAX,	See Note 4		10.4	20		10.4	20	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 of the short-circuit should not exceed one second.
NOTE 4: I_{CC} is measured with outputs open, A = B grounded, and all other inputs at 4.5 V.

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

PARAMETER¶	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B data input	A < B, A > B	1	C_L = 15 pF, R_L = 2 kΩ, See Note 7		14		ns
			2			19		
			3			24	36	
		A = B	4			27	43	
t_{PHL}	Any A or B data input	A < B, A > B	1			11		ns
			2			15		
			3			20	30	
		A = B	4			23	48	
t_{PLH}	A < B or A = B	A > B	1			14	22	ns
t_{PHL}	A < B or A = B	A > B	1			11	17	ns
t_{PLH}	A = B	A = B	2			13	20	ns
t_{PHL}	A = B	A = B	2			13	26	ns
t_{PLH}	A > B or A = B	A < B	1			14	22	ns
t_{PHL}	A > B or A = B	A < B	1			11	17	ns

¶t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 7: Load circuit and voltage waveforms are shown on page 3-11.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

recommended operating conditions

	SN54S85			SN74S85			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			−1	mA
Low-level output current, I_{OL}			20			20	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†		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		V_{CC} = MIN, I_I = −18 mA				−1.2	V
V_{OH}	High-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = −1 mA	SN54S85	2.5	3.4		V
				SN74S85	2.7	3.4		
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA				0.5	V
I_I	Input current at maximum input voltage		V_{CC} = MAX, V_I = 5.5 V				1	mA
I_{IH}	High-level input current	A < B, A > B inputs	V_{CC} = MAX, V_I = 2.7 V				50	μA
		all other inputs					150	
I_{IL}	Low-level input current	A < B, A > B inputs	V_{CC} = MAX, V_I = 0.5 V				−2	mA
		all other inputs					−6	
I_{OS}	Short-circuit output current§		V_{CC} = MAX		−40		−100	mA
I_{CC}	Supply current		V_{CC} = MAX, See Note 4			73	115	mA
			V_{CC} = MAX, T_A = 125°C, See Note 4	SN54S85W			110	

† 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 of the short-circuit should not exceed one second.
NOTE 4: I_{CC} is measured with outputs open, A = B grounded, and all other inputs at 4.5 V.

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

PARAMETER†	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Any A or B data input	A < B, A > B	1	C_L = 15 pF, R_L = 280 Ω, See Note 5		5		ns
			2			7.5		
			3			10.5	16	
		A = B	4			12	18	
t_{PHL}	Any A or B data input	A < B, A > B	1			5.5		ns
			2			7		
			3			11	16.5	
		A = B	4			11	16.5	
t_{PLH}	A < B or A = B	A > B	1			5	7.5	ns
t_{PHL}	A < B or A = B	A > B	1			5.5	8.5	ns
t_{PLH}	A = B	A = B	2			7	10.5	ns
t_{PHL}	A = B	A = B	3			5	7.5	ns
t_{PLH}	A > B or A = B	A < B	1			5	7.5	ns
t_{PHL}	A > B or A = B	A < B	1			5.5	8.5	ns

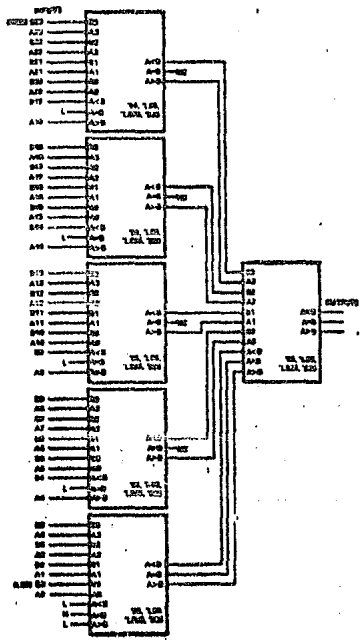
† t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 5: Load circuit and voltage waveforms are shown on page 2-10.

TYPICAL APPLICATION DATA

COMPARISON OF TWO N-BIT WORDS

This application demonstrates how these magnitude comparators can be cascaded to compare longer words. The example illustrated shows the comparison of two 24-bit words; however, the design is expandable to n-bits. As an example, one comparator can be used with five of the 24-bit comparators illustrated to expand the word length to 120-bits. Typical comparison times for various word lengths using the '85, 'L85, 'LS85, or 'S85 are:

WORD LENGTH	NUMBER OF PKGS	'85	'L85	'LS85	'S85
1-4 bits	1	23 ns	90 ns	24 ns	11 ns
5-24 bits	2-6	46 ns	180 ns	48 ns	22 ns
25-120 bits	6-31	69 ns	270 ns	72 ns	33 ns

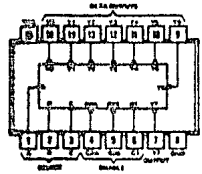


COMPARISON OF TWO EIGHT-BIT WORDS

PIN ASSIGNMENTS (TOP VIEW)

3-TO-8 LINE DECODER/DEMULTIPLEXER

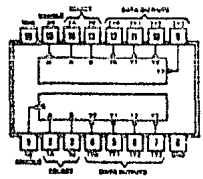
138



SN54LS138 (J, W) SN74LS138 (J, N)
SN54S138 (J, W) SN74S138 (J, N)

DUAL 2-TO-4 LINE DECODER/DEMULTIPLEXER

139

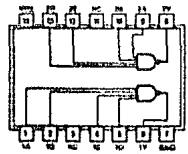


SN54LS139 (J, W) SN74LS139 (J, N)
SN54S139 (J, W) SN74S139 (J, N)

DUAL 4-INPUT POSITIVE-NAND 50-OHM LINE DRIVERS

140

Y = $\overline{ABCD}$

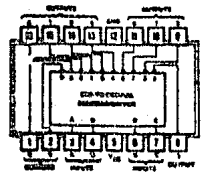


SN54S140 (J, W) SN74S140 (J, N)

NC—No internal connection

BCD-TO-DECIMAL DECODER/DRIVER

141 DRIVES COLD-CATHODE INDICATOR TUBES



SN54141 (J, W)

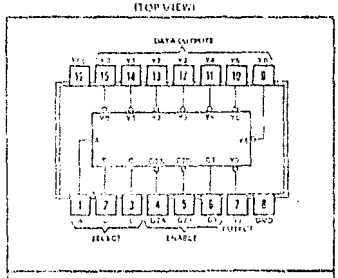
TYPES SN54LS138, SN54LS139, SN54S138, SN54S139, SN74LS138, SN74LS139, SN74S138, SN74S139 DECODERS/DEMULTIPLEXERS

BULLETIN NO. DL-S 7611804, DECEMBER 1972 – REVISED OCTOBER 1976

- Designed Specifically for High-Speed:
  Memory Decoders
  Data Transmission Systems
- 'S138 and 'LS138 3-to-8-Line Decoders Incorporate 3 Enable Inputs to Simplify Cascading and/or Data Reception
- 'S139 and 'LS139 Contain Two Fully Independent 2-to-4-Line Decoders/Demultiplexers
- Schottky Clamped for High Performance

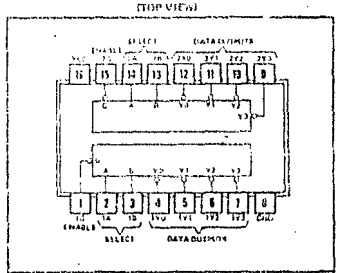
TYPE	TYPICAL PROPAGATION DELAY (3 LEVELS OF LOGIC)	TYPICAL POWER DISSIPATION
'LS138	22 ns	32 mW
'S138	8 ns	225 mW
'LS139	22 ns	34 mW
'S139	7.5 ns	300 mW

SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

SN54LS139, SN54S139 . . . J OR W PACKAGE
SN74LS139, SN74S139 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

description

These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast-enable circuit the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

The 'LS138 and 'S138 decode one-of-eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

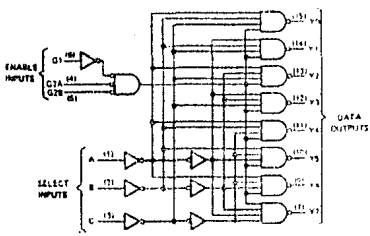
The 'LS139 and 'S139 comprise two individual two-line to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs each of which represents only one normalized Series 54LS/74LS load ('LS138, 'LS139) or one normalized Series 54S/74S load ('S138, 'S139) to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design. Series 54LS and 54S devices are characterized for operation over the full military temperature range of −55°C to 125°C; Series 74LS and 74S devices are characterized for 0°C to 70°C industrial systems.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

functional block diagrams and logic

'LS138, 'S138



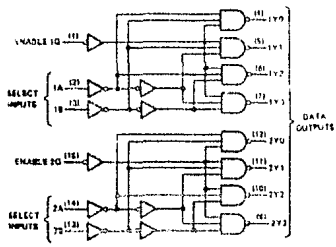
'LS138, 'S138
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*G2 = G2A + G2B

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

'LS139, 'S139

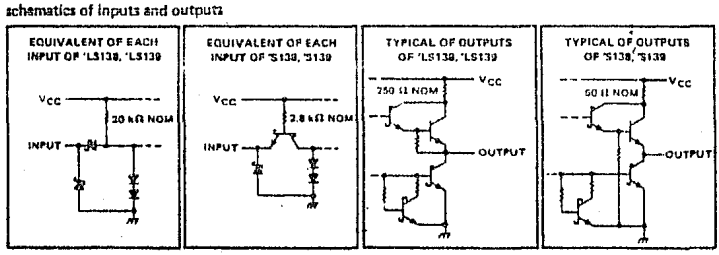


'LS139, 'S139
(EACH DECODER/DEMULTIPLEXER)
FUNCTION TABLE

INPUTS			OUTPUTS			
ENABLE	SELECT					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

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

schematics 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: SN54LS138, SN54LS139 Circuits . . . . . −55°C to 125°C
SN74LS138, SN74LS139 Circuits . . . . . 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

	SN54LS138 SN54LS139			SN74LS138 SN74LS139			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
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†		SN54LS138 SN54LS139			SN74LS138 SN74LS139			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_{IC}	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_{IH} = 2 V, V_{IL} = $V_{IL\,max}$, I_{OH} = −400 μA		2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = $V_{IL\,max}$	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		−6		−40	−6		−42	mA
I_{CC}	Supply current	V_{CC} = MAX, Outputs enabled and open	LS138		6.3	10		6.3	10	mA
			LS139		6.8	11		6.8	11	

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

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

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54LS138 SN74LS139			SN54LS139 SN74LS139			UNIT
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Binary Select	Any	2	C_L = 15 pF, R_L = 2 kΩ, See Note 2		13	20		13	20	ns
t_{PHL}						27	41		22	33	ns
t_{PLH}			3			18	27		18	29	ns
t_{PHL}						26	39		25	38	ns
t_{PLH}	Enable	Any	2			12	18		16	24	ns
t_{PHL}						21	32		21	32	ns
t_{PLH}			3			17	26				ns
t_{PHL}						25	38				ns

¶t_{PLH} = propagation delay time, low-to-high level output; t_{PHL} = propagation delay time, high-to-low-level output.
NOTE 2: Load circuits and waveforms are shown on page 3-11.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

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

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 5.5 V
Operating free-air temperature range: SN54S138, SN54S139 Circuits . . . . . −55°C to 125°C
SN74S138, SN74S139 Circuits . . . . . 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

	SN54S138 SN74S138			SN74S138 SN74S139			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			−1	mA
Low-level output current, I_{OL}			20			20	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†		SN54S138 SN74S138			SN54S139 SN74S139			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
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 = −18 mA				−1.2			−1.2	V
V_{OH} High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = −1 mA	SN54S	2.5	3.4		2.5	3.4		V
		SN74S	2.7	3.4		2.7	3.4		
V_{OL} Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA				0.5			0.5	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	V_{CC} = MAX, V_I = 2.7 V				50			50	μA
I_{IL} Low-level input current	V_{CC} = MAX, V_I = 0.5 V				−2			−2	mA
I_{OS} Short-circuit output current§	V_{CC} = MAX		−40		−100	−40		−100	mA
I_{CC} Supply current	V_{CC} = MAX, Outputs enabled and open			40	74		60	90	mA

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device 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 test should not exceed one second.

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

PARAMETER†	FROM (INPUT)	TO (OUTPUT)	LEVELS OF DELAY	TEST CONDITIONS	SN54S138, SN74S138			SN54S139 SN74S139			UNIT
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Binary select	Any	2	C_L = 15 pF, R_L = 280 Ω, See Note 3	4.5	7		5	7.5		ns
t_{PHL}					7	10.5		6.5	10		
t_{PLH}			3		7.5	12		7	12		ns
t_{PHL}					8	12		8	12		
t_{PLH}	Enable	Any	2		5	8		5	8		ns
t_{PHL}					7	11		6.5	10		
t_{PLH}			3		7	11					ns
t_{PHL}					7	11					

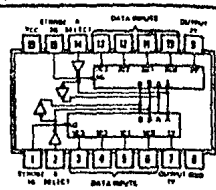
†t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output

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

PIN ASSIGNMENTS (TOP VIEW)

DUAL 4-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

153

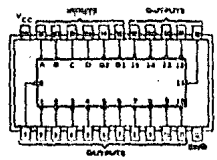


SN54153 (J, W) SN74153 (J, N)
SN54L153 (J) SN74L153 (J, N)
SN54LS153 (J, W) SN74LS153 (J, N)
SN54S153 (J, W) SN74S153 (J, N)

See page 7-169

4-LINE TO 16-LINE DECODERS/DEMULTIPLEXERS

154



SN54154 (J, W) SN74154 (J, N)
SN54L154 (J) SN74L154 (J, N)

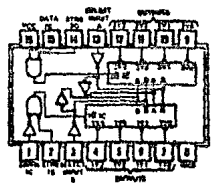
See page 7-171

DECODERS/DEMULTIPLEXERS

DUAL 2- TO 4-LINE DECODER
DUAL 1- TO 4-LINE DEMULTIPLEXER
3- TO 8-LINE DECODER
1- TO 8-LINE DEMULTIPLEXER

155 TOTEM-POLE OUTPUTS

156 OPEN-COLLECTOR OUTPUTS



SN54155 (J, W) SN74155 (J, N)
SN54LS155 (J, W) SN74LS155 (J, N)
SN54156 (J, W) SN74156 (J, N)
SN54LS156 (J, W) SN74LS156 (J, N)

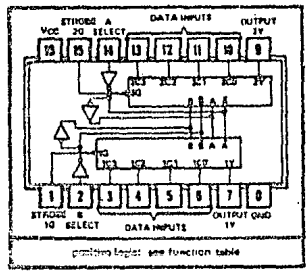
See page 7-173

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

TYPES SN54153, SN54L153, SN54LS153, SN54S153, SN74153, SN74L153, SN74LS153, SN74S153
DUAL 4-LINE-TO-1-LINE DATA SELECTORS/MULTIPLEXERS

BULLETIN NO. DL-S 7611852, DECEMBER 1972 – REVISED OCTOBER 1976

SN54153, SN54LS153, SN54S153 . . . J OR W PACKAGE
SN54L153 . . . J PACKAGE
SN74153, SN74L153, SN74LS153, SN74S153 . . . J OR N PACKAGE
(TOP VIEW)



- Parallel-to-Serial Conversion
- Strobe (Enable) Line Provided for Cascading (N lines to n lines)
- High-Fan-Out, Low-Impedance, Totem-Pole Outputs
- Fully Compatible with most TTL and DTL Circuits

TYPE	TYPICAL AVERAGE PROPAGATION DELAY TIME FROM DATA	FROM STROBE	FROM SELECT	TYPICAL POWER DISSIPATION
'153	14 ns	17 ns	22 ns	160 mW
'L153	27 ns	34 ns	44 ns	90 mW
'LS153	14 ns	19 ns	22 ns	31 mW
'S153	6 ns	9.5 ns	12 ns	235 mW

description

Each of these monolithic data selectors/multiplexers contains inverters and drivers to supply fully complementary, on-chip, binary decoding data selection to the AND-OR-invert gates. Separate strobe inputs are provided for each of the two four-line sections.

FUNCTION TABLE

SELECT INPUTS		DATA INPUTS				STROBE	OUTPUT
B	A	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

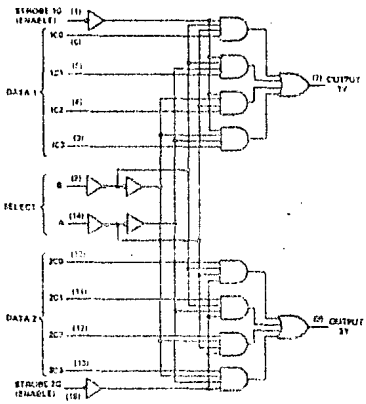
Select inputs A and B are common to both sections.
H = high level, L = low level, X = irrelevant

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

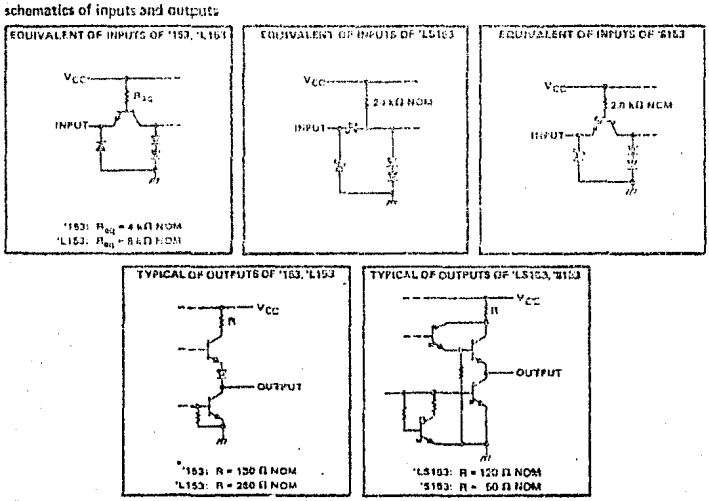
Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage: '153, 'L153, 'S153 . . . . . 5.5 V
'LS153 . . . . . 7 V
Operating free-air temperature range: SN54', SN54L', SN54LS', SN54S' Circuits . . . . . −55°C to 125°C
SN74', SN74L', SN74LS', SN74S' Circuits . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

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

functional block diagram



schematics of inputs and outputs



TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

recommended operating conditions

	SN54153			SN74153			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}			−800			−800	μA
Low-level output current, I_{OL}			16			16	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†	SN54153			SN74153			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
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	3.4		2.4	3.4		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	V_{CC} = MAX, V_I = 2.4 V			40			40	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4 V			−1.6			−1.6	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX	−20		−55	−18		−57	mA
I_{CCL}	Supply current, output low	V_{CC} = MAX, See Note 2		36	62		36	60	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.
NOTE 2: I_{CCL} is measured with the outputs open and all inputs grounded.

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

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	Y	C_L = 30 pF, R_L = 400 Ω, See Note 3		12	18	ns
t_{PHL}	Data	Y			15	23	ns
t_{PLH}	Select	Y			22	34	ns
t_{PHL}	Select	Y			22	34	ns
t_{PLH}	Strobe	Y			19	30	ns
t_{PHL}	Strobe	Y			15	23	ns

¶t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 3: Load circuit and voltage waveforms are shown on page 3-10.

recommended operating conditions

	SN54L153			SN74L153			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}			8			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†	SN54L153			SN74L153			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
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} = −400 μA	2.4	3.4		2.4	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 8 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	V_{CC} = MAX, V_I = 2.4 V			20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4 V			−0.8			−0.8	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX	−10		−20	−9		−30	mA
I_{CCL}	Supply current, output low	V_{CC} = MAX, See Note 2		18	29		18	30	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.

NOTE 2: I_{CCL} is measured with the outputs open and all inputs grounded.

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

PARAMETER*	FROM INPUT	TO OUTPUT	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	Y	C_L = 15 pF, R_L = 400 Ω, See Note 3		24	36	ns
t_{PHL}	Data	Y			30	45	ns
t_{PLH}	Select	Y			44	66	ns
t_{PHL}	Select	Y			44	66	ns
t_{PLH}	Strobe	Y			39	60	ns
t_{PHL}	Strobe	Y			30	45	ns

*t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 3: Load circuit and voltage waveforms are shown on page 3-10.

recommended operating conditions

	SN54LS153			SN74LS153			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
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†		SN54LS153			SN74LS153			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_{IH} = 2 V, V_{IL} = V_{IL} max, I_{OH} = −400 μA		2.5	3.4		2.7	3.4		V
V_{OL} Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	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_{CCL} Supply current, output low	V_{CC} = MAX, See Note 2			8.2	10		8.2	10	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 of short-circuit should not exceed one second.

NOTE 2: I_{CCL} is measured with the outputs open and all inputs grounded.

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

PARAMETER†	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	Y	C_L = 15 pF, R_L = 2 kΩ, See Note 4		10	15	ns
t_{PHL}	Data	Y			17	25	ns
t_{PLH}	Select	Y			19	29	ns
t_{PHL}	Select	Y			25	38	ns
t_{PLH}	Strobe	Y			16	24	ns
t_{PHL}	Strobe	Y			21	32	ns

† 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 voltage waveforms are shown on page 3-75.

recommended operating conditions

	SN54S153			SN74S153			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			−1	mA
Low-level output current, I_{OL}			20			20	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†		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	V_{CC} = MIN, I_I = −18 mA				−1.2	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = −1 mA	Series 54S	2.5	3.4		V
			Series 74S	2.7	3.4		
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA				0.5	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 5.5 V				1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7 V				50	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.5 V				−2	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX		−40		−100	mA
I_{CCL}	Supply current, low-level output	V_{CC} = MAX, See Note 2			45	70	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 of short-circuit should not exceed one second.
NOTE 2: I_{CCL} is measured with the outputs open and all inputs grounded.

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

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	Y	C_L = 15 pF, R_L = 280 Ω, See Note 3		6	9	ns
t_{PHL}	Data	Y			6	9	ns
t_{PLH}	Select	Y			11.5	18	ns
t_{PHL}	Select	Y			12	18	ns
t_{PLH}	Strobe	Y			10	15	ns
t_{PHL}	Strobe	Y			9	13.5	ns

¶ t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTE 3: Load circuit and voltage waveforms are shown on page 3-10.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

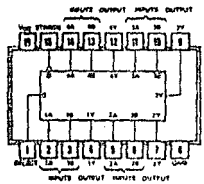
PIN ASSIGNMENTS (TOP VIEW)

QUAD 2- TO 1-LINE DATA SELECTORS/MULTIPLEXERS

157 NONINVERTED DATA OUTPUTS

158 INVERTED DATA OUTPUTS

See page 7-151

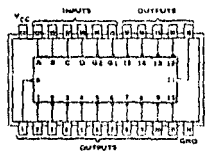


SN54157 (J, W)	SN74157 (J, N)
SN54LS157 (J)	SN74LS157 (J, N)
SN54ALS157 (J, W)	SN74ALS157 (J, N)
SN54S157 (J, W)	SN54S157 (J, N)
SN54LS158 (J, W)	SN74LS158 (J, N)
SN54S158 (J, W)	SN74S158 (J, N)

4- TO 10-LINE DECODERS/DEMULTIPLEXERS

159 OPEN-COLLECTOR OUTPUTS

See page 7-153



SN54159 (J, W) SN74159 (J, N)

SYNCHRONOUS 4-BIT COUNTERS

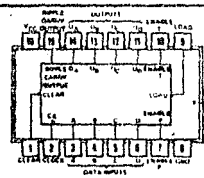
160 DECADE, DIRECT CLEAR

161 BINARY, DIRECT CLEAR

162 DECADE, SYNCHRONOUS CLEAR

163 BINARY, SYNCHRONOUS CLEAR

See page 7-160



SN54160 (J, W)	SN74160 (J, N)
SN54LS160A (J, W)	SN74LS160A (J, N)
SN54161 (J, W)	SN74161 (J, N)
SN54LS161A (J, W)	SN74LS161A (J, N)
SN54162 (J, W)	SN74162 (J, N)
SN54LS162A (J, W)	SN74LS162A (J, N)
SN54S162 (J, W)	SN74S162 (J, N)
SN54163 (J, W)	SN74163 (J, N)
SN54LS163A (J, W)	SN74LS163A (J, N)
SN54S163 (J, W)	SN74S163 (J, N)

TTL MSI

TYPES SN54160 THRU SN54163, SN54LS160A THRU SN54LS163A, SN54S162, SN54S163, SN74160 THRU SN74163, SN74LS160A THRU SN74LS163A, SN74S162, SN74S163
SYNCHRONOUS 4-BIT COUNTERS

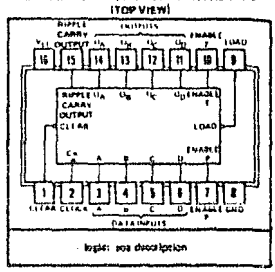
BULLETIN NO. DL-S 7711809, DECEMBER 1976 – REVISED AUGUST 1977

'160, '161, 'LS160A, 'LS161A . . . SYNCHRONOUS COUNTERS WITH DIRECT CLEAR
'162, '163, 'LS162A, 'LS163A, 'S162, 'S163 . . . FULLY SYNCHRONOUS COUNTERS

- Internal Look-Ahead for Fast Counting
- Carry Output for n-Bit Cascading
- Synchronous Counting
- Synchronously Programmable
- Load Control Line
- Diode-Clamped Inputs

TYPE	TYPICAL PROPAGATION TIME, CLOCK TO Q OUTPUT	TYPICAL MAXIMUM CLOCK FREQUENCY	TYPICAL POWER DISSIPATION
'160 thru '163	14 ns	32 MHz	305 mW
'LS160A thru 'LS163A	14 ns	32 MHz	93 mW
'S162 and 'S163	9 ns	70 MHz	475 mW

SERIES 54', 54LS', 54S' . . . J OR W PACKAGE
SERIES 74', 74LS', 74S' . . . J OR N PACKAGE
(TOP VIEW)



logic symbol description

description

These synchronous, presettable counters feature an internal carry look-ahead for application in high-speed counting designs. The '160, '162, 'LS160A, 'LS162A, and 'S162 are decade counters and the '161, '163, 'LS161A, 'LS163A, and 'S163 are 4-bit binary counters. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable inputs and internal gating. This mode of operation eliminates the output counting spikes that are normally associated with asynchronous (ripple clock) counters. A buffered clock input triggers the four flip-flops on the rising (positive-going) edge of the clock input waveform.

These counters are fully programmable; that is, the outputs may be preset to either level. As presetting is synchronous, setting up a low level at the load input disables the counter and causes the outputs to agree with the setup data after the next clock pulse regardless of the levels of the enable inputs. Low-to-high transitions at the load input of the '160 thru '163 should be avoided when the clock is low if the enable inputs are high at or before the transition. This restriction is not applicable to the 'LS160A thru 'LS163A or 'S162 or 'S163 . The clear function for the '160, '161, 'LS160A, and 'LS161A is asynchronous and a low level at the clear input sets all four of the flip-flop outputs low regardless of the levels of clock, load, or enable inputs. The clear function for the '162, '163, 'LS162A, 'LS163A, 'S162, and 'S163 is synchronous and a low level at the clear input sets all four of the flip-flop outputs low after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily as decoding the maximum count desired can be accomplished with one external NAND gate. The gate output is connected to the clear input to synchronously clear the counter to 0000 (LLLL). Low-to-high transitions at the clear input of the '162 and '163 should be avoided when the clock is low if the enable and load inputs are high at or before the transition.

The carry look-ahead circuitry provides for cascading counters for n-bit synchronous applications without additional gating. Instrumental in accomplishing this function are two count-enable inputs and a ripple carry output. Both count-enable inputs (P and T) must be high to count, and input T is fed forward to enable the ripple carry output. The ripple carry output thus enabled will produce a high-level output pulse with a duration approximately equal to the high-level portion of the Q_A output. This high-level overflow ripple carry pulse can be used to enable successive cascaded stages. High-to-low-level transitions at the enable P or T inputs of the '160 thru '163 should occur only when the clock input is high. Transitions at the enable P or T inputs of the 'LS160A thru 'LS163A or 'S162 and 'S163 are allowed regardless of the level of the clock input.

'LS160A thru 'LS163A, 'S162 and 'S163 feature a fully independent clock circuit. Changes at control inputs (enable P or T, or clear) that will modify the operating mode have no effect until clocking occurs. The function of the counter (whether enabled, disabled, loading, or counting) will be dictated solely by the conditions meeting the stable setup and hold times.

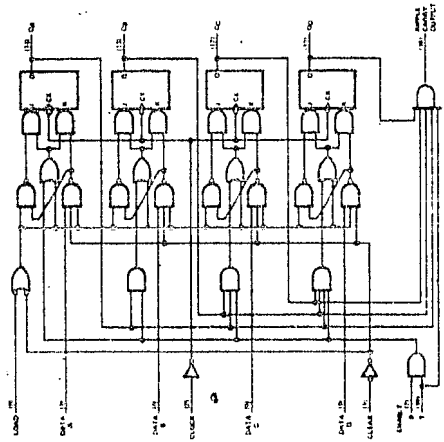
The 'LS160A thru 'LS163A are completely new designs. Compared to the original 'LS160 thru 'LS163, they feature 0-nanosecond minimum hold time and reduced input currents I_{IH} and I_{IL}.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

functional block diagrams

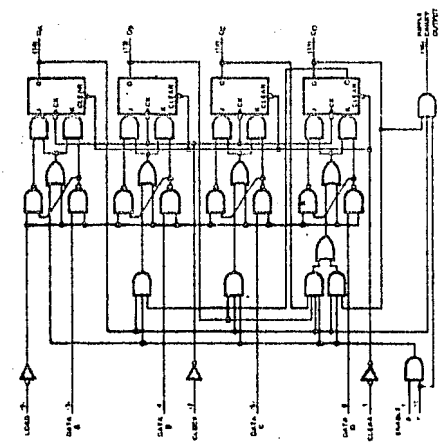
SN54163, SN74163 SYNCHRONOUS BINARY COUNTERS

SN54161, SN74161 synchronous binary counters are similar; however, the clear is synchronous, as shown for the SN54160, SN74160 decade counters at right.



SN54160, SN74160 SYNCHRONOUS DECADE COUNTERS

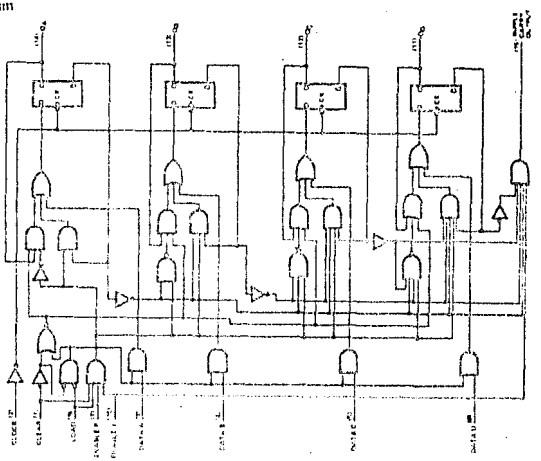
SN54162, SN74162 synchronous decade counters are similar; however, the clear is synchronous as shown for the SN54163, SN74163 binary counters at right.



functional block diagram

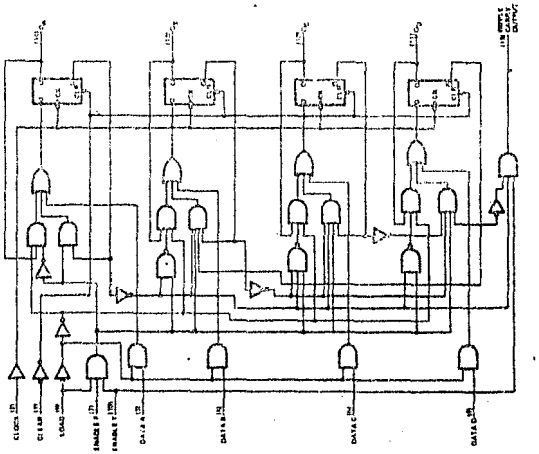
SN54LS163A, SN74LS163A SYNCHRONOUS
BINARY COUNTERS

SN54LS161A, SN74LS161A synchronous binary counters are similar; however, the clear is asynchronous as shown for the SN54LS160A, SN74LS160A decade counters at left.



SN54LS163A, SN74LS162A SYNCHRONOUS
DECADE COUNTERS

SN54LS162A, SN74LS162A synchronous decade counters are similar; however the clear is synchronous as shown for the SN54LS163A, SN74LS163A binary counters at right.

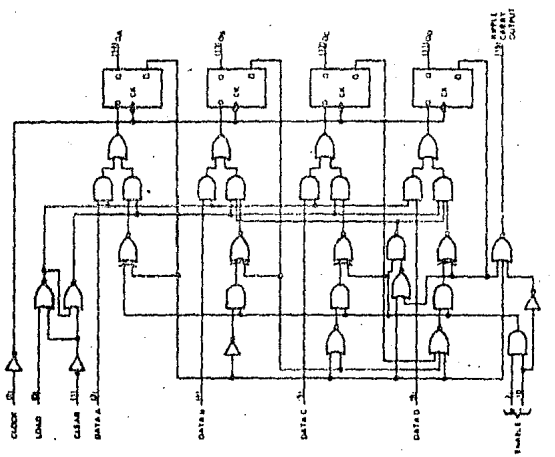


TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

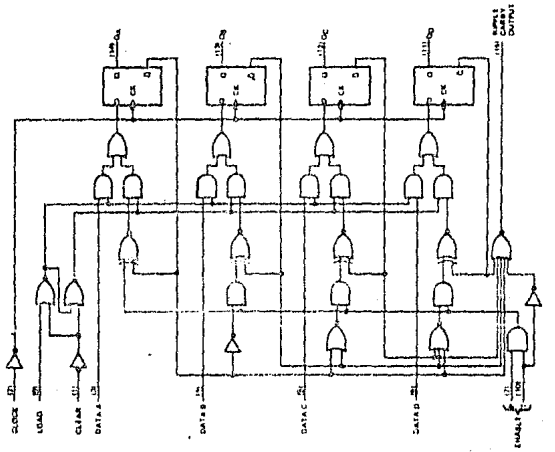
REVISED OCTOBER 1976

functional block diagrams

SN54S162, SN74S162 SYNCHRONOUS DECADE COUNTERS



SN54S163, SN74S163 SYNCHRONOUS BINARY COUNTERS

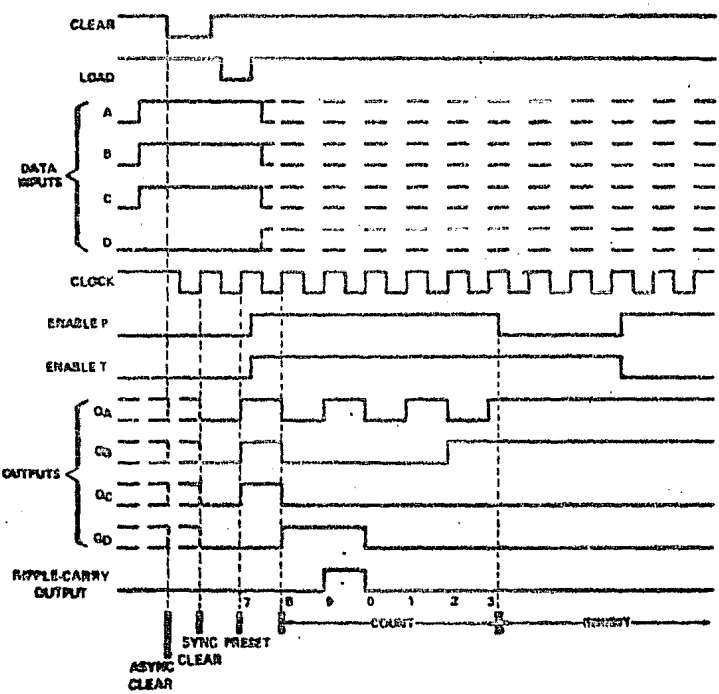


'160, '162, 'LS160A, 'LS162A, 'S162 DECADE COUNTERS

typical clear, preset, count, and inhibit sequences

Illustrated below is the following sequence:

1. Clear outputs to zero ('160 and 'LS160A are asynchronous; '162, 'LS162A, and 'S162 are synchronous)
2. Preset to BCD seven
3. Count to eight, nine, zero, one, two, and three
4. Inhibit



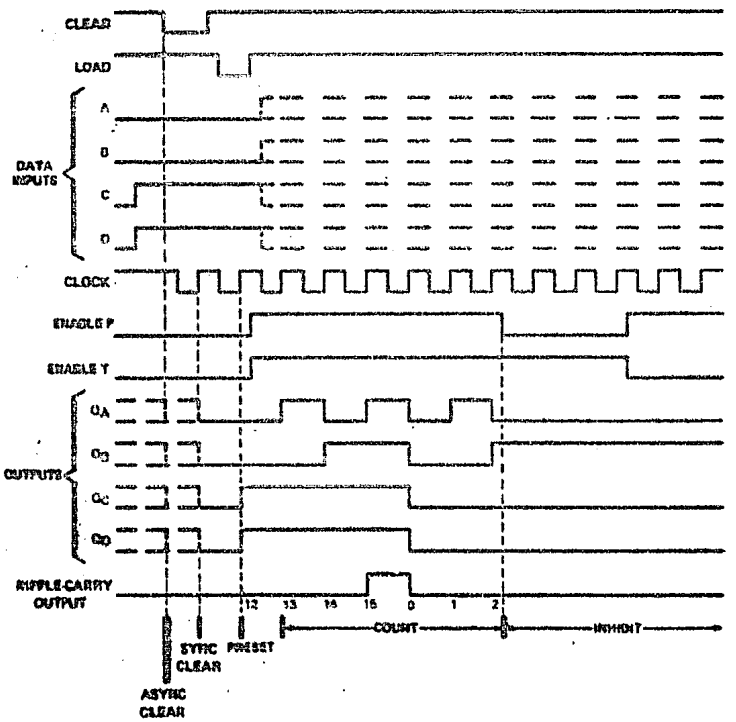
TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

'161, 'LS161A, '163, 'LS163A, 'S163 BINARY COUNTERS

typical clear, preset, count, and inhibit sequences

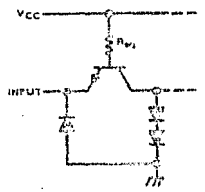
Illustrated below is the following sequence:

1. Clear outputs to zero ('161 and 'LS161A are asynchronous; '163, 'LS163A, and 'S163 are synchronous)
2. Preset to binary twelve
3. Count to thirteen, fourteen, fifteen, zero, one, and two
4. Inhibit



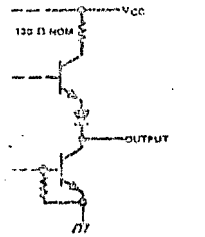
schematics of inputs and outputs

EQUIVALENT OF EACH INPUT



Clock: R_{eq} = 2.8 kΩ NOM
Enable T: R_{eq} = 2 kΩ NOM
Clear, Enable P: R_{eq} = 4 kΩ NOM
A, B, C, D: R_{eq} = 6 kΩ NOM

TYPICAL OF ALL OUTPUTS



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

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Interemitter voltage (see Note 2)	5.5 V
Operating free-air temperature range: SN54' Circuits	−55°C to 125°C
SN74' 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 these circuits, this rating applies between the count enable inputs P and T.

recommended operating conditions

		SN54160, SN54161 SN54162, SN54163			SN74160, SN74161 SN74162, SN74163			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}				−800			−800	μA
Low-level output current, I_{OL}				16			16	mA
Clock frequency, f_{clock}		0		25	0		25	MHz
Width of clock pulse, $t_{w(clock)}$		25			25			ns
Width of clear pulse, $t_{w(clear)}$		20			20			ns
Setup time, t_{su} (see Figures 1 and 2)	Data inputs A, B, C, D	20			20			ns
	Enable P	20			20			
	Load	25			25			
	Clear[a]	20			20			
Hold time at any input, t_h		0			0			ns
Operating free-air temperature, T_A		−55		125	0		70	°C

[a]This applies only for '162 and '163, which have synchronous clear inputs.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

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

PARAMETER			TEST CONDITIONS†	SN54160, SN54161 SN54162, SN54163			SN74160, SN74161 SN74162, SN74163			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
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	3.4		2.4	3.4		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	Clock or enable T	V_{CC} = MAX, V_I = 2.4 V			80			80	μA
		Other inputs				40			40	
I_{IL}	Low-level input current	Clock or enable T	V_{CC} = MAX, V_I = 0.4 V			−3.2			−3.2	mA
		Other inputs				−1.6			−1.6	
I_{OS}	Short-circuit output current§		V_{CC} = MAX	−20		−57	−18		−57	mA
I_{CCH}	Supply current, all outputs high		V_{CC} = MAX, See Note 3		59	85		59	94	mA
I_{CCL}	Supply current, all outputs low		V_{CC} = MAX, See Note 4		63	91		63	101	mA

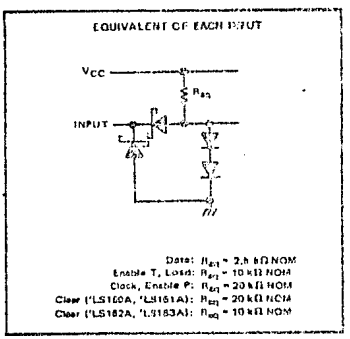
†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time.
NOTES: 3. I_{CCH} is measured with the load input high, then again with the load input low, with all other inputs high and all outputs open.
4. I_{CCL} is measured with the clock input high, then again with the clock input low, with all other inputs low and all outputs open.

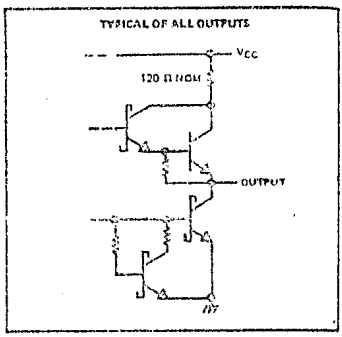
switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}			C_L = 15 pF, R_L = 400 Ω, See Figures 1 and 2 and Notes 5 and 6	25	32		ns
t_{PLH}	Clock	Ripple carry			23	35	ns
t_{PHL}					23	35	
t_{PLH}	Clock (load input high)	Any Q			13	20	ns
t_{PHL}					16	23	
t_{PLH}	Clock (load input low)	Any Q			17	25	ns
t_{PHL}					18	29	
t_{PLH}	Enable T	Ripple carry			11	16	ns
t_{PHL}					11	16	
t_{PHL}	Clear	Any Q			26	30	ns

¶f_{max} = Maximum clock frequency
t_{PLH} = Propagation delay time, low-to-high level output
t_{PHL} = Propagation delay time, high-to-low level output
NOTES: 5. Load circuit is shown on page 3-10.
6. Propagation delay for clearing is measured from the clear input for the '160 and '161 or from the clock input transition for the '162 and '163.

schematics of inputs and outputs





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

Supply voltage, V_{CC} (see Note 7)	7 V
Input voltage	7 V
Operating free-air temperature range: SN54LS' Circuits	−55°C to 125°C
SN74LS' Circuits	0°C to 70°C
Storage temperature range	−65°C to 150°C

NOTE 7: Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54LS'			SN74LS'			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
Clock frequency, f_{clock}		0		25	0		25	MHz
Width of clock pulse, $t_{w(clock)}$		25			25			ns
Width of clear pulse, $t_{w(clear)}$		20			20			ns
Setup time, t_{su} (see Figures 1 and 2)	Data inputs A, B, C, D	20			20			ns
	Enable P or T	20			20			
	Load	20			20			
	Clear[◊]	20			20			
Hold time at any input, t_h		0			0			ns
Operating free-air temperature, T_A		−55		125	0		70	°C

◊ This applies only for 'LS162 and 'LS163, which have synchronous clear inputs.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

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

PARAMETER		TEST CONDITIONS†		SN54LS'			SN74LS'			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} = −400 μA	2.5	3.4		2.7	3.4		V
V_{OL} Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	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	Data or enable P	V_{CC} = MAX,	V_I = 7 V			0.1			0.1	mA
	Load, clock, or enable T					0.2			0.2	
	Clear ('LS160A, 'LS161A)					0.1			0.1	
	Clear ('LS162A, 'LS163A)					0.2			0.2	
I_{IH} High-level input current	Data or enable P	V_{CC} = MAX,	V_I = 2.7 V			20			20	μA
	Load, clock, or enable T					40			40	
	Clear ('LS160A, 'LS161A)					20			20	
	Clear ('LS162A, 'LS163A)					40			40	
I_{IL} Low-level input current	Data or enable P	V_{CC} = MAX,	V_I = 0.4 V			−0.4			−0.4	mA
	Load, clock, or enable T					−0.8			−0.8	
	Clear ('LS160A, 'LS161A)					−0.4			−0.4	
	Clear ('LS162A, 'LS163A)					−0.8			−0.8	
I_{OS} Short-circuit output current§		V_{CC} = MAX		−20		−100	−20		−100	mA
I_{CCH} Supply current, all outputs high		V_{CC} = MAX,	See Note 3		18	31		18	31	mA
I_{CCL} Supply current, all outputs low		V_{CC} = MAX,	See Note 4		19	32		19	32	mA

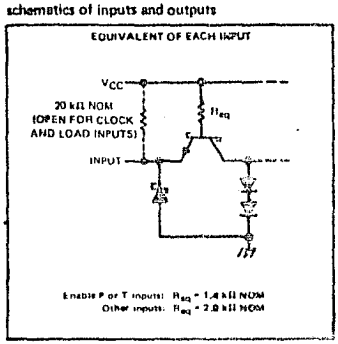
†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.
NOTES: 3. I_{CCH} is measured with the load input high, then again with the load input low, with all other inputs high and all outputs open.
4. I_{CCL} is measured with the clock input high, then again with the clock input low, with all other inputs low and all outputs open.

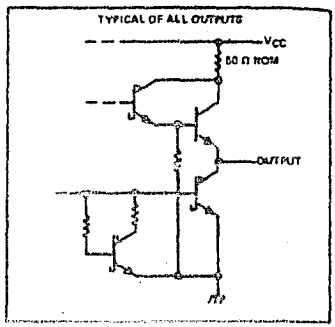
switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}			C_L = 15 pF, R_L = 2 kΩ, See Figures 1 and 2 and Notes 8 and 9	25	32		MHz
t_{PLH}	Clock	Ripple carry			20	35	ns
t_{PHL}					18	35	
t_{PLH}	Clock (load input high)	Any Q			13	24	ns
t_{PHL}					18	27	
t_{PLH}	Clock (load input low)	Any Q			13	24	ns
t_{PHL}					18	27	
t_{PLH}	Enable T	Ripple carry			9	14	ns
t_{PHL}					9	14	
t_{PHL}	Clear	Any Q			20	20	ns

¶f_{max} = Maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
NOTES: 8. Load circuit is shown on page 3-11.
9. Propagation delay for clearing is measured from the clear input for the 'LS160A and 'LS161A or from the clock transition for the 'LS162A and 'LS163A.

schematics 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 . . . . . 5.5 V
Intermitter voltage (see Note 2) . . . . . 5.5 V
Operating free-air temperature range: SN54S162, SN54S163 (see Note 10) . . . . . −55°C to 125°C
SN74S162, SN74S163 . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

recommended operating conditions

		SN54S162, SN54S163			SN74S162, SN74S163			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			−1	mA
Low-level output current, I_{OL}				20			20	mA
Clock frequency, f_{clock}		0		40	0		40	MHz
Width of clock pulse, $t_{w(clock)}$ (high or low)		10			10			ns
Width of clear pulse, $t_{w(clear)}$		10			10			ns
Setup time, t_{su} (see Figure 4)	Data inputs, A, B, C, D	4			4			ns
	Enable P or T	12			12			
	Load	14			14			
	Clear	14			14			
	Load inactive-state	12			12			
	Clear inactive-state	12			12			
Release time, $t_{release}$ (see Figure 4)	Enable P or T			4			4	ns
Hold time, t_h (see Figure 4)	Data inputs A, B, C, D	3			3			ns
	Load	0			0			
	Clear	0			0			
Operating free-air temperature, T_A (see Note 10)		−55		125	0		70	°C

NOTES: 1. Voltage values, except intermitter voltage, are with respect to network ground terminal.
2. This is the voltage between two emitters of a multiple emitter transistor. For these circuits, this rating applies between the common enable inputs P and T.
10. An SN54S162 or SN54S163 in the W package operating at free-air temperatures above 91°C requires a heat sink that provides a thermal resistance from case to free-air, $R_{\theta CA}$, of not more than 38°C/W.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

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

PARAMETER		TEST CONDITIONS†	SN54S162 SN54S163			SN74S162 SN74S163			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
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 = −18 mA			−1.2			−1.2	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = −1 mA	2.5	3.4		2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA			0.5			0.5	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, Enable T	V_{CC} = MAX, V_I = 2.7 V			100			100	μA
	High-level input current, Other inputs				50			50	
I_{IL}	Low-level input current, Enable T	V_{CC} = MAX, V_I = 0.5 V			−4			−4	mA
	Low-level input current, Other inputs				−2			−2	
I_{OS}	Short-circuit output current§	V_{CC} = MAX	−40		−100	−40		−100	mA
I_{CC}	Supply current	V_{CC} = MAX		95	160		95	160	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 of the short-circuit should not exceed one second.

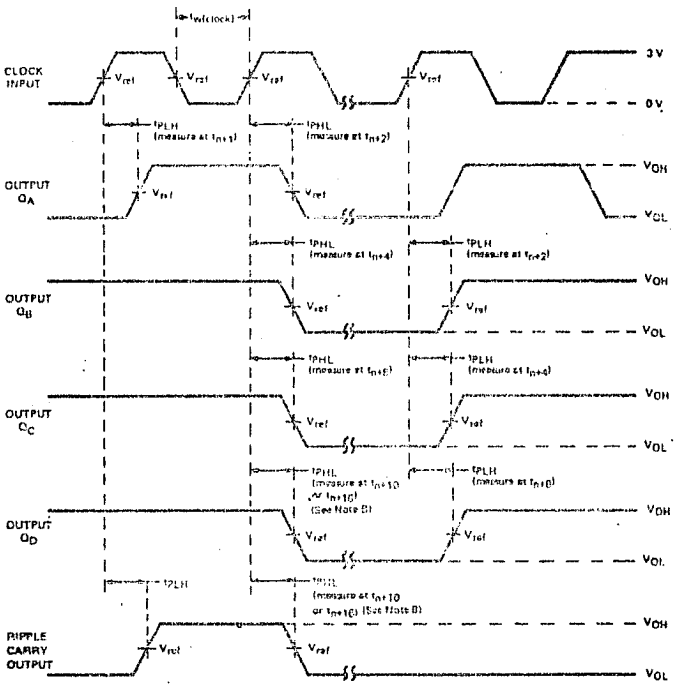
switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}				40	70		MHz
t_{PLH}	Clock	Ripple carry	C_L = 15 pF, R_L = 280 Ω, See Figure 1, 3, and 4 and Note 5		14	25	ns
t_{PHL}					17	25	
t_{PLH}	Clock	Any Q			8	15	ns
t_{PHL}					10	15	
t_{PLH}	Enable T	Ripple carry			10	15	ns
t_{PHL}					10	15	

¶f_{max} = maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output

NOTE 5: Load circuit is shown on page 3-10.

PARAMETER MEASUREMENT INFORMATION



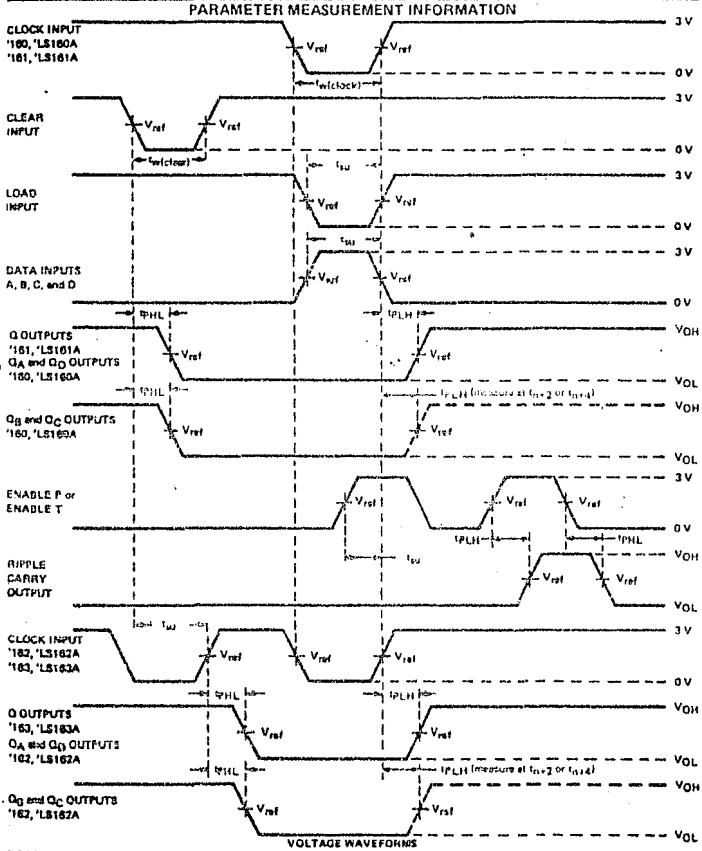
VOLTAGE WAVEFORMS

NOTES: A. The input pulses are supplied by a generator having the following characteristics: PRR ≤ 1 MHz, duty cycle ≤ 50%, $Z_{out} \approx 50\ \Omega$, for '160 thru '163, $t_r < 10$ ns, $t_f < 10$ ns; for 'LS160A thru 'LS163A $t_r < 15$ ns, $t_f < 6$ ns; and for 'S162, 'S163, $t_r < 2.5$ ns, $t_f < 2.5$ ns. Vary PRR to measure f_{max}.
B. Outputs Q_D and carry are tested at t_{n+10} for '160, '162, 'LS160A, 'LS162A and 'S162, and at t_{n+16} for '161, '163, 'LS161A 'LS163A, and 'S163, where t_n is the bit time when all outputs are low.
C. For '160 thru '163, 'S162, and 'S163, V_{ref} = 1.5 V; for 'LS160A thru 'LS163A, V_{ref} = 1.3 V.

FIGURE 1—SWITCHING TIMES

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The input pulses are supplied by generators having the following characteristics: PRR ≤ 1 MHz, duty cycle ≤ 50%, Z_{out} ≈ 50 Ω; for '160 thru '163, t_r ≤ 10 ns, t_f ≤ 10 ns; and for 'LS160A thru 'LS163A, t_r ≤ 15 ns, t_f ≤ 6 ns.
B. Enable P and enable T setup times are measured at t_{n+0}.
C. For '160 thru '163, V_{ref} = 1.5 V; for 'LS160A thru 'LS163A, V_{ref} = 1.3 V.

FIGURE 2–SWITCHING TIMES

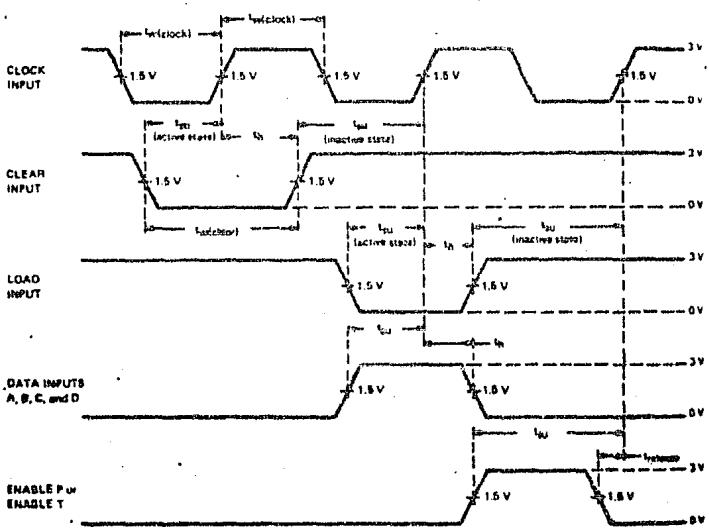
PARAMETER MEASUREMENT INFORMATION

VOLTAGE WAVEFORMS

NOTES: A. The input pulse is supplied by a generator having the following characteristics: $t_r \leq 2.5$ ns, $t_f \leq 2.5$ ns, PRR $\leq$ 1 MHz, duty cycle $\leq$ 50%, $Z_{out} \approx 50\ \Omega$.

B. t_{PLH} and t_{PHL} from enable T input to carry output assume that the counter is at the maximum count (Q_A and Q_D high for 'S162, all Q outputs high for 'S163).

FIGURE 3—PROPAGATION DELAY TIMES FROM ENABLE T INPUT TO CARRY OUTPUT



VOLTAGE WAVEFORMS

NOTE A: The input pulses are supplied by generators having the following characteristics: $t_r \leq 2.5$ ns, $t_f \leq 2.5$ ns, PRR $\leq$ 1 MHz, duty cycle $\leq$ 50%, $Z_{out} \approx 50\ \Omega$.

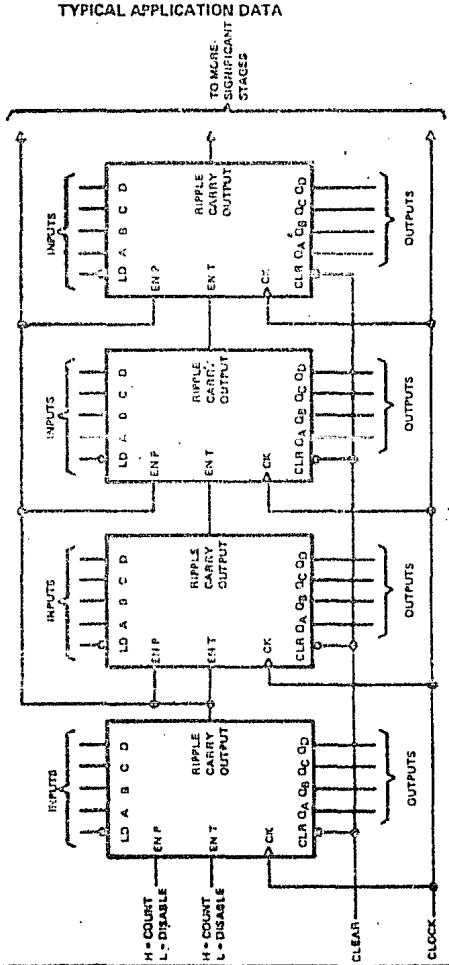
FIGURE 4—PULSE WIDTHS, SETUP TIMES, HOLD TIMES, AND RELEASE TIME

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

TYPICAL APPLICATION DATA

N-BIT SYNCHRONOUS COUNTERS

This application demonstrates how the look-ahead carry circuit can be used to implement a high-speed n-bit counter. The '160, '162, 'LS160A, 'LS162A, or 'S162 will count in BCD and the '161, '163, 'LS161A, 'LS163A or 'S163 will count in binary. Virtually any count mode (modulo-N, N1-to-N2, N1-to-maximum) can be used with this fast look-ahead circuit.



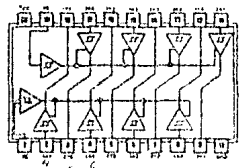
TEXAS INSTRUMENTS
INCORPORATED

PIN ASSIGNMENTS (TOP VIEWS)

OCTAL BUFFERS/LINE DRIVERS/LINE RECEIVERS

241 NONINVERTED 3-STATE OUTPUTS

See page 6-53

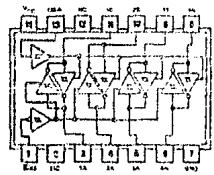


SN54LS241 (J)
SN74LS241 (J, N)
SN54S241 (J)
SN74S241 (J, N)

QUADRUPLE BUS TRANSCEIVERS

242 INVERTED 3-STATE OUTPUTS

See page 6-87

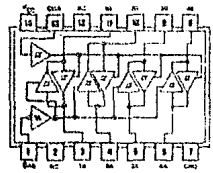


SN54LS242 (J, W)
SN74LS242 (J, N)
NC—No internal connection

QUADRUPLE BUS TRANSCEIVERS

243 NONINVERTED 3-STATE OUTPUTS

See page 6-87

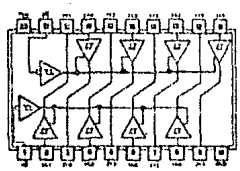


SN54LS243 (J, W)
SN74LS243 (J, N)
NC—No internal connection

OCTAL BUFFERS/LINE DRIVERS/LINE RECEIVERS

244 NONINVERTED 3-STATE OUTPUTS

See page 6-53



SN54LS244 (J)
SN74LS244 (J, N)

TYPES SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

	Typical I_{OL} (Sink Current)	Typical I_{OH} (Source Current)	Typical Propagation Delay Times		Typical Enable/ Disable Times	Typical Power Dissipation (Enabled)	
			Inverting	Noninverting		Inverting	Noninverting
SN54LS'	12 mA	−12 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN74LS'	24 mA	−15 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW
SN54S'	48 mA	−12 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW
SN74S'	64 mA	−15 mA	4.5 ns	6 ns	9 ns	450 mW	538 mW

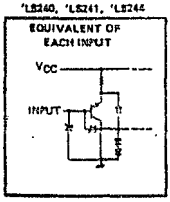
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

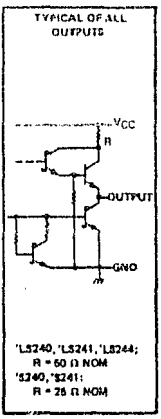
description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical $\overline{G}$ (active-low output control) inputs, and complementary G and $\overline{G}$ inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

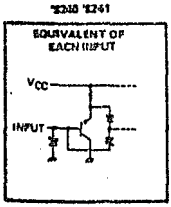
schematics of inputs and outputs

'LS240, 'LS241, 'LS244

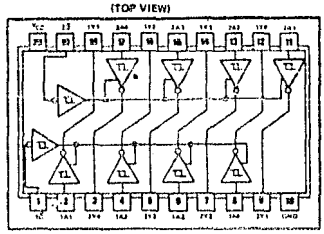




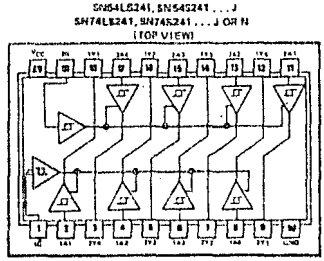
'S240 'S241



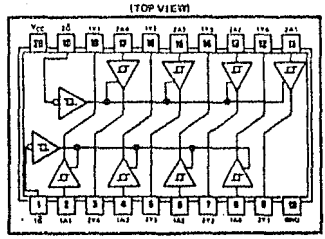
SN54LS240, SN54S240 . . . J
SN74LS240, SN74S240 . . . J OR N
(TOP VIEW)



SN54LS241, SN54S241 . . . J
SN74LS241, SN74S241 . . . J OR N
(TOP VIEW)



SN54LS244 . . . J
SN74LS244 . . . J OR N
(TOP VIEW)



recommended operating conditions

PARAMETER	SN54LS'			SN74LS'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC} (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−12			−15	mA
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

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

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

PARAMETER		TEST CONDITIONS†		SN54LS'			SN74LS'			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
	Hysteresis ($V_{T+} - V_{T-}$)	V_{CC} = MIN		0.2	0.4		0.2	0.4		V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max,	V_{IH} = 2 V, I_{OH} = −3 mA	2.4	3.4		2.4	3.4		V
		V_{CC} = MIN, V_{IL} = 0.5 V,	V_{IH} = 2 V, I_{OH} = MAX	2			2			
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	I_{OL} = 12 mA			0.4			0.4	V
			I_{OL} = 24 mA						0.5	
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, V_{IH} = 2 V, V_{IL} = V_{IL} max	V_O = 2.7 V			20			20	μA
I_{OZL}	Off-state output current, low-level voltage applied		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, any input	V_{CC} = MAX,	V_I = 2.7 V			20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX,	V_{IL} = 0.4 V			−0.2			−0.2	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX		−40		−225	−40		−225	mA
I_{CC}	Supply current	Outputs high	V_{CC} = MAX, Outputs open	All		13	23		13	23	mA
		Outputs low		'LS240		26	44		26	44	
				'LS241, 'LS244		27	46		27	46	
		All outputs disabled		'LS240		29	50		29	50	
				'LS241, 'LS244		32	54		32	54	

†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 of the short-circuit should not exceed one second.

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

PARAMETER		TEST CONDITIONS		'LS240			'LS241, 'LS244			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 45 pF, See Note 2	R_L = 667 Ω,		9	14		12	18	ns
t_{PHL}	Propagation delay time, high-to-low-level output				12	18		12	18	ns
t_{PZL}	Output enable time to low level				20	30		20	30	ns
t_{PZH}	Output enable time to high level				15	23		15	23	ns
t_{PLZ}	Output disable time from low level	C_L = 5 pF, See Note 2	R_L = 667 Ω,		15	25		15	25	ns
t_{PHZ}	Output disable time from high level				10	18		10	18	ns

NOTE 2: Load circuit and voltage waveforms are shown on page 3-11.

TYPES SN54S240, SN54S241, SN74S240, SN74S241
BUFFERS/LINE DRIVERS/LINE RECEIVERS WITH 3-STATE OUTPUTS

REVISED AUGUST 1977

recommended operating conditions

PARAMETER	SN54S'			SN74S'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC} (see Note 1)	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−12			−15	mA
Low-level output current, I_{OL}			48			64	mA
Operating free-air temperature, T_A (see Note 3)	−55		125	0		70	°C

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

3. An SN54S241J operating at free-air temperature above 116°C requires a heat sink that provides a thermal resistance from case to free-air, $R_{\theta CA}$, of not more than 40°C/W.

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

PARAMETER			TEST CONDITIONS†		'S240			'S241			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	
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 = −18 mA			−1.2			−1.2	V
	Hysteresis ($V_{T+} - V_{T-}$)		V_{CC} = MIN		0.2	0.4		0.2	0.4		V
V_{OH}	High-level output voltage	SN74S'	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V,	I_{OH} = −1 mA	2.7			2.7			V
		SN54S' and SN74S'	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V,	I_{OH} = −3 mA	2.4	3.4		2.4	3.4		
		SN54S' and SN74S'	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V,	I_{OH} = MAX	2			2			
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V,	I_{OL} = MAX			0.55			0.55	V
I_{OZH}	Off-state output current, high-level voltage applied		V_{CC} = MAX, V_{IH} = 2 V,	V_O = 2.4 V			50			50	μA
I_{OZL}	Off-state output current, low-level voltage applied		V_{IL} = 0.8 V	V_O = 0.5 V			−50			−50	
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, any input		V_{CC} = MAX,	V_I = 2.7 V			50			50	μA
I_{IL}	Low-level input current	Any A	V_{CC} = MAX,	V_I = 0.5 V			−400			−400	μA
		Any G					−2			−2	mA
I_{OS}	Short-circuit output current§		V_{CC} = MAX		−50		−225	−50		−225	mA
I_{CC}	Supply current	Outputs high	V_{CC} = MAX, Outputs open	SN54S'		80	123		95	147	mA
				SN74S'		80	135		95	160	
		Outputs low		SN54S'		100	145		120	170	
				SN74S'		100	150		120	180	
		Outputs disabled		SN54S'		100	145		120	170	
				SN74S'		100	150		120	180	

†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 of the short-circuit should not exceed one second.

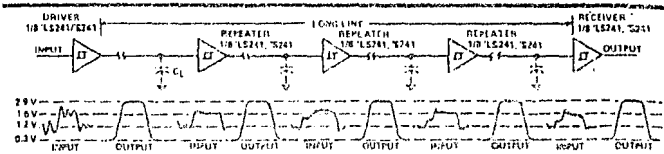
switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER		TEST CONDITIONS		'S240			'S241			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 50 pF, See Note 4	R_L = 93 Ω,		4.5	7		6	9	ns
t_{PHL}	Propagation delay time, high-to-low-level output				4.5	7		6	9	ns
t_{PZL}	Output enable time to low level				10	15		10	15	ns
t_{PZH}	Output enable time to high level				8.5	10		8	12	ns
t_{PLZ}	Output disable time from low level	C_L = 5 pF, See Note 4	R_L = 93 Ω,		10	15		10	15	ns
t_{PHZ}	Output disable time from high level				6	9		6	9	ns

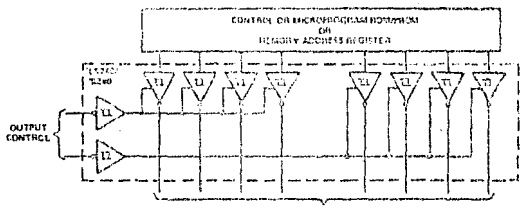
NOTE 4: Load circuit and voltage waveforms are shown on page 3-10.

TYPES SN54LS240,SN54LS241, SN54LS244,SN54S240,SN54S241,SN74LS240, SN74LS241,SN74LS244,SN74S240,SN74S241

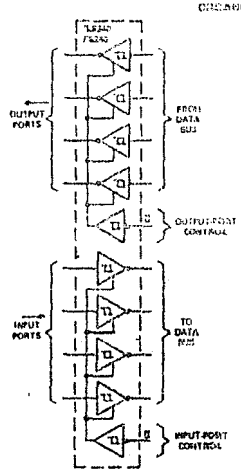
OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS



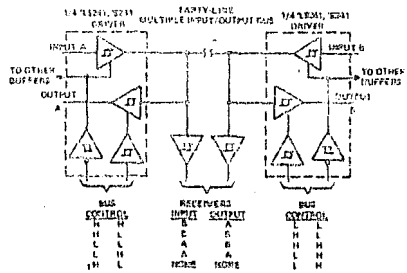
'LS241, 'S241 USED AS REPEATER/LEVEL RESTORER



'LS241, 'S240 USED AS SYSTEM ARP/OR MEMORY BUS DRIVER—4-BIT ORGANIZATION CAN BE APPLIED TO HANDLE BINARY QUAD



INDEPENDENT 4-BIT BUS DRIVERS/RECEIVERS IN A SINGLE PACKAGE



PARTY-LINE BUS SYSTEM WITH MULTIPLE INPUTS, OUTPUTS, AND RECEIVERS

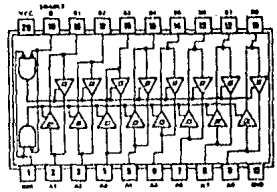
TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

PIN ASSIGNMENTS (TOP VIEWS)

OCTAL B HI TRANSCEIVERS

245 NONINVERTED 3-STATE OUTPUTS

See page 7-349



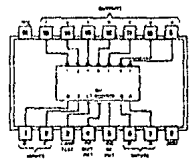
SN54LS245 (J) SN74LS245 (J, N)

BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS

246 ACTIVE-LOW, OPEN-COLLECTOR, 30-V OUTPUTS

247 ACTIVE-LOW, OPEN-COLLECTOR, 15-V OUTPUTS

See page 7-351



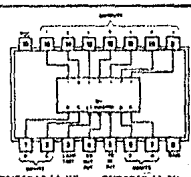
SN54246 (J, W) SN74246 (J, N)
SN54247 (J, W) SN74247 (J, N)
SN54LS247 (J, W) SN74LS247 (J, N)

BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS

248 INTERNAL PULL-UP OUTPUTS

249 OPEN-COLLECTOR OUTPUTS

See page 7-351

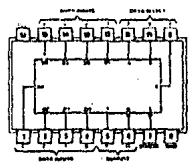


SN54248 (J, W) SN74248 (J, N)
SN54LS248 (J, W) SN74LS248 (J, N)
SN54249 (J, W) SN74249 (J, N)
SN54LS249 (J, W) SN74LS249 (J, N)

DATA SELECTORS/MULTIPLEXERS

251 TRUE AND INVERTED 3-STATE OUTPUTS

See page 7-382



SN54251 (J, W) SN74251 (J, N)
SN54LS251 (J, W) SN74LS251 (J, N)
SN54S251 (J, W) SN74S251 (J, N)

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

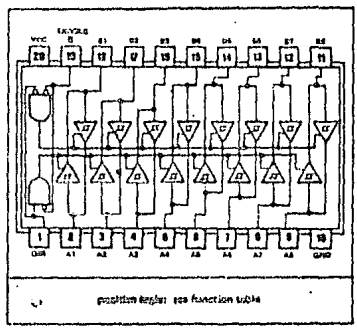
TYPES SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

BULLETIN NO. DL-S 7712471, OCTOBER 1976—REVISED AUGUST 1977

- Bi-directional Bus Transceiver in a High-Density 20-Pin Package
- 3-State Outputs Drive Bus Lines Directly
- P-N-P Inputs Reduce D-C Loading on Bus Lines
- Hysteresis at Bus Inputs Improve Noise Margins
- Typical Propagation Delay Times, Port-to-Port . . . 8 ns
- Typical Enable/Disable Times . . . 17 ns

TYPE	I_{CC} (BUS CURRENT)	I_{OH} (SOURCE CURRENT)
SN54LS245	12 mA	−12 mA
SN74LS245	24 mA	−15 mA

SN54LS245 . . . J PACKAGE
SN74LS245 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

description

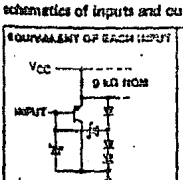
These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The device allows data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input ($\overline{G}$) can be used to disable the device so that the buses are effectively isolated.

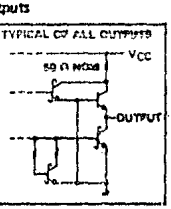
The SN54LS245 is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74LS245 is characterized for operation from 0°C to 70°C.

schematics of inputs and outputs

EQUIVALENT OF EACH INPUT



TYPICAL OF ALL OUTPUTS



FUNCTION TABLE

ENABLE $\overline{G}$	DIRECTION CONTROL DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

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

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: SN54LS245 . . . . . −55°C to 125°C
SN74LS245 . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

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

TYPES SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

REVISED AUGUST 1977

recommended operating conditions

PARAMETER	SN54LS245			SN74LS245			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}			−12			−15	mA
Low-level output current, I_{OL}			12			24	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†		SN54LS245			SN74LS245			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
	Hysteresis ($V_{T+} - V_{T-}$) A or B input		V_{CC} = MIN		0.2	0.4		0.2	0.4		V
V_{OH}	High-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	I_{OH} = −3 mA	2.4	3.4		2.4	3.4		V
				I_{OH} = MAX	2			2			
V_{OL}	Low-level output voltage		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	I_{OL} = 12 mA			0.4			0.4	V
				I_{OL} = 24 mA						0.5	
I_{OZH}	Off-state output current, high-level voltage applied		V_{CC} = MAX, $\overline{G}$ at 2 V	V_O = 2.7 V			10			10	μA
I_{OZL}	Off-state output current, low-level voltage applied			V_O = 0.4 V			−200			−200	
I_I	Input current at maximum input voltage	A or B	V_{CC} = MAX,	V_I = 5.5 V			0.1			0.1	mA
		DIR or $\overline{G}$		V_I = 7 V			0.1			0.1	
I_{IH}	High-level input current		V_{CC} = MAX,	V_{IH} = 2.7 V			20			20	μA
I_{IL}	Low-level input current		V_{CC} = MAX,	V_{IL} = 0.4 V			−0.2			−0.2	mA
I_{OS}	Short-circuit output current¶		V_{CC} = MAX		−40		−225	−40		−225	mA
I_{CC}	Supply current	Total, outputs high	V_{CC} = MAX,	Outputs open		48	70		48	70	mA
		Total, outputs low				62	90		62	90	
		Outputs at Hi-Z				64	95		64	95	

†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 of the short-circuit should not exceed one second.

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

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 45 pF,	R_L = 667 Ω, See Note 2		8	12	ns
t_{PHL}	Propagation delay time, high-to-low-level output				8	12	ns
t_{PZL}	Output enable time to low level				27	40	ns
t_{PZH}	Output enable time to high level				25	40	ns
t_{PLZ}	Output disable time from low level	C_L = 5 pF,	R_L = 667 Ω, See Note 2		15	25	ns
t_{PHZ}	Output disable time from high level				15	35	ns

NOTE 2: Load circuit and waveforms are shown on page 3-11.

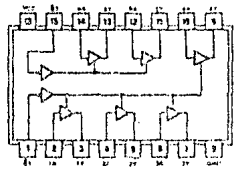
TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

PIN ASSIGNMENTS (TOP VIEWS)

HEX BUS DRIVERS

367 NONINVERTED DATA OUTPUTS
4-LINE AND 2-LINE ENABLE INPUTS
3-STATE OUTPUTS

See page 6-33

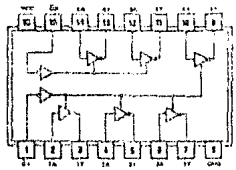


SN54367A (J, W) SN74367A (J, N)
SN54LS367A (J, W) SN74LS367A (J, N)

HEX BUS DRIVERS

368 INVERTED DATA OUTPUTS
4-LINE AND 2-LINE ENABLE INPUTS
3-STATE OUTPUTS

See page 6-35

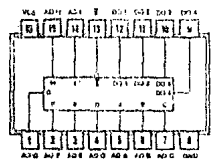


SN54368A (J, W) SN74368A (J, N)
SN54LS368A (J, W) SN74LS368A (J, N)

1024-BIT N-CHANNEL READ-ONLY MEMORIES

370 512 4-BIT WORDS
3-STATE OUTPUTS

See Bipolar Microcomputer Components Data Book, LCC4270

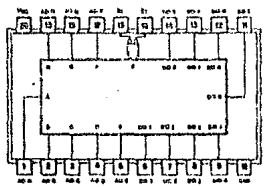


SN54S370 (J) SN74S370 (J, N)

2048-BIT READ-ONLY MEMORIES

371 256 8-BIT WORDS
3-STATE OUTPUTS

See Bipolar Microcomputer Components Data Book, LCC4270



SN54S371 (J) SN74S371 (J, N)

recommended operating conditions

	54 FAMILY / 74 FAMILY	SERIES 54 / SERIES 74 — 'S95A, 'S95A / 'S97A, 'S97A			SERIES 54LS / SERIES 74LS — 'LS365A, 'LS366A / 'LS367A, 'LS368A			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	54 Family	4.5	5	5.5	4.5	5	5.5	V
	74 Family	4.75	5	5.25	4.75	5	5.25	
High-level output current, I_{OH}	54 Family			−5.2			−1	mA
	74 Family			−5.2			−2.6	
Low-level output current, I_{OL}	54 Family			22			12	mA
	74 Family			22			24	
Operating free-air temperature, T_A	54 Family	−55		125	−55		125	°C
	74 Family	0		70	0		70	

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

PARAMETER		TEST FIGURE	TEST CONDITIONS†		SERIES 54 / SERIES 74 — 'S95A, 'S95A / 'S97A, 'S97A			SERIES 54LS / SERIES 74LS — 'LS365A, 'LS366A / 'LS367A, 'LS368A			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage	1, 2			2			2			V
V_{IL}	Low-level input voltage	1, 2					0.8			0.7	V
							0.8			0.8	
V_{IC}	Input clamp voltage	2	V_{CC} = MIN, I_I = I_{IK}				−1.5			−1.5	V
V_{OH}	High-level output voltage	1	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max, I_{OH} = MAX	54 Family	2.4	3.3		2.4	3.3		V
				74 Family	2.4	3.1		2.4	3.1		
V_{OL}	Low-level output voltage	2	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max	I_{OL} = MAX	54 Family		0.4			0.25	0.4	V
					74 Family		0.4			0.25	0.5	
				I_{OL} = 12 mA	Series 74LS					0.25	0.4	
I_{OZ}	Off-state high-impedance-state output current	10	V_{CC} = MAX, V_{IH} = 2 V, V_{IL} = V_{IL} max		V_O = 2.4 V		40			20	μA
					V_O = 0.4 V		−40			−20	
I_I	Input current at maximum input voltage	4	V_{CC} = MAX		V_I = 5.5 V		1			0.1	mA
I_{IH}	High-level input current	4	V_{CC} = MAX		V_I = 2.7 V		40			20	μA
					V_I = 2.4 V						
					V_I = 2.7 V						
I_{IL}	Low-level input current	All inputs	V_{CC} = MAX, V_I = 0.5 V, Either G input at 2 V				−40			−20	μA
			V_{CC} = MAX, V_I = 0.4 V, Both G inputs at 0.4 V				−1.6			−0.4	mA
		G inputs	V_{CC} = MAX, V_I = 0.4 V				−1.6			−0.4	
I_{OS}	Short-circuit output current§	6	V_{CC} = MAX		−40		−120	−40		−225	mA
I_{CC}	Supply current	7	V_{CC} = MAX								mA

See table on next page

† 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.
§ I_{IH} = −12 mA for SN54/74S95, and −15 mA for SN54LS/74LS, SN54S/74S.
Short-circuit output current should be applied to one output at a time, and for SN54LS/74LS and SN54S/74S, duration of the short-circuit should not exceed one second.

supply current[†]

TYPE	DATA INPUTS	OUTPUT CONTROLS	I_{CC} (mA) TYP	I_{CC} (mA) MAX
'366A, '367A	0 V	4.5 V	65	95
	4.5 V	4.5 V	90	77
LS366A, LS367A	0 V	4.5 V	14	24
	4.5 V	4.5 V	12	21
LS366A, LS368A	0 V	4.5 V	12	21
	4.5 V	4.5 V	11	19

† Maximum values of I_{CC} are given for the recommended operating ranges of V_{CC} and T_A; typical values are at V_{CC} = 5 V, T_A = 25°C.

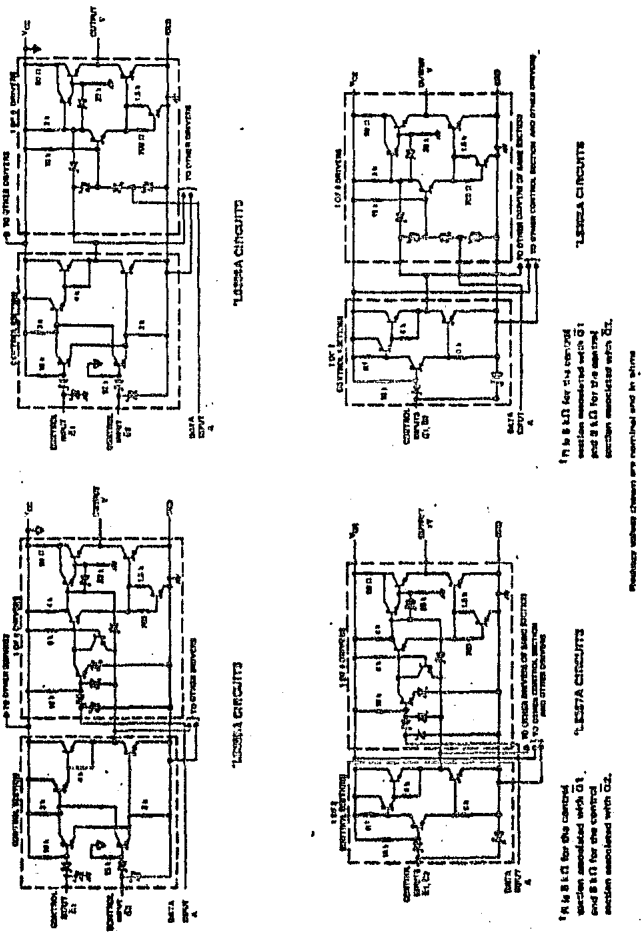
switching characteristics, V_{CC} = 5 V, T_A = 25°C, see note 1

PARAMETER*	TEST CONDITIONS	SERIES 54/74 '366A, '367A TYP	'366A, '367A MAX	SERIES 54LS/74LS LS366A, LS367A TYP	LS366A, LS367A MAX	LS368A, LS368A TYP	LS368A, LS368A MAX
t_{PLH}	C_L = 45 pF, R_L = 667 Ω	10	17	10	16	7	15
t_{PHL}		22	38	8	22	12	18
t_{PZH}		25	35	18	33	18	35
t_{PZL}		37	37	24	40	29	45
t_{PHZ}	C_L = 5 pF, R_L = 667 Ω	11	11		30		33
t_{PLZ}		27	27		25		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 1: Load circuits and voltage waveforms are shown on pages 3-10 and 3-11.

schematics

Resistor values shown are nominal and in ohms.



TEXAS INSTRUMENTS
INCORPORATED

intel

8212
8-BIT INPUT/OUTPUT PORT

- Fully Parallel 8-Bit Data Register and Buffer
- Service Request Flip-Flop for Interrupt Generation
- Low Input Load Current — .25mA Max.
- Three State Outputs
- Outputs Sink 15mA
- 3.65V Output High Voltage for Direct Interface to 8008, 8080A, or 8085A CPU
- Asynchronous Register Clear
- Replaces Buffers, Latches and Multiplexers in Microcomputer Systems
- Reduces System Package Count

The 8212 input/output port consists of an 8-bit latch with 3-state output buffers along with control and device selection logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor.

The device is multimode in nature. It can be used to implement latches, gated buffers or multiplexers. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with this device.

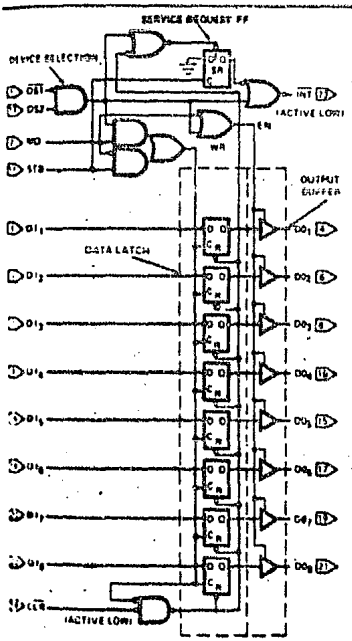
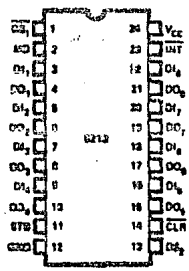


Figure 1. Logic Diagram



DI_1–DI_8	DATA IN
DO_1–DO_8	DATA OUT
$\overline{DS_1}$, DS_2	DEVICE SELECT
MD	MODE
STB	STROBE
INT	INTERRUPT (ACTIVE LOW)
CLR	CLEAR (ACTIVE LOW)

Figure 2. Pin Configuration

FUNCTIONAL DESCRIPTION

Data Latch

The 8 flip-flops that make up the data latch are of a "D" type design. The output (Q) of the flip-flop will follow the data input (DI) while the clock input (C) is high. Latching will occur when the clock (C) returns low.

The latched data is cleared by an asynchronous reset input ($\overline{CLR}$). (Note: Clock (C) Overrides Reset ($\overline{CLR}$).)

Output Buffer

The outputs of the data latch (Q) are connected to 3-state, non-inverting output buffers. These buffers have a common control line (EN); this control line either enables the buffer to transmit the data from the outputs of the data latch (Q) or disables the buffer, forcing the output into a high impedance state. (3-state)

The high-impedance state allows the designer to connect the 8212 directly onto the microprocessor bi-directional data bus.

Control Logic

The 8212 has control inputs $\overline{DS1}$, DS2, MD and STB. These inputs are used to control device selection, data latching, output buffer state and service request flip-flop.

$\overline{DS1}$, DS2 (Device Select)

These 2 inputs are used for device selection. When $\overline{DS1}$ is low and DS2 is high ($\overline{DS1} \cdot DS2$) the device is selected. In the selected state the output buffer is enabled and the service request flip-flop (SR) is asynchronously set.

MD (Mode)

This input is used to control the state of the output buffer and to determine the source of the clock input (C) to the data latch.

When MD is high (output mode) the output buffers are enabled and the source of clock (C) to the data latch is from the device selection logic ($\overline{DS1} \cdot DS2$).

When MD is low (input mode) the output buffer state is determined by the device selection logic ($\overline{DS1} \cdot DS2$) and the source of clock (C) to the data latch is the STB (Strobe) input.

STB (Strobe)

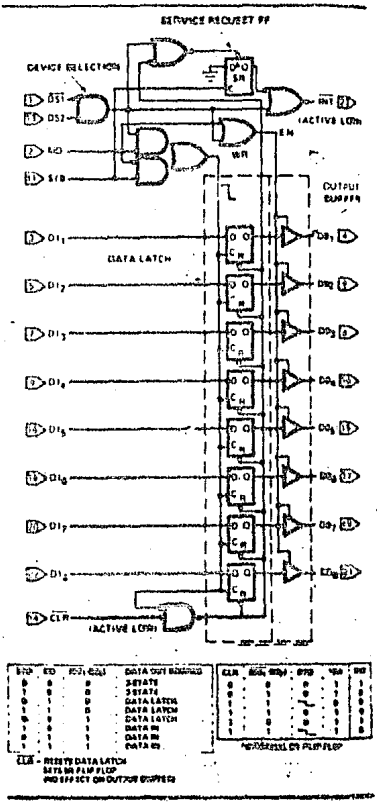
This input is used as the clock (C) to the data latch for the input mode (MD = 0) and to synchronously reset the service request flip-flop (SR).

Note that the SR flip-flop is negative edge triggered.

Service Request Flip-Flop

The (SR) flip-flop is used to generate and control interrupts in microcomputer systems. It is asynchronously set by the $\overline{CLR}$ input (active low). When the (SR) flip-flop is set it is in the non-interrupting state.

The output of the (SR) flip-flop (Q) is connected to an inverting input of a "NOR" gate. The other input to the "NOR" gate is non-inverting and is connected to the device selection logic ($\overline{DS1} \cdot DS2$). The output of the "NOR" gate (INT) is active low (interrupting state) for connection to active low input priority generating circuits



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias Plastic ..... 0°C to +70°C
Storage Temperature ..... -65°C to +160°C
All Output or Supply Voltages ..... -0.5 to +7 Volts
All Input Voltages ..... -1.0 to 5.5 Volts
Output Currents 100mA

NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may effect device reliability.

D.C. CHARACTERISTICS (T_A=0°C to +75°C, V_{CC} = +5V ± 5%)

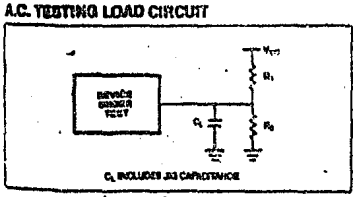
Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_F	Input Load Current, ACK, DS_2, CR, DI_1-DI_8 Inputs			-.25	mA	V_F = .45V
I_F	Input Load Current MD Input			-.75	mA	V_F = .45V
I_F	Input Load Current DS_1 Input			-1.0	mA	V_F = .45V
I_R	Input Leakage Current, ACK, DS, CR, DI_1-DI_8 Inputs			10	µA	$V_R \leq V_{CC}$
I_R	Input Leakage Current MD Input			30	µA	$V_R \leq V_{CC}$
I_R	Input Leakage Current DS_1 Input			40	µA	$V_R \leq V_{CC}$
V_C	Input Forward Voltage Clamp			-1	V	I_C = -5mA
V_{IL}	Input "Low" Voltage			.85	V	
V_{IH}	Input "High" Voltage	2.0			V	
V_{OL}	Output "Low" Voltage			.45	V	I_{OL} = 15mA
V_{OH}	Output "High" Voltage	3.65	4.0		V	I_{OH} = -1mA
I_{SC}	Short Circuit Output Current	-15		-75	mA	V_O = 0V, V_{CC} = 5V
I_{IOL}	Output Leakage Current High Impedance State			20	µA	V_O = .45V/5.25V
I_{CC}	Power Supply Current		60	130	mA	

CAPACITANCE* (f = 1MHz, V_{BIAS} = 2.5V, V_{CC} = +5V, T_A = 25°C)

Symbol	Test	Limits	
		Typ.	Max.
C_{MD}	DS_1 MD Input Capacitance	9pF	12pF
C_{IN}	$\overline{DS_2}$, $\overline{CLR}$, STB, DI_1-DI_8 Input Capacitance	5pF	8pF
C_{OUT}	DO_1-DO_8 Output Capacitance	6pF	12pF

*This parameter is sampled and not 100% tested.

A.C. TESTING LOAD CIRCUIT



SWITCHING CHARACTERISTICS

Conditions of Test

Input Pulse Amplitude = 2.5V
Input Rise and Fall Times 5ns
Between 1V and 2V Measurements made at 1.5V with 15mA and 30pF Test Load

(NOTE)
1.

Test	C_L*	R_1	R_2
t_{PD}, t_{PW}, t_R, t_F, t_C	30pF	300Ω	600Ω
t_E, ENABLE I	30pF	10KΩ	1KΩ
t_E, ENABLE I	30pF	300Ω	600Ω
t_E, DISABLE I	5pF	300Ω	600Ω
t_E, DISABLE I	5pF	10KΩ	1KΩ

*Includes probe and jig capacitance.

A.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = +5V ± 5%)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
tPW	Pulse Width	30			ns	
tPD	Data to Output Delay			30	ns	Note 1
tWE	Write Enable to Output Delay			40	ns	Note 1
tSET	Data Set Up Time	15			ns	
tH	Data Hold Time	20			ns	
tR	Reset to Output Delay			40	ns	Note 1
tS	Set to Output Delay			30	ns	Note 1
tE	Output Enable/Disable Time			45	ns	Note 1
tC	Clear to Output Delay			55	ns	Note 1

APPLICATIONS

Basic Schematic Symbols

Two examples of ways to draw the 8212 on system schematics—(1) the top being the detailed view showing pin numbers, and (2) the bottom being the symbolic view showing the system input or output as a system bus (bus containing 8 parallel lines). The output to the data bus is symbolic in referencing 8 parallel lines.

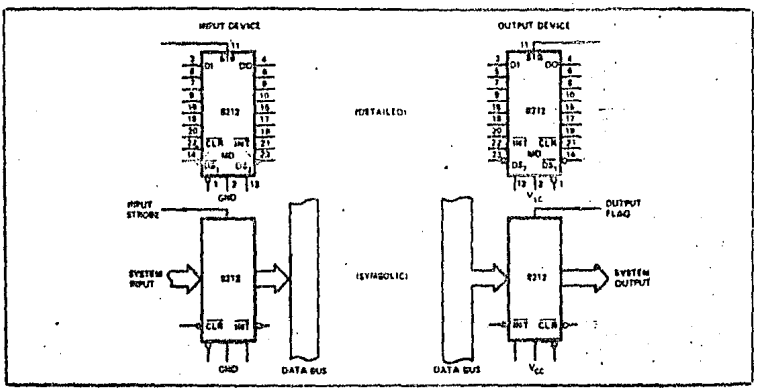


Figure 3. Basic Schematic Symbols

Gated Buffer (3-State)

The simplest use of the 8212 is that of a gated buffer. By tying the mode signal low and the strobe input high, the data latch is acting as a straight through gate. The output buffers are then enabled from the device selection logic $\overline{DS1}$ and DS2.

When the device selection logic is false, the outputs are 3-state.

When the device selection logic is true, the input data from the system is directly transferred to the output. The input data load is 250 micro amps. The output data can sink 15 milli amps. The minimum high output is 3.65 volts.

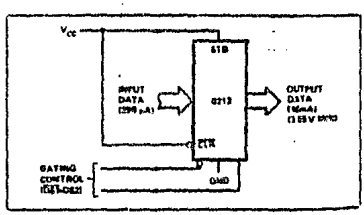


Figure 4. Gated Buffer

Bi-Directional Bus Driver

A pair of 8212's wired (back-to-back) can be used as a symmetrical drive, bi-directional bus driver. The devices are controlled by the data bus input control which is connected to $\overline{DS1}$ on the first 8212 and to DS2 on the second. One device is active, and acting as a straight through buffer the other is in 3-state mode. This is a very useful circuit in small system design.

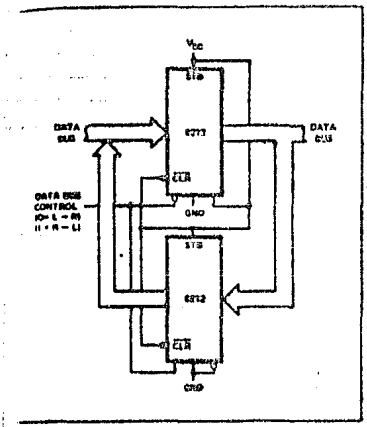


Figure 5. Bidirectional Bus Driver

Interrupting Input Port

The use of an 8212 is that of a system input port that accepts a strobe from the system input source, which in turn clears the service request flip-flop and interrupts the processor. The processor then goes through a service routine, identifies the port, and causes the device selection logic to go true — enabling the system input data onto the data bus.

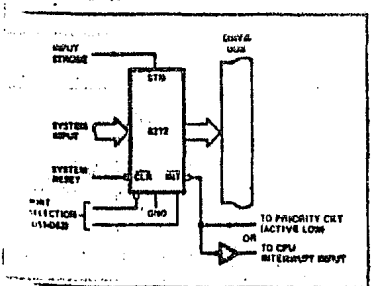


Figure 6. Interrupting Input Port

Interrupt Instruction Port

The 8212 can be used to gate the interrupt instruction, normally RESTART instructions, onto the data bus. The device is enabled from the interrupt acknowledge signal from the microprocessor and from a port selection signal. This signal is normally tied to ground. ($\overline{DS1}$ could be used to multiplex a variety of interrupt instruction ports onto a common bus).

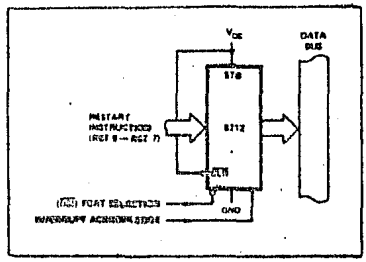


Figure 7. Interrupt Instruction Port

Output Port (With Hand-Shaking)

The 8212 can be used to transmit data from the data bus to a system output. The output strobe could be a hand-shaking signal such as "reception of data" from the device that the system is outputting to. It in turn, can interrupt the system signifying the reception of data. The selection of the port comes from the device selection logic. ($\overline{DS1}$ · DS2)

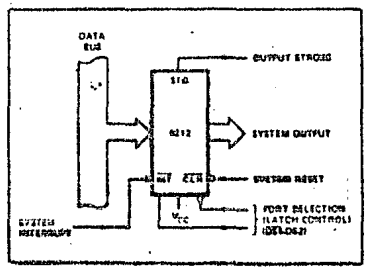
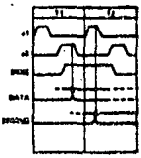
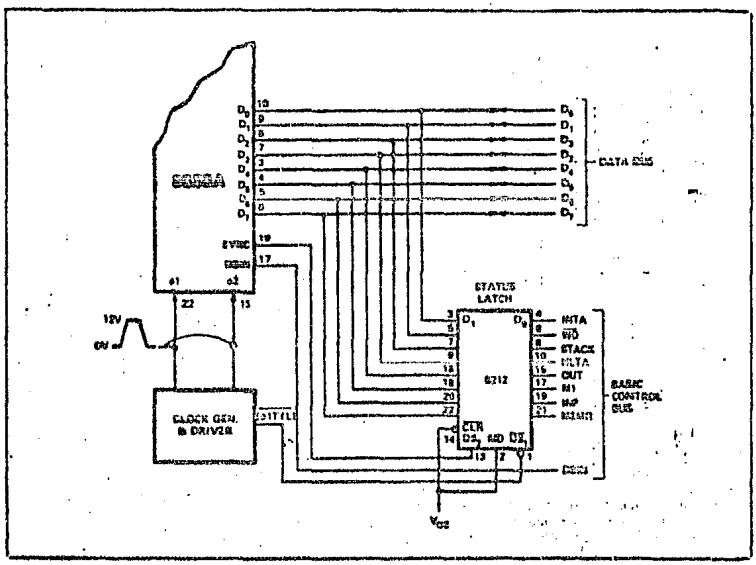


Figure 8. Output Port

8080A Status Latch

Here the 8212 is used as the status latch for an 8080A microcomputer system. The input to the 8212 latch is directly from the 8080A data bus. Timing shows that when the SYNC signal is true, which is connected to the DS2 input and the phase 1 signal is true, which is a TTL level coming from the clock generator; then, the status data will be latched into the 8212.



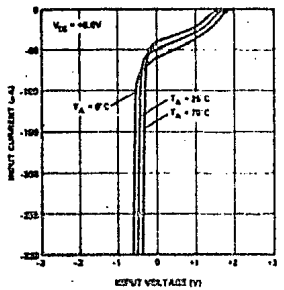


Note: The mode signal is tied high so that the output on the latch is active and enabled all the time.

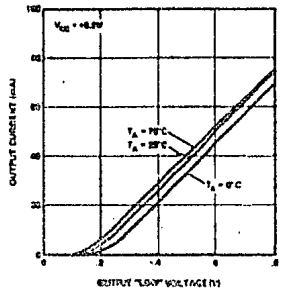
It is shown that the two areas of concern are the bi-directional data bus of the microprocessor and the control bus.

TYPICAL CHARACTERISTICS

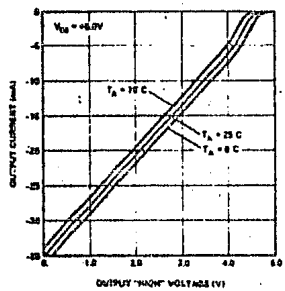
INPUT CURRENT VS. INPUT VOLTAGE



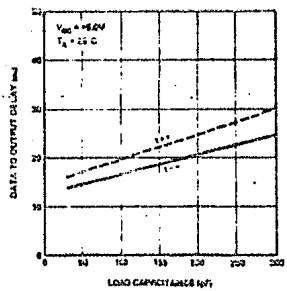
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



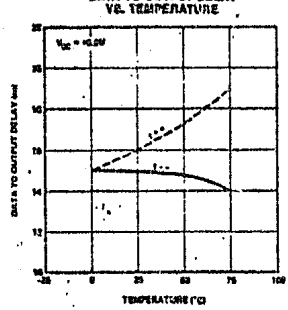
OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



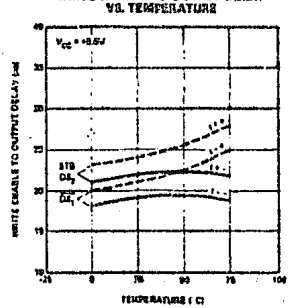
DATA TO OUTPUT DELAY VS. LOAD CAPACITANCE



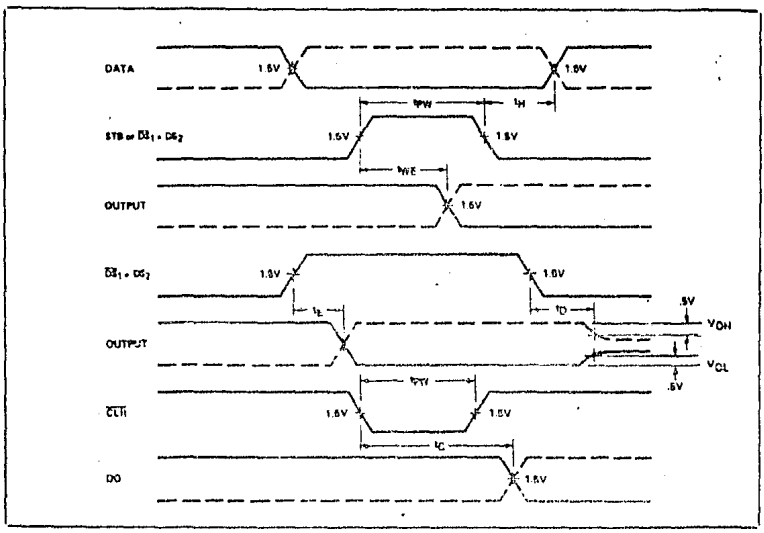
DATA TO OUTPUT DELAY VS. TEMPERATURE

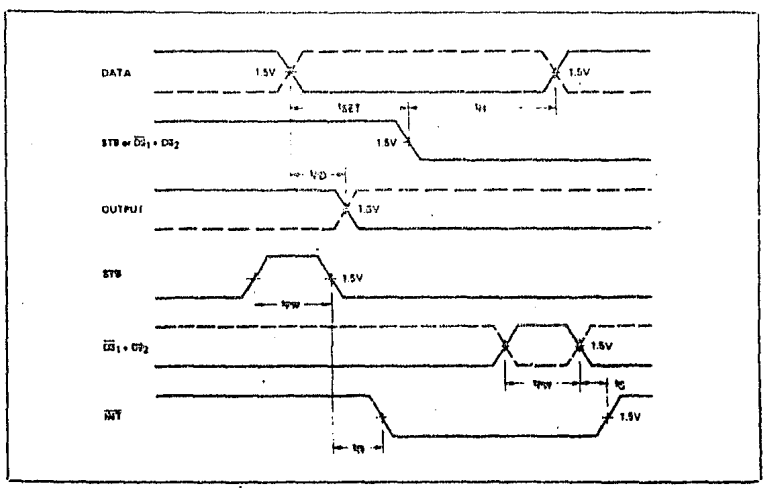


WRITE ENABLE TO OUTPUT DELAY VS. TEMPERATURE



WAVEFORMS





7-62

M8214
PRIORITY INTERRUPT CONTROL UNIT
MILITARY

- 8 Priority Levels
- Fully Expandable
- Current Status Register
- Priority Comparator
- 24-Pin Dual In-Line Package
- Military Temperature Range: −55°C to +125°C
- ±10% Power Supply Tolerance

The Intel® M8214 is an 8-level priority interrupt control unit (PICU) designed to simplify interrupt-driven microcomputer systems.

The PICU can accept 8 requesting levels, determine the highest priority, compare this priority to a software controlled current status register and issue and interrupt to the system along with vector information to identify the service routine.

The M8214 is fully expandable by the use of open collector interrupt output vector information. Control signals are also provided to simplify this function.

The PICU is designed to support a wide variety of vectored interrupt structures and reduce package count in interrupt-driven microcomputer systems.

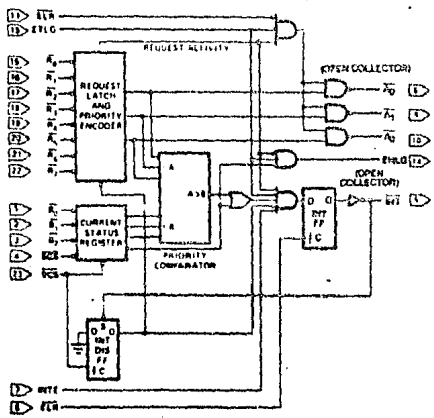


Figure 1. Logic Diagram

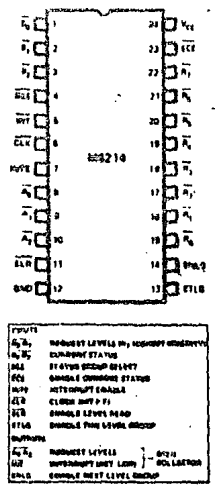


Figure 2. Pin Configuration

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias	−55°C to +125°C
Storage Temperature	−65°C to +160°C
All Output and Supply Voltages	−0.5V to +7V
All Input Voltages	−1.0V to +5.5V
Output Currents	100 mA

NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (T_A = 55°C to 125°C, V_{CC} = 5V ±10%)

Symbol	Parameter		Limits Min.	Typ.[1]	Max.	Unit	Conditions
V_C	Input Clamp Voltage (all inputs)				-1.2	V	I_C=-5mA
I_F	Input Forward Current:	ETLG input		-.15	-0.5	mA	V_F=0.45V
		all other inputs		-.09	-0.23	mA	
I_R	Input Reverse Current:	ETLG input			80	µA	V_R=5.5V
		all other inputs			40	µA	
V_{IL}	Input LOW Voltage:	all inputs			0.8	V	V_{CC}=5.0V
V_{IH}	Input HIGH Voltage:	all inputs	2.0			V	V_{CC}=5.0V
I_{CC}	Power Supply Current			90	130	mA	See Note 2.
V_{OL}	Output LOW Voltage:	all outputs		.3	.45	V	I_{OL}=10mA
V_{OH}	Output HIGH Voltage:	ENLG output	2.4	3.0		V	I_{OH}=-1mA
I_{OS}	Short Circuit Output Current:	ENLG output	-15	-35	-55	mA	V_{CC}=5.0V
I_{CEX}	Output Leakage Current:	$\overline{INT}$, $\overline{A_0}$, $\overline{A_1}$, $\overline{A_2}$			100	µA	V_{CEX}=5.5V

CAPACITANCE (V_{BIAS} = 2.5V, V_{CC} = 5V, T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Limits Min.	Typ.[1]	Max	Unit
C_{IN}	Input Capacitance		5	10	pF
C_{OUT}	Output Capacitance Except ENLG (Pin 14)		7	12	pF

A.C. CHARACTERISTICS (T_A = -55°C to +125°C, V_{CC} = +5V ±10%)

Symbol	Parameter	Limits			Unit
		Min.	Typ.[1]	Max.	
t_{CY}	$\overline{CLK}$ Cycle Time	95			ns
t_{PW}	$\overline{CLK}$, $\overline{ECS}$, $\overline{INT}$ Pulse Width	25	15		ns
t_{ISS}	INTE Setup Time to $\overline{CLK}$	16	12		ns
t_{ISH}	INTE Hold Time after $\overline{CLK}$	20	10		ns
t_{ETCS}[2]	ETLG Setup Time to $\overline{CLK}$	25	12		ns
t_{ETCH}[2]	ETLG Hold Time After $\overline{CLK}$	20	10		ns
t_{ECCS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	85	25		ns
t_{ECCH}	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{ECRS}	$\overline{ECS}$ Setup Time to $\overline{CLK}$	110	70		ns
t_{ECRH}	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			
t_{ECSS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	85	70		ns
t_{ECSH}[2]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{DCS}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{CLK}$	80	50		ns
t_{DCH}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{CLK}$	0			ns
t_{RCS}	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{CLK}$	100	60		ns
t_{RCH}	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{CLK}$	0			ns
t_{ICS}	$\overline{INT}$ Setup Time to $\overline{CLK}$	55	35		ns
t_{CI}	$\overline{CLK}$ to $\overline{INT}$ Propagation Delay		15	30	ns
t_{RIS}	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{INT}$	10	0		ns
t_{RIH}	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{INT}$	35	20		ns
t_{RA}	$\overline{R_0}$-$\overline{R_7}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		80	100	ns
t_{ELA}	$\overline{ELR}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		40	55	ns
t_{ECA}	$\overline{ECS}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		100	130	ns
t_{ETA}	ETLG to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		35	70	ns
t_{DECS}	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{ECS}$	20	10		ns
t_{DECH}	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{ECS}$	20	10		ns
t_{REN}	$\overline{R_0}$-$\overline{R_7}$ to ENLG Propagation Delay		45	70	ns
t_{ETEN}	ETLG to ENLG Propagation Delay		20	30	ns
t_{ECRN}	$\overline{ECS}$ to ENLG Propagation Delay		85	110	ns
t_{ECSN}	$\overline{ECS}$ to ENLG Propagation Delay		35	55	ns

NOTES:
1. Typical values are for T_A = 25°C, V_{CC} = 5.0V.
2. B_0-B_2, $\overline{SGS}$, CLK, $\overline{R_0}$-$\overline{R_7}$ grounded, all other inputs and all outputs open.

intel

8251A
PROGRAMMABLE COMMUNICATION INTERFACE

- Synchronous and Asynchronous Operation
- Synchronous 5–8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5–8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling
- Synchronous Baud Rate—DC to 64K Baud
- Asynchronous Baud Rate—DC to 19.2K Baud
- Full-Duplex, Double-Buffered Transmitter and Receiver
- Error Detection—Parity, Overrun and Framing
- Compatible with an Extended Range of Intel Microprocessors
- 28-Pin DIP Package
- All Inputs and Outputs are TTL Compatible
- Single +5V Supply
- Single TTL Clock

The Intel® 8251A is the enhanced version of the industry standard, Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's microprocessor families such as MCS-48, 80, 85, and iAPX-86, 88. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is fabricated using N-channel silicon gate technology.

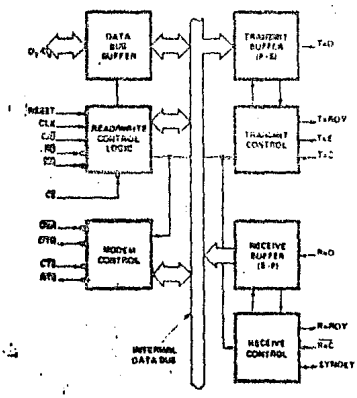


Figure 1. Block Diagram

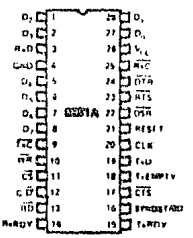


Figure 2. Pin Configuration

MAY 1981
AFN-01573C

9-165

FEATURES AND ENHANCEMENTS

The 8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.

FUNCTIONAL DESCRIPTION

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed for a wide range of Intel microcomputers such as 8048, 8080, 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support most serial data techniques in use, including IBM "bi-sync."

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The Command Status, Data-In and Data-Out registers are separate, 8-bit registers communicating with the system bus through the Data Bus Buffer.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "high" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{CY} (clock must be running).

A command reset operation also puts the device into the "Idle" state.

intel

8251A
PROGRAMMABLE COMMUNICATION INTERFACE

- Synchronous and Asynchronous Operation
- Synchronous 5–8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5–8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling
- Synchronous Baud Rate—DC to 64K Baud
- Asynchronous Baud Rate—DC to 19.2K Baud
- Full-Duplex, Double-Buffered Transmitter and Receiver
- Error Detection—Parity, Overrun and Framing
- Compatible with an Extended Range of Intel Microprocessors
- 28-Pin DIP Package
- All Inputs and Outputs are TTL Compatible
- Single +5V Supply
- Single TTL Clock

The Intel® 8251A is the enhanced version of the industry standard, Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's microprocessor families such as MCS-48, 80, 85, and iAPX-86, 88. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is fabricated using N-channel silicon gate technology.

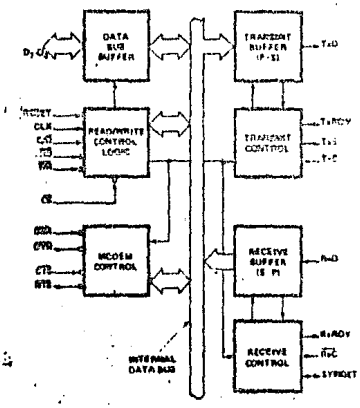


Figure 1. Block Diagram

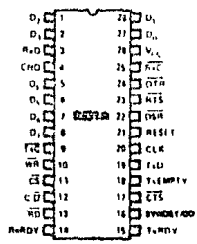


Figure 2. Pin Configuration

FEATURES AND ENHANCEMENTS

The 8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.

FUNCTIONAL DESCRIPTION

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed for a wide range of Intel microcomputers such as 8048, 8080, 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support most serial data techniques in use, including IBM "bi-sync."

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The Command Status, Data-In and Data-Out registers are separate, 8-bit registers communicating with the system bus through the Data Bus Buffer.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "high" on this input forces the 8251A into an "idle" mode. The device will remain at "idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{CY} (clock must be running).

A command reset operation also puts the device into the "idle" state.

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "low" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "low" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

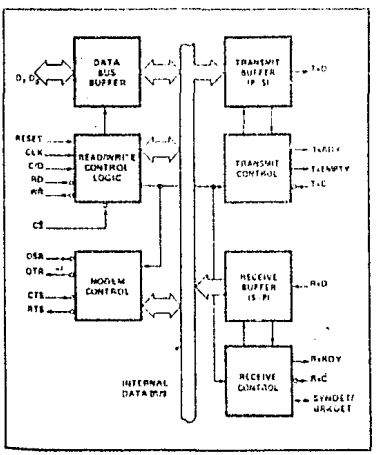


Figure 3. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → CONTROL
X	1	1	0	DATA BUS → 3 STATE
X	X	X	1	DATA BUS → 3 STATE

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.

1 = CONTROL/STATUS; 0 = DATA.

$\overline{CS}$ (Chip Select)

A "low" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus is in the float state and $\overline{RD}$ and $\overline{WR}$ have no effect on the chip.

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is a general-purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "low" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable off or $\overline{CTS}$ off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down.

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{TxC}$. The transmitter will begin transmission upon being enabled if $\overline{CTS}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable or $\overline{CTS}$ is off or the transmitter is empty.

Transmitter Control

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by TxEnable; or, for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is *not* masked by TxEnable, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to send, the TxEMPTY output will go "high." It resets upon receiving a character from the CPU if the transmitter is enabled. TxEMPTY remains low when the transmitter is disabled even if it is actually empty. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplex operational mode.

In the Synchronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers." TxEMPTY does not go low when the SYNC characters are being shifted out.

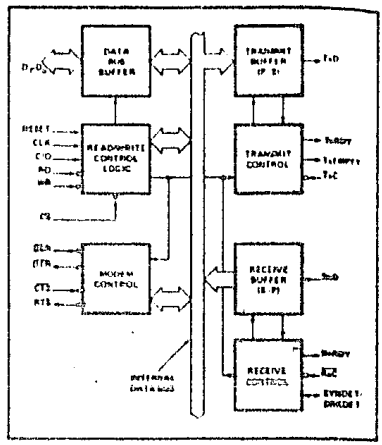


Figure 4. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

TxC (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the $\overline{TxC}$ frequency. In Asynchronous transmission mode, the baud rate is a fraction of the actual $\overline{TxC}$ frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{TxC}$.

For Example:

If Baud Rate equals 110 Baud,
$\overline{TxC}$ equals 110 Hz in the 1x mode.
$\overline{TxC}$ equals 1.72 kHz in the 16x mode.
$\overline{TxC}$ equals 7.04 kHz in the 64x mode.

The falling edge of $\overline{TxC}$ shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin, and is clocked in on the rising edge of $\overline{RxC}$.

Receiver Control

This functional block manages all receiver-related activities which consists of the following features.

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition." Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid low (Start bit) is enabled. This feature is only active in the asynchronous mode, and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the Start bit (RxD = low).

Parity error detection sets the corresponding status bit.

The Framing Error status bit is set if the Stop bit is absent at the end of the data byte (asynchronous mode).

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or, for polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

RxEnable, when off, holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be enabled to sense a Start Bit and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

RxC (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of RxC. In Asynchronous Mode, the Baud Rate is a fraction of the actual RxC frequency. A portion of the mode instruction selects this factor: 1, 1/16 or 1/64 the RxC.

For example:

Baud Rate equals 300 Baud, if
RxC equals 300 Hz in the 1x mode;
RxC equals 4800 Hz in the 16x mode;
RxC equals 19.2 kHz in the 64x mode.

Baud Rate equals 2400 Baud, if
RxC equals 2400 Hz in the 1x mode;
RxC equals 38.4 kHz in the 16x mode;
RxC equals 153.6 kHz in the 64x mode.

Data is sampled into the 8251A on the rising edge of RxC.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both TxC and RxC will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

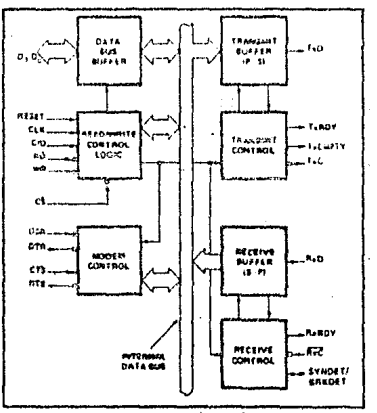


Figure 5. 8251A Block Diagram Showing Receiver Buffer and Control Functions

AFN-01573C

SYNDET (SYNC Detect/ BRKDET Break Detect)

This pin is used in Synchronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode low upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bisync), then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "high" input signal can be removed. When External SYNC Detect is programmed, Internal SYNC Detect is disabled.

BREAK (Async Mode Only)

This output will go high whenever the receiver remains low through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

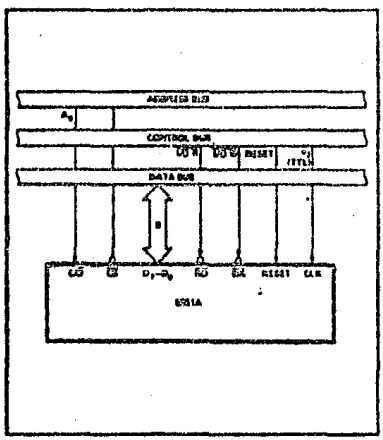


Figure 6. 8251A Interface to 8080 Standard System Bus

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "high" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.

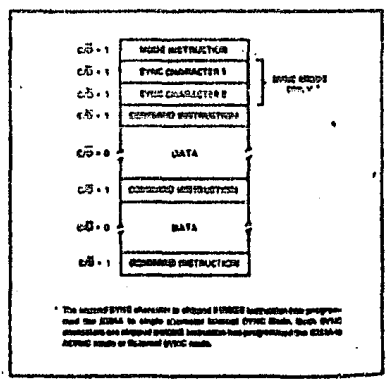


Figure 7. Typical Data Block

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

This instruction defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be written.

Command Instruction

This instruction defines a word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation (see Figure 7). The Mode Instruction must be written immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components, one Asynchronous and the other Synchronous, sharing the same package. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE: When parity is enabled it is not considered as one of the data bits for the purpose of programming the word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A, and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first), and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

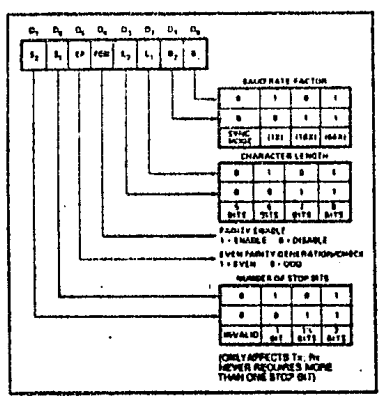


Figure 6. Mode Instruction Format, Asynchronous Mode

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the *receiver* requires only *one* stop bit, regardless of the number of stop bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset instruction. The occurrence of any of these errors will not affect the operation of the 8251A.

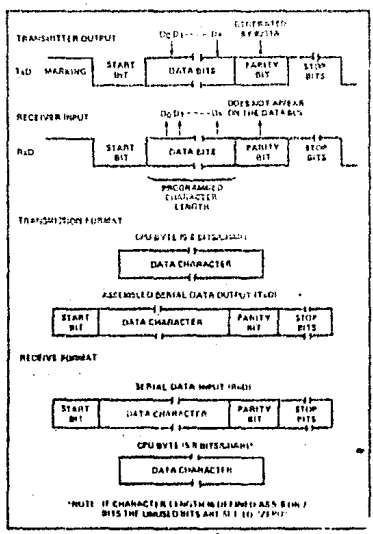
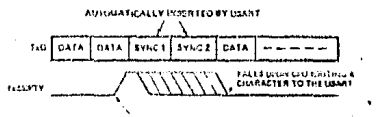


Figure 9. Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251A which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251A with a data character before the 8251A Transmitter Buffers become empty, the SYNC characters (or character if in single SYNC character mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251A is empty and SYNC characters are being sent out. TxEMPTY does not go low when the SYNC is being shifted out (see figure below). The TxEMPTY pin is internally reset by a data character being written into the 8251A.



Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the SYNC mode has been programmed, ENTER HUNT command should be included in the first command instruction word written. Data on the RxD pin is then sampled on the rising edge of $\overline{RxC}$. The content of the Rx buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ. If parity is programmed, SYNDET will not be set until the middle of the parity bit instead of the middle of the last data bit.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode. The high level can be removed after one $\overline{RxC}$ cycle. An ENTER HUNT command has no effect in the asynchronous mode of operation.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode. Parity is checked when not in Hunt, regardless of whether the Receiver is enabled or not.

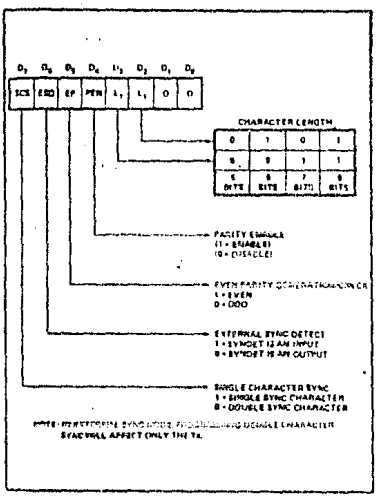


Figure 10. Mode Instruction Format, Synchronous Mode

The CPU can command the receiver to enter the HUNT mode if synchronization is lost. This will also set all the used character bits in the buffer to a "one," thus preventing a possible false SYNDET caused by data that happens to be in the Rx Buffer at ENTER HUNT time. Note that the SYNDET F/F is reset at each Status Read, regardless of whether internal or external SYNC has been programmed. This does not cause the 8251A to return to the HUNT mode. When in SYNC mode, but not in HUNT, Sync Detection is still functional, but only occurs at the "known" word boundaries. Thus, if one Status Read indicates SYNDET and a second Status Read also indicates SYNDET, then the programmed SYNDET characters have been received since the previous Status Read. (If double character sync has been programmed, then both sync characters have been contiguously received to gate a SYNDET indication.) When external SYNDET mode is selected, internal Sync Detect is disabled, and the SYNDET F/F may be set at any bit boundary.

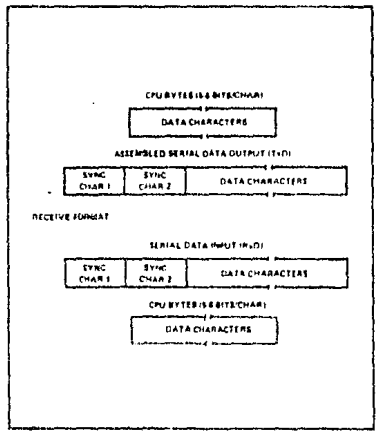


Figure 11. Data Format, Synchronous Mode

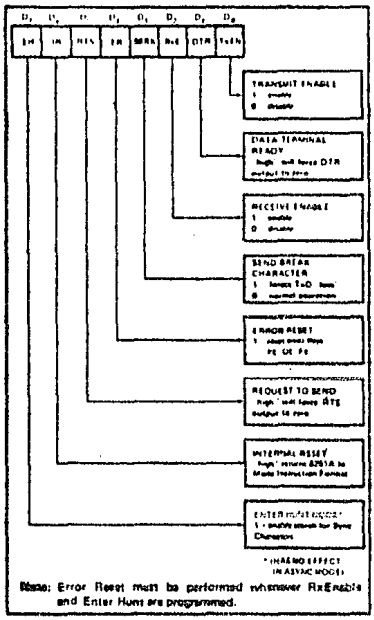
COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the sync characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, if necessary, then all further "control writes" (C/D = 1) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.

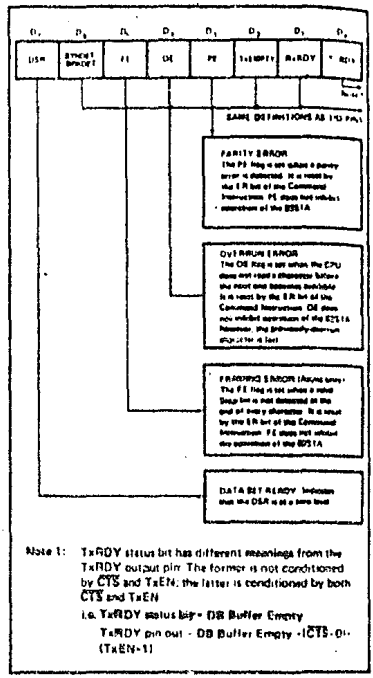
Note: Internal Reset on Power-up

When power is first applied, the 8251A may come up in the Mode, Sync character or Command format. To guarantee that the device is in the Command Instruction format before the Reset command is issued, it is safest to execute the worst-case initialization sequence (sync mode with two sync characters). Loading three 00Hs consecutively into the device with C/D = 1 configures sync operation and writes two dummy 00H sync characters. An Internal Reset command (40H) may then be issued to return the device to the "Idle" state.



Note: Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

Figure 12. Command Instruction Format



Note 1: TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by $\overline{CTS}$ and TxEN; the latter is conditioned by both $\overline{CTS}$ and TxEN

i.e. TxRDY status bit = DB Buffer Empty

TxRDY pin out = DB Buffer Empty · ($\overline{CTS}$ · (TxEN=1)

Figure 13. Status Read Format

STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (Status update is inhibited during status read.)

A normal "read" command is issued by the CPU with $C/\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251A can be used in a completely polled or interrupt-driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.

APPLICATIONS OF THE 8251A

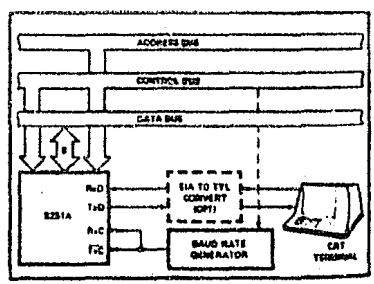


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC—9600 Baud

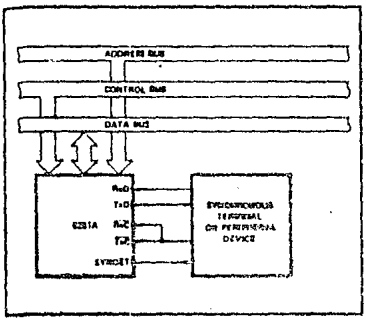


Figure 15. Synchronous Interface to Terminal or Peripheral Device

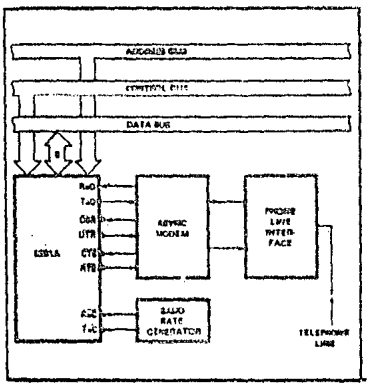


Figure 16. Asynchronous Interface to Telephone Lines

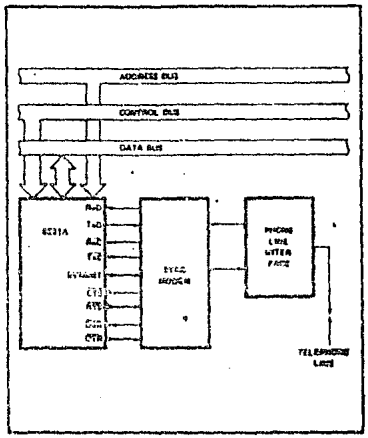


Figure 17. Synchronous Interface to Telephone Lines

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -65°C to +150°C
Voltage On Any Pin
With Respect To Ground ..... -0.5V to +7V
Power Dissipation ..... 1 Watt

NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, GND = 0V)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OL} = -400 μA
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} TO 0.45V
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} TO 0.45V
I_{CC}	Power Supply Current		100	mA	All Outputs = High

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, GND = 0V)

Bus Parameters (Note 1)

READ CYCLE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	Address Stable Before READ (CS, C/D)	0		ns	Note 2
t_{RA}	Address Hold Time for READ (CS, C/D)	0		ns	Note 2
t_{RR}	READ Pulse Width	250		ns	
t_{RD}	Data Delay from READ		200	ns	3, C_L = 150 pF
t_{DF}	READ to Data Floating	10	100	ns	

WRITE CYCLE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Stable Before WRITE	0		ns	
t_{WA}	Address Hold Time for WRITE	0		ns	
t_{WW}	WRITE Pulse Width	250		ns	
t_{DW}	Data Set-Up Time for WRITE	150		ns	
t_{WD}	Data Hold Time for WRITE	20		ns	
t_{RV}	Recovery Time Between WRITES	6		t_{CY}	Note 4

A.C. CHARACTERISTICS (Continued)

OTHER TIMINGS

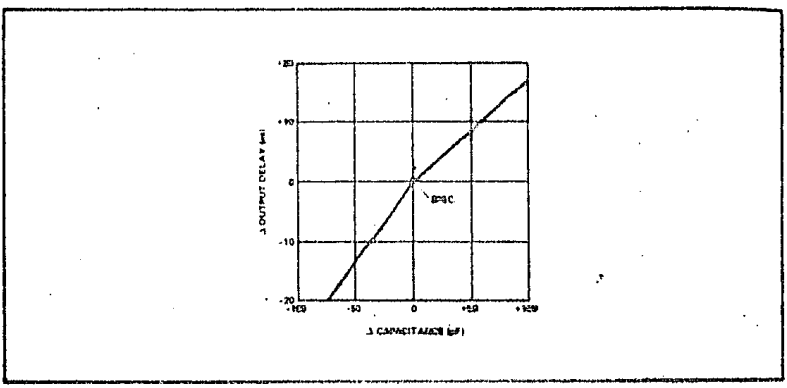
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CY}	Clock Period	320	1350	ns	Notes 5, 6
t_{ϕ}	Clock High Pulse Width	120	$t_{CY}-80$	ns	
$t_{\overline{\phi}}$	Clock Low Pulse Width	90		ns	
t_R, t_F	Clock Rise and Fall Time		20	ns	
t_{DTx}	TxD Delay from Falling Edge of $\overline{TxC}$		1	µs	
f_{Tx}	Transmitter Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{TPW}	Transmitter Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1	 	 t_{CY} t_{CY}	
t_{TPD}	Transmitter Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3	 	 t_{CY} t_{CY}	
f_{Rx}	Receiver Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{RPW}	Receiver Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1	 	 t_{CY} t_{CY}	
t_{RPD}	Receiver Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3	 	 t_{CY} t_{CY}	
t_{TxRDY}	TxRDY Pin Delay from Center of Last Bit		8	t_{CY}	Note 7
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{WR}$		400	ns	Note 7
t_{RxRDY}	RxRDY Pin Delay from Center of Last Bit		26	t_{CY}	Note 7
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{RD}$		400	ns	Note 7
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$		26	t_{CY}	Note 7
t_{ES}	External SYNDET Set-Up Time After Rising Edge of $\overline{RxC}$	18		t_{CY}	Note 7
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit	20		t_{CY}	Note 7
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)	8		t_{CY}	Note 7
t_{CR}	Control to READ Set-Up Time ($\overline{DSR}$, $\overline{CTS}$)	20		t_{CY}	Note 7

A.C. CHARACTERISTICS (Continued)

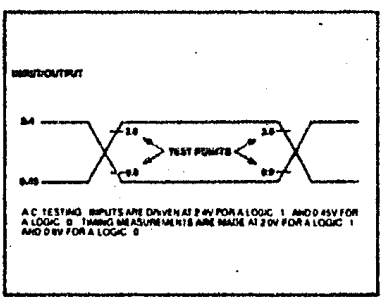
NOTES:

1. A.C. timings measured V_{OH} = 2.0 V_{OL} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.
2. Chip Select (CS) and Command/Data (C/D) are considered as Addresses.
3. Assumes that Address is valid before R_D↓.
4. This recovery time is for Mode Initialization only. Write Data is allowed only when TxRDY = 1. Recovery Time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.
5. The TxC and RxC frequencies have the following limitations with respect to CLK: For 1x Baud Rate, f_{Tx} or f_{Rx} ≤ 1/(30 t_{CY}):
   For 16x and 64x Baud Rate, f_{Tx} or f_{Rx} ≤ 1/(4.5 t_{CY}).
6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

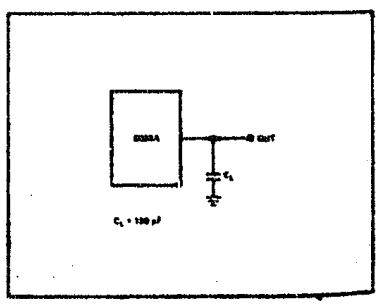
TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE (pF)



A.C. TESTING INPUT, OUTPUT WAVEFORM

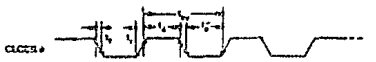


A.C. TESTING LOAD CIRCUIT

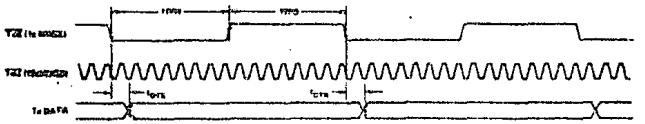


WAVEFORMS

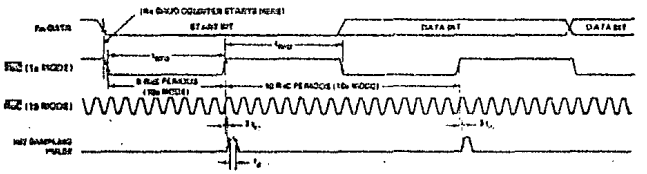
SYSTEM CLOCK INPUT



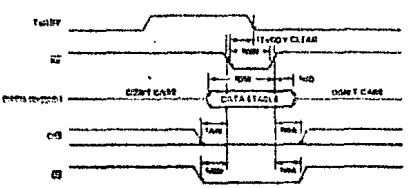
TRANSMITTER CLOCK AND DATA



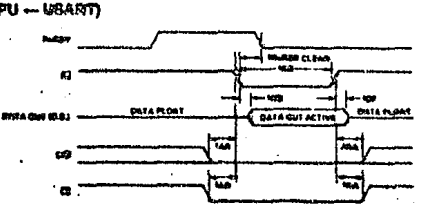
RECEIVER CLOCK AND DATA



WRITE DATA CYCLE (CPU → USART)

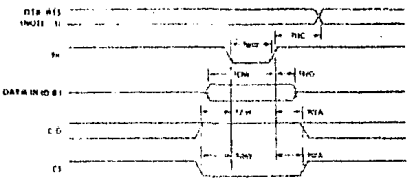


READ DATA CYCLE (CPU ← USART)

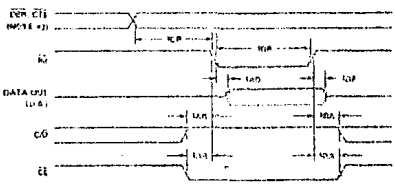


WAVEFORMS (Continued)

WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → USART)

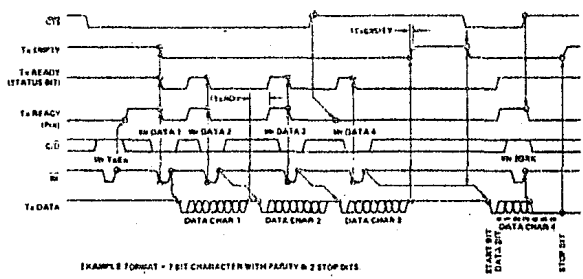


READ CONTROL OR INPUT PORT (CPU ← USART)



NOTE #1 T_{RV} INCLUDES THE RECOVERY TIMING OF A CONTROL BYTE.
NOTE #2 T_{CR} INCLUDES THE EFFECT OF CTS ON THE TxEMPL CIRCUITRY.

TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)

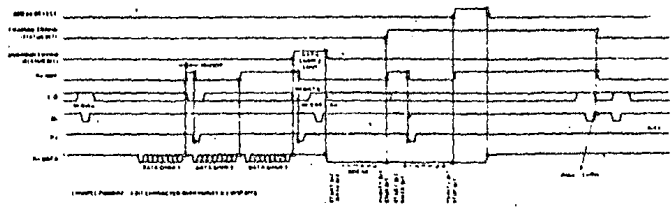


EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY & 2 STOP BITS

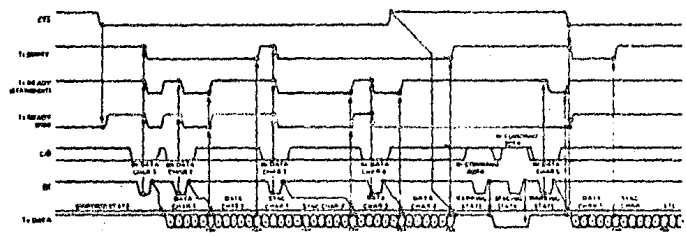
AFN-01571K

WAVEFORMS (Continued)

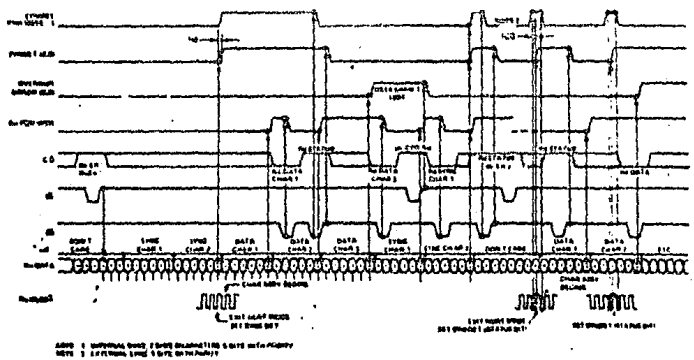
RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)



TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)



intel

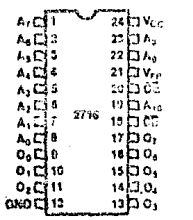
2716
16K (2K x 8) UV ERASABLE PROM

- Fast Access Time
  - — 2716-1: 350 ns Max.
  - — 2716-2: 390 ns Max.
  - — 2716: 450 ns Max.
  - — 2716-5: 490 ns Max.
  - — 2716-6: 650 ns Max.
- Single +5V Power Supply
- Low Power Dissipation
  - — Active Power: 525 mW Max.
  - — Standby Power: 132 mW Max.
- Pin Compatible to Intel 2732A EPROM
- Simple Programming Requirements
  - — Single Location Programming
  - — Programs with One 50 ms Pulse
- Inputs and Outputs TTL Compatible During Read and Program
- Completely Static

The Intel 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static standby mode, and features fast single-address programming. It makes designing with EPROMs fast, easy and economical.

The 2716, with its single 5-volt supply and with an access time up to 350 ns, is ideal for use with high-performance +5V microprocessors such as Intel's 8085 and 8086. Selected 2716-5s and 2716-6s are also available for slower speed applications. The 2716 also has a static standby mode which reduces power consumption without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 uses a simple and fast method for programming—a single TTL-level pulse. There is no need for high voltage pulsing because all programming controls are handled by TTL signals. Programming of any location at any time—either individually, sequentially or at random is possible with the 2716's single-address programming. Total programming time for all 16,384 bits is only 100 seconds.



PIN NAMES

A_0–A_{10}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0–O_7	OUTPUTS

Figure 1. Pin Configuration

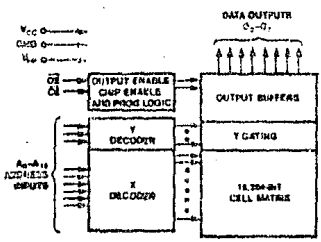


Figure 2. Block Diagram

NOVEMBER 1982
ORDER NUMBER: 210210-002

DEVICE OPERATION

The six modes of operation of the 2716 are listed in Table 1. It should be noted that inputs for all modes are TTL levels. The power supplies required are a +5V V_{CC} and a V_{PP}. The V_{PP} power supply must be at 25V during the three programming modes, and must be at 5V in the other three modes.

Read Mode

The 2716 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The 2716 has a standby mode which reduces the maximum active power dissipation by 75%, from 525 mW to 132 mW. The 2716 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because 2716s are usually used in larger memory arrays, Intel has provided a 2-line control function that accommodates this use of multiple memory connections. The two-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low-power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V_{PP} power supply is at 25V and $\overline{OE}$ is at V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active-high, TTL program pulse is applied to the $\overline{CE}$ input. A pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2716 must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Table 1. Mode Selection

Pins / Mode	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	Outputs (9–11, 13–17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Output Disable	V_{IL}	V_{IH}	+5	+5	High Z
Standby	V_{IH}	X	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

NOTES: 1. X can be V_{IL} or V_{IH}

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Program ..... +26.5V to −0.3V

*NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2716	2716-1	2716-2	2716-5	2716-6
Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ±5%	5V ±10%	5V ±5%	5V ±5%	5V ±5%
V_{PP} Power Supply[2]	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{PP1}[2]	V_{PP} Current			5	mA	V_{PP} = 5.25V
I_{CC1}[2]	V_{CC} Current (Standby)		10	25	mA	$\overline{CE}$ = V_{IH}, $\overline{OE}$ = V_{IL}
I_{CC2}[2]	V_{CC} Current (Active)		57	100	mA	$\overline{OE}$ = $\overline{CE}$ = V_{IL}
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA

A.C. CHARACTERISTICS

Symbol	Parameter	Limits (ns)										Test Conditions†
		2716		2716-1		2716-2		2716-5		2716-6		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t_{ACC}	Address to Output Delay		450		350		390		450		450	$\overline{CE}$ = $\overline{OE}$ = V_{IL}
t_{CE}	CE to Output Delay		450		350		390		490		650	$\overline{OE}$ = V_{IL}
t_{OE}[4]	Output Enable to Output Delay		120		120		120		160		200	$\overline{CE}$ = V_{IL}
t_{DF}[4,6]	$\overline{CE}$ or $\overline{OE}$ High to Output Float	0	100	0	100	0	100	0	100	0	100	$\overline{CE}$ = V_{IL}
t_{OH}	Output Hold from Addresses, CE or OE Whichever Occurred First	0		0		0		0		0		$\overline{CE}$ = $\overline{OE}$ = V_{IL}

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A high-level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2716s.

Program Inhibit

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2716s may be common. A TTL-level program pulse applied to a 27'6's $\overline{CE}$ input with V_{PP} at 25V will program that 2716. A low-level $\overline{CE}$ input inhibits the other 2716 from being programmed.

Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with V_{PP} at 25V. Except during programming and program verify, V_{PP} must be at 5V.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2716 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room-level fluorescent lighting could erase the typical 2716 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2716 window to prevent unintentional erasure.

The recommended erasure procedure for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μ W/cm^2 power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure.

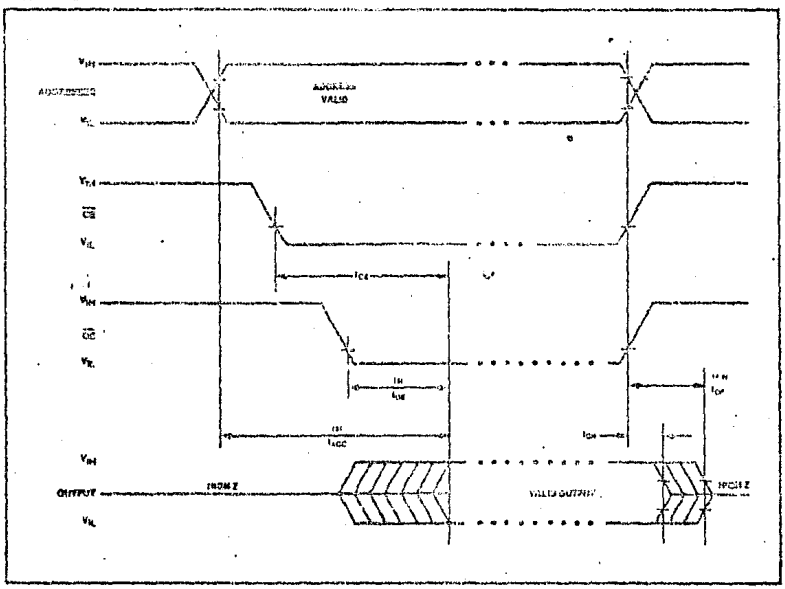
CAPACITANCE[4] (T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Typ.[3]	Max.	Units	Test Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

†A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times ..... ≤20 ns
Input Pulse Levels ..... 0.8V to 2.2V
Timing Measurement Reference Level:
Inputs ..... 0.8V and 2V
Outputs ..... 0.8V and 2V

A.C. WAVEFORMS[1]



NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for T_A = 25°C and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested.
5. OE may be delayed up to $t_{ACC}-t_{OE}$ after the falling edge of CE without impact on t_{ACC}.
6. t_{DF} is specified from OE or CE, whichever occurs first.

PROGRAMMING CHARACTERSITICS

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25°C \pm 5°C$, V_{CC}[1] $= 5V \pm 5\%$, V_{PP}[1,2] $= 25V \pm 1V$

Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions
I_{LI}	Input Current (for Any Input)			10	μA	$V_{IN} = 5.25V/0.45$
I_{PP1}	V_{PP} Supply Current			5	mA	$\overline{CE} = V_{IL}$
I_{PP2}	V_{PP} Supply Current During Programming Pulse			30	mA	$\overline{CE} = V_{IH}$
I_{CC}	V_{CC} Supply Current			100	mA	
V_{IL}	Input Low Level	-0.1		0.8	V	
V_{IH}	Input High Level	2.0		$V_{CC}+1$	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25°C \pm 5°C$, V_{CC}[1] $= 5V \pm 5\%$, V_{PP}[1,2] $= 25V \pm 1V$

Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions*
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	2			μs	
t_{OEH}	$\overline{OE}$ Hold Time	2			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}	Output Enable to Output Float Delay	0		200	ns	$\overline{CE} = V_{IL}$
t_{OE}	Output Enable to Output Delay			200	ns	$\overline{CE} = V_{IL}$
t_{PW}	Program Pulse Width	45	50	55	ms	
t_{PRT}	Program Pulse Rise Time	5			ns	
t_{PFT}	Program Pulse Fall Time	5			ns	

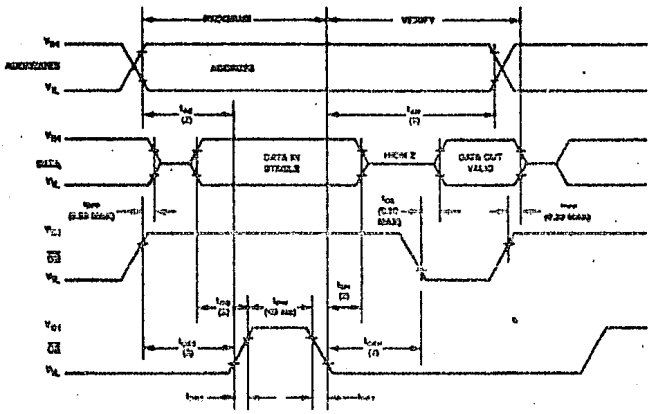
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.8 to 2.2V
Input Timing Reference Level ..... 0.8V and 2V
Output Timing Reference Level ..... 0.8V and 2V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The 2716 must not be inserted into or removed from a board with V_{PP} at 25 ±1V to prevent damage to the device.
2. The maximum allowable voltage which may be applied to the V_{PP} pin during programming is +25V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 26V maximum specification.

PROGRAMMING WAVEFORMS



NOTE:
1. ALL TIMES SHOWN IN PARENTHESIS ARE MINIMUM TIMES AND ARE ± 5% UNLESS OTHERWISE NOTED.
2. t_{DF} AND t_{OPW} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER

intel

2732A
32K (4K x 8) UV ERASABLE PROM

- 200 ns (2732A-2) Maximum Access Time . . . HMOS*-E Technology
- Compatible with High-Speed 8mHz iAPX 186...Zero WAIT State
- Two Line Control
- Compatible with 12 MHz 8051 Family
- Industry Standard Pinout . . . JEDEC Approved
- Low Standby Current...30 mA Maximum
- ±10% V_{CC} Tolerance Available
- int$_e$ligent Identifier™ Mode

The Intel 2732A is a 5V only, 32,768 bit ultraviolet erasable and electrically programmable read-only-memory (EPROM). The standard 2732A access time is 250 ns with speed selection (2732A-2) available at 200 ns. The access time is compatible with high performance microprocessors such as the 8 MHz iAPX 186. In these systems, the 2732A allows the microprocessor to operate without the addition of WAIT states.

An important 2732A feature is the separate output control, Output Enable ($\overline{OE}$), from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 2732A has a standby mode which reduces power consumption without increasing access time. The maximum active current is 125 mA, while the maximum standby current is only 35 mA, a 70% saving. The standby mode is selected by applying the TTL-high signal to the $\overline{CE}$ input.

The 2732A is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

*HMOS is a patented process of Intel Corporation.

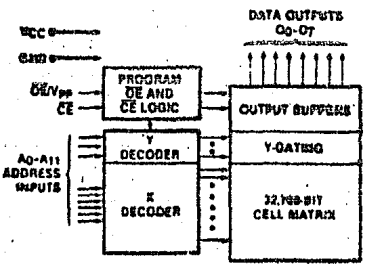


Figure 1. Block Diagram

PIN NAMES

A_0-A_{11}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}/V_{PP}$	OUTPUT ENABLE/ V_{PP}
O_0-O_7	OUTPUTS

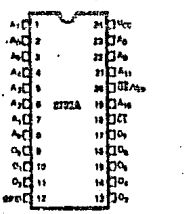


Figure 2. Pin Configuration

ERASURE CHARACTERISTICS

The erasure characteristics of the 2732A are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2732A in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2732A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2732A window to prevent unintentional erasure.

The recommended erasure procedure for the 2732A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000 μW/cm^2 power rating. The 2732A should be placed within 1 inch of the lamp tubes during erasure.

DEVICE OPERATION

The six modes of operation of the 2732A are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for $\overline{OE}/V_{PP}$ during programming and 12V on A_9 for the Intelligent Identifier™ mode. In the program mode the $\overline{OE}/V_{PP}$ input is pulsed from a TTL level to 21V.

Table 1. Mode Selection

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	A_9 (22)	V_{CC} (24)	OUTPUTS (9-11,13-17)
Read	V_{IL}	V_{IL}	X	+5	D_{OUT}
Output Disable	V_{IH}	V_{IH}	X	+5	High Z
Standby	V_{IH}	X	X	+5	High Z
Program	V_{IL}	V_{PP}	X	+5	D_{IN}
Program Inhibit	V_{IH}	V_{PP}	X	+5	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_{ID}	+5	Code

Notes: 1. X can be V_{IH} or V_{IL}.
2. V_{ID} = 12.0 ± 0.5V

Read Mode

The 2732A has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

Standby Mode

The 2732A has a standby mode which reduces the maximum active current from 125 mA to 35 mA. The 2732A is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided a 2 line control function that accommodates this use of multiple memory connection. The two line control function allows for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

PROGRAMMING

CAUTION: Exceeding 22V on Pin 20 ($\overline{OE}/V_{PP}$) will permanently damage the 2732A.

Initially, and after each erasure, all bits of the 2732A are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2732A is in the programming mode when the $\overline{OE}/V_{PP}$ input is at 21V. It is required that a 0.1 μF capacitor be placed across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time —either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2732A must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming of multiple 2732As in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2732As may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2732As.

Program Inhibit

Programming of multiple 2732As in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2732As may be common. A TTL level program pulse applied to a 2732A's $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 21V will program that 2732A. A high level $\overline{CE}$ input inhibits the other 2732As from being programmed.

Verify

A verify (Read) should be performed on the programmed bits to determine that they were correctly programmed. The verify is accomplished with $\overline{OE}/V_{PP}$ and $\overline{CE}$ at V_{IL}. Data should be verified t_{DV} after the falling edge of $\overline{CE}$.

Int$_e$ligent Identifier™ Mode

The Int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 22) of the 2732A. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 8) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during Int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2732A, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Intel will begin manufacturing 2732As during 1982 that will contain the Int$_e$ligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, preidentifier mode 2732As will respond with the current data contained in locations 0 and 1 when subjected to the Int$_e$ligent Identifier operation.

System Consideration

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

Table 2. 2732A Int$_e$ligent Identifier™ Bytes

Identifier \ Pins	A_0 (8)	O_7 (17)	O_6 (16)	O_5 (15)	O_4 (14)	O_3 (13)	O_2 (11)	O_1 (10)	O_0 (9)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	0	1	01

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +6V to −0.3V
Voltage on Pin 22 with Respect to Ground ..... +13.5V to −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming ..... +22V to −0.3V

*NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2732A/A-2/A-3/A-4	2732A-20/A-25/A-20
Operating Temperature Range	0°C–70°C	0°C–70°C
V_{CC} Power Supply	5V ± 5%	5V ± 10%

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Units	Conditions
I_{IL}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{CC1}	V_{CC} Current (Standby)			35	mA	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}	V_{CC} Current (Active)			125	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ μA

A.C. CHARACTERISTICS

Symbol	Parameter	2732A-2 2732A-20 Min.	Max.	2732A 2732A-25 Min.	Max.	2732A-3 2732A-30 Min.	Max.	2732A-4 Min.	Max.	Units	Test Conditions†
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		70		100		150		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[2]	$\overline{OE}$ High to Output Not Driven	0	60	0	60	0	130	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

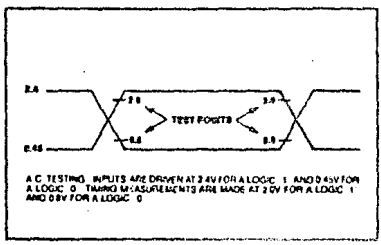
†A.C. TEST CONDITIONS

Output Load ..... 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times ..... ≤ 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Timing Measurement Reference Level:
Inputs ..... 0.8 and 2.0V
Outputs ..... 0.8 and 2.0V

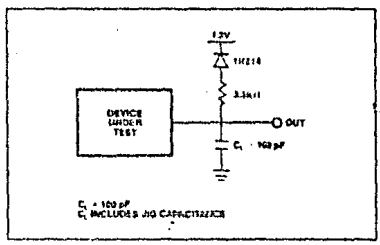
CAPACITANCE[2] (T_A = 25°C, f = 1 MHz)

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN1}	Input Capacitance Except $\overline{OE}/V_{PP}$	4	6	pF	V_{IN} = 0V
C_{IN2}	$\overline{OE}/V_{PP}$ Input Capacitance		20	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

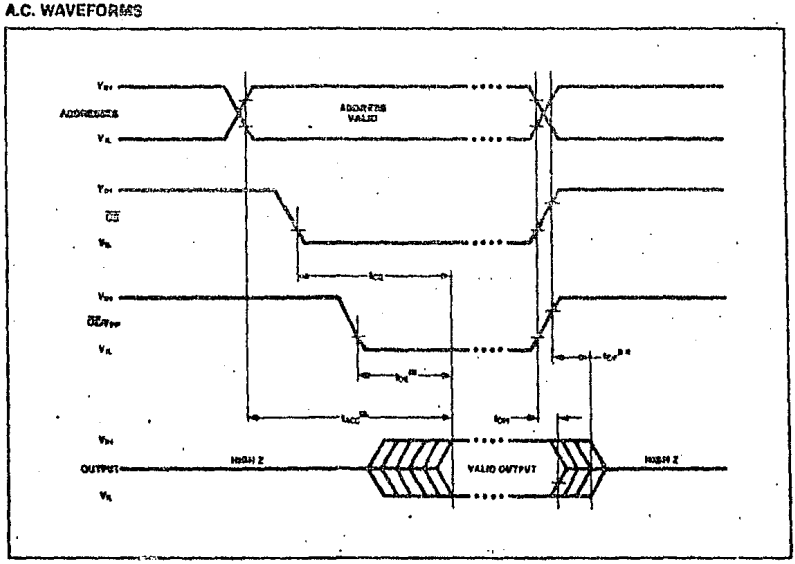
A.C. TESTING INPUT/OUTPUT WAVEFORM



A.C. TESTING LOAD CIRCUIT



A.C. WAVEFORMS



PROGRAMMING[4]

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C, V_{CC} = 5V \pm 5\%, V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.	Max.		
I_{LI}	Input Current (All Inputs)			10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify			0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4			V	$I_{OH} = -400$ μA
I_{CC}	V_{CC} Supply Current		65	125	mA	
V_{IL}	Input Low Level (All Inputs)	−0.1		0.8	V	
V_{IH}	Input High Level (All Inputs Except OE/V_{PP})	2.0		$V_{CC}+1$	V	
I_{PP}	V_{PP} Supply Current			30	mA	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{PP}$
V_{ID}	A9 Intelligent Identifier Voltage	11.5		12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C, V_{CC} = 5V \pm 5\%, V_{PP} = 21V \pm 0.5V$

Symbol	Parameter	Limits			Units	Test Conditions†
		Min.	Typ.	Max.		
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{OEH}	$\overline{OE}$ Hold Time	2			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}	Chip Enable High to Output Not Driven	0		130	ns	
t_{DV}	Data Valid from $\overline{CE}$			1	μs	$\overline{CE} = V_{IL}$, $\overline{OE} = V_{IL}$
t_{PW}	$\overline{CE}$ Pulse Width During Programming	45	50	55	ms	
t_{PHT}	$\overline{OE}$ Pulse Rise Time During Programming	50			ns	
t_{VR}	V_{PP} Recovery Time	2			μs	

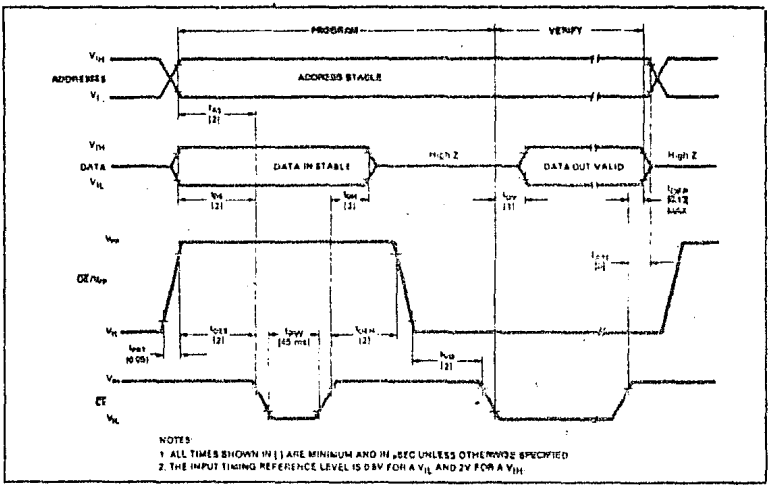
†A.C. TEST CONDITIONS

Input Rise and Fall Times (10% to 90%) ..... ≤20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. Typical values are for $T_A = 25°C$ and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested. Output float is defined as the point where data is no longer driven — see timing diagram on page 5.
3. OE may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of CE without impacting t_{ACC}.
4. When programming the 2732A, a 0.1μF capacitor is required across OE/V_{PP} and ground to suppress spurious voltage transients which may damage the device.

PROGRAMMING WAVEFORMS



intel

2764
64K (8K x 8) UV ERASABLE PROM

- 200 ns (2764-2) Maximum Access Time . . . HMOS*-E Technology
- Compatible with High-Speed 8mHz iAPX 186...Zero WAIT State
- Two Line Control
- Pin Compatible to 27128 EPROM
- Intelligent Programming™ Algorithm
- Industry Standard Pinout . . . JEDEC Approved
- Low Active Current...100mA Max.
- ±10% V_{CC} Tolerance Available

The Intel 2764 is a 5V only, 65,536-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The standard 2764 access time is 250 ns with speed selection available at 200 ns. The access time is compatible with high-performance microprocessors such as Intel's 8 mHz iAPX 186. In these systems, the 2764 allows the microprocessor to operate without the addition of WAIT states. The 2764 is also compatible with the 12 MHz 8051 family.

An important 2764 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 2764 has a standby mode which reduces power consumption without increasing access time. The maximum active current is 100 mA, while the maximum standby current is only 40 mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

±10% V_{CC} tolerance is available as an alternative to the standard ±5% V_{CC} tolerance for the 2764. This can allow the system designer more leeway with regard to his power supply requirements and other system parameters.

The 2764 is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

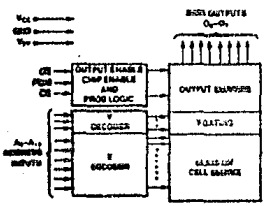


Figure 1. Block Diagram

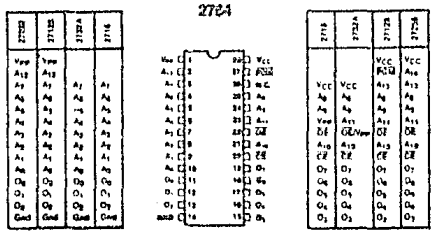


Figure 2. Pin Configurations

MODE SELECTION

Mode / Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	PGM (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (O_7-O_0, 11-13, 15-19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
Intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

1. X can be V_{IH} or V_{IL}.
2. V_H = 12.0V ± 0.5V.

PIN NAMES

A_0-A_{12}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
PGM	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation

NOVEMBER 1982
ORDER NUMBER: 210570-003

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... −10°C to +80°C
Storage Temperature ..... −65°C to +125°C
All Input or Output Voltages with Respect to Ground ..... +7.0V to −0.6V
Voltage on Pin 24 with Respect to Ground ..... +13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming ..... +22V to −0.6V

*NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2764-2	2764	2764-3	2764-4	2764-25	2764-30	2764-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits Min	Typ[3]	Max	Unit	Conditions
I_{LI}	Input Load Current			±10	µA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	µA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby			40	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active		70	100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ µA

A.C. CHARACTERISTICS

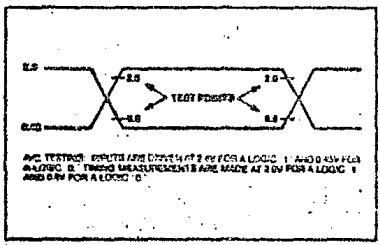
Symbol	Parameter	2764-2 & 2764 Limits Min	Max	2764-25 & 2764 Limits Min	Max	2764-30 & 2764-3 Limits Min	Max	2764-45 & 2764-4 Limits Min	Max	Unit	Test Conditions
t_{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		75		100		120		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	60	0	105	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

NOTES: 1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP}.
3. Typical values are for $T_A = 25°C$ and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on page 3.

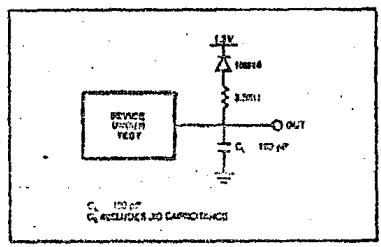
CAPACITANCE (T_A = 25°C, f = 1MHz)

Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

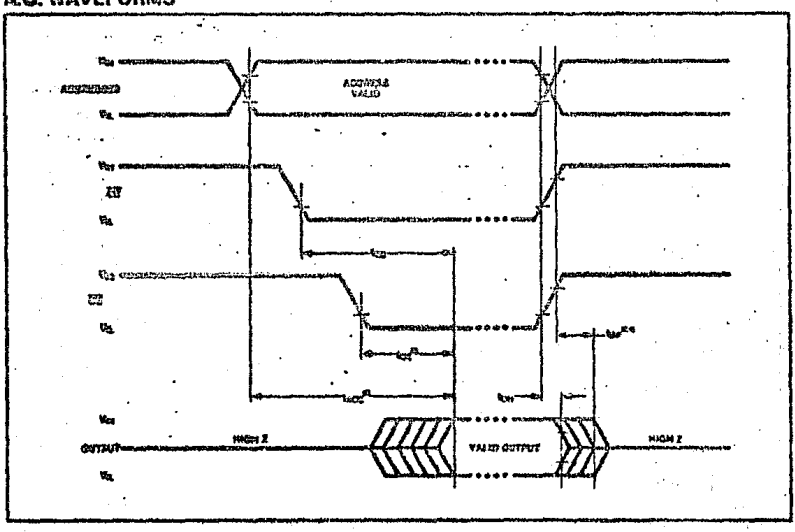
A.C. TESTING INPUT/OUTPUT WAVEFORM



A.C. TESTING LOAD CIRCUIT



A.C. WAVEFORMS



NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. OE may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of CE without impact on t_{ACC}.
4. t_{DF} is specified from OE or CE, whichever occurs first.

STANDARD PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ±5°C, V_{CC} = 5V ±5%, V_{PP} = 21V ±0.5V (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	µA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input Hight Level	2.0	V_{CC}+1	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = −400 µA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE}$ = V_{IL} = PGM
V_{ID}	A9 for Intelligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ±5°C, V_{CC} = 5V ±5%, V_{PP} = 21V ±0.5V (see Note 1)

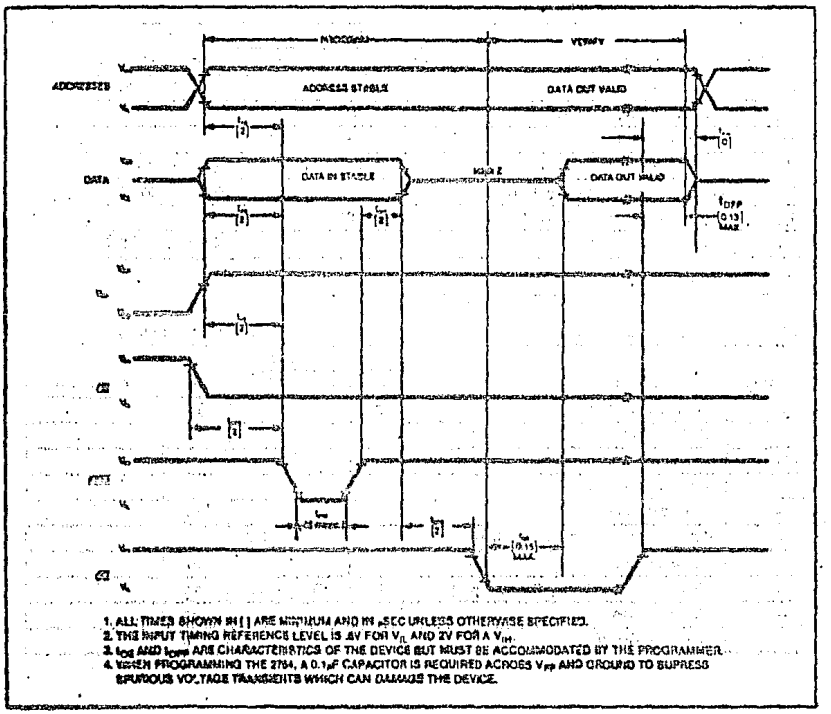
Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			µs	
t_{OES}	$\overline{OE}$ Setup Time	2			µs	
t_{DS}	Data Setup Time	2			µs	
t_{AH}	Address Hold Time	0			µs	
t_{DH}	Data Hold Time	2			µs	
t_{DFP}[2]	Output Enable to Output Float Delay	0		130	ns	
t_{VS}	V_{PP} Setup Time	2			µs	
t_{PW}	$\overline{PGM}$ Pulse Width During Programming	45	50	55	ms	
t_{CES}	$\overline{CE}$ Setup Time	2			µs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

*A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%).....20ns
Input Pulse Levels.....0.45V to 2.4V
Input Timing Reference Level.....0.8V and 2.0V
Output Timing Reference Level.....0.8V and 2.0V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on page 5.

STANDARD PROGRAMMING WAVEFORMS



1. ALL TIMES SHOWN IN [ ] ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .4V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{DF} AND t_{OPW} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 2764, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2764 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2764 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2764 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 2764 window to prevent unintentional erasure.

The recommended erasure procedure for the 2764 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 2764 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 2764 can be exposed to without damage is 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2). Exposure of the 2764 to high intensity UV light for long periods may cause permanent damage.

DEVICE OPERATION

The eight modes of operation of the 2764 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for Intelligent Identifier mode.

Table 1. MODE SELECTION

MODE \ PINS	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
Intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. $V_H = 12.0V \pm 0.5V$

READ MODE

The 2764 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 2764 has a standby mode which reduces the maximum active current from 100 mA to 40 mA. The 2764 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer — the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on *every* device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board traces.

PROGRAMMING MODES

Caution: Exceeding 22V on pin 1 (V_{PP}) will permanently damage the 2764.

Initially, and after each erasure, all bits of the 2764 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2764 is in the programming mode when V_{PP} input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Standard Programming

For programming, $\overline{CE}$ should be kept TTL-low at all times while V_{PP} is kept at 21V. When the address and data are stable, a 50 msec, active-low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec.

Programming of multiple 2764s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2764s may be connected together when they are programmed with the same data. A low-level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled 2764s.

Program Inhibit

Programming of multiple 2764s in parallel with different data is also easily accomplished by using the Program Inhibit

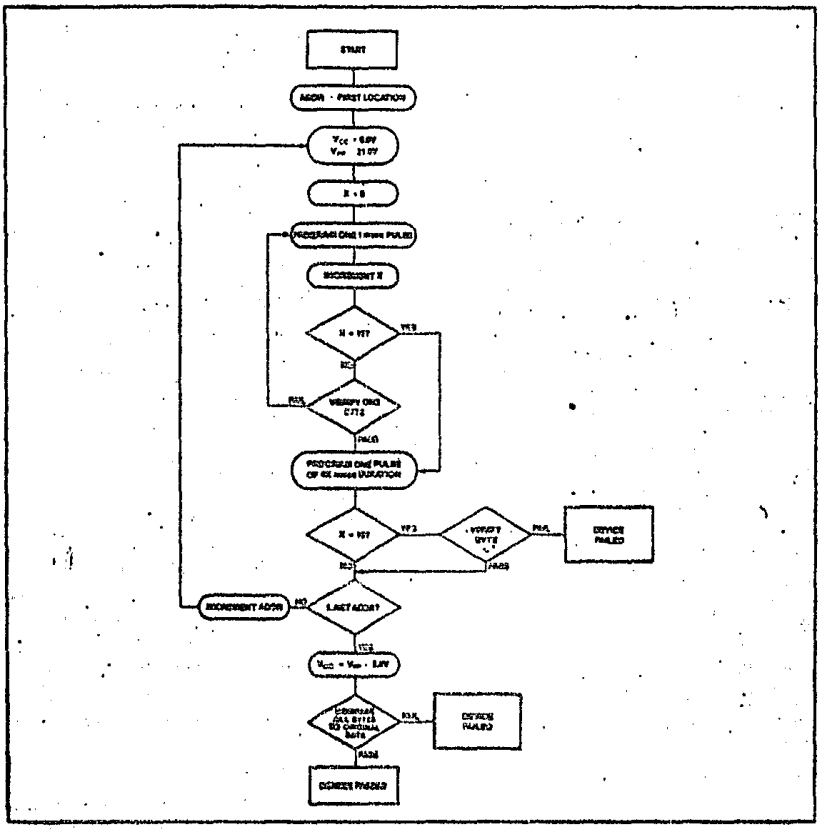


Figure 3. 2764 Intelligent Programming™ Flowchart

mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 2764s from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2764s may be common. A TTL low-level pulse applied to a 2764 $\overline{CE}$ and $\overline{PGM}$ input with V_{PP} at 21V will program that 2764.

Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify is performed with $\overline{CE}$ and $\overline{OE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 21V.

Intelligent Programming™ Algorithm

The 2764 Intelligent Programming Algorithm allows Intel 2764s to be programmed in a significantly faster time than the standard 50 msec per-byte programming routine. Typical programming times for 2764s are on the order of a minute and a half, which is a five-fold reduction in programming time from the standard method. This fast algorithm results in the same reliability characteristics as the standard 50 msec algorithm. A flowchart of the Intelligent Programming Algorithm is shown in Figure 3. This is compatible with the 27128 Intelligent Programming Algorithm.

With the standard programming method, data is programmed into a selected 2764 location by a single 50 msec, active-low, TTL pulse applied to the $\overline{PGM}$ pin. The Int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial PGM pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 2764 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the Int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

Int$_e$ligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ±5°C, V_{CC} = 6.0V ±0.25V, V_{PP} = 21V ±0.5V (see Note 1)

Symbol	Parameter	Limits Min.	Max.	Unit	Test Conditions
I_{LI}	Input Current (All Inputs)		10	µA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	-0.1	0.8	V	
V_{IH}	Input Hight Level	2.0	V_{CC}+1	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = -400 µA
I_{CC2}	V_{CC} Supply Current (Program & Varify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL} = \overline{PGM}$
V_{ID}	Ag for Int$_e$ligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ±5°C, V_{CC} = 6.0V ±0.25V, V_{PP} = 21V ±0.5V (see Note 1)

Symbol	Parameter	Limits Min.	Typ.	Max.	Unit	Test Conditions*
t_{AS}	Address Setup Time	2			µs	
t_{OES}	$\overline{OE}$ Setup Time	2			µs	
t_{DS}	Data Setup Time	2			µs	
t_{AH}	Address Hold Time	0			µs	
t_{DH}	Data Hold Time	2			µs	
t_{DF}	Output Enable to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			µs	
t_{VCS}	V_{CC} Setup Time	2			µs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			µs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

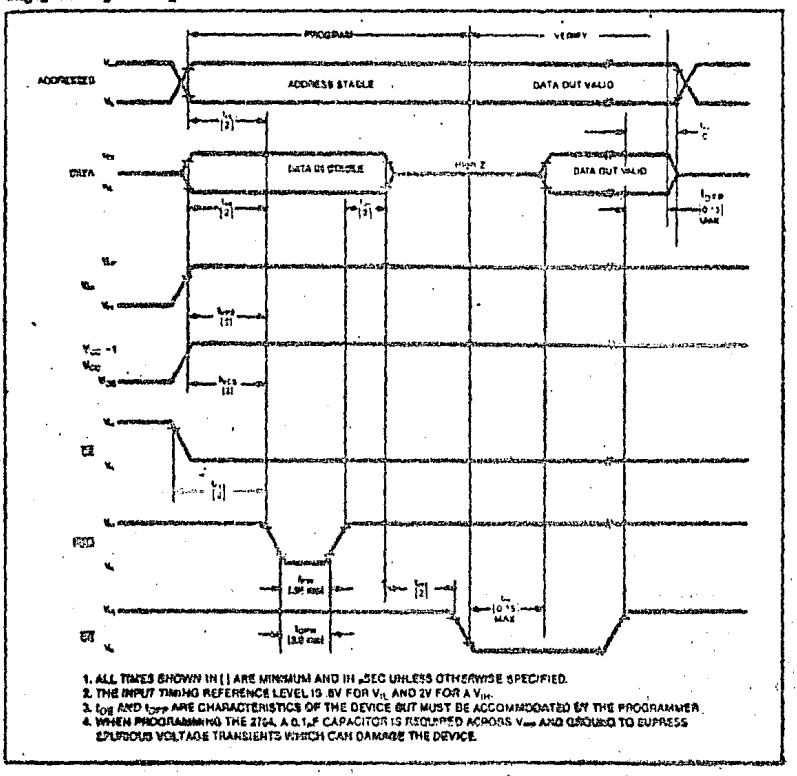
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ± 5%.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven — see timing diagram on page 9.

Intelligent Programming™ WAVEFORMS



1. ALL TIMES SHOWN IN ( ) ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE INPUT TIMING REFERENCE LEVEL IS .8V FOR V_{IL} AND 2V FOR A V_{IH}.
3. t_{DF} AND t_{OPW} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 2764, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

Int$_e$ligent Identifier™ Mode

The Int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 2764. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during Int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 2764, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

During 1982, Intel will begin manufacturing 2764s that will contain the Int$_e$ligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, pre-identifier mode 2764s will respond with the current data contained in locations 0 to 1 when subjected to the Int$_e$ligent Identifier operation.

Table 2. 2764 Int$_e$ligent Identifier™ Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	0	0	0	0	0	0	1	0	02

intel

27128
128K (16K x 8) UV ERASABLE PROM

- 250 ns Maximum Access Time . . . HMOS*-E Technology
- Compatible with High-Speed 8 MHz iAPX 186...Zero WAIT State
- Two-Line Control
- Pin Compatible to 2764 EPROM
- Industry Standard Pinout . . . JEDEC Approved
- ± 10% V_{CC} Tolerance Available
- Low Active Current . . . 100 mA Max.
- Intelligent Programming™ Algorithm

The Intel 27128 is a 5V only, 131,072-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The standard 27128 access time is 250 ns which is compatible with high-performance microprocessors such as Intel's 8 MHz iAPX 186. In these systems the 27128 allows the microprocessor to operate without the addition of WAIT states. The 27128 is also compatible with the 12 MHz 8051 family.

An important 27128 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

The 27128 has standby mode which reduces the power consumption without increasing access time. The maximum active current is 100 mA, while the maximum standby current is only 40 mA. The standby mode is selected by applying a TTL-high signal to the $\overline{CE}$ input.

±10% V_{CC} tolerance is available as an alternative to the standard ±5% V_{CC} tolerance for the 27128. This can allow the system designer more leeway with regard to his power supply requirements and other system parameters.

The 27128 is fabricated with HMOS*-E technology, Intel's high-speed N-channel MOS Silicon Gate Technology.

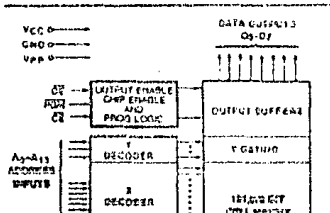
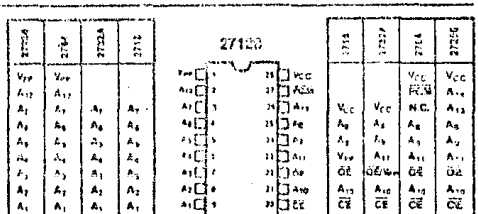


Figure 1. Block Diagram



NOTE: INTEL UNIVERSAL SITE COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 27128 PINS

Figure 2. Pin Configurations

MODE SELECTION

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	PGM (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (O_7-O_0 19-13)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	Dout
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{PP}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	Din
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	Dout
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
Intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	Din

NOTES
1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ± 0.5V

PIN NAMES

A_0-A_{13}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
PGM	PROGRAM
N.C.	NO CONNECT

*HMOS is a patented process of Intel Corporation

NOVEMBER 1982
ORDER NUMBER: 210334-003

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....−10°C to +80°C
Storage Temperature .....−65°C to +125°C
All Input or Output Voltages with Respect to Ground .....+7.0V to −0.6V
Voltage on Pin 24 with Respect to Ground .....+13.5V to −0.6V
V_{PP} Supply Voltage with Respect to Ground During Programming .....+22V to −0.6V

*NOTICE: Stresses above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	27128	27128-3	27128-4	27128-25	27128-30	27128-45
Operating Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%	5V ± 10%	5V ± 10%
V_{PP} Voltage[2]	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$	$V_{PP} = V_{CC}$

READ OPERATION

D.C. CHARACTERISTICS

Symbol	Parameter	Limits Min.	Limits Typ.[3]	Limits Max.	Units	Test Conditions
I_{LI}	Input Load Current			10	μA	$V_{IN} = 5.5V$
I_{LO}	Output Leakage Current			10	μA	$V_{OUT} = 5.5V$
I_{PP1}[2]	V_{PP} Current Read/Standby			5	mA	$V_{PP} = 5.5V$
I_{CC1}[2]	V_{CC} Current Standby		15	40	mA	$\overline{CE} = V_{IH}$
I_{CC2}[2]	V_{CC} Current Active		60	100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V_{IL}	Input Low Voltage	−.1		+.8	V	
V_{IH}	Input High Voltage	2.0		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -400$ μA

A.C. CHARACTERISTICS

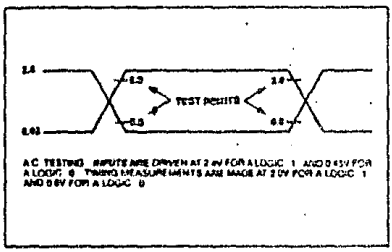
Symbol	Parameter	27128-25 & 27128 Limits Min.	27128-25 & 27128 Limits Max.	27128-30 & 27128-3 Limits Min.	27128-30 & 27128-3 Limits Max.	27128-45 & 27128-4 Limits Min.	27128-45 & 27128-4 Limits Max.	Units	Test Conditions
t_{ACC}	Address to Output Delay		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		250		300		450	ns	$\overline{OE} = V_{IL}$
t_{OE}	$\overline{OE}$ to Output Delay		100		120		150	ns	$\overline{CE} = V_{IL}$
t_{DF}[4]	$\overline{OE}$ High to Output Float	0	60	0	105	0	130	ns	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $T_A = 25°C$ and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram on page 3.

CAPACITANCE (T_A = 25°C, f = 1 MHz)

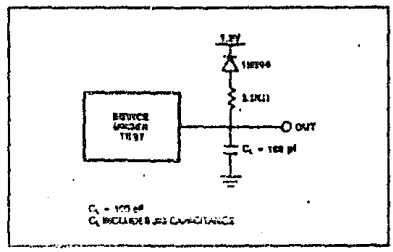
Symbol	Parameter	Typ.[1]	Max.	Unit	Conditions
C_{IN}[2]	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

A.C. TESTING INPUT/OUTPUT WAVEFORM



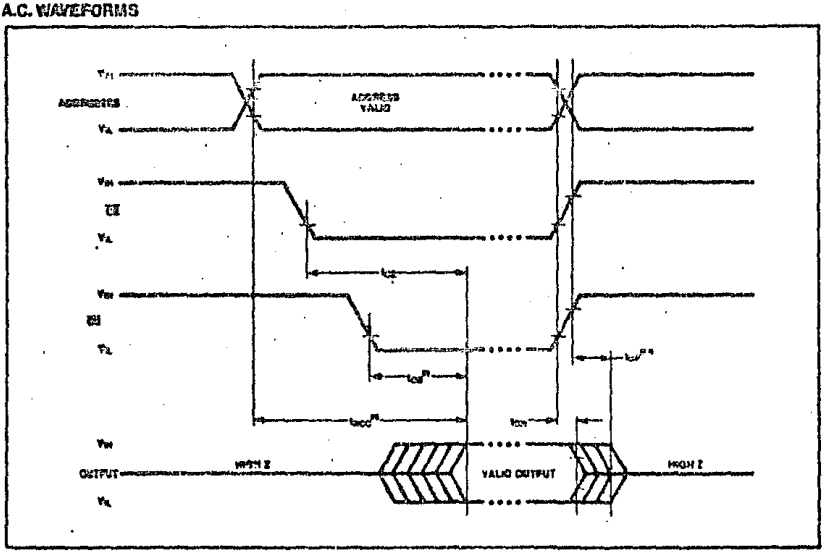
A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

A.C. WAVEFORMS



NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
4. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.

STANDARD PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

Symbol	Parameter	Limits			Test Conditions
		Min.	Max.	Unit	
I_{LI}	Input Current (All Inputs)		10	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{OL}	Output Low Voltage During Verify		0.45	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	2.4		V	$I_{OH} = -400$ μA
V_{IL}	Input Low Level (All Inputs)	-0.1	0.8	V	
V_{IH}	Input High Level	2.0	$V_{CC} + 1$	V	
I_{CC1}	V_{CC} Supply Current (Program Inhibit)		40	mA	$\overline{CE} = V_{IH}$
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL}$, $\overline{PGM}$
I_{PP3}	V_{PP} Supply Current (Verify)		5	mA	$\overline{CE} = V_{IL}$, $\overline{PGM} = V_{IH}$
I_{PP4}	V_{PP} Supply Current (Program Inhibit)		5	mA	$\overline{CE} = V_{IH}$
V_{ID}	A9 Intelligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$ (see Note 1)

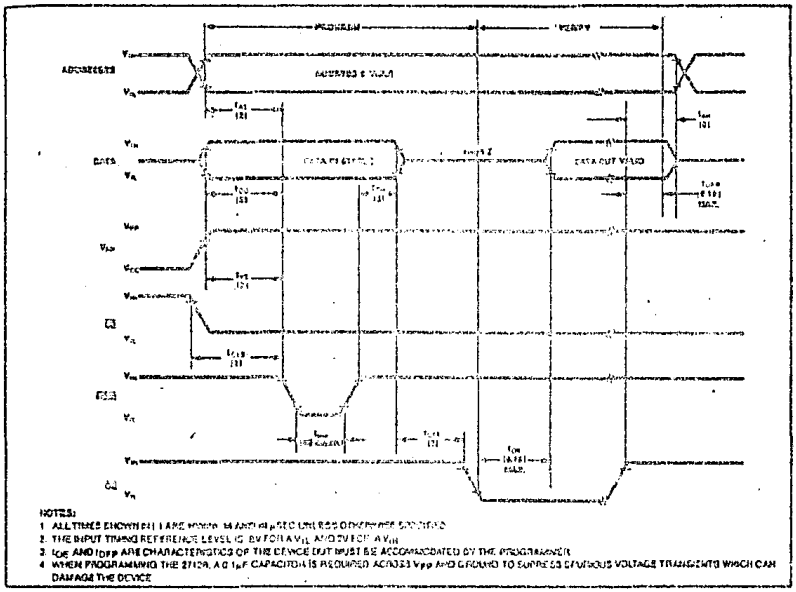
Symbol	Parameter	Limits				Test Conditions*
		Min.	Typ.	Max.	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}[2]	Output Enable to Output Float Delay	0		130	ns	
t_{VS}	V_{PP} Setup Time	2			μs	
t_{PW}	$\overline{PGM}$ Pulse Width During Programming	45	50	55	ms	
t_{CES}	$\overline{CE}$ Setup Time	2			μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram on page 5.

STANDARD PROGRAMMING WAVEFORMS



ERASURE CHARACTERISTICS

The erasure characteristics of the 27128 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 27128 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 27128 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 27128 window to prevent unintentional erasure.

The recommended erasure procedure for the 27128 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 µW/cm² power rating. The 27128 should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose a 27128 can be exposed to without damage is 7258 Wsec/cm² (1 week @ 12000 µW/cm²). Exposure of the 27128 to high intensity UV light for long periods may cause permanent damage.

DEVICE OPERATION

The eight modes of operation of the 27128 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A9 for intelligent identifier mode.

Table 1. Mode Selection

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	V_{CC}	High Z
Program	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IL}	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	V_{PP}	V_{CC}	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_{IH}	V_H	V_{CC}	V_{CC}	Code
Intelligent Programming	V_{IL}	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{IN}

NOTES:
1. X can be V_{IH} or V_{IL}
2. V_H = 12.0V ± 0.5V

READ MODE

The 27128 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 27128 has standby mode which reduces the maximum active current from 100 mA to 40 mA. The 27128 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 20) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

System Considerations

The power switching characteristics of HMOS-E EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, as detailed in Intel's Application Note, AP-72, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

PROGRAMMING MODES

Caution: Exceeding 22V on pin 1 (V_{PP}) will permanently damage the 27128.

Initially, and after each erasure, all bits of the 27128 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 27128 is in the programming mode when V_{PP} input is at 21V and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Standard Programming

For programming, $\overline{CE}$ should be kept TTL-low at all times while V_{PP} is kept at 21V. When the address and data are stable, a 50 msec. active-low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time —either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec.

Programming of multiple 27128s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 27128s may be connected together when they are programmed with the same data. A low-level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled 27128s.

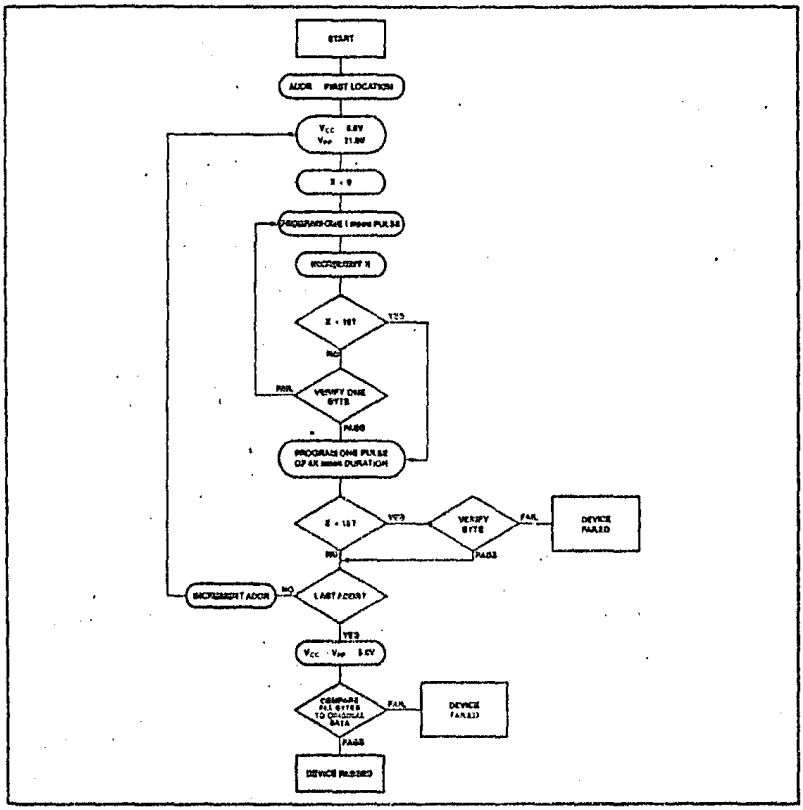


Figure 3. 27128 Int$_e$ligent Programming™ Flowchart

Program Inhibit

Programming of multiple 27128s in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other 27128s from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 27128s may be common. A TTL low-level pulse applied to the $\overline{CE}$ and $\overline{PGM}$ inputs with V_{PP} at 21V will program the selected 27128.

Verify

A verify should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{CE}$ and $\overline{OE}$ at V_{IL}, $\overline{PGM}$ at V_{IH} and V_{PP} at 21V.

Int$_e$ligent Programming™ Algorithm

The 27128 int$_e$ligent Programming Algorithm allows Intel 27128s to be programmed in a significantly faster time than the standard 50 msec per byte programming routine. Typical programming times for 27128s are on the order of two minutes, which is a six-fold reduction in programming time from the standard method. This fast algorithm results in the same reliability characteristics as the standard 50 msec algorithm. A flowchart of the 27128 int$_e$ligent

Programming Algorithm is shown in Figure 3. This is compatible with the 2764 intelligent Programming Algorithm.

With the standard programming method, data is programmed into a selected 27128 location by a single 50 msec, active-low, TTL pulse applied to the $\overline{PGM}$ pin. The intelligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{PGM}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 4X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular 27128 location, before a correct verify occurs. Up to 15 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 21.0V. When the intelligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

intelligent Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ± 5°C, V_{CC} = 6.0V ± 0.25V, V_{PP} = 21V ± 0.5V

Symbol	Parameter	Limits Min.	Max.	Unit	Test Conditions (see Note 1)
I_{LI}	Input Current (All Inputs)		10	µA	V_{IN} = V_{IL} or V_{IH}
V_{IL}	Input Low Level (All Inputs)	-0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC}+1	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	2.4		V	I_{OH} = -400µA
I_{CC2}	V_{CC} Supply Current (Program & Verify)		100	mA	
I_{PP2}	V_{PP} Supply Current (Program)		30	mA	$\overline{CE}$ = V_{IL}, $\overline{PGM}$ = V_{IL}
V_{ID}	A9 Intelligent Identifier Voltage	11.5	12.5	V	

A.C. PROGRAMMING CHARACTERISTICS: T_A = 25 ± 5°C, V_{CC} = 6.0V ± 0.25V, V_{PP} = 21V ± 0.5V

Symbol	Parameter	Limits Min.	Typ.	Max.	Unit	Test Conditions* (see Note 1)
t_{AS}	Address Setup Time	2			µs	
t_{OES}	$\overline{OE}$ Setup Time	2			µs	
t_{DS}	Data Setup Time	2			µs	
t_{AH}	Address Hold Time	0			µs	
t_{DH}	Data Hold Time	2			µs	
t_{DFP}	Output Enable to Output Float Delay	0		130	ns	
t_{VPS}	V_{PP} Setup Time	2			µs	
t_{VCS}	V_{CC} Setup Time	2			µs	
t_{PW}	$\overline{PGM}$ Initial Program Pulse Width	0.95	1.0	1.05	ms	(see Note 3)
t_{OPW}	$\overline{PGM}$ Overprogram Pulse Width	3.8		63	ms	(see Note 2)
t_{CES}	$\overline{CE}$ Setup Time	2			µs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

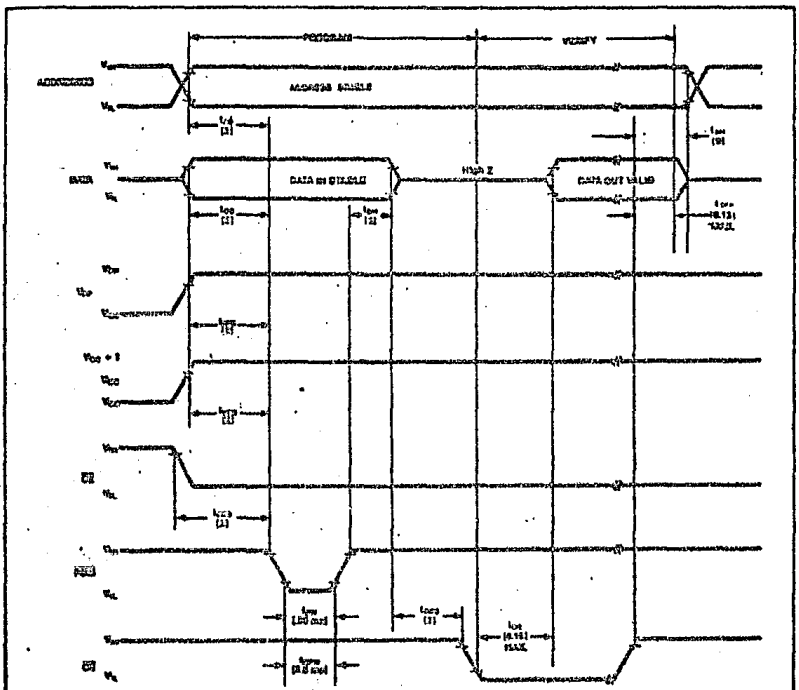
***A.C. CONDITIONS OF TEST**

Input Rise and Fall Times (10% to 90%) ..... 20 ns
Input Pulse Levels ..... 0.45V to 2.4V
Input Timing Reference Level ..... 0.8V and 2.0V
Output Timing Reference Level ..... 0.8V and 2.0V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. The length of the overprogram pulse will vary from 3.8 msec to 63 msec as a function of the iteration counter value X.
3. Initial Program Pulse width tolerance is 1 msec ± 5%.
4. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram on page 9.

Intelligent Programming™ WAVEFORMS



NOTES:
1. ALL TIMES SHOWN IN ( ) ARE MINIMUM AND IN μSEC UNLESS OTHERWISE SPECIFIED.
2. THE OUTPUT TIMING REFERENCE LEVEL IS .8V FOR A V_{IL} AND 2V FOR A V_{IH}.
3. t_{DS} AND t_{OPW} ARE CHARACTERISTICS OF THE DEVICE BUT MUST BE ACCOMMODATED BY THE PROGRAMMER.
4. WHEN PROGRAMMING THE 27128, A 0.1μF CAPACITOR IS REQUIRED ACROSS V_{PP} AND GROUND TO SUPPRESS SPURIOUS VOLTAGE TRANSIENTS WHICH CAN DAMAGE THE DEVICE.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ± 5°C ambient temperature range.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 (pin 24) of the 27128. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 (pin 10) from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. For the Intel 27128, these two identifier bytes are given in Table 2. All identifiers for manufacturer and device codes will possess odd parity, with the MSB (O_7) defined as the parity bit.

Intel will begin manufacturing 27128s during 1982 that will contain the int$_e$ligent Identifier feature. Earlier generation devices will not contain identifier information, and if erased, will respond with a "one" (V_{OH}) on each data line when operated in this mode. Programmed, pre-Identifier mode 27128s will respond with the current data contained in locations 0 and 1 when subjected to the int$_e$ligent Identifier operation.

Table 2. 27128 int$_e$ligent Identifier Bytes

Identifier \ Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	0	0	1	89
Device Code	V_{IH}	1	0	0	0	0	0	1	1	83

27256
256K (32K x 8) UV ERASABLE PROM

- Software Carrier Capability
- 250 ns Maximum Access Time
- Two-Line Control
- Intelligent Identifier™ Mode
- Industry Standard Pinout . . . JEDEC Approved
- Low Power
  - —100 mA max. Active
  - — 40 mA max. Standby
- Intelligent Programming™ Algorithm

The Intel 27256 is a 5V only, 262,144-bit ultraviolet Erasable and Electrically Programmable Read Only Memory (EPROM). Organized as 32K words by 8 bits, individual bytes are accessed in under 250ns. This is compatible with high performance microprocessors, such as the Intel 5MHz iAPX 186, allowing full speed operation without the addition of performance-degrading WAIT states.

The 27256 enables implementation of new, advanced systems with firmware intensive architectures. The combination of the 27256's high density, cost effective EPROM storage, and new advanced microprocessors having megabit addressing capability provides designers with opportunities to engineer user-friendly, high reliability, high-performance systems.

The 27256's large storage capability of 32K bytes enables it to function as a high density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27256 EPROM directly on a system's memory bus. This would permit immediate microprocessor access and execution of software and eliminate the need for time consuming disk accesses and downloads.

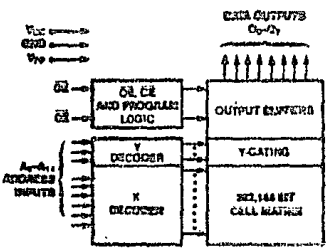
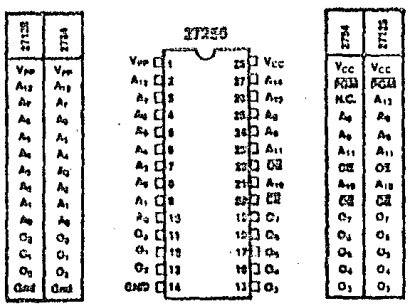


Figure 1. Block Diagram

MODE / PINS	CE (20)	OE (22)	A_9 (24)	V_{PP} (1)	V_{CC} (28)	OUTPUTS (11-13,15-19)
Read	V_{IL}	V_{IL}	X	V_{CC}	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	V_{CC}	V_{CC}	High Z
Standby	V_{IH}	X	X	V_{CC}	V_{CC}	High Z
Intelligent Programming	V_{IL}	V_{IH}	X	V_{PP}	V_{CC}	D_{IN}
Verify	V_{IH}	V_{IL}	X	V_{PP}	V_{CC}	D_{OUT}
Program Inhibit	V_{IH}	V_{IH}	X	V_{PP}	V_{CC}	High Z
Intelligent Identifier	V_{IL}	V_{IL}	V_H	V_{CC}	V_{CC}	Code

NOTES
1. X can be V_{IH} or V_{IL}.
2. V_H = 12.0V ± 0.5V.



NOTE: INTEL "UNIVERSAL GATE" COMPATIBLE EPROM PIN CONFIGURATIONS ARE SHOWN IN THE BLOCKS ADJACENT TO THE 27256 PINS.

Figure 2. Pin Configurations

PIN NAMES	
A_0–A_{14}	ADDRESSES
CE	CHIP ENABLE
OE	OUTPUT ENABLE
O_0–O_7	OUTPUTS
PGM	PROGRAM
N.C.	NO CONNECT

DECEMBER 1982
ORDER NUMBER: 210937-001

intel

UV ERASABLE PROM FAMILY

EXPRESS

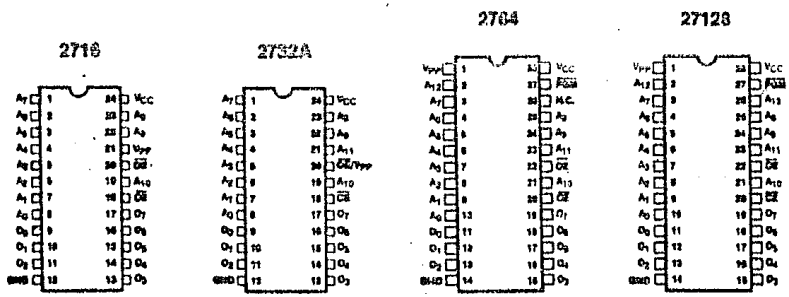
- 0–70°C Temperature Range Standard
- Extended Temperature Range −40°C – +85°C Available
- Two Line Control
- 168±8 Hour Burn-In Available
- Industry Standard Pinout . . . JEDEC Approved
- Inspected To 0.1% AQL

The Intel EXPRESS EPROM family is a series of ultraviolet erasable and electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. Intel's JEDEC approved 28 pin Universal Memory Socket provides the industry standard upgrade path to higher density EPROMs.

EXPRESS EPROM products are available with 168±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration (equivalent to MIL-STD-883B). This process exceeds or meets most industry specifications of burn-in.

The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to 85°C) EXPRESS products are available. EXPRESS products plus military grade EPROMs (−55°C to 125°C) provide the most complete choice of standard and extended temperature range EPROMs available.

Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.



PIN CONFIGURATION

EXPRESS
EPROM Product Family

Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-in 125°C (hr)
QD2716-1	2048x8	350	5V ± 10%	0 to 70	168±8
QD2716-2	2048x8	390	5V ± 5%	0 to 70	168±8
QD2716	2048x8	450	5V ± 5%	0 to 70	168±8
LD2716	2048x8	450	5V ± 5%	-40 to 85	168±8
TD2716	2048x8	450	5V ± 5%	-40 to 85	NONE
QD2732A-2	4096x8	200	5V ± 5%	0 to 70	168±8
QD2732A	4096x8	250	5V ± 5%	0 to 70	168±8
QD2732A-3	4096x8	300	5V ± 5%	0 to 70	168±8
QD2732A-4	4096x8	450	5V ± 5%	0 to 70	168±8
QD2732A-20	4096x8	200	5V ± 10%	0 to 70	168±8
QD2732A-25	4096x8	250	5V ± 10%	0 to 70	168±8
QD2732A-30	4096x8	300	5V ± 10%	0 to 70	168±8
LD2732A	4096x8	250	5V ± 5%	-40 to 85	168±8
LD2732A-4	4096x8	450	5V ± 5%	-40 to 85	168±8
LD2732A-25	4096x8	250	5V ± 10%	-40 to 85	168±8
LD2732A-45	4096x8	450	5V ± 10%	-40 to 85	168±8
TD2732A	4096x8	250	5V ± 5%	-40 to 85	NONE
TD2732A-4	4096x8	450	5V ± 5%	-40 to 85	NONE
TD2732A-25	4096x8	250	5V ± 10%	-40 to 85	NONE
TD2732A-45	4096x8	450	5V ± 10%	-40 to 85	NONE
QD2764-2	8192x8	200	5V ± 5%	0 to 70	168±8
QD2764	8192x8	250	5V ± 5%	0 to 70	168±8
QD2764-3	8192x8	300	5V ± 5%	0 to 70	168±8
QD2764-4	8192x8	450	5V ± 5%	0 to 70	168±8
QD2764-25	8192x8	250	5V ± 10%	0 to 70	168±8
QD2764-30	8192x8	300	5V ± 10%	0 to 70	168±8
QD2764-45	8192x8	450	5V ± 10%	0 to 70	168±8
LD2764	8192x8	250	5V ± 5%	-40 to 85	168±8
LD2764-4	8192x8	450	5V ± 5%	-40 to 85	168±8
LD2764-25	8192x8	250	5V ± 10%	-40 to 85	168±8
LD2764-45	8192x8	450	5V ± 10%	-40 to 85	168±8
TD2764	8192x8	250	5V ± 5%	-40 to 85	NONE
TD2764-4	8192x8	450	5V ± 5%	-40 to 85	NONE
TD2764-25	8192x8	250	5V ± 10%	-40 to 85	NONE
TD2764-45	8192x8	450	5V ± 10%	-40 to 85	NONE
QD27128	16384x8	250	5V ± 5%	0 to 70	168±8
QD27128-3	16384x8	300	5V ± 5%	0 to 70	168±8

EXPRESS
EPROM Product Family
(Cont.)

Type	Organization	Maximum Access (ns)	Power Supply	Operating Temperature (°C)	Burn-In 125°C (hr)
QD27128-4	16384x8	450	5V ± 5%	0 to 70	168 ± 8
QD27128-25 QD27128-45	16384x8 16384x8	250 450	5V ± 10% 5V ± 10%	0 to 70 0 to 70	168 ± 8 168 ± 8
LD27128 LD27128-4 LD27128-45	16384x8 16384x8 16384x8	250 450 450	5V ± 5% 5V ± 5% 5V ± 10%	−40 to 85 −40 to 85 −40 to 85	168 ± 8 168 ± 8 168 ± 8
TD27128 TD27128-4 TD27128-45	16384x8 16384x8 16384x8	250 450 450	5V ± 5% 5V ± 5% 5V ± 10%	−40 to 85 −40 to 85 −40 to 85	NONE NONE NONE

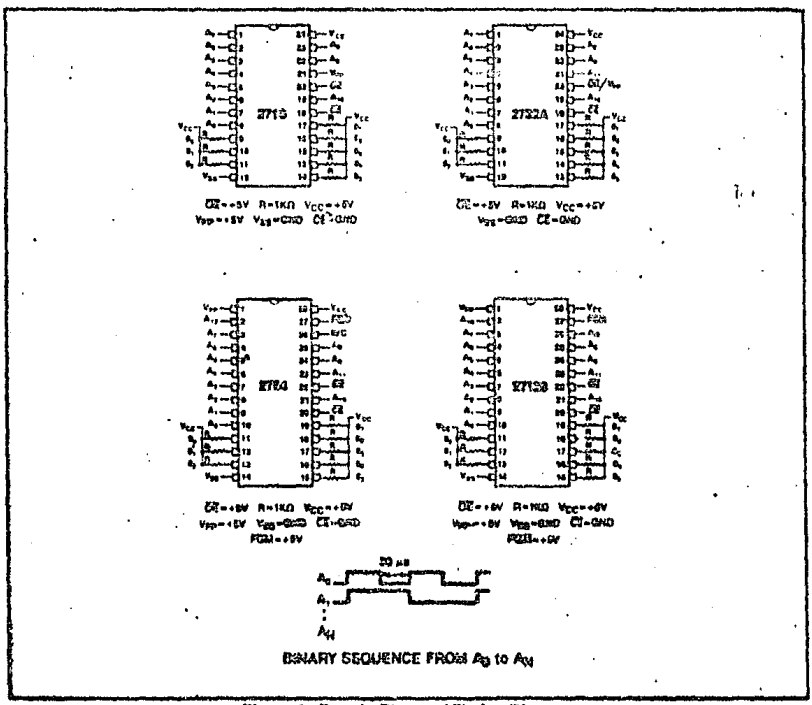


Figure 1. Burn-In Bias and Timing Diagrams

READ OPERATION

D.C. AND A.C. CHARACTERISTICS

Electrical Parameters of EXPRESS EPROM products are identical to standard data sheet parameters except for:

Symbol	Parameter	Limits								Test Conditions
		TD2716 LD2716		TD2732A LD2732A		TD27C64 LD27C64		TD27C128 LD27C128		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
T_{OE}	Output Enable to Output Delay (ns)		150							$\overline{CE} = V_{IL}$
t_{DF}	Output Enable to Output Float (ns)	0	130							$\overline{CE} = V_{IL}$
I_{CC1}	V_{CC} Standby Current (mA)				45		50		50	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC2}	V_{CC} Active Current (mA)				150		125		125	$\overline{OE} = \overline{CE} = V_{IL}$

NEC

NEC Electronics U.S.A. Inc.

Microcomputer Division

μPD4016-1
μPD4016-2
μPD4016-3
2048 × 8-BIT STATIC RAM

Description

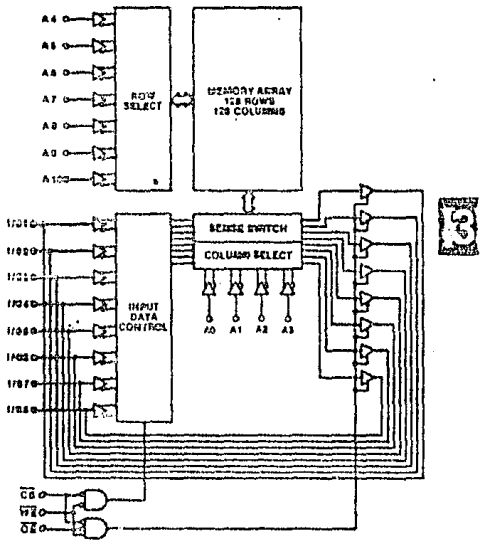
The μPD4016 is a 16384-bit static Random Access Memory device organized as 2048 words by 8 bits. Using a scaled NMOS technology, its design provides the ease-of-use features associated with non-clocked static memories. The μPD4016 has a three-state output and offers a stand-by mode with an attendant 75% savings in power consumption. It features equal access and cycle times and provides an output enable function that eliminates the need for external bus buffers. The μPD4016 is packaged in a standard 24-pin dual-in-line package and is plug-compatible with 16K EPROMS.

Features

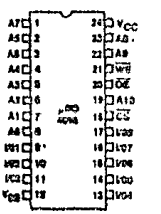
- Scaled NMOS technology
- Completely static memory: no clock, no refresh
- Equal access and cycle times
- Single +5V supply
- Automatic power-down
- All inputs and outputs directly TTL-compatible
- Common I/O capability
- $\overline{OE}$ eliminates need for external bus buffers
- Three-state outputs
- Plug-compatible with 16K 5V EPROMS
- Low power dissipation in standby mode
- Available in a standard 24-pin dual-in-line package

	Access Time	R/W Cycle
μPD4016-1	350 ns	350 ns
μPD4016-2	200 ns	200 ns
μPD4016-3	150 ns	150 ns

Block Diagram



Pin Configuration



Pin Names	
A0 - A10	Address Inputs
$\overline{WE}$	Write Enable
$\overline{CS}$	Chip Select
$\overline{OE}$	Output Enable
I/O1 - I/O8	Data Input/Output
V_{CC}	Power (+5V)
V_{SS}	Ground

Truth Table					
$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	MODE	I/O	POWER
H	X	X	Not Selected	High-Z	Standby
L	L	H	Read	Dout	Active
L	H	L	Write	Din	Active
L	L	L	Write	Din	Active

Absolute Maximum Ratings*

T_a = 25°C

Temperature Under Bias	−10°C to 85°C
Storage Temperature	−65°C to 150°C
Voltage on any pin with respect to Ground	−0.5V to 7V
D.C. Output Current	20mA
Power Dissipation	1W

*COMMENT: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Capacitance

T_a = 25°C, f = 1 MHz

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Input Capacitance	C_{IN}			5	pF	V_{IN} = 0V
I/O Capacitance	$C_{I/O}$			7	pF	$V_{I/O}$ = 0V

This parameter is sampled and not 100% tested.

REV/1
03

DC Characteristics

T_a = 0°C to 70°C, V_{CC} = 5V ± 10%

Parameter	Symbol	LIMITS Min	Typ	Max	Unit	Test Conditions
Input Leakage Current	I_{LI}			10	μA	VCC = Max VIN = GND to VCC
Output Leakage Current	I_{LO}			10	μA	VCC = Max, $\overline{CS}$ = VIH VOUT = GND to VCC
Operating Current	I_{CC}			80	mA	VCC = Max, $\overline{CE}$ = VIL Outputs Open
Standby Current	I_{SB}			15	mA	VCC = Min to Max $\overline{CS}$ = VIH
Input Low Voltage	V_{IL}	−1.5		0.8	V	
Input High Voltage	V_{IH}	2.0		6.0	V	
Output Low Voltage	V_{OL}			0.4	V	IOL = 4mA
Output High Voltage	V_{OH}	2.4			V	IOH = −1mA
Output Short Circuit Current	I_{OS}	−10		−100	mA	VOUT = GND to VCC

AC Test Conditions

Input Pulse Levels ..... 0.8V to 2.2V
Input Rise and Fall Times 10nsec
Input Timing Reference Levels 1.5V
Output Timing Reference Levels 1.5V

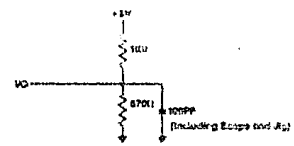


Figure 1 – Output Load

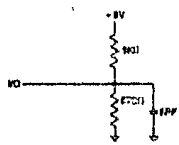


Figure 2 - Transition Load

AC Characteristics

Read Cycle

T_a = 0°C to 70°C, V_{CC} 5V ± 10%

Parameter	Symbol	LIMITS μPD4016-3 Min	Max	μPD4016-2 Min	Max	μPD4016-1 Min	Max	Unit	Notes
Read Cycle Time	t_{RC}	150		200		250		nsec	1
Address Access Time	t_{AA}		150		200		250	nsec	
Chip Select Access Time	t_{ACS}		150		200		250	nsec	2
Output Hold from Address Change	t_{OH}	10		10		10		nsec	
Chip Selection to Output in Low Z	t_{LZ}	10		10		10		nsec	3,5
Chip Deselection to Output in High Z	t_{HZ}		60		60		60	nsec	3,6
Output Enable to Output Valid	t_{OE}		70		90		110	nsec	
Output Enable to Output in Low Z	t_{OLZ}	10		10		10		nsec	3,5
Output Disable to Output in High Z	t_{OHZ}		50		60		80	nsec	3,6
Chip Selection to Power-up Time	t_{PU}	0		0		0		nsec	4
Chip Deselection to Power-down Time	t_{PD}		70		90		110	nsec	4

Write Cycle

T_a = 0°C to 70°C, V_{CC} 5V ± 10%

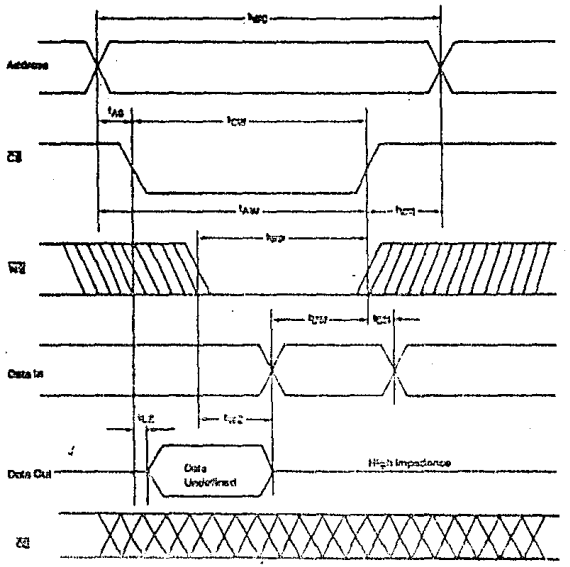
Parameter	Symbol	LIMITS μPD4016-3 Min	Max	μPD4016-2 Min	Max	μPD4016-1 Min	Max	Unit	Notes
Write Cycle Time	t_{WC}	150		200		250		nsec	
Chip Selection to End of Write	t_{CW}	120		160		200		nsec	
Address Valid to End of Write	t_{AW}	90		120		150		nsec	
Address Setup Time	t_{AS}	0		0		0		nsec	
Write Pulse Width	t_{WP}	90		100		130		nsec	5
Write Recovery Time	t_{WR}	10		10		10		nsec	
Data Valid to End of Write	t_{DW}	90		60		90		nsec	
Data Hold Time	t_{DH}	0		0		0		nsec	
Write Enabled to Output in High-Z	t_{WZ}		50		60		80	nsec	6,7
Output Active from End of Write	t_{OW}	10		10		10		nsec	6,7

Notes:

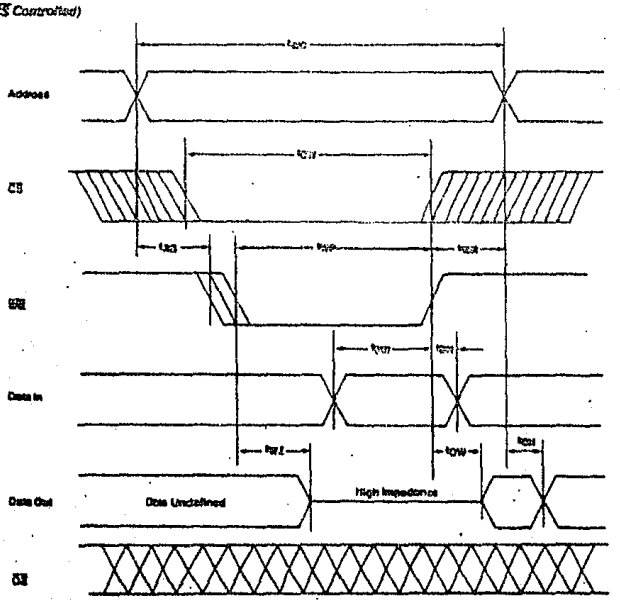
1. All Read Cycle timings are referenced from the last valid address to the first transitioning address.
2. Address valid prior to or coincident with $\overline{CS}$ transition low.
3. Transition is measured ± 200mV from steady state voltage with specified load of Figure 1.
4. This parameter is sampled and not 100% tested.
5. If $\overline{CS}$ and $\overline{OE}$ are both low before write enabled, tWP = tWZ + DW
6. Transition is measured ± 200mV from steady state voltage with specified loading in Figure 2.
7. This parameter is sampled and not 100% tested.

Timing Waveforms

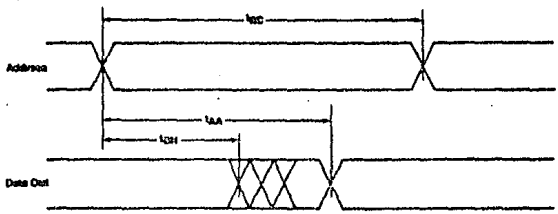
Write Cycle No. 1 ($\overline{WE}$ Controlled)



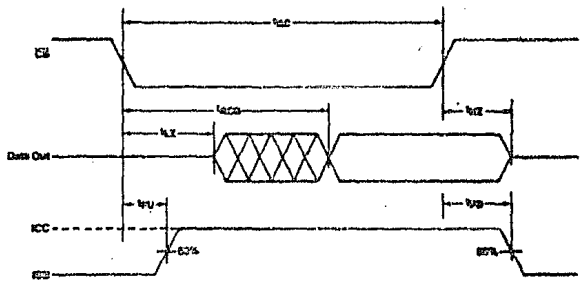
Write Cycle No. 2 ($\overline{CS}$ Controlled)



Read Cycle No. 1 ① ② ④



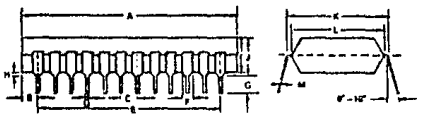
Read Cycle No. 2 ① ③ ④



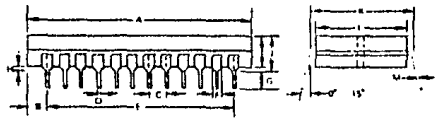
Read Cycle No. 3 ② ③

Notes: ① Address valid prior to or coincident with $\overline{CS}$ transition low.
② $\overline{WE}$ is high for Read Cycles.
③ Device is continuously selected, $\overline{CS}$ = VIL.
④ $\overline{OE}$ = VIL

Package Outlines
μPD4016C



μPD4016D

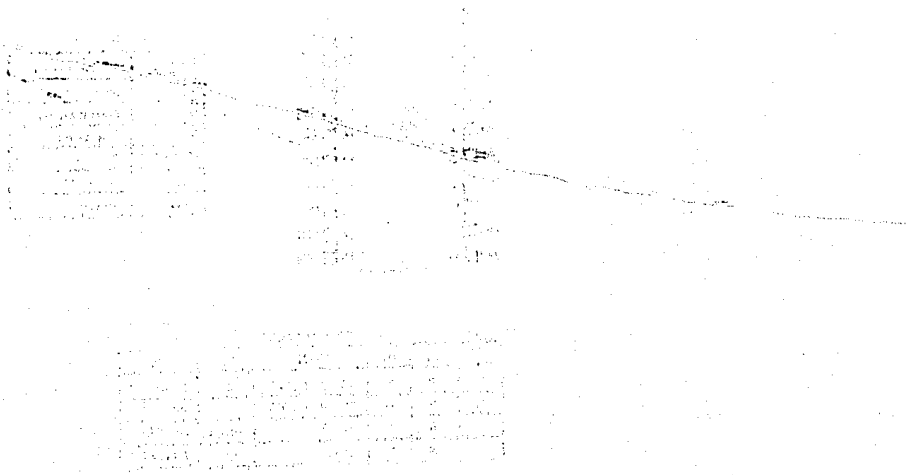


Plastic

Item	Millimeters	Inches
A	33 Max	1.3 Max
B	2.53	0.1
C	2.54	0.1
D	0.5 ± 0.1	0.02 ± 0.004
E	37.04	1.1
F	1.0	0.039
G	2.54 Min	0.1 Min
H	0.5 Min	0.02 Min
I	5.22 Max	0.205 Max
J	5.72 Max	0.225 Max
K	15.24	0.6
L	13.2	0.52
M	0.25 +0.10 −0.05	0.01 +0.004 −0.0019

Ceramic

Item	Millimeters	Inches
A	33.3 Max	1.32 Max
B	2.78	0.11
C	2.54	0.1
D	0.46	0.018
E	37.04	1.1
F	1.9	0.059
G	2.54 Min	0.1 Min
H	0.5 Min	0.019 Min
I	6.53 Max	0.181 Max
J	7.62 Max	0.2 Max
K	15.24	0.6
L	13.6	0.53
M	0.25 +0.12 −0.05	0.01 +0.004 −0.002



2048 × 8-BIT STATIC CMOS RAM

DESCRIPTION

The μPD446 is a high speed, low power, 2048 word by 8 bit static CMOS RAM fabricated using an advanced silicon gate CMOS technology. A unique circuitry technique makes the μPD446 a very low operating power device which requires no clock or refreshing to operate. Minimum standby power current is drawn by this device when $\overline{CS}$ equals V_{CC} independently of the other input levels.

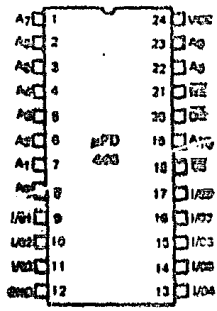
Data retention is guaranteed at a power supply voltage as low as 2V.

The μPD446 is packaged in a standard 24-pin dual-in-line package and is plug-in compatible with 16K EPROMs.

FEATURES

- Single +5V Supply
- Fully Static Operation – No Clock or Refreshing required
- TTL Compatible – All Inputs and Outputs
- Common I/O Using Three-State Output
- $\overline{OE}$ Eliminates Need for External Bus Buffers
- Max Access/Min Cycle Times Down to 150 ns
- Low power Dissipation, 18 mA Max Active/10 μA Max Standby/ 10 μA Max Data Retention
- Data Retention Voltage – 2V Min
- Standard 24-Pin Plastic and Ceramic Packages
- Plug-In Compatible with 16K EPROMs
- Operating Temperature Range – 0°C to 70°C

PIN CONFIGURATION



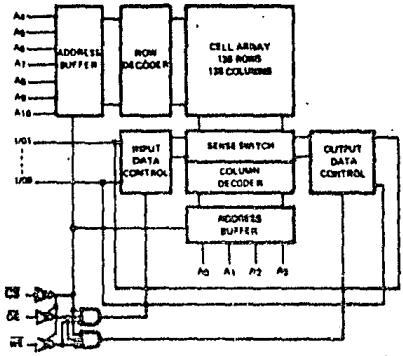
PIN NAMES

A0-A10	Address Inputs
$\overline{WE}$	Write Enable
$\overline{OE}$	Output Enable
$\overline{CS}$	Chip Select
I/O1-I/O8	Data Input/Output
V_{CC}	Power (+5V)
GND	Ground

TRUTH TABLE

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	MODE	I/O	ICC
H	X	X	NOT SELECTED	HZ	STANDBY
L	H	H	NOT SELECTED	HZ	ACTIVE
L	L	H	READ	D_{OUT}	ACTIVE
L	X	L	WRITE	D_{IN}	ACTIVE

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Supply Voltage 7.0V
Input or Output Voltage Supplied ..... −0.3 to V_{CC} + 0.3V
Storage Temperature Range ..... −65°C to 125°C
Operating Temperature Range ..... −40°C to +85°C

T_a = 25°C

*COMMENT: Stress above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

T_a = −40°C to +85°C, V_{CC} = 5.0V ± 10%

PARAMETER	SYMBOL	LIMITS MIN	TYP	MAX	UNIT	CONDITIONS
Input High Voltage	V_{IH}	2.2		V_{CC} +0.3	V	
Input Low Voltage	V_{IL}	−0.3		0.8	V	
Input Leakage Current	I_{LI}	−1.0		1.0	μA	V_{IN} = 0 ~ V_{CC}
I/O Leakage Current	I_{LO}	−1.0		1.0	μA	V_{CS} = V_{IH} $V_{I/O}$ = 0 ~ V_{CC}
Operating Supply Current	I_{CCA_1}		①	①	mA	V_{CS} = V_{IL} $I_{I/O}$ = 0 MIN TCYCLE
	I_{CCA_2}		5	10	mA	V_{CS} = V_{IL} $I_{I/O}$ = 0 DC CURRENT
	I_{CCA_3}		20	100	μA	V_{CS} = 0.2V $I_{I/O}$ = 0 V_{IN} = V_{CC} −0.2 OR 0.2V
Standby Current	I_{CC_B}			10	μA	V_{CS} = V_{CC} −0.2 V_{IN} = 0 ~ V_{CC}
Output High Voltage	V_{OH}	2.4			V	I_{OH} = −1.0 mA
Output Low Voltage	V_{OL}			0.4	V	I_{OL} = 2.0 mA

NOTE: ① μPD446: 12 mA TYP, 18 mA MAX
μPD446-1: 18 mA TYP, 25 mA MAX
μPD446-2: 20 mA TYP, 30 mA MAX
μPD446-3: 25 mA TYP, 35 mA MAX

CAPACITANCE

T_a = 25°C, f = 1.0 MHz

PARAMETER	SYMBOL	LIMITS MIN	LIMITS MAX	UNIT	TEST CONDITIONS
Input Capacitance	$C_{I/1}$		5	pF	V_{IN} = 0V
Input/Output Capacitance	$C_{I/O}$		8	pF	$V_{I/O}$ = 0V

AC CHARACTERISTICS

READ CYCLE

V_{CC} = 5.0V ± 10%, T_a = −40°C to +85°C

PARAMETER	SYMBOL	LIMITS μPD446-3 MIN	μPD446-3 MAX	μPD446-2 MIN	μPD446-2 MAX	μPD446-1 MIN	μPD446-1 MAX	μPD446 MIN	μPD446 MAX	UNIT
Read Cycle Time	t_{RC}	150		200		250		450		ns
Address Access Time	t_{AA}		150		200		250		450	ns
Chip Select Access Time	t_{ACS}		150		200		250		450	ns
Output Enable to Output Valid	t_{OE}		75		100		120		150	ns
Output Hold from Address Change	t_{OH}	15		15		15		15		ns
Chip Enable to Output in LZ	t_{CLZ}	10		10		10		10		ns
Output Enable to Output in LZ	t_{OLZ}	5		5		5		5		ns
Chip Disable to Output in HZ	t_{CHZ}		50		60		80		100	ns
Output Disable to Output in HZ	t_{OHZ}		50		60		80		100	ns

WRITE CYCLE

V_{CC} = 5.0V ± 10%, T_a = −40°C to +85°C

PARAMETER	SYMBOL	LIMITS μPD446-3 MIN	μPD446-3 MAX	μPD446-2 MIN	μPD446-2 MAX	μPD446-1 MIN	μPD446-1 MAX	μPD446 MIN	μPD446 MAX	UNIT
Write Cycle Time	t_{WC}	150		200		250		450		ns
Chip Enable to End of Write	t_{CW}	120		150		180		210		ns
Address Valid to End of Write	t_{AW}	120		150		180		210		ns
Address Set-up Time	t_{AS}	0		0		0		0		ns
Write Pulse Width	t_{WP}	90		120		150		180		ns
Write Recovery Time	t_{WR}	0		0		0		0		ns
Data Valid to End of Write	t_{DW}	60		80		80		100		ns
Data Hold Time	t_{DH}	0		0		0		0		ns
Write Enable to Output in HZ	t_{WHZ}		50		60		80		150	ns
Output Active from End of Write	t_{OW}	10		10		10		10		ns

LOW V_{CC} DATA RETENTION

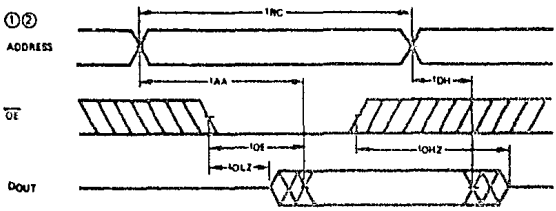
T_a = −40°C to +85°C

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
V_{CC} for Data Retention	V_{CCDR}	V_{IN} = 0 ~ V_{CC}, $\overline{CS}$ = V_{CC}	2.0			V
Data Retention Current	I_{CCDR}	V_{CC} = 3.0V, V_{IN} = 0 ~ V_{CC}, $\overline{CS}$ = V_{CC}		0.01	10	μA
Chip Deselection to Data Retention Time	t_{CDR}		0			ns
Operation Recovery Time	t_R		t_{RC}			ns

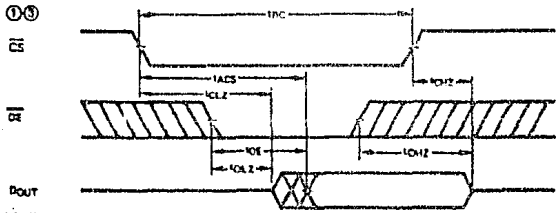
61

TIMING WAVEFORMS

READ CYCLE (1)



READ CYCLE (2)



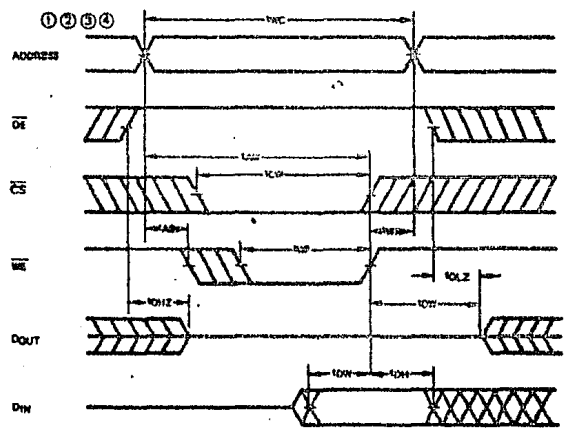
NOTES:

① WE is high for read cycles.

② Device is continuously selected, CS = V_{IL}.

③ Address valid prior to or coincident with CS transition low.

WRITE CYCLE (1)



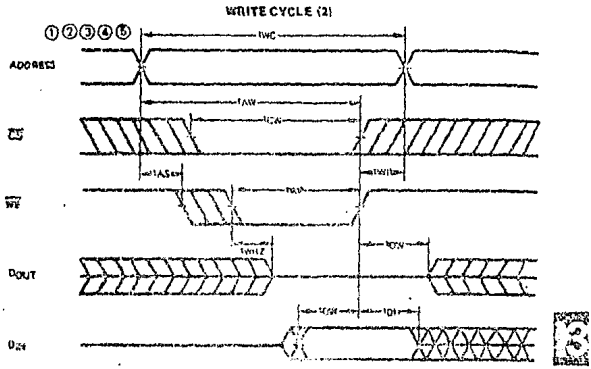
NOTES: ① WE must be high during all address transition.

② A write occurs during the overlap of a low CS and a low WE.

③ tWR is measured from the earlier of CS or WE going high to the end of write cycle.

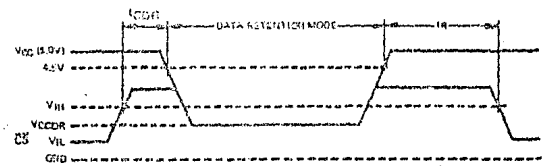
④ If the CS low transition occurs simultaneously with or after the WE low transition, output buffers remain in a high impedance state.

TIMING WAVEFORMS (CONT.)



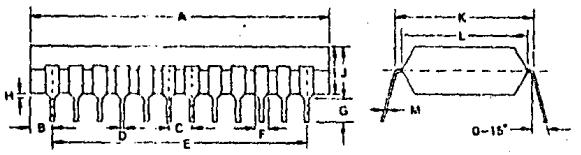
Notes: ① $\overline{WE}$ must be high during all address transition.
② A write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$.
③ t_{WR} is measured from the earlier of $\overline{CS}$ or $\overline{WE}$ going high to the end of write cycle.
④ If the $\overline{CS}$ low transition occurs simultaneously with or after the $\overline{WE}$ low transition, output buffers remain in a high impedance state.
⑤ $\overline{OE}$ is continuously low ($\overline{OE}$ = V_{IL}).

LOW V_{CC} DATA RETENTION TIMING CHART



AC TEST CONDITIONS

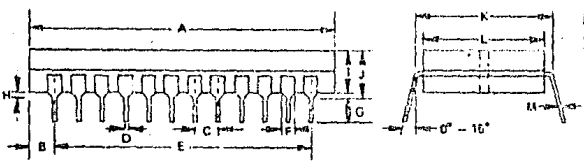
Input Pulse Levels	0.0V to 2.2V
Input Rise and Fall Times	10 ns
Input and Output Timing Reference Levels	1.5V
Output Load	1 TTL + 100 pF



PACKAGE OUTLINE
μPD446C
PLASTIC

PLASTIC

ITEM	MILLIMETERS	INCHES
A	33 MAX	1.3 MAX
B	2.53	0.1
C	2.54	0.1
D	0.5 ± 0.1	0.02 ± 0.004
E	27.94	1.1
F	1.5	0.059
G	2.54 MIN	0.1 MIN
H	0.5 MIN	0.02 MIN
I	5.22 MAX	0.205 MAX
J	5.72 MAX	0.225 MAX
K	15.24	0.6
L	13.2	0.52
M	$0.25^{+0.10}_{-0.05}$	$0.01^{+0.004}_{-0.0039}$



μPD446D
CERDIP

CERDIP

ITEM	MILLIMETERS	INCHES
A	33.5 MAX	1.32 MAX
B	2.78	0.11
C	2.54	0.1
D	0.45	0.018
E	27.94	1.1
F	1.5	0.059
G	2.54 MIN	0.1 MIN
H	0.5 MIN	0.019 MIN
I	4.58 MAX	0.181 MAX
J	5.08 MAX	0.2 MAX
K	15.24	0.6
L	13.0	0.51
M	$0.25^{+0.10}_{-0.05}$	$0.01^{+0.004}_{-0.002}$

2048 × 8-BIT STATIC CMOS RAM

DESCRIPTION

The μPD449 is a high speed, low power, 2048 word by 8-bit static CMOS RAM fabricated using an advanced silicon gate CMOS technology. A unique circuitry technique makes the μPD449 a very low operating power device which requires no clock or refreshing to operate.

Since the device has two chip enable inputs, it is suited for battery backup applications. Minimum standby power current is drawn by this device when $\overline{CE1}$ or $\overline{CE2}$ equals V_{CC} independently of the other input levels.

Data Retention is guaranteed at a power supply voltage as low as 2V.

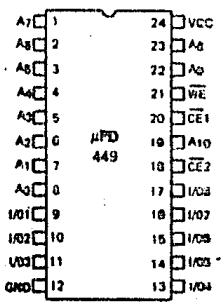
The μPD449 is packaged in a standard 24-pin dual-in-line package and is plug-in compatible with 16K EPROMs.

3

FEATURES

- Single +5V Supply
- Fully Static Operation – No Clock or Refreshing required
- TTL Compatible – All Inputs and Outputs
- Common Data Input and Output Using Three-State Output
- Two Chip Enable Inputs for Battery Operation
- Max Access/Min Cycle Times Down to 150 ns
- Low Power Dissipation; 18 mA Max Active/10 μA Max Standby/10 μA Max Data Retention
- Data Retention Voltage – 2V Min
- Standard 24-Pin Plastic and Ceramic Packages
- Plug-in Compatible with 16K EPROMs
- Operating Temperature Range −40°C to +85°C

PIN CONFIGURATION



PIN NAMES

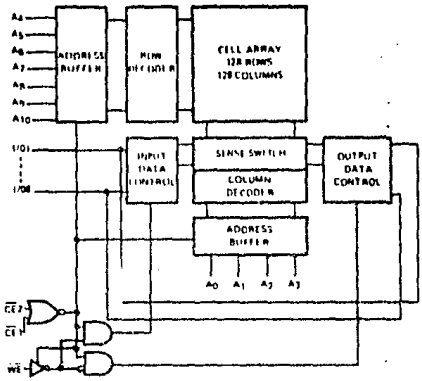
A_0-A_{10}	Address Inputs
$\overline{WE}$	Write Enable
$\overline{CE1}$, $\overline{CE2}$	Chip Enable Inputs
I/O1-I/O8	Data Input/Output
V_{CC}	Power (+5V)
GND	Ground

TRUTH TABLE

$\overline{CE1}$	$\overline{CE2}$	$\overline{WE}$	MODE	I/O	ICC
X	H	X	NOT SELECTED	HZ	STANDBY
H	L	X	NOT SELECTED	HZ	STANDBY
L	L	L	WRITE	DIN	ACTIVE
L	L	H	READ	DOUT	ACTIVE

μPD449

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Supply Voltage 7.0V
Input or Output Voltage Supplied ..... −0.3 to V_{CC} + 0.3V
Storage Temperature Range ..... −55°C to 125°C
Operating Temperature Range ..... −40°C to +85°C

T_a = 25°C

*COMMENT: Stress above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

V_{CC} = 5V ± 10%, T_a = −40°C to +85°C

PARAMETER	SYMBOL	LIMITS MIN	TYP	MAX	UNIT	TEST CONDITIONS
Input High Voltage	V_{IH}	2.2		V_{CC} +0.3	V	
Input Low Voltage	V_{IL}	0.3		0.8	V	
Input Leakage Current	I_{LI}	-1.0		1.0	μA	V_{IN} = 0 ~ V_{CC}
I/O Leakage Current	I_{LO}	-1.0		1.0	μA	$V_{\overline{CE}1}$ or $V_{\overline{CE}2}$ = V_{IH} or V_{WE} = V_{IL} $V_{I/O}$ = 0 ~ V_{CC}
Operating Supply Current	I_{CCA1}		①	①	mA	$V_{\overline{CE}1}$ and $V_{\overline{CE}2}$ = V_{IL} $I_{I/O}$ = 0 MIN TCYCLE
	I_{CCA2}		5	10	mA	$V_{\overline{CE}1}$ and $V_{\overline{CE}2}$ = V_{IL} $I_{I/O}$ = 0 DC CURRENT
	I_{CCA3}		30	10	μA	$V_{\overline{CE}1}$ and $V_{\overline{CE}2}$ = 0.2V, V_{IN} = V_{CC} −0.2V or 0.2V $I_{I/O}$ = 0
Standby Current	I_{CCS}			10	μA	$V_{\overline{CE}1}$ or $V_{\overline{CE}2}$ = V_{CC} −0.2V V_{IN} = 0 ~ V_{CC}
Output High Voltage	V_{OH}	2.4			V	I_{OH} = −1.0 mA
Output Low Voltage	V_{OL}			0.4	V	I_{OL} = 2.0 mA

NOTE: ① μPD449: 12 mA TYP, 18 mA MAX
μPD449-1: 18 mA TYP, 25 mA MAX
μPD449-2: 20 mA TYP, 30 mA MAX
μPD449-3: 25 mA TYP, 35 mA MAX

CAPACITANCE

T_a = 25°C, f = 1.0 MHz

PARAMETER	SYMBOL	LIMITS MIN	LIMITS MAX	UNIT	TEST CONDITIONS
Input Capacitance	C_{IN}		6	pF	V_{IN} = 0V
Input/Output Capacitance	$C_{I/O}$		8	pF	$V_{I/O}$ = 0V

AC CHARACTERISTICS

READ CYCLE

V_{CC} = 5.0V ± 10%, T_a = −40°C to +85°C

PARAMETER	SYMBOL	μPD449-3 MIN	μPD449-3 MAX	μPD449-2 MIN	μPD449-2 MAX	μPD449-1 MIN	μPD449-1 MAX	μPD449 MIN	μPD449 MAX	UNIT
Read Cycle Time	t_{RC}	150		200		250		450		ns
Access Time	t_A		150		200		250		450	ns
Chip Enable ($\overline{CE}$1) to Output Valid	t_{CO1}		150		200		250		450	ns
Chip Enable (CE2) to Output Valid	t_{CO2}		150		200		250		450	ns
Output Hold from Address Change	t_{OH}	15		15		15		15		ns
Chip Enable ($\overline{CE}$1) to Output in LZ	t_{LZ1}	5		5		5		5		ns
Chip Enable (CE2) to Output in LZ	t_{LZ2}	5		5		5		5		ns
Chip Enable ($\overline{CE}$1) to Output in HZ	t_{HZ1}		50		60		80		100	ns
Chip Enable (CE2) to Output in HZ	t_{HZ2}		50		60		80		100	ns

WRITE CYCLE

V_{CC} = 5.0V ± 10%, T_a = −40°C to +85°C

PARAMETER	SYMBOL	μPD449-3 MIN	μPD449-3 MAX	μPD449-2 MIN	μPD449-2 MAX	μPD449-1 MIN	μPD449-1 MAX	μPD449 MIN	μPD449 MAX	UNIT
Write Cycle Time	t_{WC}	150		200		250		450		ns
Chip Enable ($\overline{CE}$1) to End of Write	t_{CW1}	120		150		180		210		ns
Chip Enable (CE2) to End of Write	t_{CW2}	120		150		180		210		ns
Address Setup Time	t_{AW}	0		0		0		0		ns
Write Pulsewidth	t_{WP}	90		120		150		180		ns
Write Recovery Time	t_{WR}	0		0		0		0		ns
Write Enable to Output in HZ	t_{WZ}		50		60		80		100	ns
Output Active from End of Write	t_{OW}	10		10		10		10		ns
Data Valid to End of Write	t_{DW}	50		60		80		100		ns
Data Hold Time	t_{DH}	0		0		0		0		ns

LOW V_{CC} DATA RETENTION

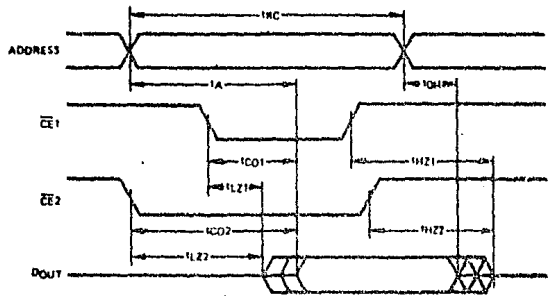
T_a = −40°C to +85°C

PARAMETER	SYMBOL	TEST CONDITIONS	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
V_{CC} for Data Retention	V_{CCDR}	V_{IN} = 0 ~ V_{CC}, $V_{\overline{CE}1}$ or V_{CE2} = V_{CC}	2.0			V
Data Retention Current	I_{CCDR}	V_{CC} = 3.0V, V_{IN} = 0 ~ V_{CC}, $V_{\overline{CE}1}$ or V_{CE2} = V_{CC}		0.01	10	μA
Chip Disable to Data Retention Time	t_{CDR}		0			ns
Operation Recovery Time	t_R		t_{RC}			ns

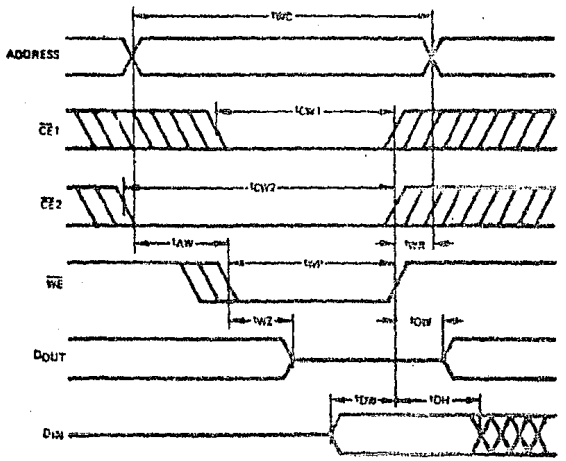
87

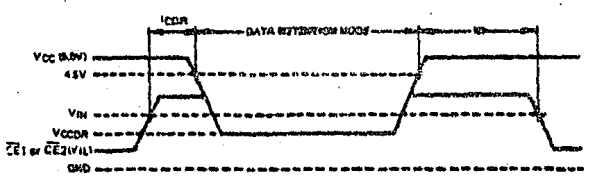
TIMING WAVEFORMS

READ CYCLE TIMING CHART



WRITE CYCLE TIMING CHART





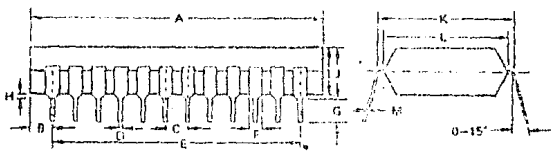
LOW VCC DATA RETENTION TIMING CHART

AC TEST CONDITIONS

Input Pulse Levels	0.8V to 2.2V
Input Rise and Fall Times	10 ns
Input and Output Timing Reference Levels	1.5V
Output Load	1 TTL + 100 pF

PACKAGE OUTLINES

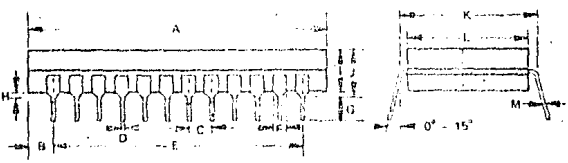
μPD449C PLASTIC



PLASTIC

ITEM	MILLIMETERS	INCHES
A	33 MAX	1.3 MAX
B	2.53	0.1
C	2.54	0.1
D	0.5 +0.1	0.02 +0.004
E	27.94	1.1
F	1.5	0.059
G	2.54 MIN	0.1 MIN
H	0.5 MIN	0.02 MIN
I	5.22 MAX	0.79 MAX
J	5.72 MAX	0.45 MAX
K	15.24	0.6
L	13.2	0.53
M	0.25 +0.13 −0.05	0.01 +0.004 −0.0019

μPD449D CERDIP



CERDIP

ITEM	MILLIMETERS	INCHES
A	33.5 MAX	1.32 MAX
B	2.28	0.11
C	2.54	0.1
D	0.46	0.018
E	27.94	1.1
F	1.5	0.059
G	2.54 MIN	0.1 MIN
H	0.5 MIN	0.019 MIN
I	4.58 MAX	0.181 MAX
J	5.08 MAX	0.2 MAX
K	15.24	0.6
L	13.5	0.53
M	0.25 +0.10 −0.05	0.01 +0.004 −0.002

449DS-1-82-CAT

National Semiconductor

PRELIMINARY

microCMOS

NMC6164/6164L 8192 × 8-Bit Static RAM

General Description

The NMC6164/6164L is a 8192-word by 8-bit, new-generation static RAM. It is fabricated with National's proprietary microCMOS double-polysilicon technology which combines high performance and high density with low power consumption and excellent reliability.

The NMC6164/6164L operates with a single 5V power supply with ± 10% tolerance. Additional battery back-up operation is available (L version) for data retention down to 2V, with low standby current.

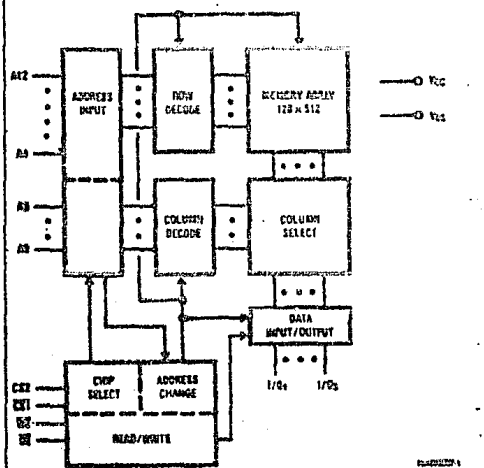
Packaging is in standard 28-pin DIP and is available in both plastic and CERDIP.

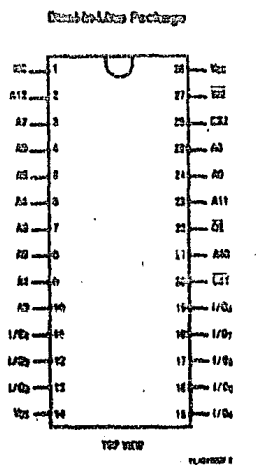
In addition to the inputs and outputs being TTL compatible, the outputs are also CMOS compatible, in that capacitive loads are driven to V_{CC} or V_{SS}.

Features

- Single power supply: 5V ± 10%
- Fast access time 100 ns/120 ns/150 ns max
- Equal access and cycle times
- Completely static RAM: no clock or timing strobe required
- Low standby power and low power operation
  - Standby: 10 μW, typical
  - Operation: 15 mW/MHz, typical
- Battery back-up operation available (L version) with data retention supply voltage: 2V–5.5V
- Common data input and output, TRI-STATE® output
- TTL compatible: all inputs and outputs
- CMOS compatible: outputs drive capacitive loads to V_{CC} or V_{SS}
- Standard 28-pin package configuration

Block and Connection Diagrams





Truth Table

Mode	$\overline{WE}$	$\overline{CS1}$	CS2	$\overline{OE}$	I/O	Current
Not Selected (Power Down)	*	H	*	*	HI-Z	I_{SB}, I_{SB1}
	*	*	L	*	HI-Z	I_{SB}, I_{SB1}
Output Disabled	H	L	H	H	HI-Z	I_{CC}, I_{CC1}
Read	H	L	H	L	D_{OUT}	I_{CC}, I_{CC1}
Write	L	L	H	*	D_{IN}	I_{CC}, I_{CC1}

*Don't care (H or L) H = Logic HIGH Level L = Logic LOW Level

Order Number NSC6164J (NMC6164LJ)
NS Package Number J28A
Order Number NMC6164N (NMC6164LN)
NS Package Number N28B

Absolute Maximum Ratings

Voltage on Any Pin Relative to V_{SS}	−0.5V to +7V
Storage Temperature, T_{STG}	−55°C to +125°C
Temperature Under Bias, T_{BIAS}	−10°C to +85°C
Power Dissipation, P_D	1.0W
Current Through Any Pin	100 mA

Recommended DC Operating Conditions

	Min	Max	Units
V_{CC} Supply Voltage	4.5	5.5	V
V_{SS} Supply Voltage	0	0	V
V_{IH}, Input High Voltage (Logic 1)			
TTL	2.2	6.0	V
CMOS	$V_{CC}-0.2$	$V_{CC}+0.2$	V
V_{IL}, Input Low Voltage (Logic 0)			
TTL	−0.3	0.8	V
CMOS	−0.2	0.2	V
T_{OPR}, Operating Temp	0	70	°C

DC Electrical Characteristics at recommended operating conditions

Symbol		Parameter	Conditions	Min	Max	Units
I_{LI}		Input Leakage Current	$V_{IN} = V_{SS}$ to V_{CC}	−2	2	μA
I_{LO}		Output Leakage Current	$\overline{CS1} = V_{IH}$ or $CS2 = V_{IL}$ or $\overline{OE} = V_{IH}$ $V_{I/O} = V_{SS}$ to V_{CC}	−2	2	μA
I_{CC}		Active Quiescent Current, TTL	All Inputs at TTL Levels $\overline{CS1} = V_{IL}$, TTL or $CS2 = V_{IH}$, TTL $I_{I/O} = 0$ mA		25	mA
I_{CC}	Std.	Active Quiescent Current, CMOS	All Inputs at CMOS Levels $\overline{CS1} = V_{IL}$, CMOS and $CS2 = V_{IH}$, CMOS $I_{I/O} = 0$ mA		2	mA
	L				100	μA
I_{CC1}		Average Operating Current, TTL	Duty Cycle = 100% All Inputs at TTL Levels		60	mA
		Average Operating Current, CMOS	Duty Cycle = 100% All Inputs at CMOS Levels		40	mA
I_{SB}	Std.	Standby Power Supply Current	$\overline{CS1} = V_{IH}$, TTL or $CS2 = V_{IL}$, TTL $I_{I/O} = 0$ mA		4	mA
	L				2	mA
I_{SB1}	Std.	Standby Power Supply Current	$\overline{CS1} = V_{IH}$, CMOS or $CS2 = V_{IL}$, CMOS		2	mA
	L				100	μA
V_{OL}		Output Low Voltage, TTL	$I_{OL} = 2.1$ mA		0.4	V
		Output Low Voltage, CMOS	$I_{OL} = \pm 10$ μA	−0.2	0.2	V
V_{OH}		Output High Voltage, TTL	$I_{OH} = -1.0$ mA	2.4		V
		Output High Voltage, CMOS	$I_{OH} = \pm 10$ μA	$V_{CC}-0.2$	$V_{CC}+0.2$	V

Capacitance

Symbol	Parameter	Conditions	Max	Units
C_{IN}	Input Capacitance	$V_{IN} = 0V$ (Note 5)	6	pF
$C_{I/O}$	Input/Output Capacitance	$V_{I/O} = 0V$ (Note 5)	8	pF

AC Electrical Characteristics (Note 1) (Standard and L Versions)

Symbol	Parameter	NMC6164/6164L						Units
		-10*		-12*		-15*		
		Min	Max	Min	Max	Min	Max	
READ CYCLE (Note 4)								
t_{RC}	Read Cycle Time	100		120		150		ns
t_{AA}	Address Access Time		100		120		150	ns
t_{CO1}	Chip Selection ($\overline{CS1}$) to Output Valid		100		120		150	ns
t_{CO2}	Chip Selection (CS2) to Output Valid		100		120		150	ns
t_{OE}	Output Enable ($\overline{OE}$) to Output Valid		50		60		70	ns
t_{LZ1}	Chip Selection ($\overline{CS1}$) to Output Active	10		10		15		ns
t_{LZ2}	Chip Selection (CS2) to Output Active	10		10		15		ns
t_{OLZ}	Output Enable ($\overline{OE}$) to Output Active	5		5		5		ns
t_{HZ1}	Chip Deselection ($\overline{CS1}$) to Output in HI-Z (Notes 2 and 3)	0	35	0	40	0	50	ns
t_{HZ2}	Chip Deselection (CS2) to Output in HI-Z (Notes 2 and 3)	0	35	0	40	0	50	ns
t_{OHZ}	Output Disable ($\overline{OE}$) to Output in HI-Z (Notes 2 and 3)	0	35	0	40	0	50	ns
t_{OHA}	Output Hold from Address Change	10		10		15		ns
WRITE CYCLE								
t_{WC}	Write Cycle Time	100		120		150		ns
t_{CW1}	Chip Selection ($\overline{CS1}$) to End of Write (Note 10)	80		85		100		ns
t_{CW2}	Chip Selection ($\overline{CS2}$) to End of Write	80		85		100		ns
t_{AS}	Address Set-Up Time (Note 7)	0		0		0		ns
t_{AW}	Address Valid to End of Write	80		85		100		ns
t_{WP}	Write Pulse Width (Note 6)	60		70		90		ns
t_{WR1}	Write Recovery Time from $\overline{CS1}$ (Note 8)	0		5		10		ns
t_{WR2}	Write Recovery Time from CS2 (Note 8)	0		5		10		ns
t_{WHZ}	Beginning of Write to Output in HI-Z (Note 9)	0	35	0	40	0	50	ns
t_{DW}	Data Valid to Write Time Overlap	35		40		50		ns
t_{DH}	Data Hold from End of Write	0		0		0		ns
t_{OHZ}	Output Disable ($\overline{OE}$) to Output in HI-Z	0	35	0	40	0	50	ns
t_{OW}	Output Active from End of Write	5		5		10		ns

*Applies to Standard and L Versions.

Note 1: AC test conditions $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 10\%$.

Note 2: t_{HZ} and t_{OHZ} are defined as the time at which the outputs achieve the open circuit condition and are not referenced to
High to TRI-STATE measured as V_{OH} (DC) − 0.1V
Low to TRI-STATE measured as V_{OL} (DC) + 0.1V

Note 3: At any given temperature and voltage condition, $t_{HZ\,MAX}$ is less than $t_{LZ\,MIN}$ both for a given device and from device to device.

Note 4: $\overline{WE}$ is high for read cycle.

Note 5: $T_A = 25°C$, f = 1.0 MHz. This parameter is sampled and not 100% tested.

Note 6: A write occurs during the overlap (t_{WP}) of a low $\overline{CS1}$ and a high CS2 and a low $\overline{WE}$.

Note 7: t_{AS} is measured from the address changes to the beginning of the write.

Note 8: t_{WR} is measured from the earliest of $\overline{CS1}$ or $\overline{WE}$ going high or CS2 going low to the end of the write cycle.

Note 9: If $\overline{CS1}$ is low and CS2 is high during this period, I/O pins are in the output state. At this time, the data input signals of opposite phase to the outputs must not be applied.

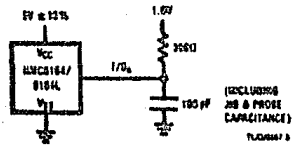
Note 10: If the $\overline{CS1}$ low transition occurs simultaneously with the $\overline{WE}$ low transition or after the $\overline{WE}$ transition, the outputs will remain in a HI-Z state.

Note 11: CS2 controls the address buffers, $\overline{WE}$ buffer, $\overline{CS1}$ buffer, D_{IN} buffer and $\overline{OE}$ buffer. When CS2 controls the data retention mode, V_{IN} level (address, $\overline{WE}$, $\overline{CS1}$, $\overline{OE}$) can be in the high impedance state. When $\overline{CS1}$ controls the data retention mode, CS2 must be at V_{IN}, CMOS. All other input levels (address, $\overline{WE}$, I/O) can be in the high impedance state.

AC Test Conditions:

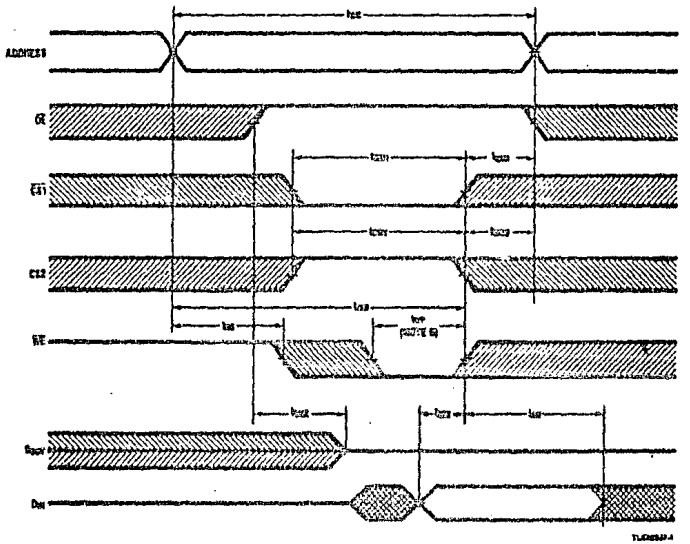
Input pulse levels $V_{IH} = 3.0V$, $V_{IL} = 0.0V$
Input rise and fall times 5 ns
All input and output timing reference levels 1.5V

AC Test Load

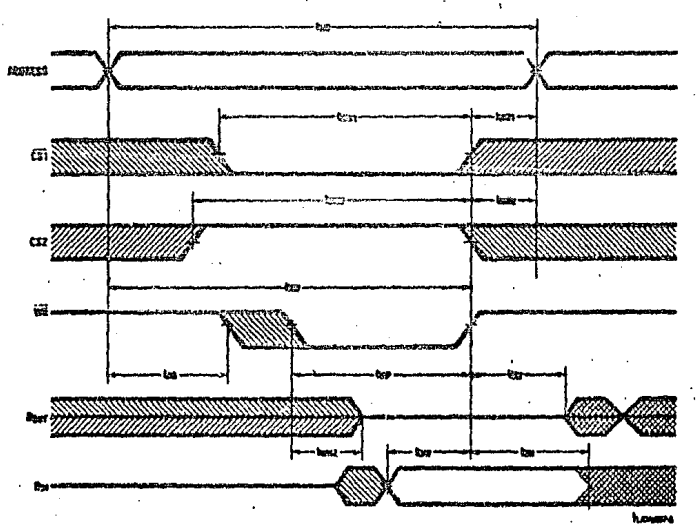


Timing Waveforms

Write Cycle 1 ($\overline{OE}$ Clocked)

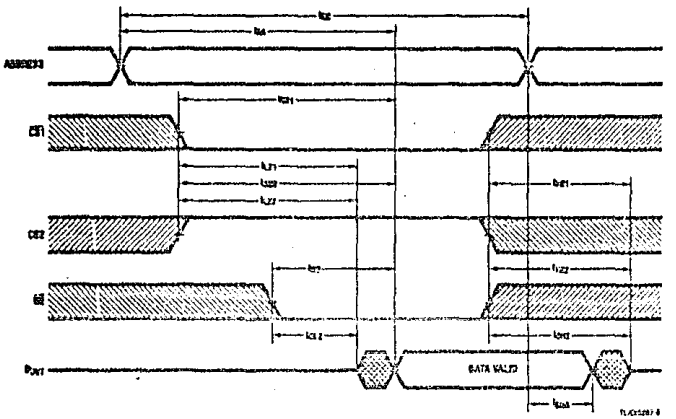


Write Cycle 2 ($\overline{WE}$ Low Fixed)



Timing Waveforms (Continued)

Read Cycle

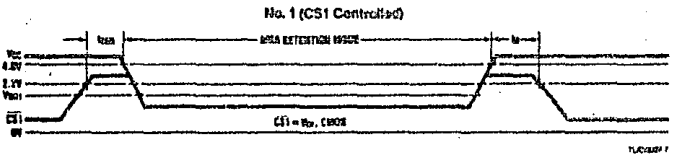


Low V_{CC} Data Retention (L Version)

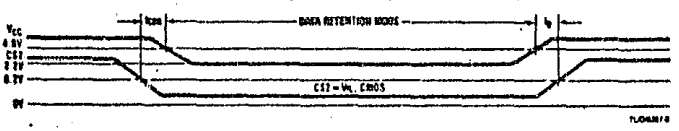
Symbol	Parameter	Conditions	Min	Max	Units
V_{DR1}	V_{CC} for Data Retention	$\overline{CS1} > V_{IH}$, CMOS $CS2 > V_{IH}$, CMOS	2.0		V
V_{DR2}	V_{CC} for Data Retention	$CS2 < V_{IL}$, CMOS	2.0		V
I_{CCDR1}	Data Retention Current (Note 11)	$V_{CC} = 2V$ $\overline{CS1} > V_{IH}$, CMOS $CS2 > V_{IH}$, CMOS		40	µA
I_{CCDR2}	Data Retention Current (Note 11)	$V_{CC} = 2V$ $CS2 < V_{IL}$, CMOS		40	µA
t_{CDR}	Chip Deselect to Data Retention Time	See Retention Waveform	0		ns
t_R	Operation Recovery Time	See Retention Waveform	t_{RC}		ns

Low V_{CC} Data Retention Waveforms

No. 1 ($\overline{CS1}$ Controlled)



No. 2 (CS2 Controlled)



1 MACROENSAMBLADOR

1.1 INTRODUCCION

El macroensamblador es un programa que traduce mnemonicos del microprocesador Z80 a codigos de maquina ejecutables. Los mnemonicos y codigos de operacion son identicos a los utilizados por MOSTEK y ZILOG en su literatura, excepto por las diferencias que representan facilidades del ensamblador.

El programa es un ensamblador de dos 'pasadas'. En la primera identifica etiquetas y variables, y en la segunda genera los codigos de operacion dando como resultado un listado con mensajes de error, facilidades de lectura y una lista de referencia de variables. Adicionalmente proporciona un archivo con un programa objeto que puede ser usado para cargas en el LINK-LOADER o bien un modulo objeto para el grabado de PROMs o EPROMs.

El programa se distingue por su capacidad de manejar macros, ensambles condicionales, direccionamiento simbolico y relativo, evaluacion de expresiones complejas, generacion de un listado de referencia y versatilidad en sus modos de direccionamiento. Adicionalmente el programa es capaz de generar datos en muchos sistemas de numeracion y codigos como ASCII o EBCDIC.

1.2 LENGUAJE ENSAMBLADOR

El lenguaje del ensamblador nos brinda una manera facil de crear programas para microsistemas con las siguientes ventajas:

* Facilidad de generacion de programas en mnemonicos.
* Facilidad de modificacion.
* Listados de facil lectura y entendimiento.
* Generacion de modulos accesables en lenguaje de maquina.

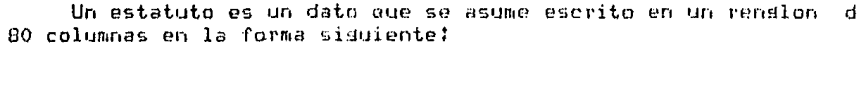
Y ha sido disenado para soportar las caracteristicas siguientes:

* Manejo de codigos de operacion de mnemonicos.
* Direcciones asignadas o referenciadas.
* Direccionamiento relativo.
* Comandos para crear datos.
* Comandos para reservar memoria.
* Comandos de control para el listado de salida.
* Direccionamientos constantes.
* Codigos de caracteres ASCII o EBCDIC.
* Comentarios para documentacion.
* Listado de referencia.
* Modulos legibles o modulos objeto.

Un programa en lenguaje ensamblador es un archivo escrito en lenguaje simbolico de maquina (mnemonicos). El programa los comprime en estatutos; un estatuto es un instruccion simbolica (mnemonicos), una direccion, una instruccion macro o un comentario.

1.2.1 ESTATUTOS

Un estatuto es un dato que se asume escrito en un renglon de 80 columnas en la forma siguiente:



El campo denominado "etiquetas" es provisto para asignar un nombre simbolico a una direccion de memoria o a un valor variable o constante. El campo de etiquetas puede empezar en cualquier columna si es terminada por una coma o empezando de la columna uno omitiendola. Un estatuto puede consistir unicamente de una etiqueta.

El campo de "operacion" es provisto para especificar una operacion simbolica, una direccion o una llamada a un macro. Si esta presente, puede empezar despues de la columna uno o estar separada del campo de etiquetas por uno o mas espacios o una coma.

El campo del "operándo" es proveido para especificar argumentos de la operacion dada en el campo anterior. El campo del operando, si esta presente, esta separado del campo de operacion por uno o mas espacios. Los argumentos en el campo de operandos no deben estar separados por blancos o por mas de una coma.

El campo de comentarios es provisto para deshabilitar el ensamblador o para poner mensajes acerca del proposito de la instruccion. El campo de los comentarios debe estar separado de el campo anterior por uno o mas blancos o por un punto y coma.

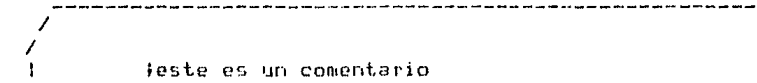
Un estatuto puede estar completamente en blanco, en cuyo caso una linea en blanco aparecera en el listado de salida.

1.2.1.1 COMENTARIOS

Un comentario es un estatuto el cual no es ensamblado, solamente es reproducido en el listado de salida. Este estatuto esta indicado por un asterisco o un punto y coma como el primer caracter no blanco en la linea. Hay que tener cuidado cuando se usa asterisco para indicar un comentario porque puede ser interpretado como una directivo del ensamblador. Si un asterisco

es usado para indicar un comentario, es recomendable iniciarlo con un espacio.

Nota!!!. Si el asterisco aparece en la columna uno el ensamblador puede reconocerlo como un directivo.



```
*       ¡este es un comentario
```

Adicionalmente, las columnas de la 73 a la 80 nunca son procesadas y pueden ser usadas para cualquier proposito.

1.2.1.2 DIRECCIONAMIENTO SIMBOLICO

Cuando se escribe un programa en mnemonicos, los operandos son expresados en muchas ocasiones en forma simbolica. El programa ensamblador reconoce este tipo de direccionamiento y para aprovecharlo solo es necesario colocar etiquetas a los estatutos y posteriormente referirse a ellas en los argumentos. Por ejemplo:

```
LOOP    LD A,B
        .
        .
        .
        JR Z,LOOP
```

Tambien es posible direccionar localidades cercanas a las etiquetas definidas utilizando los operadores + o -. Por ejemplo:

```
        JP BEGIN
        JR PE,BEGIN+4
BEGIN   LD A,B
        HALT
        LD C,'B'
        INC B
```

En el ejemplo anterior la instruccion "JP BEGIN" se refiere a la intruccion "LD A,B" mientras que la intruccion "JR PE,BEGIN+4" se refiere a la intruccion "INC B". BEGIN+4 significa la

direccion BEGIN mas cuatro bytes y no mas cuatro intrucciones. Notese que no deben existir espacios entre los operandos.

1.2.2 SINTAXIS

El lenguaje ensamblador es como cualquier otro lenguaje, es decir, tiene un set de caracteres, un vocabulario, reglas de gramatica y la posibilidad de definir nuevas palabras o elementos.

Para que un estatuto sea ensamblado debe estar correctamente escrito de acuerdo con las reglas de sintaxis descritas enseguida para poder ser reconocidas por el programa.

1.2.2.1 SET DE CARACTERES

La siguiente lista de caracteres son los que el programa puede reconocer. Son los unicos caracteres validos y cualquier otro, a excepcion de los que se usen en el campo de comentarios, generara un error. Muchos de los caracteres especiales no tienen un significado previamente definido excepto cuando se trata de un caracter constante.

CARACTERES ALFABETICOS

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

CARACTERES NUMERICOS

0 1 2 3 4 5 6 7 8 9

CARACTERES ESPECIALES

blanco	/ diagonal
> mayor que	$ pesos
< menor que	* asterisco
' comilla simple	(parentesis izquierdo
, coma	) parentesis derecho
+ mas	@ arroba
- menos	. punto
& ampersand	: dos puntos
! admiracion	; punto y coma
" comillas dobles	= igual
# sharp	? interrogacion
% porciento	_ underbar
¦ barra vertical	\ diagonal inversa

1.2.2.2 CONSTANTES

Una constante es un valor que permanece invariable y existen muchas maneras de especificarlas en el lenguaje ensamblador.

Constantes decimales pueden ser especificadas como una secuencia de caracteres numericos, precedidos por un signo menos o mas. Si no esta signado, el valor es asumido positivo.

Los valores negativos se codifican en complemento a dos y son evaluadas en cantidades de 16 bits, es decir, en modulo 65336 por lo que no se debe intentar evaluar numeros demasiado grandes o un error sera generado.

La siguiente lista indica los decriptores disponibles y su significado. Si no se da descriptor el numero es asumido decimal.

B - Binario
O - Octal
Q - Octal
D - Decimal
H - Hexadecimal

Usando los descriptores anteriores, debe colocarse un cero a la izquierda cuando se utilicen numeros hexadecimales que empiecen de la A a la F. Ejemplo de constantes son los siguientes:

```
10011B      25      0FFH      1377Q      255D
```

Un caracter constante en ASCII o en EBCDIC puede ser especificado encerrado con comillas simples precedido por una A para codificacion ASCII o una E para una codificacion EBCDIC. Si no se da un descriptor el caracter se asume ASCII. Ejemplos de esto se dan enseguida:

```
        LD  A,'I'
        LD  C,E'A'
        OR  'O'
```

Para algunas intrucciones, la constante puede consistir de dos caracteres. Los caracteres son cargados en los registros en el orden en que, esto es, el estatuto:

```
        LD   HL,'-1'
```

asignara el codigo ASCII "-" al registro H y el ASCII "1" al registro L.

Una conJunto de caracteres puede ser especificado usando

DEFB, DB, DATA o DEFW. Los caracteres declarados mediante estos operadores se definen de igual manera para ASCII o EBCDIC, por ejemplo:

```
A'CODIGO DE TELETIPO'
E'CODIGO DE TERMINAL'
 ' 123.7'
```

Notese que un byte de memoria es ocupado por cada codigo de cada caracter en forma secuencial con el primero disponible cuando son definidos mediante DEFB, DB, DATA O DEFW.

Para generar el codigo de una comilla simple se especifica con dos comillas simples consecutivas, ejemplo:

```
'DON''T'
```

1.2.2.3 SIMBOLOS

-Un simbolo es una secuencia de caracteres. El primer caracter en un simbolo debe ser alfabetico, signo de interrogacion, arroba, signo de admiracion, porcentaje o underbar. Los caracteres especiales, exceptuando los enunciados anteriormente, no deben ser usados en la definicion de simbolos. Tampoco son permitidos blancos intermedios.

Solo los primeros seis caracteres de un simbolo son usados por el programa para definirlo; los restantes pueden ser usados para documentacion. La tabla de simbolos del programa puede contener hasta 200 diferentes definiciones.

Los simbolos son usados para representar valores aritmeticos, direcciones de memoria, arreglos de bits, etc. Ejemplos de simbolos validos son los siguientes:

```
LAB 1
MASK
LOOP_NUM          (El simbolo solo es LOOP_N)
```

1.2.2.4 SIMBOLOS ESPECIALES

El ensamblador tiene 20 simbolos reservados que corresponden precisamente a los nombres de los registros internos y codigos condicionales definidos por ZILOG. Estos simbolos especiales no son guardados en la tabla de simbolos por lo que pueden ser usados como etiquetas. Dichos simbolos son los siguientes:

A	B	C	D
E	F	H	L
BC	DE	HL	SP
AF	AF'	IX	IY
I	R	Z	NZ
C	NC	PE	PO
P	M		

El ensamblador generara un error si se atenta usar uno de estos simbolos donde no es requerido o si no es usado donde se requiere.

Exiten acicionalmente dos simbolos que denotan el stack y segmentos de memoria de un programa, estos son:

STACK MEMORY

Finalmente, existe un simbolo predefinido llamado 'NARG' y es usado para representar el numero de argumentos pasados en una llamada de macro.

1.2.2.5 CONTADOR DE PROGRAMA DEL ENSAMBLADOR

Durante el proceso de ensamblado se mantiene una variable que contiene la direccion de la siguiente localidad de memoria llamada CONTADOR DE PROGRAMA del ensamblador. Es usada por este para asignar direcciones a los bytes ensamblados, sin embargo, esta disponible tambien para el programador.

El caracter '$' es el nombre simbolico del contador de programa y puede ser usado como cualquier otro simbolo, pero no aparecer en el campo de etiquetas. Cuando se usa el '$' el programa asume que $ = mi misma direccion, por ejemplo:

```
3F  JR      $
```

La intruccion de salto relativo se encuentra en la direccion

3F y es codificada como un salto a si mismo. El CONTADOR DE PROGRAMA tiene en este caso el valor de 3F.

1.2.2.6 EXPRESIONES

Una expresion es una secuencia de uno o mas simbolos, constantes o de otras expresiones. Las expresiones son evaluadas de izquierda a derecha sujetas a las prioridades que se muestran abajo. Los parentesis hacen que las prioridades no sean tomadas en cuenta. Las operaciones y prioridades son mostradas enseguida de la mayor a la menor.

PRIORIDAD	OPERADOR	
1	+	(signo positivo)
	-	(signo-negativo)
2	-**	(exponenciacion)
3	*	(multiplicacion)
	/	(division)
	.MOD.	(modulo)
	.SHR.	(corrimiento a la derecha)
	.SHL.	(corrimiento a la izquierda)
4	+	(suma)
	-	(resta)
5	,.NOT.	(NOT logico)
6	&,.AND.	(AND logico)
7	!,.OR.	(OR logico)
	.XOR.	(OR exclusiva)
8	=,.EQ.	(igualdad)
	>,.GT.	(mayor que)
	<,.LT.	(menor que)
	.UGT.	(mayor que sin signo)
	.ULT.	(menor que sin signo)
9	.RES.	(resultado)
10	.LOW.	(8 LSB)
	.HIGH.	(8 MSB)

Las operaciones de comparacion dan como resultado todos los bits "1" si la comparacion es verdadera, y todos los bits "0" si la comparacion es falsa. Los operadores .GT. y .LT. realizan la comparacion con numeros signados por lo que si son mayores de 32767 seran tratados como negativos. Los operadores .UGT. y .ULT. realizan las comparaciones si tomar en cuenta el signo.

El operador resultado, .RES., no realiza ninguna funcion pero se da para compatibilidad con otros ensambladres.

- Los operadores de corrimiento (.SHR. y .SHL.) realizan su funcion tantas veces como lo indique el argumento antes del

operador sobre el argumento especificado despues de el. Los lugares dejados a la izquierda o derecha son llenados con ceros. Ejemplo:

```
DEFB 2.SHL.BIT
```

Los operadores .HIGH. y .LOW. se dan para ayudar al programador a definir una direccion de dos bytes en forma individual si se desea.

Una expresion debe resolver un valor unico, consecuentemente no se permiten argumentos caracteres (a excepcion de longitud 2) debido a que son procesados en modulo 65536. Atentar contra esto generara un mensaje de error. Ejemplos de expresiones validas se dan enseguida:

```
PAM+3                   LOOP+(ADDR.SHR.8)
(PAM+45H)/CAL           VAL1.EQ.VAL2
IDAM.AND.255            .LOW.EXPR
```

Como puede verse, los blancos intermedios no son permitidos.

Para ciertos mnemonicos, una expresion encerrada en parentesis indica una direccion de memoria. Para indicar valores inmediatos en una expresion encerrada en parentesis se puede hacer colocando un signo mas al principio del valor.

1.2.2.7 DIRECCIONAMIENTO RELATIVO

Para aquellas intrucciones que usan direccionamiento relativo (JR, DJNZ), el contador de programa, $, puede ser sustraido o no de la direccion relativa dependiendo si se especifica LIST o NLIST (explicado mas adelante). De esta manera el usuario puede definir el direccionamiento relativo en cualquiera de las siguientes formas:

```
DJNZ MAIN               DJNZ MAIN-$
```

Aunque el default es "$" y no es necesario especificarlo, se recomienda usarlo para decir explicitamente al ensamblador que va a hacer, ademas de que ciertos errores pueden ser detectados solo con esta opcion.

1.2.3 COMANDOS DEL ENSAMBLADOR

Los comandos del ensamblador son directivos que empiezan con un asterisco en la columna uno. La columna dos identifica el tipo de comando. El usuario debe ser cauteloso con el uso de estos comandos cuando denota comentarios con un asterisco en la columna uno. Dependiendo de que caracter se encuentre en la columna 2, puede ser interpretado como comando o no. Los comandos equivalentes a los directivos correspondientes se dan a continuacion.

*EJECT	EJECT
*HEADING S	TITLE 'S'
*LIST ON	LIST S
*LIST OFF	NLIST S
*MACLIST ON	LIST M
*MACLIST OFF	NLIST M

1.2.4 DIRECTIVOS

Los directivos del programa son escritos como estatutos ordinarios en el lenguaje ensamblador, pero no son transladados a lenguaje de maquina sino que son identificados como comandos para el control del ensamblado.

A traves de estos directivos, el ensamblador reservara memoria, definira bytes de datos, asignara valores a simbolos, controlara los listados de salida, etc.

Esta seccion describe todos los directivos excepto aquellos asociados con el ensamblado de macros y relocalizacion. Algunos directivos tales como ORG se aplican a ensamblado absoluto o relocalizado y se discute en ambas secciones.

Los directivos descritos en esta seccion son:

ORG	Coloca el origen del programa.
END	Fin del ensamblado.
EQU	Iguala un simbolo a una expresion.
DEFL	Define o redefine una etiqueta.
DEFB	Define bytes.
DB	Define bytes(como DEFB).
DATA	Define bytes (como DEFB).
DEFW	Define palabras.
DW	Define palabras(como DEFW).
DDB	Define bytes dobles.
DEFS	Define memoria.

DS	Define memoria(como DEFS).
DEFM	Define mensajes.
SPAC	Da lineas en el listado.
EJECT	Avanza el listado a la siguiente pagina.
TITLE	Coloca el encabezado del programa.
LIST	Lista los elementos especificados.
NLIST	No lista los elementos especificados.
COND	Ensamble condicional.
IF	Ensamble condicional(como COND)
ELSE	Ensamble condicional alternativo.
ENDC	Fin del ensamble condicional.
ENDIF	Fin del ensamble condicional(como ENDC).

En las siguientes descripciones los parentesis () son usados para indicar opcionalidad.

1.2.4.1 ORG

Coloca el origen del programa (En modo no-relocalizable).

El directivo ORG es usado para informar al ensamblador la direccion con la cual debera ser asignado el siguiente byte ensamblado. Los subsecuentes bytes seran asignados con direcciones secuenciales empezando con esta direccion.

Si el programa no tiene un ORG como primer estatuto, un "ORG 0" es asumido y el ensamblador empieza en la direccion cero en el segmento absoluto del programa, esto es, el programa no sera relocalizable, a menos que se especifique lo contrario.

```
 ---------------------------------------------------
/
/
: (etiqueta)   ORG    expresion
```

donde:

Etiqueta -Es un simbolo opcional el cual,si esta presente, sera igualado a la expresion dada.

Expresion -Un valor la cual reemplazara el contenido del contador de programa del ensamblador con el valor especificado. Cualquier simbolo utilizado debe estar previamente definido.

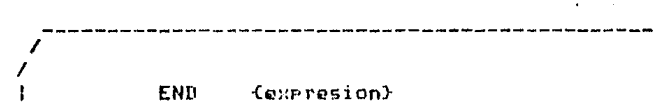
1.2.4.2 END

El directivo END es usado para informar al ensamblador que es el ultimo estatuto que debe ser leido e indicar la direccion de comienzo del modulo de carga. Cualquier estatuto que siga al directivo END no sera procesado.

Especificando una direccion de cargado en este directivo tambien informa al cargador que este es el programa principal. Si son combinados multiples modulos de carga por el Linking Loader, solo un modulo puede ser especificado con una direccion de cargado y de esta manera sea un programa principal.

Ejemplo:

```
END MAIN
```



donde:

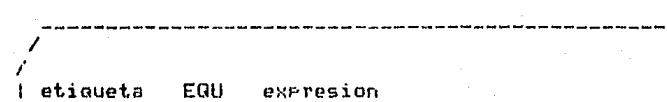
Expresion -Es una direccion que va a ser puesta en el final de el record del modulo de carga e informa al cargador donde va a empezar la ejecucion del programa. Si la expresion no esta especificada, la direccion de cargado es puesta a cero.

1.2.4.3 EQU

El directivo EQU causa que el ensamblador asigne un valor particular a una etiqueta no definida. Este valor puede ser absoluto o relocalizable.

Ejemplo:

```
SEVEN EQU 7
```



donde:

Etiqueta -Es un simbolo definido por este estatuto.

Expresion -Es una expresion la cual valuara a la etiqueta dada durante todo el ensamblado. Si se intenta revaluar la misma etiqueta, se generara un mensaje de error. Cualquier simbolo usado en esta expresion debe estar previamente definido. No pueden ser usados aqui simbolos externos.

1.2.4.4 DEFL

El directivo DEFL asigna a un simbolo un valor particular, pero a diferencia del directivo EQU puede haber multiples DEFL al mismo simbolo durante el ensamblado del mismo programa. La ultima definicion de el simbolo es la que especifica su valor.

Ejemplo:

```
GO    DEFL   5
GO    DEFL   GO+10
```


donde:

Etiqueta -Es un simbolo definido por este estatuto.

Expresion -Es un valor que se le sera asignado a la etiqueta dada a menos que sea cambiado por otro DEFL. Cualquier simbolo usado en esta expresion debe estar previamente definido. Los simbolos externos no pueden ser usados aqui.

1.2.4.5 DEFB DATA DB

Los directivos DEFB, DATA, DB son usados para definir hasta 70 bytes de datos. El ensamblador colocara un byte si se da una expresion y varios si una palabra de dos o mas caracteres se especifica. Todas las expresiones deberan valuar un solo byte o un mensaje de error se generara.Los valores negativos se codificaran en complemento a dos. En una expresion relocalizable puede usarse los operadores .HIGH. y .LOW..Si cualquier otro operador es usado, sera generado un error y el operador .LOW. sera asumido.

Ejemplo:

```
7A 11 00      ITEM  DEFB 122,17,0
06 1F 42 1A         DATA 6,1FH,'A'+1,32Q
```

```
 ---------------------------------------------------
/
/
|    (etiqueta)  DEFB    operando1,(operando2,...)
|                DATA
|                DB
```

donde:

Etiqueta -Es un simbolo opcional el cual sera asignado con la direccion del primer byte definido.

Operando -Es una expresion evaluada conteniendo un byte o un caracter constante o una variable alfanumerica en ASCII o en EBCDIC de hasta 70 caracteres.

1.2.4.6 DEFW DW

Estos directivos informan al ensamblador la localizacion de dos bytes por operando. Cada operando es guardado en pares sucesivos de bytes. Estos son guardados con los 8 bits menos significativos en la primera direccion disponible y los 8 bits mas significativos en la siguiente direccion. Los valores negativos son codificados en complemento a dos.

Ejemplo:

```
-1B 00 28 00      ADD1   DEFW 1BH,24H+4,0
 00 00
```

```
E8 03 10 27              DW    1000,10000
```

```
 ----------------------------------------------
/
|  {etiqueta}   DEFW     operando1{,operando2...}
|               DW
```

donde:

Etiqueta - Es un simbolo opcional a el que se le asignara el valor de la direccion del primer byte codificado.

Operando -Es una expresion evaluable conteniendo dos bytes o una cadena de caracteres de longitud 1 o 2. Un total de 70 bytes pueden ser alojados en este directivo.

1.2.4.7 DDB

Este directivo es similar al directivo DEFW excepto por el orden en el que los valores de 16 bits de cada operando es guardado. Los 8 bits menos significativos del operando son guardados en el segundo byte y los 8 bits mas significativos son guardados en el primer byte. Los valores negativos son codificados en complemento a dos.

Nota!!! Este directivo no debe ser usado con operandos relocalizables debido a que se generaran errores al momento del cargado. El equivalente de "DDB exp" seria "DB .HIGH.exp,.LOW.exp"

Ejemplo:

```
03 E8 27 10    REV1     DDB  1000,10000
```

```
 ----------------------------------------------
/
|    {etiqueta}  DDB   operando1{,operando2,...}
```

donde:

Etiqueta -Es un simbolo el cual sera asignado con un valor igual a la direccion del primer byte codificado en esta expresion.

Operando -Es una expresion evaluable contenida en dos bytes.

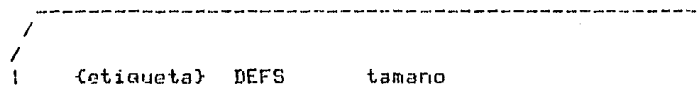
Un total de 70 bytes pueden se alojados por este directivo.

1.2.4.8 DEFS DS

Los directivos DEFS y DS son usados para reservar un bloque secuencial de bytes. Este directivo realmente causa que el contador de programa avance y de este modo el contenido de los bytes reservados es impredecible.

Ejemplo:

```
JAKE      DEFS   62H
```



donde:

Etiqueta -Es un simbolo opcional el cual sera asignado con la direccion de el primer byte alojado por este directivo.

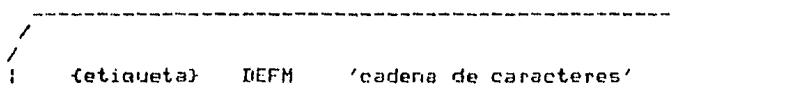
Tamano -Es un valor que especifica el numero de bytes que seran alojados por este directivo. Cualquier simbolo usado en este directivo debe ser previamente definido. La expresion final no puede contener terminos relocalizables.

1.2.4.9 DEFM

Este directivo es usado para definir hasta 70 bytes de cadenas de caracteres en ASCII o en EBCDIC. Este directivo es equivalente a usar el comando DEFB con la diferencia de que la cadena de caracteres es un solo operando.

Ejemplo:

```
DEFM   'SYSTEM SHUTDOWN'
```



donde:

Etiqueta -Es un simbolo opcional el cual sera asignado con la direccion de el primer byte alojado por este directivo.

Cadena de caracteres -Es una cadena de hasta 70 caracteres. Esta debe estar encerrada en comillas simples. Una comilla simple puede ser representada por dos de ellas seguidas. Puede estar precedido por una A para que se codifique en ASCII o una E para que lo haga en EBCDIC. Si no se da caracter alguno, la codificacion es asumida ASCII.

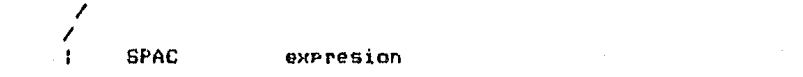
1.2.4.10 SPAC

Este directivo causa una o mas lineas en blanco en el listado de salida. Esto habilita al programador para formatear los listados y que estos sean de facil lectura. Este directivo no aparece en el listado.

El programador puede insertar estatutos en blanco para lograr el mismo efecto.

Ejemplo:

```
SPAC    7
```



donde:

Expresion -Es un valor el cual determina cuantas lineas en blanco apareceran en el listado. Este valor no puede ser relocalizable.

1.2.4.11 EJEC

Este directivo le indica al programa que debe saltar a la parte superior de la siguiente pagina. El proposito de esto es hacer listados de facil lectura. Algunos programadores prefieren empezar las subrutinas en nuevas paginas.

Ejemplo:

```
        ------------------------------------------------
       /
      /
      !       EJEC
```

1.2.4.12 TITLE

Este directivo es usado para imprimir encabezados en el comienzo de cada pagina de el listado. El encabezado default es "Z80 ASSEMBLER VER _._ MR.". Para que el usuario pueda usar encabezados propios, este directivo debe ser el primer estatuto en el programa.

Ejemplo:

```
                TITLE 'DISPLAY PROGRAM'
```

```
        ------------------------------------------------
       /
      /
      !       TITLE     'encabezado'
```

donde:

Encabezado -Es el titulo que sera puesto en el comienzo de cada pagina. Se pueden usar hasta 50 caracteres, los caracteres adicionales seran ignorados. El encabezado esta limitado por comillas simples, pero si no esta presente la comilla de terminacion, solo los primeros 50 caracteres son usados. El encabezado puede contener o no caracteres, en tal caso, el encabezado sera puesto en blanco.

Nota: El comando del ensamblador *HEADING S es similar al directivo TITLE con las siguientes diferencias:

-*HEADING tambien causa un cambio de pagina.
-El titulo impreso con *HEADING comienza con el primer

caracter no blanco en el operando.
-*HEADING no es impreso en el listado de salida.

1.2.4.13 LIST

El directivo puede ser usado para generar listados de elementos especificos. Los defaults en el programa son: El texto fuente, la tabla de simbolos, expansiones de macro y los estatutos condicionales que no fueron ensamblados. Un modulo objeto en formato relocalizable es producido y la tabla de simbolos no es puesta en el modulo objeto.

Los mensajes de errores son siempre listado sin hacer caso de los elementos especificados. En particular, la opcion E puede ser usado para crear un archivo separado consistente solo de los mensajes de errores.

Ejemplo:

```
LIST X,B                    El listado tiene
                            una tabla de referencia y pone la
                            tabla de simbolos en el modulo
                            objeto.
```

```
 ----------------------------------------------
/
/
|      LIST     (A,B,D,E,G,I,M,O,R,S,T,X)
```

donde:

A -Especifica que sera un ensamblado absoluto. Todos los directivos relocalizables son distinguidos como errores y el modulo objeto es dado en formato Hexadecimal Intel, esta opcion debera estar habilitada y debe especificarse antes de generar cualquier codigo objeto. Cuando es usada esta opcion, debera ser el unico argumento de este directivo.

B -Especifica que la tabla de simbolos sera colocada en el modulo objeto, absoluto o relocalizable, y puede ser usado para busqueda de errores.

D -Especifica que cualquier linea de datos, despues del primero, generado por DEFB, DB,DEFW, DW o DDB sera listado (Default).

E -Especifica que los errores seran listados en un archivo

aparte. Esta opcion es muy usual cuando el programador quiere un listado sin mensajes de error y estos aparecen en la terminal del usuario. Si solo se quiere ver los errores puede deshabilitar el listado y los errores apareceran en la terminal.

G -Especifica que cualquier simbolo generado sera listado en la tabla de simbolos o en la tabla de referencia y en el modulo objeto (si B es especificado).

I -Especifica que las instrucciones no ensambladas debido a condicionamientos seran listados (Default).

M -Especifica que la expansion de macros sera impresa en el listado (Default).

O -Especifica que se va a crear un modulo objeto (Default).

R -Especifica que el usuario debe sustraer el contador de programa, 'S', cuando se haga un direccionamiento relativo (Default).

S -Especifica que el texto fuente sera listado (Default)

T -Especifica que la tabla de simbolos sera listada (Default).

X -Especifica que la tabla de referencia sera listada. Este parametro deshabilita la opcion T si esta especificada.

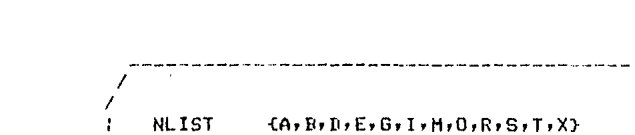
Nota. Si el usuario especifica las opciones B o G, estas deberan estar antes de generarse cualquier codigo objeto.

1.2.4.14 NLIST

El directivo NLIST instruye al programa para suprimir el listado de los elementos especificados. El listado puede ser habilitado por el directivo LIST. Los errores generados son siempre listados. Para generar un listado de errores solamente se debe especificar la opcion 'NLIST S' al principio del programa, el usuario puede usar la opcion 'E' posteriormente.

Ejemplo:

```
NLIST   O
```



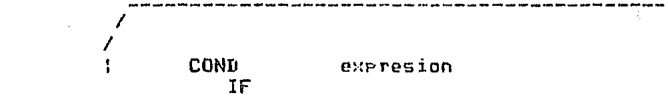
Las opciones especificadas tienen el mismo significado que en el directivo LIST, a excepcion que este directivo suprime los listados.

1.2.4.15 COND IF

Los directivos COND e IF pueden ser usados para condiconar ensambles del texto entre estos directivos y ELSE, ENDIF o ENDC. Cuando la expresion evaluada en el campo del operando es no-cero, el codigo sera ensamblado. Si la expresion evaluada es cero, el codigo no se ensamblara. Pueden hacerse anidamientos con estos directivos hasta 16 niveles.

Ejemplo:

```
COND    SYSTEM
IF      DATA.EQ.7FH
```



donde:

Expresion -Evalua la expresion la cual determina si se va a ensamblar o no los estatutos entre este directivo y el siguiente ELSE, ENDC o ENDIF. Cualquier simbolo usado en esta expresion debera estar previamente definida. La expresion no puede ser relocalizable.

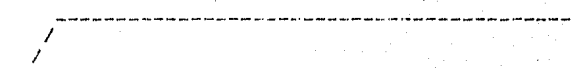
1.2.4.16 ELSE

El directivo ELSE es usado en conjuncion con los directivos IF o COND. Si la expresion en el operando de estos directivos fuese cero, todos los estatutos que siguen al ELSE son ensamblados hasta el siguiente ENDIF o ENDC.

El directivo ELSE es opcional y solo puede aparecer en un bloque IF-ENDIF o COND-ENDC.

Ejemplo:

```
            IF    MAIN
            .
            .
            ELSE
            .
            .
            ENDIF
```



```
      ELSE
```

1.2.4.17 ENDC ENDIF

Los directivos ENDC y ENDIF son usados para informar al ensamblador donde termina el codigo fuente que se sujeto a un condicionamiento. En el caso de anidamiento, el ENDC o ENDIF es apareado con el ultimo COND o IF.

Ejemplo:

```
                  IF SUM-4
ensamblado si     OR 200H
SUM-4 es no-      ADD A,VALOR
cero.             LD A,47
                  ELSE
ensamblado si     OR 07FH
SUM-4 es cero     ADD A,C
                  ENDIF
```

```
                ;-----------------------------------------------
               /
              /
             1      ENDIF
```

1.3 MACROS

Un MACRO es una secuencia de instrucciones que puede ser automaticamente insertada en el texto fuente del programa varias veces con el codificado de una sola instruccion, la llamada de MACRO. La definicion del macro es escrita solo una vez y puede ser llamada cuantas veces se requiera. Dicha definicion puede contener parametros a los que es posible cambiarlos en cada llamada. La facilidad del MACRO simplifica la codificacion de programas, reduce las posibilidades de errores del programador y permite hacer programas faciles de entender dando oportunidad de hacer cambios en una sola parte del programa.

Una definicion de MACRO consiste en tres partes: un encabezado, un cuerpo y un terminador; la definicion puede estar en cualquier lugar pero siempre antes del primer llamado. Un Macro puede ser redefinido en cualquier momento y la definicion mas reciente sera la que se use. Un mnemonico estandar del ensamblador, por ejemplo RIT, puede ser redefinido dandole este nombre a un MACRO; en este caso, los usos subsecuentes de esta instruccion en el programa causara una expansion de este ultimo.

1.3.1 ENCABEZADO DEL MACRO

El encabezado, el cual consiste en el directivo MACRO o MACR da al MACRO un nombre y define cualquier parametro formal.

```
 ------------------------------------------
/
/
| etiqueta    MACRO    (lista de parametros)
|             MACR
```

La etiqueta especifica el nombre del MACRO y puede ser cualquier simbolo definido por el usuario. Este nombre puede ser el mismo que el de un simbolo definido por el programa y el significado dependera del campo en donde se encuentre. Por ejemplo, TAB puede ser el nombre de un simbolo y a la vez de un MACRO.

Si el nombre del MACRO es identico al de un mnemonico o un directivo del ensamblador este sera redefinido por el MACRO; una vez que esto ha sucedido, no hay forma de regresar a la definicion

original, sin embargo, el MACRO puede ser redefinido con un nuevo cuerpo.

El campo del operando del directivo MACRO contiene el nombre de los parametros formales en el orden en el cual ellos aparecen en la llamada del MACRO. Cada parametro es separado por comas y debe empezar con un signo sharp (#). El parametro puede consistir de un texto arbitrario, por ejemplo #12XYZ. La lista de parametros es terminada por un blanco, un tab o un punto y coma. Los parametros son leidos de izquierda a derecha; asi pues, el usuario debe tener cuidado de no usar nombres de parametros los cuales sean prefijos uno de otro, por ejemplo #AB y #ABC.

1.3.2 CUERPO DEL MACRO

La primera linea de codigos despues del directivo MACRO el cual no es un directivo LOCAL es el inicio del cuerpo del MACRO. Estos estatutos son puestos en el archivo de MACROS (MCFILE, destruido cuando termina el programa) para su uso cuando es llamado. Durante una llamada un error sera generado si un MACRO es definido dentro de otro MACRO. Los estatutos del MACRO no son ensamblados cuando encuentra una definicion de este incluyendo directivos o comandos del ensamblador.

El nombre de un parametro formal especificado en el directivo MACRO puede aparecer en el cuerpo en cualquier campo. Si un parametro existe, es marcado y substituido por el parametro real cuando el MACRO es llamado. Los parametros pueden existir en cualquier campo dentro del cuerpo, incluso en comentarios. Un parametro formal dentro del cuerpo del MACRO es indicado por un sharp (#) igual que en el encabezado de este.

Para todas la definiciones de MACRO existe un parametro internamente definido indicado por ##YM; este parametro puede ser referenciado dentro del cuerpo del MACRO pero no debe aparecer en la lista de parametros formales. Cuando el MACRO es llamado, cada ocurrencia de ##YM en el cuerpo de este es reemplazado por una cadena de caracteres representando una constante hexadecimal de cuatro digitos. La cadena de 4 digitos es constante sobre un nivel dado de la expansion y se incrementa en uno en cada llamado. El uso tipico de ##YM es proveer etiquetas unicas a un MACRO que es expandido varias veces para evitar cometer el error de duplicarlas. Esto puede lograrse tambien con el uso del directivo LOCAL.

1.3.3 TERMINADOR DEL MACRO

El directivo ENDM termina la definicion del MACRO; durante la definicion de este, un ENDM debe encontrarse antes de otro directivo MACRO. Un directivo END que es encontrado durante la definicion de un MACRO terminara su definicion asi como el ensamblado total. El formato del directivo ENDM es el siguiente.

```
        ----------------------------------------
       /
      /
      !    (etiqueta)    ENDM
```

donde:

Etiqueta -Es una etiqueta opcional la cual la direccion simbolica del primer byte de memoria seguida de la insersion del MACRO.

1.3.4 LLAMADA DE MACRO

Un MACRO puede ser llamado codificando el nombre de este en el campo de operacion de un estatuto; el formato es el siguiente:

```
        ----------------------------------------
       /
      /
      !   (etiqueta) Nombre  (lista de parametros)
```

donde:

Etiqueta -Es una etiqueta opcional la cual sera asignada con un valor igual al del actual contador de programa.

Nombre -Es el nombre del MACRO llamado. Este nombre tiene que estar definido por el directivo MACRO anteriormente o un mensaje de error sera generado.

Lista de Parametros -Es una lista de parametros separados por comas. Estos parametros pueden ser constantes, simbolos, expresiones, cadenas de caracteres o cualquier otro texto separados por comas.

Los parametros en la llamada del MACRO son los actuales y sus nombres pueden ser diferentes que los parametros formales usados en la definicion de este. Los parametros actuales sustituiran a los parametros formales en el orden en que estan escritos. Las comas pueden ser usadas para reservar la posicion de un parametro. En este caso el parametro sera nulo, esto es, no contendra caracteres actuales; el parametro formal correspondiente a un nulo es removido simplemente en la expansion. Cualquier parametro que no se especifique sera nulo. La lista de parametros es terminada por un blanco, un tab o un punto y coma.

Todos los parametros actuales son pasados como cadenas de caracteres dentro de la definicion, y por nombre y no por valor. En otras palabras, si el valor de un simbolo es cambiado en el MACRO, su valor cambiara tambien fuera de este. Asi los directivos DEFL dentro del cuerpo del MACRO pueden alterar el valor de los parametros pasados a este.

Durante la expansion de macro, el ensamblador reconoce ciertos caracteres que tienen significado especial. El ampersand (&) es usado para concatenar el texto en la linea de definicion y cualquier parametro actual. Durante la expansion, un ampersand inmediatamente precedido o siguiendo a un parametro formal es removido y la sustitucion de este ocurre en ese momento. Si el ampersand no esta adyacente al parametro, no sera removido y permanece como parte de la linea de definicion. Ampersands dentro de cadenas de caracteres no son reconocidos como concatenadores.

Las combinaciones simples son usadas para delimitar los parametros actuales que pueden contener otros delimitadores. Todos los caracteres entre comillas son removidos antes de ser sustituidos en los parametros formales. Las comillas es el unico camino para pasar parametros que continen blancos, comas u otros delimitadores. Por ejemplo, para usar la instruccion 'LD HL,0' como un parametro actual requiere que se de como: 'LD HL,0' en la lista de parametros. Un parametro nulo puede consistir de comillas sin caracteres. Una comilla dentro del parametro es representada por dos comillas consecutivas.

Un ejemplo de una llamada de MACRO y su expansion es mostrada a continuacion. Note el uso de concatenacion y el parametro especial @$YM. El codigo expandido es marcado con signos mas (+).

```
Definicion:     GET       MACR    @X,@Y,@Z
                          LD B,$X&.AND.0FH
                          @Y
                  @Z      JP C,MAIN
                          ADD HL,HL
                L&@YM     SET 0,C
                          ADD A,C
                          ENDM
                           -
                           -
Llamado:                  SCF
                LOOP      GET 200,'INC B',ENTRY
```

```
                        JR NZ,GO
                        -

Expansion:              SFC
            LOOP        GET 200,'INC B',ENTRY
          +             LD B,200.AND.0FH
          +             INC B
          +   ENTRY     JP C,MAIN
          +             ADD HL,HL
          +   L001      SET 0,C
          +             ADD A,C
                        JR NZ,GO
```

1.3.5 DIRECTIVOS

Los directivos que se definen enseguida corresponde unicamente a los usados en la definicion del MACRO. Algunos de estos coinciden con los directivos del ensamblador, sin embargo son enfatizados aqui.

1.3.5.1 LOCAL

Debido a que todas las etiquetas, incluso las que se encuentran en los MACROS, son globales en todo el programa, un MACRO que las contiene y que es llamado varias veces puede generar errores por duplicacion de etiquetas. Para evitar este problema el usuario puede declarar etiquetas 'locales' dentro de los MACROS. Cada vez que se llame al MACRO , el ensamblador asignara a cada simbolo local un simbolo de la forma ??nnnn. De esta manera, la primera vez que sea llamado el MACRO el simbolo local sera ??0000, la segunda vez ??0001, etc. El ensamblador no empieza con ??0000 para cada MACRO, sino que incrementa el contador para cada simbolo encontrado. Los simbolos definidos por este directivo son tratados como parametros formales y de esta manera pueden ser usados en el campo del operando de las instrucciones. El campo del operando del directivo LOCAL puede no contener cualquier parametro formal definidos en- la linea del directivo MACRO. Se pueden usar tantos directivos LOCAL-como sea necesario pero deben estar despues del directivo MACRO y antes de la primera linea del cuerpo de este. Si se encuentra un directivo

LOCAL fuera de la definicion de un MACRO se generara un mensaje de error.

Ejemplo:

```
Definicion:        WAIT     MACRO  #TIME
                            LOCAL  #LAB1
                            LD B,#TIME
                   #LAB1    DEC B
                            JR NZ,#LAB1
                            ENDM

1a. Llamada: +              LD B,5
             +      ??0000  DEC B
             +              JR NZ,??0000

2a. Llamada: +              LD B,0FFH
             +      ??0001  DEC B
             +              JR NZ,??0001
```

Uso:

```
        ----------------------------------------
       /
      /
     |          LOCAL   lista de simbolos
```

donde:

Lista de simbolos -Es una lista de simbolos separados por comas que son definidas como locales en el MACRO.

1.3.5.2 EXITM

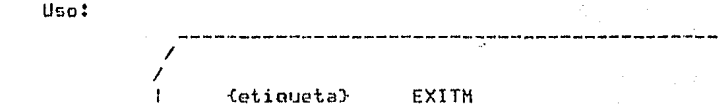
El directivo EXITM provee un metodo alternativo para terminar una expansion de MACRO. Durante una expansion, un directivo EXITM causa que esta cese hasta encontrar un ENDM y los codigos entre estos dos directivos son ignorados. Si los MACROS estan anidados, este directivo causara que la generacion de codigos regrese al nivel de expansion anterior. Note que un EXITM o ENDM puede ser usado para terminar una expansion, pero solo un ENDM termina la definicion.

Ejemplo:

```
STORE     MACRO #DATA
           -
           -
          IF #DATA
            EXITM
```

```
                    ENDIF
                     -
                     -
                    ENDM
```

Uso:



donde:

Etiqueta -Es una etiqueta opcional la cual tomara el valor de la direccion actual del contador de programa.

1.3.6 Parametro de Cuenta del MACRO

El simbolo especial NARG puede ser usado por aquellos usuarios que necesitan saber el numero de parametros pasados en la llamada del MACRO a este. Este simbolo es usado como cualquier otro simbolo y representa el numero actual de parametros pasados al MACRO, no de parametros formales definidos en este. El simbolo debera usarse solamente dentro del MACRO ya que fuera de este valdra cero (0). El uso tipico de este argumento es cuando se forman tablas dentro de los MACROS con ensambles condicionales. Esta cuenta representa solamente los parametros que no son nulos. No debe estar definido en el encabezado.

Ejemplo:

```
Definicion:   GEN  MACRO #P1,#P2,#P3
                   IF NARG
                   DEFB #P1
                   GEN #P2,#P3
                   ELSE
                   ENDIF
                   ENDM

Llamada:           GEN ADD1,ADD2

Expansion: + IF 2
           + DEFB ADD1
           + GEN ADD2,
           + IF 1
           + DEFB ADD2
           + GEN ,
           + IF 0
```

```
+ ELSE
+ ENDIF
+ ENDIF
+ ENDIF
```

1.4 RELOCALIZACION

El modulo objeto producido por el ensamblador puede estar en formato relocalizable. Esto permite al usuario escribir programas cuya direccion final pueda ser ajustada por el LINKER-LOADER y tambien que los modulos individuales sean cambiados sin necesidad de reensamblar el programa completo. Los modulos objeto separados pueden ser concatenados y cargados en un programa final.

La programacion relocalizable provee muchas ventajas al usuario. La direccion actual de memoria no es definida hasta el momento de la carga final. Los programas muy grandes pueden ser separados en pequenos segmentos, desarrollados en forma separada y posteriormente cargarlos juntos. Si un segmento contiene un error, solo este necesita ser reensamblado. Una vez desarrollada una libreria de rutinas, esta puede ser usada por muchos usuarios. El cargador ajustara la direccion para los requerimientos de cada usuario.

1.4.1 PROGRAMAS SEGMENTADOS

Para tomar ventaja de la relocabilidad, el usuario debe comprender el concepto de programas segmentados y como estos pueden ser cargados juntos. Un segmento de programa es una parte de este el cual contiene su propio contador de programa y es una seccion logica distinta del programa total. En el instante de la carga la direccion de cada segmento puede ser especificada por separado.

El ensamblador provee cuatro segmentos de programa distintos. El segmento de codigos (CSEG) es generalmente el segmento que contiene las instrucciones de maquina. En un sistema RAM/ROM este debe ser colocado en ROM. El area de datos es localizada usualmente en el segmento de datos (DSEG); generalmente reside en RAM. El segmento de datos puede contener intrucciones de maquina asi como el segmento de codigos puede contener datos.

Dos segmentos adicionales son provistos para facilitar la programacion. El segmento de stack puede ser usado para contener el area de stack del programa y reside en RAM. Generalmente solo el programa principal puede hacer referencias al segmento de stack y especificar su tamano. Pueden hacerse referencias a este segmento con el simbolo reservado "STACK". El segmento de memoria es aquel espacio de la porcion de memoria que no aloja los otros segmentos; las referencias a este segmento se pueden hacer con el simbolo reservado "MEMORY". Ambos segmentos son comunes a todos los programas y su direccion o tamano es definida generalmente al momento de la carga y no en el ensamblado.

Asi como los ensambladores no relocalizables, los usuarios pueden tambien especificar un ensamblado absoluto. En este caso, los modulos objeto aun cuando esten en formato relocalizable, contendran instrucciones o datos los cuales residiran en localizaciones especificas de memoria.

1.4.2 CARGADO

Los modulos objeto del ensamblador son combinados o cargados juntos por el LINKER-LOADER. El LOADER convierte todas las direcciones relocalizables en direcciones absolutas y resuelve referencias entre modulo y otro. El cargado entre modulos es provisto por definiciones externas (PUBLIC) y referencias externas (EXTRN). Las definiciones externas son definidas en un modulo objeto y son puestas como disponibles a todos los demas modulos por el LOAIER. Las referencias externas son simbolos referenciados en un modulo pero definidos en otro

El LINKER-LOADER combina las definiciones externas de un programa con las refernias externas de otros para obtener la direccion final. Un programa puede contener referencias y definiciones externas.

1.4.3 SIMBOLOS RELOCALIZABLES

Cada simbolo en el ensamblador esta asociado con un tipo el cual denota si es absoluto o relocalizable. Si es relocalizable, este tambien indica el segmento al que pertenece. Los simbolos que no dependen del origen del programa son llamados absolutos, los simbolos cuyo valor cambia al cambiar el origen del programa se les llama relocalizables. Los simbolos reservados 'STACK' y 'MEMORY' discutidos antes son considerados como relocalizables, a pesar de que el valor final pueda ser absoluto. Los simbolos locales y relocalizables pueden aparecer en programas absolutos o relocalizables. Note que los simbolos relocalizables pueden no aparecer en un programa cuando se especifica un 'LIST A'.

Los simbolos relocalizables son definidos como sigue:

1. Un simbolo en el campo de etiquetas de una instruccion cuando el programa se ensambla en un segmento de codigos o de datos relocalizable.

2. Un simbolo hecho igual a una expresion relocalizable por

EQU o DEFL.

3. Los simbolos reservados 'STACK' y 'MEMORY'.

4. Las referencias externas.

5. Una referncia al contador de programa ($) cuando el segmento se ensambla en forma relocalizable.

Los simbolos relocalizables son clasificados tambien como codigos, datos, stack o memoria dependiendo de como esten definidos.

1.4.4 EXPRESIONES RELOCALIZABLES

La relocabilidad de una expresion es determinada por la relocalizacion de los simbolos que ella compromete. Todas las constantes numericas son consideradas absolutas. Las expresiones relocalizables pueden ser combinadas para producir una expresion absoluta, relocalizable o incluso, bajo ciertas circunstancias, expresiones ilegales. La siguiente lista muestra aquellas expresiones cuyo resultado es relocalizable (ABS denota un simbolo absoluto, constante o expresion absoluta y REL denota un simbolo o expresion relocalizable).

```
ABS+REL   REL+ABS   REL-ABS  .LOW.REL   .HIGH.REL
```

Adicionalmente, las siguientes expresiones son validas y producen expresiones absolutas. Ambas subexpresiones deben ser relocalizables en el mismo segmento del programa.

```
REL.EQU.REL          REL.UGT.REL          REL.GT.REL
REL.LT.REL           REL.ULT.REL          REL-REL
```

Los simbolos relocalizables que aparecen en expresiones con cualquier otro operador causara un error, por ejemplo, REL*REL. Cualquier combinacion de dos simbolos relocalizables de difernetes segmentos, incluyendo referencias externas (EXTRN) es erronea.

1.4.5 DIRECTIVOS RELOCALIZABLES

Las siguientes paginas describen aquellos directivos en el ensamblador que permiten una primera relocalizacion. La nomenclatura es la misma que la descrita anteriormente. Los directivos son:

ASEG	Especifica Segmento Absoluto.
CSEG	Especifica Segmento de Codigos.
DSEG	Especifica Segmento de Datos.
ORG	Especifica Origen.
PUBLIC	Especifica Definicion Externa.
EXTRN	Especifica Referencia Externa.
NAME	Especifica Nombre del Modulo.
STKLN	Especifica la Longitud del Stack.

1.4.5.1 ASEG

El directivo ASEG especifica al ensamblador que los siguientes elementos deben ser ensamblados en modo absoluto. El ASEG permanece en efecto hasta que el directivo CSEG o DSEG es ensamblado. La direccion inicial para el contador de programa es cero. Al inicio del ensamble el ensamblador asume que un directivo ASEG ha sido especificado y el ensamble se hace en modo absoluto.

El usuario puede usar el directivo 'LIST A' para generar un listado absoluto con un modulo objeto absoluto.

Nota. Este directivo causa que un codigo absoluto sea generado dentro del programa relocalizable y por lo cual el modulo objeto esta en formato relocalizable.

Uso:

```
          ---------------------------------------------
         /
        /
        |    {etiqueta}       ASEG
```

donde:

Etiqueta -Es una etiqueta opcional que es asignada con la direccion actual del contador de programa antes de

ponerse en modo absoluto. Generalmente no es usada esta opcion.

1.4.5.2 CSEG

El directivo CSEG especifica al ensamblador que los siguientes estatutos deberan ser ensamblados en modo relocalizable usando el contador de programa del segmento codificado. Este segmento permanece en efecto hasta que otro directivo de segmento es ensamblado. Inicialmente, el contador de programa es cero.

Este directivo puede especificar un operando indicando el tipo de relocalizacion. Los tipos son pasados al LOADER y no tienen efecto en el ensamblado.

El usuario puede alternar entre varios segmentos con multiples directivos para segmentos dados dentro de un programa. El ensamblador mantendra el valor del actual contador de programa para cada segmento.

Ejemplo: CSEG P

Uso:

```
        ------------------------------------------------
       /
      /
     !   <etiqueta> CSEG  <espacio,P,I>
```

donde:

Etiqueta -Es una etiqueta opcional la cual sera asignada con la direccion actual del contador de programa.

Espacio -Especifica que el segmento del programa puede ser relocalizado en el siguiente byte disponible.

P -Especifica que el segmento de codigos debe empezar en la frontera de una pagina (0,100H,200H,etc.) cuando se relocalize por el LOADER.

I -Especifica que el segmento codificado debe estar codificado dentro de una sola pagina. El LOADER comenzara el segmento en la siguiente pagina si este no cabe en la actual.

Si se especifica varias veces el directivo CSEG en el ensamblador, cada uno debe especificar el mismo operando.

1.4.5.3 DSEG

El directivo DSEG especifica que los siguientes estatutos deben ser ensamblados en modo relocalizable usando el contador de programa del segmento de datos, el cual sera inicializado con cero (0). Este segmento permanece en efecto hasta que otro directivo de segmento sea ensamblado.

El directivo puede especificar un operando indicando el tipo de relocalizacion, este es pasado al LOADER y no tiene efecto en el ensamblado.

El usuario puede alternar entre varios segmentos con multiples directivos de relocalizacion para un segmento dado dentro de un programa. El ensamblador mantiene el valor actual del contador de programa para cada segmento.

Ejemplo: DSEG I

Uso:

```
    /
   /
   |   (etiqueta)  DSEG  (espacio,P,I)
```

donde:

Etiqueta -Especifica una etiqueta opcional la cual sera asignada con la direccion actual del contador de programa antes de empezar el segmento de datos.

Espacio -Especifica que el segmento de datos puede ser relocalizado con el siguiente byte disponible.

P -Especifica que el segmento de datos debe empezar en la frontera de una pagina (0,100H,200H,etc.) cuando se relocalice por el LOADER.

I -Especifica que el segmento de datos debe ser colocado en una sola pagina cuando se relocalice. El LOADER empezara el segmento en la frontera de la siguiente pagina si no cabe dentro de la actual.

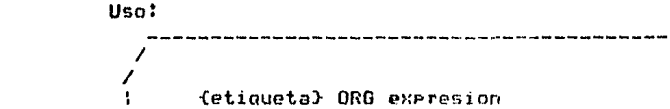
Si son especificados multiples directivos DSEG en el mismo ensamble, cada uno debe especificar el mismo operando.

1.4.5.4 ORG

El directivo ORG es usado para informar al ensamblador de la direccion de memoria la cual sera asignada al siguiente byte ensamblado. Este directivo cambia el contador de programa del segmento que esta siendo ensamblado. Cuando ORG se encuentra en un segmento relocalizable, la direccion dada puede ser una expresion absoluta o relocalizable, la cual a su vez sera relocalizada dentro del segmento.

Ejemplo: ORG $+30

Uso:



⟨etiqueta⟩ ORG expresion

donde:

Etiqueta -Es una etiqueta opcional la cual sera igualada a la expresion.

Expresion -Es un valor el cual reemplazara el contenido del contador de programa del segmento actual. Cualquier simbolo usado debe estar previamente definido.

1.4.5.5 PUBLIC

Este directivo especifica una lista de simbolos los cuales daran atributos y definicion externamente. Estos simbolos estaran disponibles a otros modulos para que el LOADER haga la concatenacion necesaria entre estos. Solo aquellos simbolos definidos por este directivo y definidos en el ensamble seran colocados en el modulo objeto.

Este directivo puede aparecer en cualquier lugar dentro del programa y cada simbolo debe estar definido solo una vez.

Los simbolos definidos con este directivo pero no definidos en el programa seran declarados indefinidos en el listado de salida.

Ejemplo: PUBLIC SCAN,LABEL,COSINE

Uso:

donde:

Lista de Simbolos -Es una lista de simbolos separados por comas la cual especifica los nombres definidos en este modulo y disponibles a otros.

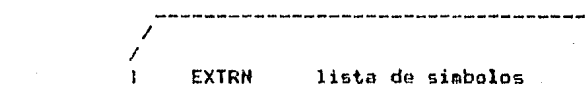
1.4.5.6 EXTRN

Este directivo especifica una lista de simbolos los cuales tendran atributos con referencia externas. Estos son simbolos que son referenciados en el programa pero definidos en otro modulo. Este directivo provee el enlace entre dos o mas modulos a travez del LOADER.

Este directivo puede aparecer en cualquier parte del programa y cada simbolo debe ser declarado solo una vez.

Ejemplo: EXTRN INPUT,OUTPUT

Uso:



donde:

Lista de simbolos -Es una lista de simbolos separados por comas la cual especifica los nombres de las variables externas disponibles en otros modulos.

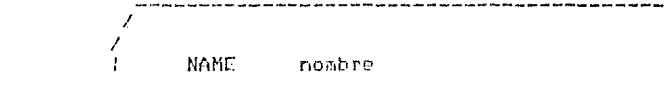
1.4.5.7 NAME

El directivo NAME es usado para asignar un nombre al modulo objeto producido por el ensamblador. Solo un directivo NAME debe aparecer en el programa. El nombre del modulo es usado por el LOADER cuando combina programas y dicho nombre aparece en el mapa de carga.

Si el directivo NAME no es especificado por el usuario, sera usado el nombre 'MODULE'.

Ejemplo: NAME MULT

Uso:



donde:

Nombre -Es el nombre que sera puesto en el modulo objeto denotando el nombre de este al LOADER. Este nombre debe seguir las reglas de los simbolos.

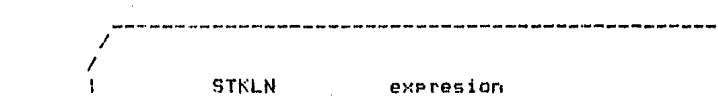
1.4.5.8 STKLN

El directivo STKLN especifica el tamano del segmento de stack generado por el LOADER. Generalmente este directivo solo es usado en el programa principal, pero otros programas pueden definirlo.

Si el usuario no especifica este directivo, el ensamblador asume la longitud del stack como cero (0). Mas de un directivo STKLN puede ser puesto en el programa, pero solo la ultima definicion es la usada.

Ejemplo: STKLN 20H

Uso:



donde:

Expresion -Es una expresion la cual indica el tamano del segmento de stack. Esta expresion puede no ser relocalizable y todos los simbolos usados en la expresion deben estar previamente definidos.

1.5 USO DEL ENSAMBLADOR

Para hacer uso del Macroensamblador Z80 relocalizable de la PRIME es necesario:

a) Escribir un programa utilizando mnemonicos y directivos, codificando los argumentos con constantes, etiquetas, direcciones simbolicas, etc.

b) Transferir el programa fuente a un archivo utilizando el EDITOR de la PRIME.

c) Ejecutar el programa invocandolo a nivel PRIMOS con el comando MACRO.

El programa pedira el nombre del archivo creado de la forma:

---------- ARCHIVO DE ENTRADA:

este archivo debe existir, de lo contrario volvera a pedirlo. Posteriormente pide el nombre del listado de salida de la forma:

---------- ARCHIVO DE SALIDA :

que es el archivo que contendra la informacion acerca del ensamblado, mensajes de error, simbolos no definidos, tablas de referencia, etc. Este archivo no debe existir previamente o de lo contrario volvera a pedirlo, esto con el fin de evitar el borrar archivos por error. Por ultimo pide el nombre del archivo objeto de salida de la forma:

---------- ARCHIVO OBJETO :

en donde el ensamblador colocara el archivo objeto de salida para su utilizacion directa o bien para su posterior cargado a travez del LOADER.

Al terminar el programa dara el mensaje de STOP y podra entonces revisar el listado de salida y, en su caso, corregir el programa fuente para volver a ensamblarlo.

MACROENSAMBLADOR Z80

1.0 INTERPRETACION DE LISTADOS

Durante la segunda pasada del proceso de ensamblado, un listado es producido. El listado contiene informacion pertinente al programa ensamblado, los estatutos fuentes de los mnemonicos y datos que el usuario dio.

El listado puede ser usado como herramienta documentativa a travez de la inclusion de comentarios que describan el funcionamiento particular del programa.

El proposito principal del listado es llevar toda la informacion pertinente acerca del ensamblado del programa, esto es, las direcciones de memoria y sus valores ya que el modulo objeto, aunque tiene la misma informacion, se encuentra en un formato no facil de leer.

El listado en el apendice A es un ejemplo tipico de un programa ensamblado. Refiriendose a este apendice se da la siguiente informacion.

- Cuando el ensamblador detecta condiciones de error durante el proceso de ensamblado, la columna titulada ERR contendra el o los codigos describiendo el tipo de error asociado a la linea correspondiente. Una explicacion de los codigos de error individual es proporcionada en el apendice B.

- La columna titulada LINE contiene numeros decimales los cuales estan asociados con las lineas del listado fuente. Estos numeros son usados en las tablas de referencia.

- La columna titulada ADDR contiene un valor el cual representa la primera direccion de memoria de los datos mostrados en bytes en la linea o bien el valor de un directivo EQU o DEFL. El numero hexadecimal bajo B1 representa un byte de datos guardado en la direccion de memoria. Si hay un dato bajo B2, este representa un byte guardado en la direccion de memoria dada anteriormente mas uno. Las columnas B3 y B4, si contienen valores, representan datos similares que seran guardados en la misma direccion de memoria mas dos y mas tres respectivamente.

- A la derecha de los bytes de datos se encuentran los tipos de relocalizacion de cualquier operando relocalizable. Los tipos son catalogados por las siguientes letras: C=Codigos, D=Datos, S=Stack, M=Memoria, E=Externa.

- Los estatutos orisinales del usuario son reproducidos a la derecha de la informacion anterior. Las expansiones de MACROS son precedidas por un signo mas (+).

- Al final del listado el ensamblador muestra el mensaje "ASSEMBLER ERRORS =" con un contador acumulativo de errores. El ensamblador sustituye con cuatro bytes de NOP cuando no encuentra un mnemonico particular y asi provee lugar para corregir el

programa manualmente si es posible.

- Una tabla de simbolos o tabla de referencia es generada al final del listado. La tabla lista todos los simbolos utilizados en orden alfabetico con cualquier tipo de relocalizacion como se describio anteriormente.

LINKER-LOADER

2 LINKER-LOADER

2.1 INTRODUCCION

Este manual describe el LINKER-LOADER del Z80 que complementa el Macroensamblador Z80 Relocalizable. El LOADER puede ser usado para combinar varios modulos objeto relocalizables independientemente ensamblados. Las referencias externas entre los modulos son resueltas y cada simbolo es sustituido por su valor final.

El LOADER no solo provee el cargado de varios modulos y ajusta las direcciones relocalizables en direcciones absolutas, sino que ademas permite que las direcciones de los segmentos sean especificadas, las definiciones externas sean definidas y el orden del cargado sea especificado.

LINKER-LOADER

2.2 CARGADO

Muchos programas son tan grandes que es conveniente ensamblarlos en modulos sencillos, para evitar tiempos muy largos de proceso o para reducir el tamano de las tablas de simbolos. Estos programas pueden dividirse en pequenos segmentos, ensamblarlos en forma separada y conjuntarlos posteriormente con el LOADER. Despues de que los segmentos han sido cargados y conjuntados, el modulo de salida aparece como si se hubiese generado por un solo ensamble.

Las funciones primarias del LINKER-LOADER son:

1. Resolver referencias externas entre modulos y checar las referencias no definidas.

2. Ajustar todas las direcciones relocalizables para su propio direccionamiento absoluto.

3. Proporcionar un modulo objeto final.

2.3 SEGMENTOS

Para entender el proceso de cargado y habilitar al usuario del Ensamblador y del LINKER-LOADER (Referido como LOADER en adelante) en forma efectiva, el usuario debe comprender los tipos de segmentos que componen un programa y las direcciones de estos a cargar. A pesar de que son descritos en el manual del ensamblador, a continuacion se da un descripcion de cada uno de ellos.

SEGMENTO ABSOLUTO

Este segmento es la parte del programa en ensamblador que contiene informacion no relocalizable pero que puede ser cargado en localidades fijas de memoria. Los codigos absolutos son colocados en el modulo de salida tal como se lee de los archivos objeto de entrada.

SEGMENTO DE CODIGOS

El Segmento de codigos contiene la parte del programa el cual comprende las instrucciones de maquina actuales y que generalmente residira en ROM. Las instrucciones en el segmento de codigos pueden hacer referencia a cualquier otro programa.

SEGMENTO DE DATOS

Los segmentos de datos contienen especificaciones para las partes del programa del usuario que generalmente son datos obtenidos en el momento de correrlo y que usualmente residen en RAM. No existen restricciones en este segmento para contener instrucciones ejecutables.

SEGMENTO DE STACK

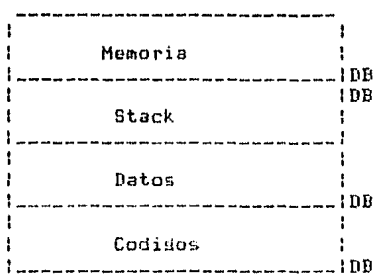
El segmento de STACK es usado tal como el programa del usuario lo requiera y su comportamiento es similar al del STACK del Z80.

SEGMENTO DE MEMORIA

El segmento de memoria es generalmente la direccion mas alta de la porcion de memoria la cual no aloja los otros segmentos. Las tablas de datos pueden ser expandidas dentro del segmento de memoria pero el ensamblador no tiene la facilidad de causar que las instrucciones se carguen dentro de este segmento. El inicio del segmento de memoria es determinada en el momento de la carga.

El LOADER permite al usuario cargar los segmentos en un programa continuo o especificando la direccion inicial (Direccion Base) de cualquiera de los segmentos. El orden en el cual los segmentos son colocados en la memoria puede ser especificado tambien. La organizacion predefinida usada por el LOADER es la siguiente:

Direccion alta



DB = Direccion Base

Esta es una organizacion de memoria tipica en la mayor parte de los programas, pero algunos usuarios prefieren colocar el segmento de STACK despues del segmento de codigos, de esta manera el segmento de datos puede expanderse dentro del segmento de memoria durante la ejecucion del programa.

La direccion base (DB) para todos los segmentos, excepto para el segmento de STACK, es la direccion mas baja de estos. Cuando un usuario especifica la direccion inicial de un segmento con comandos del LOADER, esta es la direccion base especificada. La direccion base del segmento de STACK se mueve tipicamente hacia las direcciones mas bajas durante la ejecucion del programa.

2.4 TIPOS DE RELOCALIZACION

Los tipos de relocalizacion (paginado) de cualquier segmento del programa es determinado en el ensamblador, por el directivo CSEG o DSEG; estos pueden ser especificados tambien por comandos del LOADER. El efecto de los tres tipos de relocalizacion en el LOADER se da a continuacion:

Relocalizacion de Byte

Este implica que no fue especificado un operando en el directivo de segmento en el ensamblador. En este caso, el segmento del modulo objeto sera localizado inmediatamente despues del mismo segmento desde el modulo objeto anterior y esto causara que no sean desperdiciadas porciones de memoria.

Relocalizacion Paginizada

Este tipo de relocalizacion es especificado por el operando "P" en el directivo del ensamblador o en el comando del LOADER. Esto implica que el segmento del programa deba empezar en la frontera de la siguiente pagina disponible, esto es 0,100H,200H,etc. despues del modulo objeto anterior.

Relocalizacion No Paginizada

Este es especificada por el operando "I" en el directivo apropiado del ensamblador o del LOADER. Esto implica que el segmento del programa debe residir en una pagina si es posible. Si el LOADER determina que no cabe en la pagina corriente, este empieza el segmento en la siguiente pagina como si fuera un "P" relocalizable.

En la secuencia tipica de cargado, el LOADER coloca todos los segmentos de codigo contiguos en la memoria seguidos por los segmentos de datos, no existen bytes adicionales entre estos segmentos. Sin embargo, si cualquier segmento de datos especifica una relocalizacion que no sea de byte, el LOADER lo colocara en la frontera de la localidad reservada. Para evitar cualquier desperdicio de memoria el usuario puede siempre definir la direccion inicial. El mismo problema existe si el segmento de codigos sigue al segmento de datos y el primero tiene definido un tipo de relocalizacion paginizada o no paginizada.

Cuando se esta desarrollando inicialmente un programa y se esta probando, es recomendable especificar cada segmento en cada ensamblado como relocalizacion paginizada; esto forza que la direccion inicial de cada modulo a terminar en 00H y hace mas facil al usuario seguir el flujo del programa. En este caso la salida del ensamblador contiene la direccion de memoria exacto, excepto por un offset que sera adicionado al byte alto de la direccion. La relocalizacion paginizada puede ser especificada tambien en el LOADER por los comandos CPAGE o DPAGE.

LINKER-LOADER

2.5 COMANDOS

El LOADER lee una secuencia de comandos del dispositivo de entrada (archivo) y la ultima linea debe ser un EXIT o un END.

Los modulos objeto son leidos del dispositivo de entrada o de archivos especificados por el comando LOAD.

La salida del LOADER consiste en un modulo absoluto cargable en el microprocesador en uso. El modulo de salida es escrito en el formato INTEL hexadecimal.

Todos los comando deben comenzar en la column uno (1). Los argumentos de los comandos pueden empezar en cualquier columna y deben estar separados de estos por un espacio al menos. Los comentarios pueden colocarse en cualquier lugar del archivo indicandolos con un asterisco en la columna uno (1).

Las siguientes paginas describen los comandos del LOADER. En la descripcion de los comandos, las llaves {} indican opcionalidad en los argumentos. Un sumario de los comandos es dado a continuacion:

CODE	Da la direccion base del segmento de codigos.
DATA	Da la direccion base del segmento de datos.
STACK	Da la direccion base del segmento de Stack.
MEMORY	Da la direccion base del segmento de Memoria.
CPAGE	Relocaliza para un segmento de codigos.
DPAGE	Relocaliza para un segmento de datos.
ORDER	Especifica el orden de los segmentos.
START	Especifica la direccion inicial del modulo de salid a.
STKLN	Especifica el tamano del Stack.
NAME	Especifica el nombre del modulo de salida.
LOAD	Especifica modulos objeto cargables.
PUBLIC	Especifica la definicion de simbolos.
LIST	Especifica los elementos a listar.
NLIST	Especifica los elementos que no se listaran.
EXIT	Termino de Sesion.
END	Termino de comandos y carga.
*	Comentario.

Los argumentos de los comandos que son numericos pueden ser dados en decimal o hexadecimal. Las constantes hexadecimales deben ser terminadas con una "H" y no necesitan empezar con "0" a menos que el primer digito sea un caracter hexadecimal de la "A" a la "F".

Los comandos pueden ser leidos en cualquier orden y el mismo comando puede darse mas de una vez. El ultimo uso del comando determina los parametros de este. Los comandos pueden ser puestos

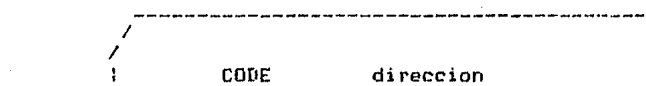
antes o despues del comando LOAD excepto por CODE, DATA, STACK y MEMORY los cuales deben preceder al primer LOAD en el archivo.

2.5.1 CODE

El comando CODE es usado para especificar la direccion inicial del segmento de codigos relocalizable. Si no es especificado, la direccion inicial es asumida cero (0) o empezando despues del segmento precedente, sujeto a cualquier tipo de relocalizacion, si este no es el primer segmento en la memoria. Este comando debe especificarse antes del primer comando LOAD.

Ejemplo: CODE 400H

Uso:



donde:

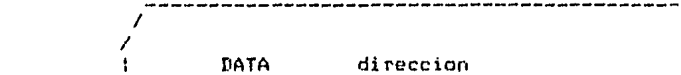
Direccion -Especifica la direccion base del segmento de codigos.

2.5.2 DATA

El comando DATA es usado para especificar la direccion inicial del segmento de datos relocalizable. Si no es especificado, la direccion inicial es asumida cero (0) o empezando despues del segmento precedente, sujeto a cualquier tipo de relocalizacion, si este no es el primer segmento en la memoria. Este comando debe especificarse antes del primer comando LOAD.

Ejemplo: DATA 1000H

Uso:



donde:

Direccion -Especifica la direccion inicial del segmento de datos.

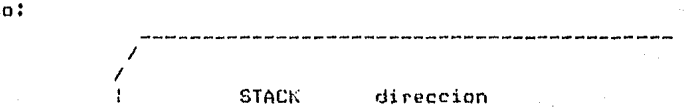
2.5.3 STACK

Este comando es usado para especificar la direccion inicial del segmento de Stack; la longitud de este es especificado por el comando STKLN en el ensamblador o en el LOADER. Si la direccion del Stack no es especificada, esta se iniciara inmediatamente despues del segmento en la memoria precedente o empezara en cero (0) si es el primer segmento. Este comando debe especificarse antes del primer comando LOAD.

Nota. La direccion inicial especificada en este comando es la parte alta (tope) del segmento de Stack.

Ejemplo: STACK 1FFH

Uso:



donde:

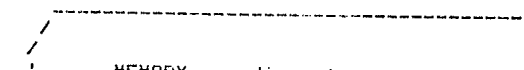
Direccion -Especifica la direccion inicial del segmento de Stack.

2.5.4 MEMORY

El comando MEMORY es usado para especificar la direccion inicial del segmento de memoria. La longitud de este segmento sera especificada como cero (0) en el mapa de memoria, pero la longitud se actualiza con la memoria disponible remanente en el sistema del usuario despues de que los otros segmentos se han cargado. Si no se especifica, la direccion inicial sera la inmediata despues del anterior segmento o cero (0) si es el primero. Este comando debe especificarse antes del primer comando LOAD.

Ejemplo: MEMORY 8000H

Uso:



MEMORY direccion

donde:

Direccion -Especifica la direccion inicial del segmento de memoria.

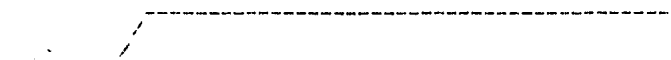
2.5.5 CPAGE

Este comando puede ser usado para modificar los tipos de relocalizacion de los segmentos de codigo en los modulos objeto de entrada. Como se explico en los tipos de relocalizacion, el ensamblador le indica al LOADER el tipo especifico (byte, paginizada, no paginizada) en cada segmento de cada modulo objeto. Este comando permite al usuario redefinir el tipo de relocalizacion especificado por el ensamblador.

El uso especifico de este comando es permitir al usuario empezar cada modulo en la frontera de una pagina, para propositos de rastreo de errores, y entonces especificar el programa final con una relocalizacion de Byte, paginizada o no paginizada o bien especificada por el ensamblador. Este comando puede ser cambiado para cada modulo leido por el LOADER. El ultimo uso de CPAGE determina el tipo de relocalizacion.

Ejemplo: CPAGE P

Uso:



donde:

Espacio -Especifica que el tipo de relocalizacion esta dada por el ensamblador (Default).

B -Especifica relocalizacion de byte.

P -Especifica relocalizacion paginizada.

I -Especifica relocalizacion no paginizada.

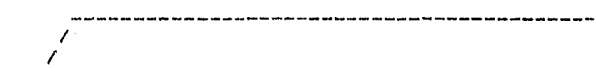
2.5.6 DPAGE

Este comando puede ser usado para modificar el tipo de relocalizacion para segmentos de datos en los modulos objeto de entrada. Este comando es usado de la misma manera que el comando CPAGE y permite al usuario especificar el tipo de relocalizacion de cada modulo, de byte, paginizada o no paginizada, o bien definida por el ensamblador.

Este comando puede ser cambiado para cada modulo leido por el LOADER. El ultimo uso de este comando especifica sus parametros.

Ejemplo: DPAGE

Uso:



donde:

Espacio -Especifica que el tipo de relocalizacion esta definida por el ensamblador (Default).

B -Especifica relocalizacion de byte.

P -Especifica relocalizacion paginizada.

I -Especifica relocalizacion no paginizada.

2.5.7 ORDER

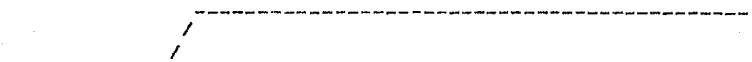
Como se describio en la operacion del LOADER, el orden normal de los segmentos de memoria es: Codigo, Datos, Stack y Memoria. El comando ORDER es provisto para aquellos usuarios que no necesitan especificar la direccion inicial pero si el orden de los segmentos en la memoria en forma diferente.

Si el usuario especifica la direccion inicial de los segmentos, el orden de estos puede no tener particular importancia. Sin embargo, es bueno recordar que si se especifica la direccion inicial para cierto segmento, los subsecuentes seran cargados en forma progresiva. Asi, si el usuario lo desea, el segmento de datos puede residir en la localidad 8000H y todos los otros segmentos puestos juntos en la porcion de memoria mas baja; pero el segmento de datos tendra que ser el ultimo parametro de los argumentos del comando. Si este no es especificado asi, entoces cualquier segmento cargado despues del segmento de datos residira en la parte alta de la memoria despues de este.

El usuario especifica el orden de los segmentos con letras representativas de estos, separados por comas. Todos los segmentos deben ser especificados en el comando o un mensaje de error sera generado ignorandose el comando.

Ejemplo: ORDER D,C,S,M

Uso:



donde:

Seg -Especifica uno de los cuatro segmentos como sigue:

C -Segmento de Codigos

D -Segmento de Datos

M -Segmento de Memoria

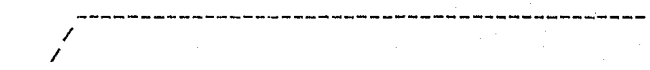
S -Segmento de Stack

2.5.8 START

Este comando es usado para especificar la direccion inicial que sera puesta en el record terminador del modulo objeto. Si no se especifica, la direccion inicial es obtenida del record final del programa principal de los modulos objeto de entrada. En caso de no haberse leido el programa principal, la direccion inicial sera cero (0).

EJemplo: START 7FCH

Uso:



donde:

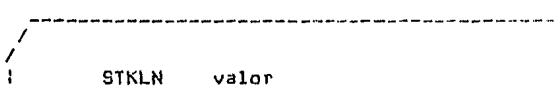
Valor - Especifica la direccion inicial a usar en el modulo objeto de salida.

2.5.9 STKLN

El comando STKLN es usado para especificar la longitud del segmento de Stack al LOADER. Si no se especifica, la longitud de este es determinada por el tamano dado por las definiciones en los modulos objeto de entrada.

EJemplo: STKLN 20H

Uso:



STKLN valor

donde:

Valor -Especifica la longitud del segmento de Stack.

2.5.10 NAME

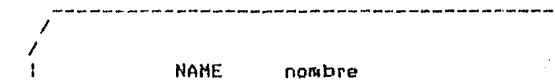
Especifica el nombre del modulo objeto de salida final. Generalmente este comando no realiza ninguna funcion sobre este modulo, el cual esta escrito en formato hexadecimal INTEL que no contiene el nombre. Este directivo es usado cuando el modulo objeto de salida puede ser relocalizable.

El nombre especificado por el usuario puede ser cualquier simbolo estandar de hasta 6 caracteres. Si el usuario no especifica nombre, este sera tomado del primer modulo objeto de entrada.

Este nombre no se refiere al del archivo de salida o el objeto, estos son dados en el inicio de la sesion con el LOADER.

Ejemplo: NAME LECTOR

Uso:



NAME nombre

donde:

Nombre -Es un simbolo que especifica el nombre del modulo objeto de salida.

LINKER-LOADER

2.5.11 LOAD

El comando LOAD es usado para especificar uno o mas modulos objeto a ser cargados. Si el operando del comando es un numero, este es asumido como un dispositivo logico de entrada/salida (lectora de cinta perforada por ejemplo). Si el operando no es un numero, se asume que se dio el nombre de un archivo y que el modulo objeto sera leido de este. Si a cualquier operando le precede un signo menos, esto indica que varios modulos seran leidos del dispositivo o del archivo hasta encontrar la marca de fin de archivo. En este caso, el usuario no necesita especificar un nombre para cada modulo objeto.

Los modulos objeto pueden ser leidos en combinaciones de archivos y dispositivos de entrada/salida y pueden o no ser leidos hasta el fin de archivo. Los modulos objeto son cargados en el orden especificado y cada uno de ellos es colocado inmediatamente despues del anterior o bien como se haya especificado la relocalizacion. El usuario puede usar tantos comandos LOAD como necesite.

Ejemplo: LOAD 7,-FILE1

Uso:

```
        ------------------------------------
       /
      /
     !      LOAD    modulo1{,modulo2,...,moduloN}
```

donde:

Modulo -Especifica el numero de unidad logica del dispositivo de entrada o el nombre del archivo en el cual el modulo objeto reside. Cualquier modulo especificado con un signo menos causara que se lea hasta encontra la marca de fin de archivo. Los modulos son separados con comas.

Nota -El LOADER disponible en PRIME no tiene habilitada la facilidad de leer de unidades logicas debido a que no tiene dispositivos de entrada salida viables como lectora de tarjetas o de cinta perforada.

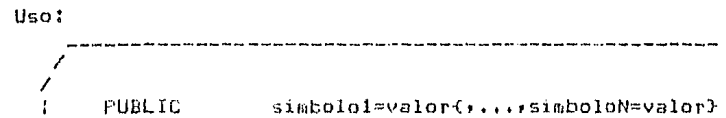
2.5.12 PUBLIC

Este comando es usado para definir y/o cambiar el valor de una definicion externa. Si un simbolo especificado por este comando tiene una definicion externa (Definido por el directivo PUBLIC del ensamblador en el modulo objeto de entrada), el valor del simbolo es cambiado al especificado por el usuario. Si el simbolo no esta definido, este sera colocado en la tabla de simbolos del LOADER con el valor especificado y estara disponible para satisfacer referencias externas de los modulos.

Este comando permite al usuario especificar el valor de algunos simbolos externos al momento de cargar los modulos y la posibilidad de evitar un reensamblado. Para cambiar el valor de un simbolo que se encuentra en una definicion en el modulo objeto, este comando debe especificarse antes de ser cargado por el comando LOAD. Puede ser colocado inmediatamente antes del comando END.

Ejemplo: INPUT=2FH,OUTPUT=0ACH

Uso:



donde:

Simbolo -Es un simbolo externo definido por el usuario.

Valor -Es el valor que sera asignado al simbolo.

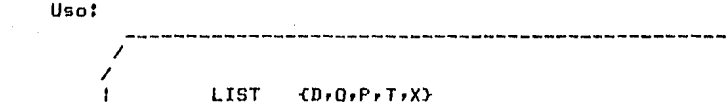
2.5.13 LIST

El comando LIST puede ser usado para generar listados de los elementos especificados. Los defaults son: Tabla de simbolos no listado, modulo objeto producido, los simbolos no son colocados en el modulo objeto de salida y los simbolos locales no son pursados de los modulos objeto de entrada. Los simbolos locales son los colocados en el modulo objeto por el ensamblador que no sòn definiciones externas. El usuario debe notar que colocando definiciones externas y locales en el modulo objeto de salida puede causar la repeticion de simbolos. Tipicamente solo los simbolos locales puestos por el ensamblador o las definiciones

externas seran colocadas en el modulo objeto de salida. Es por esto que el directivo LIST B en el ensamblador forma una tabla de simbolos donde incluye a las definiciones locales y externas.

Ejemplo: LIST T,X

Uso:



donde:

D -Especifica que las definiciones extenas seran puestas en el modulo objeto de salida.

O -Especifica que un modulo objeto sera producido (Default).

P -Especifica que cualquier simbolo local en el modulo de entrada sera puesto en la tabla de simbolos del LOADER (Default).

S -Especifica que la tabla de simbolos locales sera escrita en el modulo objeto y puede ser usado para rastreo de errores.

T -Especifica que la tabla de simbolos sera listada en el archivo de salida.

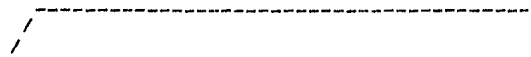
X -Especifica que la tabla de definiciones externas sera listado en el archivo de salida.

2.5.14 NLIST

El comando NLIST es el contrario del comando LIST y es usado para suprimir el listado de los elemntos especificados en el argumento. Los elementos pueden ser puestos a listar nuevamente usando LIST.

Ejemplo: NLIST O

Uso:



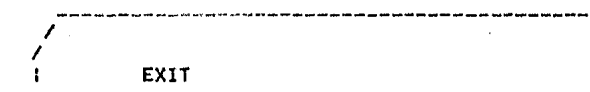
donde:

D,O,P,S,T,X -Tienen el significado opuesto al dado en el comando LIST. Los elementos que son default en el comando anterior, no los son en este y viceversa.

2.5.15 EXIT

Este comando es usado cuando se esta en modo interactivo (no habilitado en PRIME) para terminar una sesion. Este comando es muy usado cuando el usuario encuentra un error y requiere regresar al sistema operativo para corregirlo. Este actua como el comando END excepto que la carga final no toma lugar y no es producido el modulo objeto.

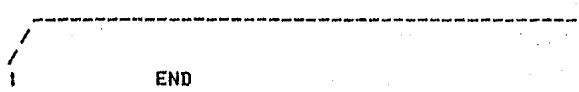
Uso:



2.5.16 END

Este debe ser el ultimo comando en cada archivo de entrada. Inicia los pasos finales del proceso de carga.

Uso:

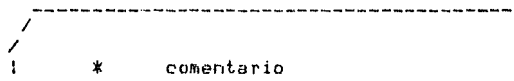


2.5.17 * (Comentario)

Un asterisco puede ser usado para especificar un comentario en el archivo de entrada. Este debe estar en la columna uno (1).

Ejemplo: * ESTE ES UN COMENTARIO

Uso:



2.6 USO DEL LOADER

Para hacer uso del LINK-LOADER de la PRIME es necesario tener:

a) Uno o mas archivos objeto generados por el ensamblador o bien por el LOADER.

b) Un archivo con comandos para el LOADER en donde se indique cuales archivos seran cargados y la forma en que se hara.

El programa es invocado como comando de la PRIME mediante el tecleo de LKLD. Esto hace que el programa corra en la cuenta especifica del usuario. El programa empieza preguntando por un archivo de entrada:

--------- ARCHIVO DE ENTRADA:

que es precisamente el archivo de comandos formado por el usuario. Este debe ser un archivo existente o de lo contrario volvera a pedirse. Posteriormente pide el nombre que tendra el listado de salida:

-------- ARCHIVO DE SALIDA :

que es el archivo donde el LOADER coloca los resultados y mensajes de error de la carga en caso de haberlos, asi como las tablas de simbolos si son pedidas; este archivo no debe existir o de lo contrario se pedira nuevamente, esto con el el fin de no borrar o reescribir archivos por error. Por ultimo pide el nombre del archivo objeto de salida:

---------- ARCHIVO OBJETO :

en donde el LOADER colocara el modulo objeto de salida; este archivo tampoco debe existir o volvera a pedirse. Al terminar dara un mensaje de STOP. En el listado de salida se podra ver los resultados de la carga. En caso de resultar sucesiva y sin errores aparecera el mensaje siguiente en dicho archivo:

LOAD COMPLETE

En la PRIME no estan implementadas las lecturas en

dispositivos de Entrada/Salida por no haber ninguno apropiado como lo seria la lectora de tarjetas o cinta perforada.

Asimismo no esta implementado el modo interactivo por ser poco usual, sin embargo puede hacerse uso de otros sistemas existentes como el BATCH para complementarlo.

1 EL ESTANDAR RS-232C.

El estandar RS-232C es uno de los diversos estandars o recomendaciones encaminadas a facilitar la conexion entre computadora y terminal, modems o redes de computadoras. Actualmente el estandar RS-232C es uno de los mas populares utilizados para conectar computadoras a modems y a terminales. El titulo oficial para este estandar es: Interfase entre equipo terminal de datos y equipo de datos con circuito-terminal utilizando interface binaria en serie. La C en RS-232C indica que ha sido revisado. Este estandar incluye muchos otros aspectos ademas de los cables de transmision y recepcion de datos que se utilizan para conectar una terminal a una computadora.

El estandar RS-232C consta de cuatro partes principales:

a) Caracteristicas electricas de la senal.
b) Caracteristicas mecanicas de la interface.
c) Descripcion funcional de las senales.
d) Una lista de subconjuntos de senales estandars para interfaces de tipo especifico.

La primera parte define los voltajes a ser utilizados y sus interpretaciones como ceros y unos. La segunda especifica el tamano del conector y la disposicion de las terminales. La tercera, la cual se describira posteriormente con mas detalle, proporciona una descripcion funcional de las 21 senales que conforman el estandar RS-232C, y la cuarta parte, enlista

alrededor de 14 subgrupos de estas 21 senales que son utilizadas en diferentes tipos de modems.

El estandar RS-232C fue establecido por una organizacion de los estados unidos, la asociacion de industrias de la electronica (EIA), y son casi identicas a las recomendaciones V.24 del CCITT (Comite Consultatif International Telephonique Et Telegraphique), Comite de la Union Internacional de Telecomunicaciones, el cual es una agencia de la Organizacion de las Naciones Unidas sin embargo las caracteristicas electricas de la senal se especifican separadamente en la recomendacion V.28.

Cuando se accesa una computadora atravez de lineas telefonicas, la computadora se debe conectar a un modem, este a su vez se comunica atravez de la red telefonica a otro modem el cual debera estar conectado a una terminal. Esta configuracion involucra dos interfaces RS-232C: una entre la computadora y su modem y otra entre la terminal y su modem. La terminologia oficial denomina tanto a la computadora como a la terminal DTE's (equipo terminal de datos) y a los modems DCE (equipo de comunicacion de datos).

Debido a que frecuentemente es deseable poder escoger entre conectar una terminal con un modem o directamente al puerto de salida de una computadora, el estandar RS-232C provee frecuentemente ambas conexiones. Estrictamente hablando, el estandar RS-232C no fue pensado para conectar directamente un dispositivo DTE a otro DTE, y cuando esto se

lleva a cabo, la mayoria de sus senales son innecesarias. Cuando un fabricante especifica que un producto es compatible con el estandar RS-232C generalmente se refiere a que el equipo acepta y genera unicamente un pequeno grupo de las 21 senales del estandar y no viola ninguna otra parte del mismo.

Generalmente, el estandar RS-232C cubre senales tales como las del protocolo para contestar llamadas y para controlar modems para invertir la direccion de la transmision en un enlace HALF DUPLEX.

La principal desventaja del RS-232C es su limitacion en lo que se refiere a distancia de transmision; unicamente 15 metros. En la practica se puede rebasar considerablemente, pero siempre bajo nuestro riesgo. Una segunda desventaja es su velocidad maxima de transmision. Si bien esta generalmente no es una limitante en comunicacion entre terminal y computadora, ya que mientras la velocidad limite del RS-232C es de 19,200 Bits por segundo, la velocidad de los datos entre computadora-terminal es generalmente de 9600 Bits por segundo en el mejor de los casos, y es muy dificil transmitir datos aun a estas bajas velocidades atravez de redes telefonicas conmutadas.

La restriccion de distancia no es una seria desventaja si se utiliza un modem para accesar a una computadora en forma remota. Los modems generalmente se ubican a un lado de la computadora o terminal, y lo largo de la transmision se lleva a cabo entre los modems a lo largo de la linea telefonica.

Sin embargo, para aplicaciones locales es muy frecuente encontrar interfaces RS-232C conectando directamente terminales a computadoras, simplemente porque es mas conveniente utilizar la misma interface terminal-computadora, se utilice o no una conexion por modem, y es aqui donde el limite de los 15 metros se vuelve restrictivo, ademas, los niveles de voltajes no son particularmente convenientes dado que no son los mismos que los que se utilizan en las tecnologias estandars que dominan actualmente en la implementacion de computadoras (TTL y MOS), esto significa que es necesario una fuente de alimentacion adicional al utilizar dicho estandar.

Debido a la serie de desventajas antes mencionadas del estandar RS-232C, la interface por medio de un anillo de corriente, hecho famoso por los teletipos originales, se ha puesto de moda nuevamente, particularmente con las computadoras caseras de bajo costo, esta interface no es propiamente estandar, popularizandose las de 20 y 60 mA, y generalmente trabaja sobre distancias de hasta 450 metros a velocidades de hasta 9600 Bits por segundo. Desafortunadamente, esta interface es completamente incompatible con el RS-232C y requiere el uso de circuiteria conmutable, circuitos de conversion o una dualidad de circuitos para ambos estandares. Mas aun, la interface viene en dos sabores; activa, la cual genera por si misma la corriente y pasiva, la cual detecta la corriente o permite el

paso de corriente abriendo o cerrando el circuito. Circuitos de conversion permiten que dispositivos pasivos se comunicuen con dispositivos activos. Por ejemplo, un microcomputador generalmente contiene la interface activa y una terminal la pasiva, lo cual significa que se debera contar con un circuito de conversion activa-a-activa para conectar dos microcomputadores directamente.

SOBRELLEVANDO LOS DEFECTOS.

Para resolver los problemas del RS-232C y para incorporar y mejorar las ventajas de la interface de anillo de corriente, la EIA introdujo los estandares RS-422A, RS-423A y RS-449. Un cambio principal fue el de desenredar las especificaciones electricas, mecanicas y funcionales del RS-232C. En los estandares RS-422A y RS-423A unicamente se mencionan las especificaciones electricas. Para permitir velocidades mayores en la transmision de datos, el RS-422A utiliza dos cables para cada una de las senales, este arreglo, conocido como transmision balanceada, duplica el numero de alambres en el cable. El RS-423A transmite a velocidades mas lentas y utiliza un cable como trayectoria de retorno comun de todas las senales, a lo cual se le conoce como transmision desbalanceada y es muy similar al diseno del RS-232C. El estanda RS-423A opera en cualquiera de los ambientes RS-232C o RS-422A, y de esta manera se provee al usuario del equipa actual con una via para emigrar lentamente al nuevo regimen

rel RS-422A.

La EIA ha introducido el estandar RS-449 como el posible sucesor del RS-232C. Dicho estandar provee una descripcion funcional completa de las senales necesarias para el control de modems, lo mismo que de las especificaciones mecanicas de clavijas y soquets. Las especificaciones electricas para la mayoria de las senales son iguales a las del estandar RS-423A, aunque las del RS-422A estan disponibles para operaciones de alta velocicad si es necesario. El RS-449 tiene un tremendo numero de cables (46, en contraste con los 25 del RS-232C) en dos conectores, uno con 37 terminales y otro con 9. Afortunadamente, la mayoria de las aplicaciones no requieren las senales del conector de 9 terminales. Ademas de las mejoras en la velocidad y distancia, el RS-449 ofrece algunas otras funciones de mayor valia sobre el RS-232C en lo que se refiere a la prueba automatica de modem y provisionamiento para canales en estado estable, pero aun no incorpora la capacidad de automarcacion. El exito del RS-449 en el mercado comercial aun esta por verse.

EL ESTANDAR RS-232C.

La siguiente tabla muestra las 21 senales del estandar RS-232C en tres modalidades:

a) De acuerdo al numero de la terminal que ocupa en el conector estandar de 25 terminales.

b) Segun la codificacion asignada por la EIA.

c) Conforme la numeracion del CCITT V.24.

PIN	EIA	CCITT	Signal	Source
1	AA	101	Protective Ground	
7	AB	102	Signal Ground	
2	BA	103	Transmitted Data	
4	CA	105	Request To Send	
20	CD	108.2	Data Terminal Ready	
23	CH	111	Data Signaling Rate Selector (DTE source)	DTE (computer interface)
24	DA	113	Transmitter Signal Element Timing (DTE source)	
14	SBA	118	Secondary Transmitted Data	
19	SCA	120	Secondary Request To Send	
3	BB	104	Received Data	
5	CB	106	Clear To Send	
6	CC	107	Data Set Ready	
22	CE	125	Ring Indicator	
8	CF	109	Received Line Signal Detector	
21	CG	110	Signal Quality Detector	
23	CI	112	Data Signaling Rate Selector (DCE source)	DCE (modem or terminal)
15	DB	114	Transmitter Signal Element Timing (DCE source)	
17	DD	115	Receiver Signal Element Timing (DCE source)	
16	SBB	119	Secondary Received Data	
13	SCB	121	Secondary Clear To Send	
12	SCF	122	Secondary Received Line Signal Detector	

Table 2: *RS-232C signals.*

2 DESCRIPCION DE LAS SENALES.

TIERRA DE PROTECCION.-

Esta senal es para seguridad y se conecta al chasis del equipo en ambos extremos del enlace.

TIERRA DE SENAL.-

Esta senal establece un voltaje de tierra de referencia comun para todas las senales de datos.

TRANSMISION DE DATOS.

Es la trayectoria de los datos desde la terminal hasta la interface de la computadora.

RECEPCION DE DATOS.-

Es la trayectoria de los datos en direccion opuesta.

DATA SET READY.-

Senal generada por un modem para indicar que esta preparado para recibir datos para su transmision.

Cuando se interconecta un dispositivo lento a un computacor se puede utilizar esta terminal para controlar el

flujo de datos, aunque se puede utilizar otras lineas tal como el "CLEAR TO SEND" para el mismo proposito.

RECEIVED LINE SIGNAL DETECTOR.-

Esta senal es llamada con frecuencia "CARRIER DETECT" y se utiliza para indicar a la computadora que alguien esta tratando de hacer contacto en esta linea, y se puede utilizar para disparar a la computadora para generar una invitacion de acceso (LOGIN).

REQUEST TO SEND.-

Esta linea en union con la de "CLEAR TO SEND" controlan la direccion de la transmision en una operacion HALF-DUPLEX.

La computadora genera la senal "REQUEST TO SEND" cuando desea transmitir, y la senal "CLEAR TO SEND" le indica que el modem esta listo para recibir los caracteres para la transmision.

DATA TERMINAL READY.-

Esta senal se utiliza para indicar que la computadora esta preparada para recibir indicadores de llamada (RING INDICATOR).

RING INDICATOR.-

Esta senal se hace "0" logico y "1" logico al ritmo del timbre telefonico de tal manera que la computadora puede

contestar despues de un numero especifico de llamadas.

DATA SIGNALING RATE SELECTOR (DTE SOURCE).-

Senal utilizada por algunos modems para conmutar entre dos velocidades de transmision. La computadora en el extremo en el cual se origina la llamada utiliza esta senal para fijar la velocidad de transmision de la linea.

DATA SIGNALING RATE SELECTOR (DCE SOURCE).-

Esta senal es generada por el modem en el extremo de la linea que recibe la llamada y le indica a su computadora la velocidad de transmision, de acuerdo a la senal recibida del otro extremo de la linea.

RECEIVER SIGNAL ELEMENT TIMING (DCE SOURCE).-

Senal generada en el extremo receptor por los modems de funcionamiento sincronico para proveer de un reloj e incrementar la confiabilidad de la transmision.

TRANSMITTER SIGNAL ELEMENT TIMING (DCE SOURCE).-

Para la transmision de los datos en forma sincronica el reloj se puede generar en el modem, en cuyo caso el reloj recibe este nombre.

TRANSMITTER SIGNAL ELEMENT TIMING (DTE SOURCE).-

Este es el nombre que recibe la senal de temporizacion de

datos cuando se genera en la computadora.

SIGNAL QUALITY DETECTOR.-

Senal generada en los modems sincronicos y utilizada para indicar si existe una alta probabilidad de error en los datos recibidos.

LINEAS DE CANAL SECUNDARIO.-

Algunos modems proveen cinco lineas que forman un canal secundario de comunicacion. El canal principal generalmente se utiliza a una alta velocidad (1200 Bits por segundo) para transmitir en una direccion y un canal secundario de velocidad mucho mas baja (75 Bits por segundo) en la direccion inversa. El canal secundario se utiliza para 'escuchar' y confirmar la recepcion o para interrumpir al transmisor.

SENALES DEL CANAL SECUNDARIO:

1.- Secondary transmited data.

2.- Secondary received data.

3.- Secondary request to send.

4.- Secondary clear to send.

5.- Secondary received line signal detector.

3 ESPECIFICACIONES ELECTRICAS DEL ESTANDAR RS-232C.

Un transmisor RS-232C debe generar un voltaje superior a los 5 volts positivos para indicar una condicion de linea llamada 'espacio', y un voltaje por debajo de los 5 volts negativos para indicar la otra condicion, llamada marca. Para producir estos voltajes generalmente es necesario utilizar una fuente de alimentacion de +/- 12 Volts. Un receptor debe reconocer voltajes por encima de +3 volts como espacios y voltajes por debajo de los -3 volts como marcas, ver figura.

El cambio de la senal de un estado a otro debe tomarle

cuando mucho, 4 por ciento del periodo de un Bit (2 microsegundos a la maxima velocidad permisible de 19,200

BAUDS) en la region de transicion.

Estos requerimientos limitan la cantidad de capacitancia dispersa permisible en el enlace de transmision debido a que las capacitancias suavizan las transiciones rapidas. El estandar RS-232C especifica que la capacitancia no debe exceder los 2500 PF; y debido a que los cables tienen una capacitancia de 40 o 50 PF por pie, el RS-232C limita la longitud del cable a 50 pies.

Una segunda dificultad del RS-232C es su arreglo de tierras con dos lineas separadas: tierra de proteccion y tierra de senal. Desafortunadamente el estandar no establece claramente como deben utilizarse estas senales. En muchas implementaciones, la tierra de proteccion simplemente no es conectada.

La conexion de tierras para sistemas analogos distribuidos es una materia notoriamente dificil. Para dar una idea sencilla de los problemas que pueden ocurrir, imagine un enlace RS-232C entre equipos en los cuales las tierras de proteccion no estan conectadas pero donde la tierra de senal esta conectada a la tierra real en ambos extremos. Diferentes potenciales de tierra en los extremos del enlace causaran que exista un flujo de corriente a traves del cable de tierra de senal. La resistencia inevitable en este cable asegura la existencia de una diferencia de potencial entre las tierras de senal que podria, si la distancia es lo suficientemente grande, causar que los datos sean recibidos incorrectamente.

```
1         TITLE 'S I S T E M A   G R - 1 0 0 0'
2         LIST A

4   ********************************************************************
5   ***************** SISTEMA DE DESARROLLO GR-1000 ********************
6   ********************************************************************

8   *     El presente programa fue desarrollado en la Unidad de Inge-
9   * nieria Especializada de la Comision Federal de Electricidad por
10  * Hector Diaz Marenco, formando este el cuerpo basico, en cuanto
11  * a Programacion se refiere, que permite que el Sistema de desa--
12  * rrollo GR-1000, funcione como tal, convirtiendose en una herra-
13  * mienta bastante valiosa para el desarrollo de circuitos micro--
14  * procesados que manejen el microprocesador Z80.
15  *
16  *     La Organizacion del programa ha sido dividida en cuatro
17  * secciones, quedando establecido de la siguiente manera:
18  *
19  * 1. DIRECTIVUS DEL PROGRAMA, Tabla de direcciones o Flujos
20  * correspondientes a objetos localizados a lo largo del
21  * programa principal.
22  *
23  * 2. PROGRAMA PRINCIPAL, Rutinas que contienen los Bloques
24  * basicos para el manejo del Sistema GR-1000. Estas incluyen
25  * la Inicializacion, el Manejo de Pantallas, etc..
26  *
27  * 3.- SUBRUTINAS AUXILIARES, Rutinas de proposito general
28  * que sirven como soporte al Programa Principal, Entre estas
29  * se incluyen Manejo de bloques de datos, Traslado de Bloques
30  * de datos a la pantalla, etc.
31  *
32  * 4.- MENSAJES Y PANTALLAS, Contiene las tablas de datos en
33  * codigo ASCII, que son necesarios para formar las diferentes
34  * pantallas y mensajes del Sistema GR-1000.
35  *
36  *     El programa esta escrito para poder ser corrido en el
37  * Programa MACROENSAMBLADOR, del que se habla en la seccion
38  * aduacente del presente apendice.
```

```
     40                      ********************************************************************
     41                      ******************  DIRECTIVOS DEL PROGRAMA  ***********************
     42                      ********************************************************************


     44  0000                RUT_06 EQU 00000H
     45  0000                RUT_07 EQU 00000H
     46  0000                RUT_08 EQU 00000H
     47  0000                RUT_16 EQU 00000H
     48  0000                RUT_17 EQU 00000H
     49  0000                RUT_18 EQU 00000H
     50  F000                IN_RAM EQU 0F000H
     51  FFFF                RMA_20 EQU 0FFFFH
     52  FFFF                RMA_21 EQU RMA_20
     53  FFFF                RMA_22 EQU 0FFFFH
     54  FFFF                RMA_23 EQU 0FFFFH
     55  FFFF                RMA_24 EQU 0FFFFH
     56  FFFF                RMA_25 EQU 0FFFFH
     57  FFFF                RMA_26 EQU 0FFFFH
     58  FFFF                RMA_30 EQU 0FFFFH
     59  FFFF                RMA_31 EQU 0FFFFH
     60  FFFF                RMA_32 EQU 0FFFFH
     61  FFFF                RMA_33 EQU 0FFFFH
     62  FFFF                RMA_34 EQU 0FFFFH
     63  FFFF                RMA_35 EQU 0FFFFH
     64  FFFF                RMA_40 EQU 0FFFFH
     65  FFFF                RMA_41 EQU 0FFFFH
     66  FFFF                RMA_42 EQU 0FFFFH
     67  FFFF                RMA_43 EQU 0FFFFH
     68  FFFF                RMA_44 EQU 0FFFFH
     69  FFFF                RMA_45 EQU 0FFFFH
     70  FFFF                RMA_46 EQU 0FFFFH
     71  FFFF                COM_GR EQU 0FFFFH
     72  FFFF                COM_DS EQU 0FFFFH
     73  FFFF                GRB_MM EQU 0FFFFH
     74  FFFF                GRB_CP EQU 0FFFFH
     75  FFFF                GRB_GR EQU 0FFFFH
     76  FFFF                GRB_DS EQU 0FFFFH
     77  FFFF                DSK_MM EQU 0FFFFH
     78  FFFF                DSK_CP EQU 0FFFFH
     79  FFFF                DSK_GR EQU 0FFFFH
     80  FFFF                DSK_DS EQU 0FFFFH
     81  FFFF                MEM_CP EQU 0FFFFH
     82  FFFF                MEM_GR EQU 0FFFFH
     83  FFFF                MEM_DS EQU 0FFFFH
     84  FFFF                LFEED2 EQU 0FFFFH
     85  FFFF                LFEED3 EQU 0FFFFH
     86  FFFF                LFEED4 EQU 0FFFFH
     87  F6D9                REX_01 EQU 0F6D9H
     88  F600                REX_02 EQU 0F600H
```

```
                                ;*****************************************************
                                ;**********************  PROGRAMA PRINCIPAL  *********
                                ;*****************************************************

0000 06 03              RST_00  LD B,03H        'MONITOR Y PROGRAMADOR Z80'.
0002 AF                         XOR A
0003 C3 40 00                   JP 0040H
0006                            DEFS 50
0038 00 00 00           RST_38  DEFB 00,00,00
003B C3 B9 F4                   JP RFX 01
003E                            DEFS 2
0040 D3 03              LOOP01  OUT (03H),A     BORRA USART'S DE PIO'S. 03 Y 01.
0042 D3 01                      OUT (01H),A
0044 05                         DEC B
0045 20 F9                      JR NZ,LOOP01
0047 3E 40                      LD A,40H        PREPARA PIO'S. PARA INSTRUCCION.
0049 D3 03                      OUT (03H),A
004B D3 01                      OUT (01H),A
004D 31 00 00                   LD SP,0000H     CARGA STACK CON DIRECCION 0000.
0050 3F FE                      LD A,0FEH       CARGA VECTOR DE INTERRUPCIONES.
0052 ED 47                      LD I,A
0054 ED 5E                      IM 2            PONE EN MODO 2 DE INTERRUPCIONES.
0056 AF                         XOR A
0057 D3 05                      OUT (05H),A     PONE MODO DE TRABAJO DEL CDYT.
0059 3E 37                      LD A,37H
005B D3 05                      OUT (05H),A     PROGRAMA RETO1 DEL CDYT.
005D 3E D0                      LD A,0D0H
005F D3 05                      OUT (05H),A     BORRA REGS. INTERNOS DEL CDYT Y
0061 0E 32                      LD C,32H        RESETEA EL DISPLAY.
0063 C3 70 00                   JP 0070H        BRINCA PA RESTART 56H.
0066                            DEFS 10
0070 0B                 LOOP02  DEC C
0071 20 FD                      JR NZ,LOOP02
0073 3E 90                      LD A,90H
0075 D3 05                      OUT (05H),A     PONE CDYT EN MODO DE ESCRITURA.
0077 3E 7A                      LD A,7AH
0079 D3 03                      OUT (03H),A     PONE INSTRUCCION MODO PA PTO. 03.
007B 11 00 10                   LD DE,MENT00    CARGA CON DIR. DE MENSAJE 00.
007E 06 FF                      LD B,0FFH       CARGA CONTADOR DE CARACTERES.
0080 DD 21 96 00                LD IX,REG_00    CARGA EN IX DIR. DE RETORNO 00.
0084 21 00 0C                   LD HL,SUBT01
0087 22 F2 FE                   LD (0FEF2H),HL  HABILITA TRANSMISION EN SUBT01.
008A 3E 31                      LD A,31H
008C D3 03                      OUT (03H),A     TRANSMISION A TERMINAL.
008E 3E 05                      LD A,05H
0090 D3 06                      OUT (06H),A     HABILITA CPI.
0092 FB                         EI              HABILITA INTERRUPCIONES PA Z80.
0093 76                 HALT    HALT            ESTADO DE ESPERA.
0094 18 FD                      JR HALT
```

```
  140  0096  11 FF 10        REG_00 LD DE,MENT01     CARGA CON MENSAJE 01.
  141  0099  06 6A                 LD B,6AH         CARGA CONTADOR DE CARACTERES.
  142  009B  DD 21 A2 00            LD IX,REG_01     CARGA DIR. DE RETORNO.
  143  009F  C3 0C 0C               JP MENTER
  144  00A2  0E 00           REG_01 LD C,00H         CONTADOR.
  145  00A4  21 AA 00        RECT_1 LD HL,RECT01     RECEPCION DE TERMINAL EN RECT01.
  146  00A7  C3 20 0C               JP RECTER        VA A RECTER.
  147  00AA  DB 02           RECT01 IN A,(02H)       DEJA DATO DESDE LA TERMINAL.
  148  00AC  B1                     OR C
  149  00AD  4F                     LD C,A           SALVA DATO Y BANDERA.
  150  00AE  FD 21 B5 00            LD IY,RET_01     DIRECCION DE RETORNO.
  151  00B2  C3 22 0C               JP DET_PO        BRINCA A PARCHE POSICION.
  152  00B5  FD 21 BE 00     RET_01 LD IY,VUELVE     DIRECCION DE RETORNO.
  153  00B9  CB 79                  BIT 7,C          PRUEBA BANDERA.
  154  00BB  C2 7E 0C               JP NZ,BORRAS     SI BANDERA=1, VA A BORRAR.
  155  00BE  21 BE 11        VUELVE LD HL,0118H      CARGA DIRECCION.
  156  00C1  79                     LD A,C
  157  00C2  FE 43                  CP 43H
  158  00C4  28 3C                  JR Z,SELECC      SI A='C', BRINCA A SELECC
  159  00C6  FE 50                  CP 50H
  160  00C8  28 19                  JR Z,BACK10      SI A='P', BRINCA A BACK10
  161  00CA  FE 53                  CP 53H
  162  00CC  28 23                  JR Z,TAB         SI A='S', BRINCA A TAB
  163  00CE  11 6D 11               LD DE,MENT05     CARGA CON MENSAJE 05.
  164  00D1  06 0B                  LD B,0BH
  165  00D3  DD 21 DA 00            LD IX,REG_03     RETORNA EN REG_03.
  166  00D7  C3 0C 0C               JP MENTER
  167  00DA  0E 00           REG_03 LD C,00H         PONE BANDERA DE FECHA.
  168  00DC  FD 21 A4 00            LD IY,RECT_1
  169  00E0  C3 47 0C               JP SET_PO        PONE CURSOR EN TERMINAL.
  170  00E3  35              BACK10 DEC (HL)         HACE EL BACK SPACE.
  171  00E4  35                     DEC (HL)
  172  00E5  7E                     LD A,(HL)
  173  00E6  FE 26                  CP 26H
  174  00E8  28 10                  JR Z,REAJ_1      SI A='&', REAJUSTA CURSOR.
  175  00EA  FD 21 A2 00     DIRIGE LD IY,REG_01
  176  00EE  C3 47 0C               JP SET_PO        PONE CURSOR EN TERMINAL
  177  00F1  34              TAB    INC (HL)         EFECTUA EL TAB.
  178  00F2  34                     INC (HL)
  179  00F3  7E                     LD A,(HL)
  180  00F4  FE 32                  CP 32H
  181  00F6  28 06                  JR Z,REAJ_2      SI A='2', REAJUSTA CURSOR.
  182  00F8  18 F0                  JR DIRIGE
  183  00FA  36 30           REAJ_1 LD (HL),30H      REAJUSTA POSICION.
  184  00FC  18 EC                  JR DIRIGE
  185  00FE  36 20           REAJ_2 LD (HL),20H      REAJUSTA POSICION.
  186  0100  18 E8                  JR DIRIGE
  187  0102  7E              SELECC LD A,(HL)        RUTINA DE SELECCION.
  188  0103  E6 0F                  AND 0FH          MASCARA EL NIBBLE ALTO.
  189  0105  CB 27                  SLA A            RECORRE LA INFORMACION.
  190  0107  26 0D                  LD H,0DH         CARGA CON BYTE DE ALTO ORDEN
  191  0109  6F                     LD L,A           CARGA BYTE DE BAJO ORDEN.
  192  010A  08                     EX AF,AF'        SALVA EN REGS. ALTERNADOS.
  193  010B  D9                     EXX
```

```
ERR LINEA ADDR  B1 B2 B3 B4        S I S T E M A     O R - 1 0 0 0                    PAGINA  5

    194  010C  11 97 14                LD DE,MENT90      CARGA MENSAJE 90.
    195  010F  06 3B                   LD B,3BH
    196  0111  DD 21 1B 01             LD IX,REG_EX      SALVA DIR. DE RETORNO.
    197  0115  C3 0C 0C                JP MENTEP
    198  0118  D9                  REG_EX EXX            RECUPERA DE REGS. ALTERNADOS.
    199  0119  08                        EX AF,AF'
    200  011A  E9                        JP (HL)         BRINCA A TABLA DE RUTINAS.
    201  011B                            DEFS 3
```

```
205  011E  11 06 11      EX_MEM LD DE,MENTIO     ENVIA MENSAJE 10.
206  0121  06 FF                 LD B,0FFH       CARGA CONTADOR.
207  0123  DD 21 2A 01           LD IX,REG_10    RETORNA EN REG 10.
208  0127  C3 0C 0C       MENT_3 JP MENTER
209  012A  11 0F 12       REG_10 LD DE,MENTII_1  ENVIA MENSAJE 08.
210  012D  06 33                 LD B,33H
211  012F  DD 21 35 01           LD IX,REG_11    RETORNA EN REG 11.
212  0133  18 F2                 JR MENT_3
213  0135  06 04          REG_11 LD B,04H        CARGA CONTADOR = 4.
214  0137  3E 00                 LD A,00H
215  0139  32 B0 FF       CT_RUN LD (0FFB0H),A   CARGA REG. INDICADOR = 00.
216  013C  21 FF FF       REC_10 LD HL,0FFFFH
217  013F  70                    LD (HL),B       Guarda el CONTADOR.
218  0140  11 B0 FF              LD DE,0FFB0H    CARGA DIRECCION DE MEMORIA.
219  0143  0E 66          RET_18 LD C,066H       RESETEA BANDERA Y DATO.
220  0145  21 4B 01       HRCT10 LD HL,RECT10    RECTIN EN RECTIO DE TERMINAL.
221  0148  C3 20 0C              JP RECTIF
222  014B  DB 02          RECTIO IN A,(02H)      METE DATO DESDE TERMINAL.
223  014D  B1                    OR C            SALVA EN REGISTRO C.
224  014E  ED 21 BC 01           LD IY,RET_15    CARGA DIR. DE RETORNO.
225  0152  4F                    LD C,A          SALVA EL DATO.
226  0153  CB 79                 BIT 7,C         PRUEBA BANDERA DE T.I.
227  0155  C2 27 0C              JP NZ,RET_PR    SI HAMMING = 0A A BOMBAS.
228  0158  FE 08          RET_15 CP 08H
229  015A  28 6C                 JR Z,BACKSP     SI ES BACKSPACE RETORNA A BORRAR.
230  015C  FE 40                 CP 40H
231  015E  CA D4 03              JP Z,NONITO     SI ES NUMERO, VA A RUTINA.
232  0161  21 B0 FF              LD HL,0FFB0H    REVISA REGISTRO INDICADOR.
233  0164  CB 46                 BIT 0,(HL)
234  0166  C2 FF 02              JP NZ,RUT_01
235  0169  CB 4E                 BIT 1,(HL)
236  016B  C2 09 03              JP NZ,RUT_02
237  016E  CB 56                 BIT 2,(HL)
238  0170  C2 13 03              JP NZ,RUT_03
239  0173  CB 5E                 BIT 3,(HL)
240  0175  C2 91 01              JP NZ,RETURN
241  0178  CB 66                 BIT 4,(HL)
242  017A  C2 D5 05              JP NZ,RUT_05
243  017D  CB 6E                 BIT 5,(HL)
244  017F  C2 00 00              JP NZ,RUT_06
245  0182  CB 76                 BIT 6,(HL)
246  0184  C2 00 00              JP NZ,RUT_07
247  0187  CB 7E                 BIT 7,(HL)
248  0189  C2 00 00              JP NZ,RUT_08
249  018C  FE 52                 CP 52H
250  018E  CA A5 03              JP Z,RUN.       SI ES 'R' VA A RUTINA.
251  0191  21 A0 01       RETURN LD HL,ERROR1    CARGA DIRECCION DE ERROR.
252  0194  CD 8D 0C              CALL CR_ASC     IMPRIMA 2 RUTINA EN ASC.
253  0197  12                   LD (DE),A       CARGA DATO EN MEMORIA.
```

```
254  0198 79                  LD A,C          RECUPERA DATO RECIBIDO.
255  0199 DD 21 B5 01         LD IX,RET_11    CARGA DIR. DE RETORNO.
256  019D C3 5E 0C            JP SACA_T       VA A RUTINA SACA A_ TERMINAL.
257  01A0 FD 21 A7 01 ERROR1  LD IY,RET_12    CARGA DIR. DE RETORNO.
258  01A4 C3 27 0C            JP DET_FO       BRINCA A RUTINA DET_FO.
259  01A7 FD 21 AE 01 RET_12  LD IY,RET_13    CARGA DIR. DE RETORNO.
260  01AB C3 6B 0C            JP PONE_T       LLAMA A RUTINA PONE_T.
261  01AE FD 21 45 01 RET_13  LD IY,HRCT10    CARGA DIR. DE RETORNO.
262  01B2 C3 47 0C            JP SET_FO       LLAMA A RUTINA SET_FO.
263  01B5 0E 00               RET_11 LD C,00H RESETEA REGISTRO DE DATO Y BAND.
264  01B7 18 23               JR RET_14
265  01B9 79                  RET_17 LD A,C   RECUPERA DATO DE TERMINAL.
266  01BA 18 9C               JR RET_15
267  01BC FD 21 C3 01 RET_10  LD IY,RET_16    CARGA DIR. DE RETORNO.
268  01C0 C3 7F 0C            JP BORRAS
269  01C3 FD 21 B9 01 RET_16  LD IY,RET_17    CARGA DIR. DE RETORNO.
270  01C7 C3 47 0C            JP SET_FO
271  01CA 78                  BACKSP LD A,B   PRUEBA CONTADOR.
272  01CB 21 CF FF            LD HL,0FFCFH
273  01CE BE                  CP (HL)
274  01CF CA 43 01            JP Z,RET_10     SI CONTADOR=4, VA A RET_10.
275  01D2 1B                  DEC DE          DECREMENTA DIR. DE MEMORIA.
276  01D3 04                  INC B           INCREMENTA CONTADOR.
277  01D4 79                  LD A,C          RECUPERA EL DATO.
278  01D5 DD 21 43 01         LD IX,RET_10    CARGA DIR. DE RETORNO.
279  01D9 C3 5E 0C            JP SACA_T       ENVIA A TERMINAL.
280  01DC 13                  RET_14 INC DE   INCREMENTA DIR. DE MEMORIA.
281  01DD 05                  DEC B           DECREMENTA CONTADOR.
282  01DE C2 45 01            JP NZ,HRCT10    SICONTADOR>0, VA A HRCT10.
283  01E1 21 D0 FF            LD HL,0FFD0H    REVISA REGISTRO INDICADOR.
284  01E4 CB 46               BIT 0,(HL)
285  01E6 C2 16 03            JP NZ,RUT_11
286  01E9 CB 4E               BIT 1,(HL)
287  01EB C2 C5 03            JP NZ,RUT_12
288  01EE CB 56               BIT 2,(HL)
289  01F0 C2 05 04            JP NZ,RUT_13
290  01F3 CB 5E               BIT 3,(HL)
291  01F5 C2 6B 04            JP NZ,RUT_14
292  01F8 CB 66               BIT 4,(HL)
293  01FA C2 6B 04            JP NZ,RUT_14
294  01FD CB 6E               BIT 5,(HL)
295  01FF C2 00 00            JP NZ,RUT_16
296  0202 CB 76               BIT 6,(HL)
297  0204 C2 00 00            JP NZ,RUT_17
298  0207 CB 7E               BIT 7,(HL)
299  0209 C2 00 00            JP NZ,RUT_18
300  020C 11 B8 12            LD DE,MENT12    ENVIA MENSAJE 12.
301  020F 06 D0               LD B,0D0H
302  0211 DD 21 18 02         LD IX,REG_12    DIRECCION DE RETORNO.
303  0215 C3 0C 0C            JP MENTER
304  0218 D9                  REG_12 EXX
305  0219 0E 0B               LD C,0BH        CARGA NO. DE LINEAS (ALT).
306  021B D9                  EXX
307  021C 21 DD FF            LD HL,0FFDDH    LOC. DE DATOS HEXADECIMALES.
```

```
     308  021F  11 D6 FF          LD DE,OFFD6H     LOC. DE BYTES HEXADECIMALES.
     309  0222  06 02             LD B,02H         NO. DE CONVERSIONES.
     310  0224  CD A8 0C          CALL HEX.BY
     311  0227  2A D6 FF          LD HL,(OFFD6H)   RECOGE POSICION.
     312  022A  11 05 00    N_DATO LD DE,0005H
     313  022D  ED 52             SBC HL,DE
     314  022F  22 D4 FF    RET_21 LD (OFFD4H),HL   SALVA NUEVA DIRECCION.
     315  0232  7E                LD A,(HL)        CARGA EN EL ACUMULADOR EL DATO.
     316  0233  32 D3 FF          LD (OFFD3H),A    SALVA EL DATO.
     317  0236  21 D4 FF          LD HL,OFFD4H     TRANSFIERE DATOS DE FERALFDS
     318  0239  11 D1 FF          LD DE,OFFD1H     EN FFD1_FFD2.
     319  023C  01 02 00          LD BC,0002H
     320  023F  ED B0             LDIR
     321  0241  21 D5 FF          LD HL,OFFD5H     LOC. DE BYTES HEXADECIMALES.
     322  0244  11 D8 FF          LD DE,OFFD8H     LOC. DE CARACTERES ASCII.
     323  0247  06 03             LD B,03H         NO. DE CONVERSIONES.
     324  0249  CD B5 0C          CALL BY.ASC
     325  024C  11 D8 FF          LD DE,OFFD8H     ENVIA MENSAJE DESDE FFD8.
     326  024F  06 06             LD B,06H
     327  0251  DD 21 5B 02       LD IX,RET_19     DIRECCION DE RETORNO.
     328  0255  C3 0C 0C          JP MENTER
     329  0258  D9           RET_19 EXX
     330  0259  0D                DEC C            DECREMENTA CONTADOR (ALT).
     331  025A  CA 65 02          JP Z,RET_20      SI CONTADOR=0, VA A RET_20.
     332  025D  D9                EXX
     333  025E  2A D1 FF          LD HL,(OFFD1H)   RECUPRA DIRECCION.
     334  0261  23                INC HL           INCREMENTA DIRECCION.
     335  0262  C3 2F 02          JP  RET_21       REGRESA A LA RUTINA.
     336  0265  D9           RET_20 EXX
     337  0266  0E 0B             LD C,0BH         CARGA CONTADOR DE LINEAS.
     338  0268  D9                EXX
     339  0269  06 02             LD B,02H         CARGA CONTADOR DE DIGITOS.
     340  026B  0E 00             LD C,00H         CARGA REGISTRO DE BANDERAS=00.
     341  026D  21 73 02     HRCT11 LD HL,RECT11     RECEPCION EN RECT11.
     342  0270  C3 20 0C          JP RECTER
     343  0273  DB 02        RECT11 IN A,(02H)       METE DATO DESDE TERMINAL.
     344  0275  08                EX AF,AF'        GUARDA EL DATO.
     345  0276  FD 21 87 02       LD IY,RET_25     CARGA DIR. DE RETORNO.
     346  027A  CB 79             BIT 7,C
     347  027C  C2 7F 0C          JP NZ,BORRAS     SI BIT 7=1, VA A BORRAS.
     348  027F  21 CF FF          LD HL,OFFCFH
     349  0282  CB 7E             BIT 7,(HL)
     350  0284  C2 BB 03          JP NZ,BORRA2     SI BIT 6=1, VA A BORRA2.
     351  0287  08           RET_25 EX AF,AF'        REGRESA EL DATO.
     352  0288  2A D6 FF          LD HL,(OFFD6H)   CARGA HL, CON LA POSICION.
     353  028B  FE 49             CP 49H
     354  028D  CA D3 02          JP Z,RESTAR      SI ES 'I', VA A RESTAR.
     355  0290  FE 4D             CP 4DH
     356  0292  CA D4 03          JP Z,MONITO      SI ES 'M', VA A MONITO.
     357  0295  FE 4E             CP 4EH
     358  0297  CA CB 02          JP Z,NEXT        SI ES 'N', VA A NEXT.
     359  029A  FE 4F             CP 4FH
     360  029C  CA DB 02          JP Z,SUMAR       SI ES 'O', VA A SUMAR.
     361  029F  FE 50             CP 50H
```

```
LOC   CODE           SENTENCIA
02A1  CA CF 02              JP Z,PREVIO      SI ES "P", VA A PREVIO.
02A4  FE 51                 CP 51H
02A6  CA 9A 03              JP Z,N_DIRE      SI ES "Q", VA A N_DIRE.
02A9  FE 52                 CP 52H
02AB  CA A6 03              JP Z,_RUN        SI ES "R", VA A _RUN.
02AE  21 47 03              LD HL,ERROR2
02B1  CD 8D 0C              CALL CP_ASC      LLAMA A RUTINA CP_ASC.
02B4  21 DA FF              LD HL,OFFDAH
02B7  ED 6F                 RLD              CARGA EL DIGITO.
02B9  05                    DEC B            DECREMENTA CONTADOR.
02BA  C2 60 02              JP NZ,NDCIT1     SI CONTADOR<>0, VA A NDCIT1.
02BD  7E                    LD A,(HL)
02BE  47                    LD B,A           CARGA REGISTRO DE COMPARACION
02BF  2A D6 FF              LD HL,(OFFD6H)   RECARGA POSICION.
02C2  77                    LD (HL),A        CARGA NUEVO DATO.
02C3  7E                    LD A,(HL)        RECOBRA DATO
02C4  B8                    CP B
02C5  C2 5C 03              JP NZ,ERROR3     SI NO ES IGUAL, VA A ERROR3.
02C8  23             NEXT   INC HL           INCREMENTA LA POSICION.
02C9  22 D6 FF       CARGAR LD (OFFD6H),HL   CARGA NUEVA POSICION.
02CC  C3 2A 02              JP M_DATO        VA A MOSTRAR.
02CF  2B             PREVIO DEC HL           DECREMENTA DIRECCION.
02D0  C3 C9 02              JP CARGAR        VA A CARGAR.
02D3  3E 81          RESTAR LD A,81H         RUTINA PARA RETRASAR POSICION.
02D5  32 D0 FF              LD (OFFD0H),A    CARGA REG. INDICADOR 81.
02D8  11 D6 13              LD DE,MENT14     ENVIA MENSAJE 14.
02DB  18 08                 JR IGUAL
02DD  3E 01          SUMAR  LD A,01H         RUTINA PARA ADELANTAR POSICION.
02DF  32 D0 FF              LD (OFFD0H),A    CARGA REGISTRO INDICADOR 01
02E2  11 88 13              LD DE,MENT13     ENVIA MENSAJE 13.
02E5  06 17          IGUAL  LD B,17H
02E7  DD 21 EE 02           LD IX,REG_13     DIRECCION DE RETORNO.
02EB  C3 0C 0C              JP MENTER
02EE  11 9F 13       REG_13 LD DE,MENT14     ENVIA MENSAJE 14.
02F1  06 1B                 LD B,1BH
02F3  DD 21 FA 02           LD IX,REG_14     DIRECCION DE RETORNO.
02F7  C3 0C 0C              JP MENTER
02FA  06 02          REG_14 LD B,02H         CONTADOR DE DIGITOS.
02FC  C3 3C 01              JP REG_10
02FF  FE 49          RUT_01 CP 49H           CONTINUA RUTINA DE COMPARACION.
0301  CA D3 02              JP Z,RESTAR      SI ES "I", VA A RESTAR.
0304  FE 4F                 CP 4FH
0306  CA DD 02              JP Z,SUMAR       SI ES "O", VA A SUMAR.
0309  FE 51          RUT_02 CP 51H
030B  CA 9A 03              JP Z,N_DIRE      SI ES "Q", VA A N_DIRE.
030E  FE 52                 CP 52H
0310  CA A6 03              JP Z,_RUN        SI ES "R", VA A _RUN.
0313  C3 91 01              JP RETURN        RETORNA A LA RUTINA.
0316  11 BA 13       RUT_11 LD DE,MENT15     ENVIA MENSAJE 15.
0319  06 1C                 LD B,1CH
031B  DD 21 22 03           LD IX,REG_15     DIRECCION DE RETORNO.
031F  C3 0C 0C              JP MENTER
0322  21 DB FF       REG_15 LD HL,OFFDBH     DIR. DE DATOS DESCARGADOS.
0325  11 DC FF              LD DE,OFFDCH     DIR. DE DATOS ANTERIORES.
```

```
  470  03B5  06 23           LD B,23H
  471  03B7  DD 21 BE 03     LD IX,REG_21      DIRECCION DE RETORNO.
  472  03BB  C3 0C 0C        JP MENTER
  473  03BE  06 04     REG 21 LD B,04H          CONTADOR=04.
  474  03C0  3E 02           LD A,02H          PONE REGISTRO DE RUTINA.
  475  03C2  C3 39 01        JP ET_RUN
  476  03C5  21 D0 FF  RUT_12 LD HL,0FFD0H      DIRECCION DE DIGITOS.
  477  03C8  11 D6 FF        LD DE,0FFD6H      DIRECCION DE BYTES.
  478  03CB  06 02           LD B,02H          NO. DE BYTES.
  479  03CD  CD A8 0C        CALL HEX_BY
  480  03D0  2A D6 FF        LD HL,(0FFD6H)
  481  03D3  E9              JP (HL)           BRINCA A EJECUTAR.
  482  03D4  11 00 10  MONITO LD DE,MENT00      CARGA MENSAJE 00.
  483  03D7  06 FF           LD B,0FFH
  484  03D9  DD 21 96 00     LD IX,REG_00      CARGA DIR. DE RETORNO.
  485  03DD  C3 0C 0C        JP MENTER
```

```
    416  0328  06 01              LD B,01H        NO. DE CONVERSIONES.
    417  032A  CD A0 0C           CALL HEX_BY     LLAMA A RUTINA DE CONV. HEX_BY.
    418  032D  1B                 DEC DE
    419  032E  1A                 LD A,(DE)       RECUPERA EL BYTE Y
    420  032F  5F                 LD E,A          LO CARGA EN EL REGISTRO E.
    421  0330  16 00              LD D,00H
    422  0332  2A D6 FF           LD HL,(OFFD6H)  RECUPERA POSICION.
    423  0335  3A D0 FF           LD A,(OFFD0H)
    424  0338  CB 7F              BIT 7,A         PRUEBA BANDERA DE ADC/SBC.
    425  033A  C2 42 03           JP NZ,RESTA
    426  033D  ED 5A              ADC HL,DE       EFECTUA LA SUMA.
    427  033F  C3 C9 02           JP CARGAR       VA A CARGAR
    428  0342  ED 52        RESTA  SBC HL,DE       EFECTUA LA RESTA.
    429  0344  C3 C9 02           JP CARGAR       VA A CARGAR.
    430  0347  FD 21 4E 03  ERROR2 LD IY,RET_22    CARGA DIR. DE RETORNO.
    431  034B  C3 27 0C           JP DET_PO
    432  034E  FD 21 55 03  RET_22 LD IY,RET_23    CARGA DIR. DE RETORNO.
    433  0352  C3 6B 0C           JP PONE_T
    434  0355  FD 21 6B 02  RET.23 LD IY,HRCT11    CARGA DIR. DE RETORNO.
    435  0359  C3 47 0C           JP SET_PO
    436  035C  FD 21 2A 02  ERROR3 LD IY,N_DATO    CARGA DIR. DE RETORNO.
    437  0360  7C                 LD A,H
    438  0361  21 CF FF           LD HL,OFFCFH
    439  0364  CB FE              SET 7,(HL)      CARGA BANDERA DE ERROR.
    440  0366  2A D6 FF           LD HL,(OFFD6H)  CARGA LA POSICION EN HL.
    441  0369  FE 0B              CP 0BH
    442  036B  38 0B              JR C,ERROR4     SI DIR. ALTA>02, VA A ERROR4.
    443  036D  D9                 EXX
    444  036E  11 E0 13           LD DE,MENT17    ENVIA MENSAJE 17.
    445  0371  06 2C              LD B,2CH
    446  0373  DD 21 44 0C        LD IX,REG_04
    447  0377  C3 0C 0C           JP MENTER
    448  037A  3F           ERROR4 CCF             RESETEA BANDERA DE CARRY.
    449  037B  D9                 EXX
    450  037C  11 19 14           LD DE,MENT18    ENVIA MENSAJE 18.
    451  037F  06 2C              LD B,2CH
    452  0381  DD 21 44 0C        LD IX,REG_04
    453  0385  C3 0C 0C           JP MENTER
    454  0388  21 CF FF     BORRA2 LD HL,OFFCFH    RESETEA BIT 6 DE BANDERA5.
    455  038B  CB BE              RES 7,(HL)
    456  038D  D9                 EXX
    457  038E  11 45 14           LD DE,MENT19    ENVIA MENSAJE 19.
    458  0391  06 0F              LD B,0FH
    459  0393  DD 21 44 0C        LD IX,REG_04
    460  0397  C3 0C 0C           JP MENTER
    461  039A  11 54 14     N_DIRE LD DE,MENT20    CARGA CON MENSAJE 20.
    462  039D  06 33              LD B,33H
    463  039F  DD 21 35 01        LD IX,REG_11    DIRECCION DE RETORNO.
    464  03A3  C3 0C 0C           JP MENTER
    465  03A6  11 87 14     _RUN_ LD DE,MENT22    CARGA CON MENSAJE 22.
    466  03A9  06 10              LD B,10H
    467  03AB  DD 21 B2 03        LD IX,REG_22    DIRECCION DE RETORNO.
    468  03AF  C3 0C 0C           JP MENTER
    469  03B2  11 64 14     REG_22 LD DE,MENT21    CARGA CON MENSAJE 21.
```

```
     407                        ************** RUTINA DEL TRASLADO DE ARCHIVOS **************

     409  03E0  11 70 15        TRASLA LD DE,MENT71     RUTINAS DE TRASLADO DE ARCHIVOS!
     490  03E3  06 A3                  LD B,0A3H
     491  03E5  DD 21 EC 03            LD IX,REG_71
     492  03E9  C3 0C 0C               JP MENTER
     493  03EC  06 02           REG_71 LD B,02H
     494  03EE  3E 04                  LD A,04H
     495  03F0  C3 39 01               JP LT_RUN
     496  03F3  FE 45           RUT_03 CP 45H
     497  03F5  D2 A0 01               JP NC,ERROR1
     498  03F8  FE 41                  CP 41H
     499  03FA  DA A0 01               JP C,ERROR1
     500  03FD  12                     LD (DE),A
     501  03FE  DD 21 B5 01            LD IX,RET_11
     502  0402  C3 5C 0C               JP SACA_T
     503  0405  1B              RUT_13 DEC DE
     504  0406  1A                     LD A,(DE)
     505  0407  CB 27                  SLA A
     506  0409  CB 27                  SLA A
     507  040B  12                     LD (DE),A
     508  040C  EB                     EX DE,HL
     509  040D  11 D0 FF               LD DE,OFFD0H
     510  0410  06 01                  LD B,01H
     511  0412  CD A0 0C               CALL HEX_BY
     512  0415  EB                     EX DE,HL
     513  0416  2B                     DEC HL
     514  0417  6E                     LD L,(HL)
     515  0418  26 0E                  LD H,0EH          * ASIGNAR DIRECCION *
     516  041A  E9                     JP (HL)
     517  041B  FF FF FF               DATA OFFH,OFFH,OFFH
     518  041E  11 13 16        COM_CP LD DE,MENT72
     519  0421  06 09                  LD B,09H
     520  0423  DD 21 2A 04            LD IX,REG_72
     521  0427  C3 0C 0C               JP MENTER
     522  042A  11 13 16        REG_72 LD DE,MENT72
     523  042D  06 09                  LD B,09H
     524  042F  DD 21 36 04            LD IX,REG_73
     525  0433  C3 10 0C               JP MENCOM
     526  0436  3E 36           REG_73 LD A,36H
     527  0438  C3 12 0C               JP MENTER106

     529                        ;RUTINA DE MEMORIA-MEMORIA.

     531  043B  FD 21 42 04     MEM_MM LD IY,RMM_00      --- RETORNO ---
     532  043F  C3 65 0D               JP M_OPC1         BRINCA A MENSAJE OPC1.
     533  0442  11 39 16        RMM_00 LD DE,OPC_02      CARGA MENSAJE OPC_02
     534  0445  06 19                  LD B,25
     535  0447  DD 21 4E 04            LD IX,RMM_01      --- RETORNO ---
```

```
     536  044B  C3 0C 0C            JP MENTER
     537  044E  FD 21 55 04  RMM_01 LD IY,RMM_02     --- RETORNO ---
     538  0452  C3 91 0D            JP M_OPCS        BRINCA A MENSAJE OPCS
     539  0455  06 0C        RMM_02 LD B,0CH         CARGA CONTADOR DE CARACTERES
     540  0457  11 C1 FF            LD DE,0FFC1H     INICIO DE CARGA
     541  045A  3E 01                LD A,01H         BANDERA DE INDICACION
     542  045C  32 C0 FF            LD (0FFC0H),A
     543  045F  21 CF FF            LD HL,0FFCFH
     544  0462  70                  LD (HL),B        SALVA EL CONTADOR
     545  0463  3C 08                LD A,08H         CARGA REGISTRO INDICADOR
     546  0465  32 D0 FF            LD (0FFD0H),A
     547  0468  C3 43 01            JP RET_10        BRINCA A RUTINA RET.10
     548  046B  11 92 16     RUT_14 LD DE,MENT73     CARGA CON MENSAJE 73
     549  046E  06 2D                LD B,45
     550  0470  DD 21 76 04          LD IX,REG_73     --- RETORNO ---
     551  0474  18 D5                JR RMM_00109     ENVIA MENSAJE
     552  0476  21 7C 04     REG_73 LD HL,HRCT70     RECIBE DE LA TERMINAL
     553  0479  C3 20 0C            JP RECTER        EN HRCT70
     554  047C  DB 02        HRCT70 IN A,(02)        RECIBE EL DATO
     555  047E  FE 4D                CP 4DH
     556  0480  CA B4 03            JP Z,MONITO      SI A=4D, VA A MONITO
     557  0483  FE 0D                CP 0DH
     558  0485  28 4D                JR Z,RMM_06      SI A=0D, VA A RMM_06
     559  0487  FE 08                CP 08H
     560  0489  28 1A                JR Z,RMM_05      SI A=08, VA A RMM_05
     561  048B  FE 1A                CP 1AH
     562  048D  CA 49 05            JP Z,RMM_09      SI A=1A, VA A RMM_09
     563  0490  FD 21 97 04          LD IY,RMM_03     RUTINA DE ERROR
     564  0494  C3 27 0C            JP RET_PO              .
     565  0497  FD 21 9E 04  RMM_03 LD IY,RMM_04           .
     566  049B  C3 6B 0C            JP PONE_T              .
     567  049E  FD 21 76 04  RMM_04 LD IY,REC.73           .
     568  04A2  C3 47 0C            JP SET_PO              .
     569  04A5  3A C0 FF     RMM_05 LD A,(0FFC0H)    CARGA BANDERA INDICADORA
     570  04A8  FE 01                CP 01H           Y SIGUE RUTINA
     571  04AA  28 16                JR Z,RMM_5A            .
     572  04AC  FE 02                CP 02H                 .
     573  04AE  CA FF FF             JP Z,RMM_20            .
     574  04B1  FE 03                CP 03H                 .
     575  04B3  CA FF FF             JP Z,RMM_21            .
     576  04B6  FE 04                CP 04H                 .
     577  04B8  CA FF FF             JP Z,RMM_22            .
     578  04BB  FE 05                CP 05H                 .
     579  04BD  CA FF FF             JP Z,RMM_23            .
     580  04C0  00 00                DEFB 00,00
     581  04C2  11 DF 16     RMM_5A LD DE,MENT74     RUTINA BACKSPACE
     582  04C5  06 18                LD B,24
     583  04C7  DD 21 CD 04          LD IX,REG_74     --- RETORNO ---
     584  04CB  18 71                JR REG_75-03     ENVIA MENSAJE 74
     585  04CD  06 01        REG_74 LD B,01H         INICIALIZA CONTADOR
     586  04CF  11 CC FF            LD DE,0FFCCH     Y DIRECCION DE CARGA
     587  04D2  18 94                JR RUT_14-03     VA A RET_10
     588  04D4  3A C0 FF     RMM_06 LD A,(0FFC0H)    CARGA BANDERA INDICADORA
     589  04D7  FE 01                CP 01H           Y SIGUE RUTINA
```

```
 590  04D9  28 25              JR Z,RMM_07          .
 591  04DB  FE 02              CP 02H               .
 592  04DD  CA FF FF           JP Z,RMM_30          .
 593  04E0  FE 03              CP 03H               .
 594  04E2  CA FF FF           JP Z,RMM_31          .
 595  04E5  FE 04              CP 04H               .
 596  04E7  CA FF FF           JP Z,RMM_32          .
 597  04EA  FE 05              CP 05H               .
 598  04EC  CA FF FF           JP Z,RMM_33          .
 599  04EF  FE 06              CP 06H               .
 600  04F1  CA FF FF           JP Z,RMM_34          .
 601  04F4  FE 07              CP 07H               .
 602  04F6  CA FF FF           JP Z,RMM_35          .
 603  04F9  FE 08              CP 08H               .
 604  04FB  CA E0 05           JP Z,RMM_36          .
 605  04FE  00 90              DEFB 00,90H
 606  0500  11 DF 16     RMM_07 LD DE,MENT24         RUTINA RETURN
 607  0503  06 12              LD B,18
 608  0505  DD 21 0B 05        LD IX,RMM_00         --- RETORNO ---
 609  0509  18 33              JR REG_75-03         ENVIA MENSAJE 74
 610  050B  37           RMM_00 SCF
 611  050C  3F                 CCF                  RESETEA EL CARRY
 612  050D  21 CC FF           LD HL,0FFCCH         DIRECCION DE DATOS HEXA
 613  0510  11 D1 FF           LD DE,0FFD1H         DIRECCION DE BYTES
 614  0513  06 04              LD B,04H             NUMERO DE CONVERSIONES
 615  0515  CD A8 0C           CALL HEX_BY          LLAMA A HEX-BY
 616  0518  2A D3 FF           LD HL,(0FFD3H)       CARGA HL CON DIR. MAXIMA
 617  051B  ED 5B D5 FF        LD DE,(0FFD5H)       CARGA DE CON DIR. DE INICIO
 618  051F  ED 52              SBC HL,DE            EFECTUA LA DIFERENCIA
 619  0521  44                 LD B,H               CARGA EN "BC" EL NUMERO
 620  0522  4D                 LD C,L               DE TRASPASOS
 621  0523  03                 INC BC               INCREMENTA EL CONTADOR
 622  0524  EB                 EX DE,HL             CARGA "HL" CON DIR. DE INICIO
 623  0525  ED 5B D1 FF        LD DE,(0FFD1H)       CARGA "DE" CON DIR. DESTINO
 624  0529  1B                 DEC DE
 625  052A  13           INCREM INC DE               INCREMENTA EL DESTINO
 626  052B  7E                 LD A,(HL)            ATRAPA EL DATO
 627  052C  12                 LD (DE),A            LO CARGA EN "(DE)"
 628  052D  1A                 LD A,(DE)            COMPRUEBA LA CARGA
 629  052E  ED A1              CPI                  CONTRA EL DATO ORIGINAL
 630  0530  20 55              JR NZ,ERROR          SI (DE) DIF. (HL), VA A ERROR
 631  0532  EA 2A 05            JP PE,INCREM        SI "BC" DIF. DE 00, VA A INCREM
 632  0535  11 D7 16            LD DE,MENT25        RUTINA DE CONCLUSION
 633  0538  06 61              LD B,97
 634  053A  DD 21 41 05        LD IX,REG_75         --- RETORNO ---
 635  053E  C3 0C 0C           JP MENTER            ENVIA MENSAJE 75
 636  0541  21 C0 FF     REG_75 LD HL,0FFC0H
 637  0544  36 80              LD (HL),80H          CARGA BANDERA INDICADORA
 638  0546  C3 76 04           JP REG_73            BRINCA A REG_73
 639  0549  3A C0 FF     RMM_09 LD A,(0FFC0H)       COMPARA CON BANDERA INDICADORA
 640  054C  FE 80              CP 80H               Y BRINCA A RUTINA
 641  054E  28 26              JR Z,RMM_10-11       .
 642  0550  FE 81              CP 81H               .
 643  0552  CA FF FF            JP Z,RMM_40         .
```

```
644  0555  FE 82              CP 82H              .
645  0557  CA FF FF           JP Z,RMM_41         .
646  055A  FE 83              CP 83H              .
647  055C  CA FF FF           JP Z,RMM_42         .
648  055F  FE 84              CP 84H              .
649  0561  CA FF FF           JP Z,RMM_43         .
650  0564  FE 85              CP 85H              .
651  0566  CA FF FF           JP Z,RMM_44         .
652  0569  FE 86              CP 86H              .
653  056B  CA FF FF           JP Z,RMM_45         .
654  056E  FE 87              CP 87H              .
655  0570  CA FF FF           JP Z,RMM_46         .
656  0573  C3 90 04           JP RMM_03-07     BRINCA A RUTINA DE ERROR
657  0576  11 D4 12           LD DE,MENT12-04  ENVIA UN "ESC , 2"
658  0579  06 03              LD B,03H
659  057B  DD 21 01 05        LD IX,RMM_10     --- RETORNO ---
660  057F  18 BD              JR REG_75-03
661  0581  21 1C 0D        RMM_10 LD HL,0D1CH    CARGA "HL" CON DIR. PARA TABLA
662  0584  C3 0B 01           JP REG_EX-13     BRINCA A RUTINA DE SELECCION
663  0587  11 D7 16        ERROR  LD DE,MENT75     CARGA MENSAJE 75
664  058A  06 40              LD B,64
665  058C  DD 21 92 05        LD IX,REG_76     --- RETORNO ---
666  0590  18 AC              JR REG_75-03
667  0592  11 38 17         REG_76 LD DE,MENT76     CARGA MENSAJE 76
668  0595  06 21              LD B,33
669  0597  DD 21 41 05        LD IX,REG_75     --- RETORNO ---
670  059B  18 A1              JR REG_75-03

672                           ;RUTINA DE COMPUTADORA A MEMORIA.

674  059D  FD 21 A4 05     COM_MM LD IY,RCM_01
675  05A1  C3 65 0D           JP M_OPC1
676  05A4  FD 21 AB 05     RCM_01 LD IY,RCM_02
677  05A8  C3 78 0D           JP M_OPC3
678  05AB  00 00 00        RCM_02 DEFB 00,00,00
679  05AE  00 00              DEFB 00,00
680  05B0  00 00 00 00        DEFB 00,00,00,00
681  05B4  C3 00 08           JP JMP_00
682  05B7  11 C5 FF        RCM_03 LD DE,0FFC5H
683  05BA  3E 4B              LD A,4BH
684  05BC  12                 LD (DE),A
685  05BD  13                 INC DE
686  05BE  3E 20              LD A,20H
687  05C0  12                 LD (DE),A
688  05C1  13                 INC DE
689  05C2  06 06              LD B,06H
690  05C4  3E 0B              LD A,0BH
691  05C6  32 C0 FF           LD (0FFC0H),A
692  05C9  21 CF FF           LD HL,0FFCFH
693  05CC  70                 LD (HL),B
694  05CD  3E 10              LD A,10H
695  05CF  32 D0 FF           LD (0FFD0H),A
```

```
 696  05D2  C3 43 01          JP RET_18
 697  05D5  12          RUT_05 LD (DE),A
 698  05D6  DD 21 B5 01        LD IX,RET_11
 699  05DA  C3 5E 0C           JP SACA_T
 700  05DD  00 00 00           DEFB 00,00,00
 701  05E0  21 C1 FF     RAM_36 LD HL,0FFC1H
 702  05E3  36 00              LD (HL),00
 703  05E5  11 79 17           LD DE,MENT79116
 704  05E8  06 0D              LD B,0DH
 705  05EA  DD 21 F1 05        LD IX,RCM_04
 706  05EE  C3 0C 0C           JP MENTER
 707  05F1  00 C3 62     RCM_04 DEFB 00,0C3H,62H
 708  05F4  06 08              LD B,08H
 709  05F6  DD 21 FC 05        LD IX,RCM_05
 710  05FA  10 F2              JR RCM_04-03
 711  05FC  00 00 C3     RCM_05 DEFB 00,00,0C3H
 712  05FF  0C 08              DEFB 0CH,08H
 713  0601  12                 LD (DE),A
 714  0602  06 09              LD B,09H
 715  0604  DD 21 0B 06        LD IX,RCM_06
 716  0608  C3 10 0C           JP MENCOM
 717  060B  11 06 17     RCM_06 LD DE,MENT79129
 718  060E  06 02              LD B,02H
 719  0610  DD 21 16 06        LD IX,RCM_07
 720  0614  10 F2              JR RCM_06-03
 721  0616  DD 21 44 06  RCM_07 LD IX,RCM_10
 722  061A  21 20 06           LD HL,RCOMP1
 723  061D  C3 B7 0C           JP RC_TIM
 724  0620  DB 02        RCOMP1 IN A,(02H)
 725  0622  FE 0A               CP 0AH
 726  0624  C2 5B 06            JP NZ,RCM_12
 727  0627  21 C1 FF            LD HL,0FFC1H
 728  062A  CB 7E              BIT 7,(HL)
 729  062C  C2 EB 0C           JP NZ,RC_SEC
 730  062F  CB 76              BIT 6,(HL)
 731  0631  C2 FF FF           JP NZ,LFEED2
 732  0634  CB 6E              BIT 5,(HL)
 733  0636  C2 FF FF           JP NZ,LFEED3
 734  0639  CB 66              BIT 4,(HL)
 735  063B  C2 FF FF           JP NZ,LFEED4
 736  063E  21 60 06           LD HL,RCOMP2
 737  0641  C3 CF 0C           JP RECOMP
 738  0644  11 D7 16     RCM_10 LD DE,MENT75
 739  0647  06 40              LD B,64
 740  0649  DD 21 50 06        LD IX,RCM_11
 741  064D  C3 0C 0C           JP MENTER
 742  0650  11 DB 17     RCM_11 LD DE,MENT78
 743  0653  06 22              LD B,34
 744  0655  DD 21 41 05        LD IX,REG_75
 745  0659  10 F2              JR RCM_11-03
 746  065B  21 20 06     RCM_12 LD HL,RCOMP1
 747  065E  18 E1              JR RCM_10-03
 748  0660  00 00 00     RCOMP2 DEFB 00,00,00
 749  0663  C3 71 06           JP 00671H
```

```
     750  0666  FF FF FF              DEFB 0FFH,0FFH,0FFH             *** AGREGADO ***
     751  0669  11 C7 FF              LD DE,0FFC7H                      *
     752  066C  06 06                 LD B,06                           *
     753  066E  C3 F6 05              JP 005F6H                         *
     754  0671  DB 02                 IN A,(02H)          RECIBE EL PRIMER CARACTER
     755  0673  33                    INC SP
     756  0674  33                    INC SP
     757  0675  FE 3A                 CP 3AH
     758  0677  28 01                 JR Z,RCM_00
     759  0679  DD 21 69 07           LD IX,RCM_09
     760  067D  06 04                 LD B,04H
     761  067F  21 C1 FF              LD HL,0FFC1H
     762  0682  36 00                 LD (HL),00H
     763  0684  11 C2 FF              LD DE,0FFC2H
     764  0687  C3 5B 06              JP RCM_12
     765  068A  DD 21 96 06     RCM_00 LD IX,RCM_13        INICIA RUTINA DE CARGA
     766  068E  06 02                 LD B,02H
     767  0690  11 DA FF              LD DE,0FFDAH
     768  0693  C3 EB 0C              JP RC_SEC
     769  0696  FD 21 92 07     RCM_13 LD IY,ERROR5        CARGA ERROR DE ESCRITURA.
     770  069A  06 01                 LD B,01H
     771  069C  1B                    DEC DE
     772  069D  1B                    DEC DE
     773  069E  21 B1 FF              LD HL,0FFB1H
     774  06A1  CD BA 0B              CALL ASC_BY
     775  06A4  7E                    LD A,(HL)
     776  06A5  ED 44                 NEG
     777  06A7  23                    INC HL
     778  06A8  77                    LD (HL),A
     779  06A9  DD 21 B3 06           LD IX,RCM_14
     780  06AD  06 04                 LD B,04H
     781  06AF  18 DF                 JR RCM_13-06
     782  06B1  06 02           RCM_14 LD B,02             CARGA DIRECCION DE INICIO
     783  06B3  1E DA                 LD E,0DAH
     784  06B5  21 D3 FF              LD HL,0FFD3H
     785  06B8  CD BA 0B              CALL ASC_BY
     786  06BB  23                    INC HL
     787  06BC  1E D3                 LD E,0D3H
     788  06BE  1A                    LD A,(DE)
     789  06BF  77                    LD (HL),A
     790  06C0  ED 44                 NEG
     791  06C2  1B                    DEC DE
     792  06C3  47                    LD B,A
     793  06C4  1A                    LD A,(DE)
     794  06C5  80                    ADD A,B
     795  06C6  12                    LD (DE),A
     796  06C7  2B                    DEC HL
     797  06C8  7E                    LD A,(HL)
     798  06C9  ED 44                 NEG
     799  06CB  47                    LD B,A
     800  06CC  1A                    LD A,(DE)
     801  06CD  80                    ADD A,B
     802  06CE  12                    LD (DE),A
     803  06CF  DD 21 D5 06           LD IX,RCM_15
```

```
    804  06D3  18 B9              JR RCM_08+04
    805  06D5  1B                 RCM_15 DEC DE
    806  06D6  1A                 LD A,(DE)
    807  06D7  FE 31              CP 31H
    808  06D9  CA 27 07           JP Z,FIN_LI
    809  06DC  DD 21 E2 06        LD IX,RCM_15+13
    810  06E0  18 AC              JR RCM_08+04
    811  06E2  06 01              LD B,01H            CARGA DE DATOS DESDE LA COMPUTADORA
    812  06E4  1B                 DEC DE
    813  06E5  1B                 DEC DE
    814  06E6  21 D6 FF           LD HL,0FFD6H
    815  06E9  CD BA 0D           CALL ASC_BY
    816  06EC  2A D4 FF           LD HL,(0FFD4H)
    817  06EF  3A D6 FF           LD A,(0FFD6H)
    818  06F2  77                 LD (HL),A
    819  06F3  46                 LD B,(HL)
    820  06F4  23                 INC HL
    821  06F5  22 D4 FF           LD (0FFD4H),HL
    822  06F8  B0                 CP B
    823  06F9  C2 98 07           JP NZ,ERROR6        VA A ERROR DE CARGADO.
    824  06FC  ED 44              NEG
    825  06FE  21 D2 FF           LD HL,0FFD2H
    826  0701  86                 ADD A,(HL)
    827  0702  77                 LD (HL),A
    828  0703  2B                 DEC HL
    829  0704  35                 DEC (HL)
    830  0705  20 B5              JR NZ,RCM_15+07
    831  0707  DD 21 0D 07        LD IX,RCM_16
    832  070B  18 81              JR RCM_08+04
    833  070D  06 01              RCM_16 LD B,01H
    834  070F  1B                 DEC DE
    835  0710  1B                 DEC DE
    836  0711  21 D1 FF           LD HL,0FFD1H
    837  0714  CD BA 0D           CALL ASC_BY
    838  0717  7E                 LD A,(HL)
    839  0718  23                 INC HL
    840  0719  46                 LD B,(HL)
    841  071A  B8                 CP B
    842  071B  C2 9E 07           JP NZ,ERROR7        VA A ERROR DE CHEKSUM.
    843  071E  DD 21 0A 06        LD IX,RCM_0B
    844  0722  06 03              LD B,03H
    845  0724  C3 90 06           JP RCM_08+06
    846  0727  FD 21 59 07        FIN_LI LD IY,RCM_18+15
    847  072B  06 01              LD B,01H
    848  072D  DD 21 34 07        LD IX,RCM_17
    849  0731  C3 90 06           JP RCM_08+06
    850  0734  01 01 00           RCM_17 LD BC,0001H
    851  0737  21 BC 17           LD HL,MENT7A-05
    852  073A  DD 21 65 07        LD IX,RCM_19
    853  073E  CD B0 0D           CALL CP_SEC
    854  0741  06 04              LD B,04H
    855  0743  DD 21 4A 07        LD IX,RCM_18
    856  0747  C3 90 06           JP RCM_08+06
    857  074A  01 04 00           RCM_18 LD BC,0004H
```

```
     858  074D  21 DD 17           LD HL,MENT7A-04
     859  0750  DD 21 65 07        LD IX,RCM_19
     860  0754  CD B0 0D           CALL CP_SEC
     861  0757  FD E9              JP (IY)
     862  0759  DD 21 35 05        LD IX,REG_75-12
     863  075D  11 C1 17           LD DE,MENT7A
     864  0760  06 06              LD B,06
     865  0762  C3 0C 0C           JP MENTER
     866  0765  33           RCM_19 INC SP
     867  0766  33                 INC SP
     868  0767  18 C2              JR RCM_17-09
     869  0769  01 04 00     RCM_09 LD BC,0004H
     870  076C  21 DB 17           LD HL,MENT7A-09
     871  076F  DD 21 A4 07        LD IX,ERROR8        CARGA ERROR DE FORMATO.
     872  0773  CD B0 0D           CALL CP_SEC
     873  0776  DD 21 7C 07        LD IX,RCM_20
     874  077A  18 E1              JR RCM_19-08
     875  077C  11 D7 16     RCM_20 LD DE,MENT75
     876  077F  06 40              LD B,64
     877  0781  DD 21 87 07        LD IX,RCM_21
     878  0785  18 DB              JR RCM_19-03
     879  0787  11 C7 17     RCM_21 LD DE,MENT7A+06
     880  078A  06 22              LD B,34
     881  078C  DD 21 41 05        LD IX,REG_75
     882  0790  18 B0              JR RCM_19-03
     883  0792  00 00 00     ERROR5 DEFB 00,00,00
     884  0795  C3 AB 07           JP ERR05A
     885  0798  00 00 00     ERROR6 DEFB 00,00,00
     886  079B  C3 B9 07           JP ERR06A
     887  079E  00 00 00     ERROR7 DEFB 00,00,00
     888  07A1  C3 BF 07           JP ERR07A
     889  07A4  00 00 00     ERROR8 DEFB 00,00,00
     890  07A7  D9           ERR00A EXX
     891  07A8  21 F9 07           LD HL,CONT_4
     892  07AB  18 04              JR ERR05A+04
     893  07AD  D9           ERR05A EXX
     894  07AE  21 DE 07           LD HL,CONT_1
     895  07B1  D9                 EXX
     896  07B2  FD 21 C5 07        LD IY,RCM_22
     897  07B6  C3 2B 07           JP FIN_LI+04
     898  07B9  D9           ERR06A EXX
     899  07BA  21 E9 07           LD HL,CONT_2
     900  07BD  18 F2              JR ERR05A+04
     901  07BF  D9           ERR07A EXX
     902  07C0  21 F4 07           LD HL,CONT_3
     903  07C3  18 EC              JR ERR05A+04
     904  07C5  11 C1 17     RCM_22 LD DE,MENT7A
     905  07C8  06 06              LD B,06
     906  07CA  DD 21 D1 07        LD IX,RCM_23
     907  07CE  C3 0C 0C           JP MENTER
     908  07D1  11 D7 16     RCM_23 LD DE,MENT75
     909  07D4  06 40              LD B,64
     910  07D6  DD 21 DC 07        LD IX,RCM_24
     911  07DA  18 F2              JR RCM_23-03
```

```
     912  07DC  D9              RCM_24 EXX
     913  07DD  E9                     JP (HL)
     914  07DE  11 92 0F       CONT_1 LD DE,MENT7B
     915  07E1  06 24                  LD B,36
     916  07E3  DD 21 41 05             LD IX,REG_75
     917  07E7  18 E5                   JR RCM_23-03
     918  07E9  11 B6 0F       CONT_2 LD DE,MENT7C-02
     919  07EC  06 02                  LD B,02
     920  07EE  DD 21 92 05             LD IX,REG_74
     921  07F2  18 DA                   JR RCM_23-03
     922  07F4  11 B8 0F       CONT_3 LD DE,MENT7C
     923  07F7  18 E8                   JR CONT_1403
     924  07F9  11 DC 0F       CONT_4 LD DE,MENT7D
     925  07FC  18 E3                   JR CONT_1403
     926  07FE                          DEFS 2
     927  0000  11 59 17       JMP_00 LD DE,MENT77
     928  0003  06 10                  LD B,010H
     929  0005  DD 21 B7 05             LD IX,RCM_03
     930  0009  C3 0C 0C                JP MENTER
     931  000C                          DEFS 3
     932  000F  11 AA 17                LD DE,MENT7A-23
     933  0012  06 0E                   LD B,00EH
     934  0014  DD 21 1D 08             LD IX,CONT99
     935  0018  C3 0C 0C                JP MENTER
     936  001B  11 C4 FF       CONT99 LD DE,0FFC4H
     937  001E  3E 3F                   LD A,03FH
     938  0020  C3 01 06                JP RCM_0545
```

```
940                     ********************************************************************
941                     ********************  SUBRUTINAS AUXILIARES  ***********************
942                     ********************************************************************
943                             ORG 0C00H

945  0C00  1A           SUBT01 LD A,(DE)        'TRANSMISION NO. 1'.
946  0C01  D3 02                OUT (02H),A      ENVIA DATO A PTO. 02.
947  0C03  13                   INC DE
948  0C04  05                   DEC B            DECREMENTA CONTADOR.
949  0C05  28 62                JR Z,RETN01      SI B=00, BRINCA A VUELVE.
950  0C07  CD 79 0C             CALL TIEMPO      LLAMA A RUTINA DE TIEMPO.
951  0C0A  18 0E                JR ESPERA
952  0C0C  3E 31        MENTER  LD A,31H         ENVIA MENSAJE A TERMINAL.
953  0C0E  18 02                JR TRANSM
954  0C10  3E 13        MENCOM  LD A,13H         ENVIA MENSAJE A COMPUTADORA.
955  0C12  21 00 0C     TRANSM  LD HL,SUBT01
956  0C15  22 F2 FE     ENVIA   LD (0FEF2H),HL   HABILITA TRANSMISION EN SUBT01.
957  0C18  D3 03        SACAR   OUT (03H),A
958  0C1A  3E 05        ESPERA  LD A,05H
959  0C1C  D3 06                OUT (06H),A      HABILITA CPI.
960  0C1E  FB                   EI               HABILITA INTERRUPCIONES.
961  0C1F  C9                   RET
962  0C20  22 F0 FE     RECTER  LD (0FFF0H),HL
963  0C23  3E 34                LD A,34H
964  0C25  18 F1                JR SACAR
965  0C27  D9           DET_PO  EXX              DETECTA POSICION EN LA TERMINAL.
966  0C28  11 69 11             LD DE,MENT02
967  0C2B  06 04                LD B,04H
968  0C2D  DD 21 33 0C          LD IX,REG_02
969  0C31  18 D9        MENT_X  JR MENTER
970  0C33  11 DE FF     REG_02  LD DE,0FFDEH
971  0C36  06 02                LD B,02H
972  0C38  21 3D 0C     RECT_2  LD HL,RECT02
973  0C3B  18 E3                JR RECTER
974  0C3D  DB 02        RECT02  IN A,(02H)
975  0C3F  12                   LD (DE),A        SALVA LA POSICION.
976  0C40  13                   INC DE
977  0C41  05                   DEC B
978  0C42  20 F4                JR NZ,RECT_2
979  0C44  D9           REG_04  EXX
980  0C45  FD E9                JP (IY)
981  0C47  D9           SET_PO  EXX              PONE CURSOR EN TERMINAL.
982  0C48  11 78 11             LD DE,MENT05
983  0C4B  06 04                LD B,04H
984  0C4D  DD 21 53 0C          LD IX,REG_05
985  0C51  18 DE                JR MENT_X
986  0C53  11 DE FF     REG_05  LD DE,0FFDEH
987  0C56  06 02                LD B,02H
988  0C58  DD 21 44 0C          LD IX,REG_04
989  0C5C  18 D3                JR MENT_X
990  0C5E  08           SACA_T  EX AF,AF'        ENVIA DATO A TERMINAL.
```

```
 991  0C5F  21 46 0C          LD HL,SUBT02
 992  0C62  3F 31             LD A,31H
 993  0C64  18 AF             JR ENVIA
 994  0C66  08                SUBT02 EX AF,AF'
 995  0C67  D3 02             OUT (02H),A
 996  0C69  DD E9             RETN01 JP (IX)
 997  0C6B  0E B0             PONE_T LD C,B0H         PONE MENSAJE 'T, I,'
 998  0C6D  D9                EXX
 999  0C6E  11 6D 11          LD DE,MENT03
1000  0C71  06 0B             LD B,0BH
1001  0C73  DD 21 44 0C       LD IX,REG_04
1002  0C77  18 B9             JR MENT_X
1003  0C79  26 B0             TIEMPO LD H,B0H         RUTINA DE TIEMPO.
1004  0C7B  25                LOOP03 DEC H
1005  0C7C  20 FD             JR NZ,LOOP03
1006  0C7E  C9                RET
1007  0C7F  CB D9             BORRAS RES 7,C          BORRA 'TECLA INVALIDA'.
1008  0C81  D9                EXX
1009  0C82  11 7C 11          LD DE,MENT06
1010  0C85  06 0B             LD B,0BH
1011  0C87  DD 21 44 0C       LD IX,REG_04
1012  0C8B  18 A4             JR MENT_X
1013  0C8D  FE 47             CP_ASC CP 47H           COMPARA SI ES CARACTER ASCII.
1014  0C8F  30 0F             JR NC,NO_ASC
1015  0C91  FE 41             CP 41H
1016  0C93  30 0C             JR NC,VALIDA
1017  0C95  FE 3A             CP 3AH
1018  0C97  30 07             JR NC,NO_ASC
1019  0C99  FE 30             CP 30H
1020  0C9D  38 03             JR C,NO_ASC
1021  0C9D  E6 0F             AND 0FH
1022  0C9F  C9                RET
1023  0CA0  33                NO_ASC INC SP           SI NO ES ASCII,
1024  0CA1  33                INC SP                  INCNCA A RUTINA DE ERROR.
1025  0CA2  E9                JP (HL)
1026  0CA3  C6 09             VALIDA ADD A,09H        PARA CARACTERES A,F.
1027  0CA5  E6 0F             AND 0FH
1028  0CA7  C9                LISTO  RET
1029  0CA8  7E                HEX_BY LD A,(HL)        RUTINA PARA CONVERTIR
1030  0CA9  2B                DEC HL                  DATOS HEXADECIMALES EN BYTES
1031  0CAA  ED 6F             RLD                     TAMBIEN HEXADECIMALES.
1032  0CAC  7E                LD A,(HL)
1033  0CAD  12                LD (DE),A
1034  0CAE  13                INC DE
1035  0CAF  05                DEC B
1036  0CB0  28 F5             JR Z,LISTO
1037  0CB2  2B                DEC HL
1038  0CB3  18 F3             JR HEX_BY
1039  0CB5  0E 02             BY_ASC LD C,02H         RUTINA PARA CONVERTIR
1040  0CB7  3E 30             LD A,30H                BYTES A CARACTERES ASCII.
1041  0CB9  ED 6F             REPITE RLD
1042  0CBB  FE 3A             CP 3AH
1043  0CBD  D4 CC 0C          CALL NC,VALORA
1044  0CC0  12                LD (DE),A
```

```
    1045  0CC1  3E 30                    LD A,30H
    1046  0CC3  13                       INC DE
    1047  0CC4  0D                       DEC C
    1048  0CC5  20 F2                    JR NZ,REPITE
    1049  0CC7  2B                       DEC HL
    1050  0CC8  05                       DEC B
    1051  0CC9  20 EA                    JR NZ,BY_ASC
    1052  0CCB  C9                       RET
    1053  0CCC  C6 07             VALORA ADD A,07H          REAJUSTA VALOR ASCII.
    1054  0CCE  C9                       RET
    1055  0CCF  3E 16             RECOMP LD A,16H
    1056  0CD1  22 F0 FE                 LD (0FEF0H),HL
    1057  0CD4  C3 10 0C                 JP SACAR
    1058  0CD7  CD CF 0C          RC_TIM CALL RECOMP
    1059  0CDA  0E 0D                    LD C,0DH
    1060  0CDC  26 FF                    LD H,0FFH
    1061  0CDE  2E FF                    LD L,0FFH
    1062  0CE0  2B                       DEC L
    1063  0CE1  20 FD                    JR NZ,RC_TIM109
    1064  0CE3  25                       DEC H
    1065  0CE4  20 F8                    JR NZ,RC_TIM107
    1066  0CE6  0D                       DEC C
    1067  0CE7  20 F3                    JR NZ,RC_TIM105
    1068  0CE9  DD E9                    JP (IX)
    1069  0CEB  21 F0 0C          RC_SEC LD HL,RC_SEC105
    1070  0CEE  18 BF                    JR RECOMP
    1071  0CF0  DB 02                    IN A,(02H)
    1072  0CF2  12                       LD (DE),A
    1073  0CF3  13                       INC DE
    1074  0CF4  05                       DEC B
    1075  0CF5  20 F4                    JR NZ,RC_SEC
    1076  0CF7  DD E9                    JP (IX)
    1077  0CF9                          DEFS 7
```

```
1079                            ****************  TABLAS Y RUTINAS AUXILIARES  ****************

1081  0D00  C3 42 0D                   JP PRG_EP       ; "PROGRAMADOR DE EPROM'S".
1082  0D03                             DEFS 13
1083  0D10  C3 1E 01                   JP EX_MEM       ; "EXAMEN DE MEMORIA".
1084  0D13  FF                         DEFB OFFH
1085  0D14  C3 20 0D                   JP EXM_RG       ; "EXAMEN DE REGISTROS".
1086  0D17  FF                         DEFB OFFH
1087  0D18  C3 2C 0D                   JP EXM_PT       ; "EXAMEN DE PUERTOS".
1088  0D1B  FF                         DEFB OFFH
1089  0D1C  C3 37 0D                   JP TRS_AR       ; "TRASLADO DE ARCHIVOS".
1090  0D1F  FF                         DEFB OFFH
1091  0D20  11 D4 14          EXM_RG LD DE,MENT30     PRESENTA EXAMEN DE REGISTROS.
1092  0D23  06 26                    LD B,26H
1093  0D25  DD 21 4D 0D              LD IX,REG_30     RETORNA EN REG 30.
1094  0D29  C3 0C 0C          MENT   JP MENTER
1095  0D2C  11 FA 14          EXM_PT LD DE,MENT50     PRESENTA EXAMEN DE PUERTOS.
1096  0D2F  06 22                    LD B,22H
1097  0D31  DD 21 53 0D              LD IX,REG_50     RETORNA EN REG_50.
1098  0D35  18 F2                    JR MENT
1099  0D37  11 1C 15          TRS_AR LD DE,MENT70     PRESENTA TRASLADO DE ARCHIVOS.
1100  0D3A  06 20                    LD B,20H
1101  0D3C  DD 21 59 0D              LD IX,REG_70     RETORNA EN REG_70.
1102  0D40  18 E7                    JR MENT
1103  0D42  11 44 15          PRG_EP LD DE,MENT80     PRESENTA PROGRAMADOR DE EPROMS.
1104  0D45  06 2C                    LD B,2CH
1105  0D47  DD 21 5F 0D              LD IX,REG_80     RETORNA EN REG_80.
1106  0D4B  18 BC                    JR MENT
1107  0D4D  00 00 00          REG_30 DEFB 00,00,00
1108  0D50  C3 00 F6                 JP REX_02
1109  0D53  C3 00 F0          REG_50 JP IN_RAM
1110  0D56  FF FF FF                 DEFB OFFH,OFFH,OFFH
1111  0D59  00 00 00          REG_70 DEFB 00,00,00
1112  0D5C  C3 E0 03                 JP TRASLA
1113  0D5F  C3 00 F0          REG_80 JP IN_RAM
1114  0D62  FF FF FF                 DEFB OFFH,OFFH,OFFH

1116                            ;OPCIONES DE MENSAJES:

1118  0D65  11 1C 16          M_OPC1 LD DE,OPC_01     FUENTE Y DESTINO.
1119  0D68  06 1D                    LD B,29
1120  0D6A  DD 21 AE 0D              LD IX,RG_OPC
1121  0D6E  C3 0C 0C                 JP MENTER
1122  0D71  11 39 16          M_OPC2 LD DE,OPC_02     DIRECCIONES (DESTINO).
1123  0D74  06 2B                    LD B,43
1124  0D76  18 F2                    JR M_OPC1+05
1125  0D78  11 64 16          M_OPC3 LD DE,OPC_03     ARCHIVO (FUENTE).
1126  0D7B  06 22                    LD B,34
1127  0D7D  18 EB                    JR M_OPC1+05
```

```
1128  0D7F  11 39 16          M_OPC4 LD DE,OPC_02     ARCHIVO (DESTINO).
1129  0D82  06 04                    LD B,04H
1130  0D84  DD 21 0A 0D              LD IX,RG_OP4
1131  0D88  18 E4                    JR M_OPC1+09
1132  0D8A  11 68 16          RG_OP4 LD DE,OPC_03+04
1133  0D8D  06 1E                    LD B,30
1134  0D8F  18 D9                    JR M_OPC1+05
1135  0D91  11 64 16          M_OPC5 LD DE,OPC_03     DIRECCIONES (FUENTE).
1136  0D94  06 04                    LD B,04H
1137  0D96  DD 21 9C 0D              LD IX,RG_OP5
1138  0D9A  18 D2                    JR M_OPC1+09
1139  0D9C  11 38 16          RG_OP5 LD DE,OPC_02+04
1140  0D9F  06 27                    LD B,39       ;27H#
1141  0DA1  DD 21 A7 0D              LD IX,RG_50P
1142  0DA5  18 C7                    JR M_OPC1+09
1143  0DA7  11 86 16          RG_50P LD DE,OPC_03+34
1144  0DAA  06 0C                    LD B,12
1145  0DAC  18 BC                    JR M_OPC1+05
1146  0DAE  FD E9             RG_OPC JP (IY)
1147  0DB0  1B                CP_SEC DEC DE
1148  0DB1  1A                       LD A,(DE)
1149  0DB2  ED A1                    CPI
1150  0DB4  C2 E9 0C                 JP NZ,RC_SEC-02
1151  0DB7  C0                       RET PO
1152  0DB8  18 F6                    JR CP_SEC
1153  0DBA  0E 02             ASC_BY LD C,02H
1154  0DBC  1A                       LD A,(DE)
1155  0DBD  D9                       EXX
1156  0DBE  21 D0 0D                 LD HL,ASC_BY+22
1157  0DC1  CD 0D 0C                 CALL CP_ASC
1158  0DC4  D9                       EXX
1159  0DC5  ED 6F                    RLD
1160  0DC7  13                       INC DE
1161  0DC8  0D                       DEC C
1162  0DC9  20 F1                    JR NZ,ASC_BY+02
1163  0DCB  05                       DEC B
1164  0DCC  C0                       RET Z
1165  0DCD  23                       INC HL
1166  0DCE  18 EA                    JR ASC_BY
1167  0DD0  FD E9                    JP (IY)

1169                          ;TABLA DE TRASLADOS:
1170  0DD2                           DEFS 62

1172  0E10  C3 9D 05                 JP COM_MM
1173  0E13  FF                       DATA OFFH
1174  0E14  C3 1E 04                 JP COM_CP
1175  0E17  FF                       DATA OFFH
1176  0E18  C3 FF FF                 JP COM_GR
1177  0E1B  FF                       DATA OFFH
1178  0E1C  C3 FF FF                 JP COM_DS
1179  0E1F  FF                       DATA OFFH
```

```
1180  0E20  C3 FF FF        JP GRB_MM
1181  0E23  FF              DATA OFFH
1182  0E24  C3 FF FF        JP GRB_CP
1183  0E27  FF              DATA OFFH
1184  0E28  C3 FF FF        JP GRB_GR
1185  0E2B  FF              DATA OFFH
1186  0E2C  C3 FF FF        JP GRB_DS
1187  0E2F  FF              DATA OFFH
1188  0E30  C3 FF FF        JP DSK_MM
1189  0E33  FF              DATA OFFH
1190  0E34  C3 FF FF        JP DSK_CP
1191  0E37  FF              DATA OFFH
1192  0E38  C3 FF FF        JP DSK_GR
1193  0E3B  FF              DATA OFFH
1194  0E3C  C3 FF FF        JP DSK_DS
1195  0E3F  FF              DATA OFFH
1196  0E40  C3 38 04        JP MEM_MM
1197  0E43  FF              DATA OFFH
1198  0E44  C3 FF FF        JP MEM_CP
1199  0E47  FF              DATA OFFH
1200  0E48  C3 FF FF        JP MEM_GR
1201  0E4B  FF              DATA OFFH
1202  0E4C  C3 FF FF        JP MEM_DS
```

```
    1204                        ;*****************************************************
    1205                        ;***************   MENSAJES DEL SISTEMA GR-1000   ***************
    1206                        ;*****************************************************
    1207                                ORG 00F92H

    1209  0F92  1B 27 1B 3D    MENT2B DEFB 1BH,'''',1BH,'=','0','8'
    1210  0F96  30 3B
    1211  0F98  20 2A 2A 2A            DEFM ' **** ERROR DE ESCRITURA **** '
    1212  0F9C  2A 20 45 52
    1213  0FA0  52 4F 52 20
    1214  0FA4  44 45 20 45
    1215  0FA8  53 43 52 49
    1216  0FAC  54 55 52 41
    1217  0FB0  20 2A 2A 2A
    1218  0FB4  2A 20
    1219  0FB6  1B 27                DEFB 1BH,''''
    1220  0FB8  1B 37 1B 3D    MENT7C DEFB 1BH,'''',1BH,'=','0','8'
    1221  0FBC  30 3B
    1222  0FBE  20 2A 2A 2A            DEFM ' ****  ERROR DE CHEKSUM  **** '
    1223  0FC2  2A 20 20 45
    1224  0FC6  52 52 4F 52
    1225  0FCA  20 44 45 20
    1226  0FCE  43 48 45 4B
    1227  0FD2  53 55 4D 20
    1228  0FD6  20 2A 2A 2A
    1229  0FDA  2A 20
    1230  0FDC  1B 27 1B 3D    MENT7D DEFB 1BH,'''',1BH,'=','0','8'
    1231  0FE0  30 3B
    1232  0FE2  2A 20 45 4C            DEFM '* EL ARCHIVO NO ES HEX-INTEL *'
    1233  0FE6  20 41 52 43
    1234  0FEA  48 49 56 4F
    1235  0FEE  20 4E 4F 20
    1236  0FF2  45 53 20 48
    1237  0FF6  45 58 2D 49
    1238  0FFA  4E 54 45 4C
    1239  0FFE  20 2A
    1240  1000  1B 2A 1B 29    MENF00 DEFB 1BH,2AH,1BH,')',1AH,1BH,'G','4'
    1241  1004  1A 1B 47 34
    1242  1008  1B 3D 20 6E            DEFB 1BH,'=',' ',6EH,1BH,'G','4'
    1243  100C  1B 47 34
    1244  100F  1B 47 32 1B            DEFB 1BH,'G','2',1BH,'=','"',' '
    1245  1013  3D 22 20
    1246  1016  1B 47 34 1B            DEFB 1BH,'G','4',1BH,'=','"',6EH
    1247  101A  3D 22 6E
    1248  101D  1B 47 34 1B            DEFB 1BH,'G','4',1BH,'G','2'
    1249  1021  47 32
    1250  1023  1B 3D 21 20            DEFB 1BH,'=','!',' ',1BH,'G','4'
    1251  1027  1B 47 34
    1252  102A  20 20 53 20            DEFM '  S I S T E M A   M O N I T O R   '
    1253  102E  49 20 53 20
    1254  1032  54 20 4F 20
```

```
1255  1036  4D 20 41 20
1256  103A  20 20 4D 20
1257  103E  4F 20 4E 20
1258  1042  49 20 54 20
1259  1046  4F 20 52 20
1260  104A  20 20
1261  104C  59 20 20 20        DEFM 'Y  PROGRAMADOR Z80  V1.0 '
1262  1050  50 20 52 20
1263  1054  4F 20 47 20
1264  1058  52 20 41 20
1265  105C  4D 20 41 20
1266  1060  44 20 4F 20
1267  1064  52 20 20 20
1268  1068  5A 20 38 20
1269  106C  30 20 20 20
1270  1070  56 20 31 2E
1271  1074  30 20 20
1272  1077  1B 47 34 1B        DEFB 1BH,'G','4',1BH,'G','2',1BH,'=','(','8'
1273  107B  47 32 1B 3D
1274  107F  28 40
1275  1081  45 58 41 4D        DEFM 'EXAMEN DE MEMORIA'
1276  1085  45 4E 20 44
1277  1089  45 20 4D 45
1278  108D  4D 4F 52 49
1279  1091  41
1280  1092  1B 3D 2A 40        DEFB 1BH,'=','*','@'
1281  1096  45 58 41 4D        DEFM 'EXAMEN DE REGISTROS'
1282  109A  45 4E 20 44
1283  109E  45 20 52 45
1284  10A2  47 49 53 54
1285  10A6  52 4F 53
1286  10A9  5C 52 55 4E        DEFB 5CH,'R','U','N'
1287  10AD  1B 3D 2C 40        DEFB 1BH,'=',',','@'
1288  10B1  45 58 41 4D        DEFM 'EXAMEN DE PUERTOS'
1289  10B5  45 4E 20 44
1290  10B9  45 20 50 55
1291  10BD  45 52 54 4F
1292  10C1  53
1293  10C2  1B 3D 2E 40        DEFB 1BH,'=','.','@'
1294  10C6  54 52 41 53        DEFM 'TRASLADO DE ARCHIVOS'
1295  10CA  4C 41 44 4F
1296  10CE  20 44 45 20
1297  10D2  41 52 43 48
1298  10D6  49 56 4F 53
1299  10DA  1B 3D 30 40        DEFB 1BH,'=','0','@'
1300  10DE  50 52 4F 47        DEFM 'PROGRAMADOR DE EPROMS'
1301  10E2  52 41 4D 41
1302  10E6  44 4F 52 20
1303  10EA  44 45 20 45
1304  10EE  50 52 4F 4D
1305  10F2  53
1306  10F3  1B 3D 34 29        DEFB 1BH,'=','4',')'
1307  10F7  53 20 3D 20        DEFM 'S = SIGU'
1308  10FB  53 49 47 55
```

```
    1309  10FF  49 45 4E 54       MENT01 DEFM 'IENTE'
    1310  1103  45
    1311  1104  1B 3D 34 42              DEFB 1BH,'=','4','B'
    1312  1108  50 20 3D 20              DEFM 'P = PREVIO'
    1313  110C  50 52 45 56
    1314  1110  49 4F
    1315  1112  1B 3D 34 56              DEFB 1BH,'=','4','V'
    1316  1116  43 20 3D 20              DEFM 'C = SELECCION'
    1317  111A  53 45 4C 45
    1318  111E  43 43 49 4F
    1319  1122  4F
    1320  1123  1B 3D 36 3D              DEFB 1BH,'=','6','=',1BH,'G','1'
    1321  1127  1B 47 31
    1322  112A  54 20 45 20              DEFM 'T E C L A   I N V A L I D A'
    1323  112E  43 20 4C 20
    1324  1132  41 20 20 20
    1325  1136  49 20 4E 20
    1326  113A  56 20 41 20
    1327  113E  4C 20 49 20
    1328  1142  44 20 41
    1329  1145  1B 47 30 1B              DEFB 1BH,'G','0',1BH,'('
    1330  1149  28
    1331  114A  1B 3D 28 3E              DEFB 1BH,'=','(','>','1'
    1332  114E  31
    1333  114F  1B 3D 29 3E              DEFB 1BH,'=',')','>','2'
    1334  1153  32
    1335  1154  1B 3D 2C 3E              DEFB 1BH,'=',',','>','3'
    1336  1158  33
    1337  1159  1B 3D 2E 3E              DEFB 1BH,'=','.','>','4'
    1338  115D  34
    1339  115E  1B 3D 30 3E              DEFB 1BH,'=','0','>','5'
    1340  1162  35
    1341  1163  1B 26 09 1B              DEFB 1BH,'&',09H,1BH,',','2'
    1342  1167  2C 32
    1343  1169  1B 42 1B 3F       MENT02 DEFB 1BH,'B',1BH,'?'
    1344  116D  1B 27 1B 3D       MENT03 DEFB 1BH,'''',1BH,'=','6','='
    1345  1171  36 3D
    1346  1173  1B 47 32 1B              DEFB 1BH,'G','2',1BH,'&'
    1347  1177  26
    1348  1178  1B 43 1B 3D       MENT05 DEFB 1BH,'C',1BH,'='
    1349  117C  1B 27 1B 3D       MENT06 DEFB 1BH,'''',1BH,'=','6','='
    1350  1180  36 3D
    1351  1182  1B 47 31 1B              DEFB 1BH,'G','1',1BH
    1352  1186  26 1B 2A          MENT10 DEFB '&',1BH,2AH
    1353  1189  1B 29 1A 1B              DEFB 1BH,')',1AH,1BH,'=',' ','>'
    1354  118D  3D 20 3E
    1355  1190  1B 47 34 1B              DEFB 1BH,'G','4',1BH,'=',' ',6EH
    1356  1194  3D 20 6E
    1357  1197  1B 47 34 1B              DEFB 1BH,'G','4',1BH,'G','2'
    1358  119B  47 32
    1359  119D  1B 3D 22 3E              DEFB 1BH,'=','"','>',1BH,'G','4'
    1360  11A1  1B 47 34
    1361  11A4  1B 3D 22 6E              DEFB 1BH,'=','"',6EH,1BH,'G','4'
    1362  11A8  1B 47 34
```

```
1363  11AB  1B 47 30 1B        DEFB 1BH,'G','0',1BH,'=','I',' '
1364  11AF  3D 21 20
1365  11B2  1B 47 31           DEFB 1BH,'G','1'
1366  11B5  44 49 52 45        DEFM 'DIRECCION'
1367  11B9  43 43 49 4F
1368  11BD  4E
1369  11BE  1B 3D 21 2D        DEFB 1BH,'=','!','-'
1370  11C2  44 41 54 4F        DEFM 'DATO'
1371  11C6  1B 3D 21 3E        DEFB 1BH,'=','!','>',1BH,'G','4'
1372  11CA  1B 47 34
1373  11CD  1B 3D 21 46        DEFB 1BH,'=','!','F'
1374  11D1  45 20 58 20        DEFM 'E X A M E N  D E  M E M O R I A'
1375  11D5  41 20 4D 20
1376  11D9  45 20 4F 20
1377  11DD  20 20 44 20
1378  11E1  45 20 20 20
1379  11E5  4D 20 45 20
1380  11E9  4D 20 4F 20
1381  11ED  52 20 49 20
1382  11F1  41
1383  11F2  1B 3D 21 6E        DEFB 1BH,'=','!',6EH,1BH,'G','4'
1384  11F6  1B 47 34
1385  11F9  1B 47 32 1B        DEFB 1BH,'G','2',1BH,'=','$','T'
1386  11FD  3D 24 54
1387  1200  4D 20 45 20        DEFM 'M E N U :'
1388  1204  4E 20 55 20
1389  1208  3A
1390  1209  1B 3D 2A 3E        DEFB 1BH,'=','*','>'
1391  120D  50 4F 53 49        DEFM 'POSICION'
1392  1211  43 49 4F 4E
1393  1215  1B 3D 2A 54        DEFB 1BH,'=','*','T'
1394  1219  4F 20 3D 20        DEFB 'O',' ','=',' ','A',64H,65H,6CH
1395  121D  41 64 65 6C
1396  1221  61 6E 74 61        DEFB 61H,6CH,74H,61H,72H,' ','N','P'
1397  1225  72 20 4E 4E
1398  1229  1B 3D 2B 54        DEFB 1BH,'=','I','T'
1399  122D  4D 20 3D 20        DEFB 'M',' ','=',' ','M'
1400  1231  4D
1401  1232  6F 6E 69 74        DEFB 6FH,6EH,69H,74H,6FH,72H
1402  1236  6F 72
1403  1238  1B 3D 2C 54        DEFB 1BH,'=',',','T'
1404  123C  51 20 3D 20        DEFB 'Q',' ','=',' ','D',69H,72H
1405  1240  44 69 72
1406  1243  65 63 63 69        DEFB 65H,63H,63H,69H,6FH,6EH
1407  1247  6F 6E
1408  1249  1B 3D 30 54        DEFB 1BH,'=','0','T'
1409  124D  52 20 3D 20        DEFB 'R',' ','=',' ','R',75H,6EH
1410  1251  52 75 6E
1411  1254  1B 3D 2D 54        DEFB 1BH,'=','-','T'
1412  1258  50 20 3D 20        DEFB 'P',' ','=',' ','P'
1413  125C  50
1414  125D  72 65 76 69        DEFB 72H,65H,76H,69H,6FH
1415  1261  6F
1416  1262  1B 3D 2E 54        DEFB 1BH,'=','.','T'
```

```
1417  1266  49 20 3D 20          DEFB 'I',' ','=',' ','R',65H,74H
1418  126A  52 65 74
1419  126D  72 61 73 61          DEFB 72H,61H,73H,61H,72H,' ',6EH,6EH
1420  1271  72 20 6E 6E
1421  1275  1B 3D 2F 54          DEFB 1BH,'=','/','T'
1422  1279  4E 20 3D 20          DEFB 'N',' ','=',' ','S',69H,67H
1423  127D  53 69 67
1424  1280  75 69 65 6E          DEFB 75H,69H,65H,6EH,74H,65H
1425  1284  74 65
1426  1286  1B 3D 36 3D    MENT11 DEFB 1BH,'=','6','=',1BH,'G','1'
1427  128A  1B 47 31
1428  128D  54 20 45 20          DEFM 'T E C L A  I N V A L I D A'
1429  1291  43 20 4C 20
1430  1295  41 20 20 20
1431  1299  49 20 4E 20
1432  129D  56 20 41 20
1433  12A1  4C 20 49 20
1434  12A5  44 20 41
1435  12A8  1B 28 1B 3D          DEFB 1BH,'(',1BH,'=','*','I'
1436  12AC  2A 49
1437  12AE  50 50 50 50          DEFM 'XXXX'
1438  12B2  1B 26 1B 2C          DEFB 1BH,'&',1BH,',','2',09H
1439  12B6  32 09
1440  12B8  1B 27 1B 29    MENT12 DEFB 1BH,'''',1BH,')'
1441  12BC  1B 3D 2A 3E          DEFB 1BH,'=','*','>',1BH,'G','1'
1442  12C0  1B 47 31
1443  12C3  1B 3D 2A 49          DEFB 1BH,'=','*','I'
1444  12C7  20 20 20 20          DEFM '    '
1445  12CB  1B 47 30 1B          DEFB 1BH,'G','0',1BH,'=','1',' '
1446  12CF  3D 21 20
1447  12D2  1B 47 30 1B          DEFB 1BH,'G','0',1BH,'=','#','$'
1448  12D6  3D 23 23
1449  12D9  1B 28 20 20          DEFB 1BH,'(',' ',' ',' ',' ',' '
1450  12DD  20 20
1451  12DF  1B 3D 23 2E          DEFB 1BH,'=','$',',',' ',' ',' '
1452  12E3  20 20
1453  12E5  1B 3D 25 23          DEFB 1BH,'=','%'
1454  12E9  20 20 20 20          DEFM '    '
1455  12ED  1B 3D 25 2E          DEFB 1BH,'=','%'
1456  12F1  20 20
1457  12F3  1B 3D 27 23          DEFB 1BH,'=','''','#'
1458  12F7  20 20 20 20          DEFM '    '
1459  12FB  1B 3D 27 2E          DEFB 1BH,'=','''',',',' ',' '
1460  12FF  20 20
1461  1301  1B 3D 29 23          DEFB 1BH,'=',')','#'
1462  1305  20 20 20 20          DEFM '    '
1463  1309  1B 3D 29 2E          DEFB 1BH,'=',')',',',' ',' '
1464  130D  20 20
1465  130F  1B 3D 2B 23          DEFB 1BH,'=','+','#'
1466  1313  20 20 20 20          DEFM '    '
1467  1317  1B 3D 2B 2E          DEFB 1BH,'=','+',',',' ',' '
1468  131B  20 20
1469  131D  1B 3D 2D 22          DEFB 1BH,'=','-','"',1BH,'G','4'
1470  1321  1B 47 34
```

```
1471  1324  20 20 20 20          DEFM '    '
1472  1328  1B 47 34 1B          DEFB 1BH,'G','4',1BH,'G','0'
1473  132C  47 30
1474  132E  1B 3D 2D 2D          DEFB 1BH,'=','-','-',1BH,'G','4'
1475  1332  1B 47 34
1476  1335  20 20 1B 47          DEFB ' ',' ',1BH,'G','4',1BH,'G','0'
1477  1339  34 1B 47 30
1478  133D  1B 3D 2F 23          DEFB 1BH,'=','/','#'
1479  1341  20 20 20 20          DEFM '    '
1480  1345  1B 3D 2F 2E          DEFB 1BH,'=','/','.',' ','.' '
1481  1349  20 20
1482  134B  1B 3D 31 23          DEFB 1BH,'=','1','#'
1483  134F  20 20 20 20          DEFM '    '
1484  1353  1B 3D 31 2E          DEFB 1BH,'=','1','.',' ','.' '
1485  1357  20 20
1486  1359  1B 3D 33 23          DEFB 1BH,'=','3','#'
1487  135D  20 20 20 20          DEFM '    '
1488  1361  1B 3D 33 2E          DEFB 1BH,'=','3','.',' ','.' '
1489  1365  20 20
1490  1367  1B 3D 35 23          DEFB 1BH,'=','5','#'
1491  136B  20 20 20 20          DEFM '    '
1492  136F  1B 3D 35 2E          DEFB 1BH,'=','5','.',' ','.' '
1493  1373  20 20
1494  1375  1B 3D 37 23          DEFB 1BH,'=','7','#'
1495  1379  20 20 20 20          DEFM '    '
1496  137D  1B 3D 37 2E          DEFB 1BH,'=','7','.',' ','.' '
1497  1381  20 20
1498  1383  1B 26 1B 2F          DEFB 1BH,'&',1BH,'.','0'
1499  1387  30
1500  1388  1B 27 1B 2E    MENT13 DEFB 1BH,''',1BH,'.','2'
1501  138C  32
1502  138D  1B 3D 32 4A          DEFB 1BH,'=','2','J',1BH,'G','0',1BH,')'
1503  1391  1B 47 30 1B
1504  1395  29
1505  1396  41 44 45 4C          DEFM 'ADELANTAR'
1506  139A  41 4E 54 41
1507  139E  52
1508  139F  1B 3D 32 58    MENT14 DEFB 1BH,'=','2','X'
1509  13A3  44 49 52 45          DEFM 'DIRECCIONES'
1510  13A7  43 43 49 4F
1511  13AB  4E 45 53
1512  13AE  1B 3D 32 55          DEFB 1BH,'=','2','U',1BH,'(','X','X'
1513  13B2  1B 28 58 58
1514  13B6  1B 26 00 00          DEFB 1BH,'&',00H,00H
1515  13BA  1B 27 1B 2E    MENT15 DEFB 1BH,''',1BH,'.','0'
1516  13BE  30
1517  13BF  1B 3D 32 4A          DEFB 1BH,'=','2','J',1BH,'0','1'
1518  13C3  1B 47 31
1519  13C6  1B 3D 32 55          DEFB 1BH,'=','2','U',1BH,')',' ','.' '
1520  13CA  1D 29 20 20
1521  13CE  1B 28 1B 26          DEFB 1BH,'(',1BH,'&',1BH,'=','#','#'
1522  13D2  1B 3D 23 23
1523  13D6  1B 27 1B 2E    MENT16 DEFB 1BH,''',1BH,'.','2'
1524  13DA  32
```

```
1525  13DD  1D 3D 32 4A              DEFB 1DH,'=','2','J',1DH,'6','0',1DH,')'
1526  13DF  1D 47 30 1D
1527  13E3  29
1528  13E4  20 52 45 54              DEFM ' RETRASAR'
1529  13E8  52 41 53 41
1530  13EC  52
1531  13ED  1D 27 1D 29        MENT17 DEFB 1DH,'''',1DH,')'
1532  13F1  1D 3D 2D 32              DEFB 1DH,'=','-','2',1DH,'0','2'
1533  13F5  1D 47 32
1534  13F8  3C 2D 2D 2D              DEFM '<--- RAM NO DISPONIBLE'
1535  13FC  20 52 41 4D
1536  1400  20 4E 4F 20
1537  1404  44 49 53 50
1538  1408  4F 4E 49 42
1539  140C  4C 45
1540  140E  1D 47 30 1D              DEFB 1DH,'G','0',1DH,'('
1541  1412  28
1542  1413  1D 26 1D 3D              DEFB 1DH,'&',1DH,'=','$','$'
1543  1417  23 23
1544  1419  1D 27 1D 29        MENT18 DEFB 1DH,'''',1DH,')'
1545  141D  1D 3D 2D 37              DEFB 1DH,'=','-','7',1DH,'6','2'
1546  1421  1D 47 32
1547  1424  3C 2D 2D 20              DEFM '<--- LOCALIDAD DE RAM '
1548  1428  20 4C 4F 43
1549  142C  41 4C 49 44
1550  1430  41 44 20 44
1551  1434  45 20 52 4F
1552  1438  4D 20
1553  143A  1D 47 30 1D              DEFB 1DH,'G','0',1DH,'('
1554  143E  28
1555  143F  1D 26 1D 3D              DEFB 1DH,'&',1DH,'=','$','$'
1556  1443  23 23
1557  1445  1D 27 1D 3D        MENT19 DEFB 1DH,'''',1DH,'=','-','7'
1558  1449  2D 37
1559  144B  1D 47 31 1D              DEFB 1DH,'G','1',1DH,'&',1DH,'=','$','$'
1560  144F  26 1D 3D 23
1561  1453  23
1562  1454  1D 27 1D 29        MENT20 DEFB 1DH,'''',1DH,')',1DH,'=','*','>'
1563  1458  1D 3D 2A 3E
1564  145C  50 4F 53 49              DEFM 'POSICION'
1565  1460  43 49 4F 4E
1566  1464  20 20 20           MENT21 DEFM '   '
1567  1467  1D 28                     DEFB 1DH,'('
1568  1469  58 58 58 58              DEFM 'XXXX'
1569  146D  1D 29 1D 3D              DEFB 1DH,')',1DH,'=','2','U',' ',' '
1570  1471  32 55 20 20
1571  1475  1D 3D 32 4A              DEFB 1DH,'=','2','J',1DH,'6','1'
1572  1479  1D 47 31
1573  147C  1D 28 1D 26              DEFB 1DH,'(',1DH,'&'
1574  1480  1D 3D 2A 49              DEFB 1DH,'=','*','I',1DH,',','2'
1575  1484  1D 2E 32
1576  1487  1D 27 1D 29        MENT22 DEFB 1DH,'''',1DH,')',1DH,'=','*','>'
1577  148B  1D 3D 2A 3E
1578  148F  20 20 20 20              DEFM '     RUN'
```

```
1579  1493  20 52 55 4E
1580  1497  1B 43 1B 2A      MENT90 DEFB 1BH,'C',1BH,'*',1BH,')'
1581  149B  1B 29
1582  149D  1A 1B 3D 20            DEFB 1AH,1BH,'=',' ','/',1BH,'G','4'
1583  14A1  2F 1B 47 34
1584  14A5  1B 3D 20 61            DEFB 1BH,'=',' ',61H,1BH,'G','4'
1585  14A9  1B 47 34
1586  14AC  1B 47 30 1B            DEFB 1BH,'G','0',1BH,'=','!','/'
1587  14B0  3D 21 2F
1588  14B3  1B 47 34 1B            DEFB 1BH,'G','4',1BH,'=','!',61H
1589  14B7  3D 21 61
1590  14BA  1B 47 34 1B            DEFB 1BH,'G','4',1BH,'G','0'
1591  14BE  47 30
1592  14C0  1B 3D 22 2F            DEFB 1BH,'=','"','/',1BH,'G','4'
1593  14C4  1B 47 34
1594  14C7  1B 3D 22 61            DEFB 1BH,'=','"',61H
1595  14CB  1B 47 34 1B            DEFB 1BH,'G','4',1BH,'G','0'
1596  14CF  47 30
1597  14D1  1B 3D 21               DEFB 1BH,'=','!'
1598  14D4  36                MENT30 DEFB '6'
1599  14D5  45 20 58 20            DEFM 'E X A M E N  D E  R E G I S T R O S'
1600  14D9  41 20 4D 20
1601  14DD  45 20 4E 20
1602  14E1  20 20 44 20
1603  14E5  45 20 20 20
1604  14E9  52 20 45 20
1605  14ED  47 20 49 20
1606  14F1  53 20 54 20
1607  14F5  52 20 4F 20
1608  14F9  53
1609  14FA  38                MENT50 DEFB '8'
1610  14FB  45 20 58 20            DEFM 'E X A M E N  D E  P U E R T O S'
1611  14FF  41 20 4D 20
1612  1503  45 20 4E 20
1613  1507  20 20 44 20
1614  150B  45 20 20 20
1615  150F  50 20 55 20
1616  1513  45 20 52 20
1617  1517  54 20 4F 20
1618  151B  53
1619  151C  35                MENT70 DEFB '5'
1620  151D  54 20 52 20            DEFM 'T R A S L A D O  D E  A R C H I V O S'
1621  1521  41 20 53 20
1622  1525  4C 20 41 20
1623  1529  44 20 4F 20
1624  152D  20 20 44 20
1625  1531  45 20 20 20
1626  1535  41 20 52 20
1627  1539  43 20 48 20
1628  153D  49 20 56 20
1629  1541  4F 20 53
1630  1544  33                MENT80 DEFB '3'
1631  1545  50 20 52 20            DEFM 'P R O G R A M A D O R  D E  E P R O M ' S'
1632  1549  4F 20 47 20
```

```
1633  154D  52 20 41 20          
1634  1551  4D 20 41 20          
1635  1555  44 20 4F 20          
1636  1559  52 20 20 20          
1637  155D  44 20 45 20          
1638  1561  20 20 45 20          
1639  1565  50 20 52 20          
1640  1569  4F 20 4D 20          
1641  156D  27 20 53                    DEFB ''',' ','S'
1642                            ;EL SIGUIENTE ES EL MENSAJE DE LAS OPCIONES.
1643  1570  1B 3D 24 59          MENT71 DATA 1BH,'=','$','Y'
1644  1574  4F 50 43 49                 DEFM 'OPCIONES:'
1645  1578  4F 4E 45 53          
1646  157C  3A                   
1647  157D  1B 3D 26 59                 DATA 1BH,'=','&','Y'
1648  1581  41 2E 20 43                 DEFM 'A. COMPUTADORA'
1649  1585  4F 4D 50 55          
1650  1589  54 41 44 4F          
1651  158D  52 41                
1652  158F  1B 3D 27 59                 DATA 1BH,'=',''','Y'
1653  1593  42 2E 20 47                 DEFM 'B. GRABADORA'
1654  1597  52 41 42 41          
1655  159B  44 4F 52 41          
1656  159F  1B 3D 28 59                 DATA 1BH,'=','(','Y'
1657  15A3  43 2E 20 44                 DEFM 'C. DISKETTE'
1658  15A7  49 53 4B 45          
1659  15AB  54 54 45             
1660  15AE  1B 3D 29 59                 DATA 1BH,'=',')','Y'
1661  15B2  44 2E 20 4D                 DEFM 'D. MEMORIA'
1662  15B6  45 4D 4F 52          
1663  15BA  49 41                
1664  15BC  1B 3D 2A 59                 DATA 1BH,'=','*','Y'
1665  15C0  4D 2E 20 4D                 DEFM 'M. MONITOR'
1666  15C4  4F 4E 49 54          
1667  15C8  4F 52                
1668  15CA  1B 3D 36 3D                 DATA 1BH,'=','6','=',1BH,'G','1'
1669  15CE  1B 47 31             
1670  15D1  54 20 45 20                 DEFM 'T E C L A   I N V A L I D A'
1671  15D5  43 20 4C 20          
1672  15D9  41 20 20 20          
1673  15DD  49 20 4E 20          
1674  15E1  56 20 41 20          
1675  15E5  4C 20 49 20          
1676  15E9  44 20 41             
1677  15EC  1B 47 30 1B                 DATA 1BH,'G','0',1BH,'=',''','F'
1678  15F0  3D 27 46             
1679  15F3  44 45 53 54                 DEFM 'DESTINO: '
1680  15F7  49 4E 4F 3A          
1681  15FB  20                   
1682  15FC  1B 28 58 1B                 DATA 1BH,'(','X',1BH,')',1BH,'=',''','9'
1683  1600  29 1B 3D 27          
1684  1604  39                   
1685  1605  46 55 45 4E                 DEFM 'FUENTE: '
1686  1609  54 45 3A 20          
```

```
1687  160D  1B 28 58 1B        DATA 1BH,'(','X',1BH,'&',00H
1688  1611  26 08
1689  1613  1B 42 1B 2A    MENT22 DATA 1BH,'B',1BH,'*',1BH,'C',0DH,0AH,00H
1690  1617  1B 43 0D 0A
1691  161B  00

1693                             ;MENSAJE DE OPCIONES

1695  161C  1B 27 1B 29    OPC_01 DATA 1BH,'''',1BH,')',1BH,'=','0','/'
1696  1620  1B 3D 29 2F
1697  1624  46 55 45 4E           DEFM 'FUENTE: '
1698  1628  54 45 3A 20
1699  162C  1B 3D 2C 2F           DATA 1BH,'=',',','/'
1700  1630  44 45 53 54           DEFM 'DESTINO: '
1701  1634  49 4E 4F 3A
1702  1638  20
1703  1639  1B 3D 2D 35    OPC_02 DATA 1BH,'=','-','5'
1704  163D  44 49 52 20           DEFM 'DIR INICIAL: '
1705  1641  49 4E 49 43
1706  1645  49 41 4C 3A
1707  1649  20
1708  164A  1B 28 58 58           DATA 1BH,'(','X','X','X','X',1BH,')'
1709  164E  58 58 1B 29
1710  1652  20 20 20 46           DEFM '   FINAL: '
1711  1656  49 4E 41 4C
1712  165A  3A 20
1713  165C  1B 28 58 58           DATA 1BH,'(','X','X','X','X',1BH,')'
1714  1660  58 58 1B 29
1715  1664  1B 3D 2A 35    OPC_03 DATA 1BH,'=','*','5'
1716  1668  4E 4F 4D 42           DEFM 'NOMBRE DEL ARCHIVO: '
1717  166C  52 45 20 44
1718  1670  45 4C 20 41
1719  1674  52 43 48 49
1720  1678  56 4F 3A 20
1721  167C  1B                    DATA 1BH
1722  167D  28 58 58 58           DEFM '(XXXXXX'
1723  1681  58 58 58
1724  1684  1B 29                 DATA 1BH,')'
1725  1686  1B 26 1B 28           DATA 1BH,'&',1BH,'('
1726  168A  08 08 08 08           DATA 08,08,08,08,08,08,08,08
1727  168E  08 08 08 08
1728  1692  1B 2E 30 1D    MENT23 DATA 1BH,',','0',1BH,'''',1BH,'=','0','5'
1729  1696  27 1B 3D 30
1730  169A  35
1731  169B  1B 29 1B 47           DATA 1BH,')',1BH,'G','2'
1732  169F  32
1733  16A0  50 41 52 41           DEFM 'PARA EJECUTAR OPRIMA *RETURN*'
1734  16A4  20 45 4A 45
1735  16A8  43 55 54 41
1736  16AC  52 20 4F 50
1737  16B0  52 49 4D 41
1738  16B4  20 22 52 45
```

```
1739  16B8  54 55 52 4E
1740  16BC  22
1741  16BD  1B 26                    DATA 1BH,'&'
1742  16BF  1B 27 1B 3D       MENT74 DATA 1BH,'''',1BH,'=','0','5',1BH,'0','1'
1743  16C3  30 35 1B 47
1744  16C7  31
1745  16C8  1B 3D 36 3D              DATA 1BH,'=','6','=',1BH,'0','1',1BH,'&'
1746  16CC  1B 47 31 1B
1747  16D0  26
1748  16D1  00 1B 28 1B              DATA 00H,1BH,'(',1BH,',','2'
1749  16D5  2E 32
1750  16D7  1B 27 1B 3D       MENT75 DATA 1BH,'''',1BH,'=','0','9',1BH,'0','4'
1751  16DB  30 39 1B 47
1752  16DF  34
1753  16E0  1B 3D 30 5A              DATA 1BH,'=','0','Z',1BH,'0','4',1BH,'B'
1754  16E4  1B 47 34 1B
1755  16E8  47
1756  16E9  30 1B 3D 33              DATA '0',1BH,'=','3','6',1BH,'0','2'
1757  16ED  36 1B 47 32
1758  16F1  50 41 52 41              DEFM 'PARA NUEVA OPCION OPRIMA '
1759  16F5  20 4E 55 45
1760  16F9  56 41 20 4F
1761  16FD  50 13 49 4F
1762  1701  4E 20 4F 50
1763  1705  53 49 4D 41
1764  1709  20
1765  170A  22 43 4C 45              DEFM '"CLEAR SPACE"'
1766  170E  41 52 20 53
1767  1712  50 41 43 45
1768  1716  22
1769  1717  1B 3D 30 3B              DATA 1BH,'=','0',';'
1770  171B  20 2A 2A 2A              DEFM ' **** TRASLADO CONCLUIDO ****'
1771  171F  2A 20 54 52
1772  1723  41 53 4C 41
1773  1727  44 4F 20 43
1774  172B  4F 4E 43 4C
1775  172F  55 49 44 4F
1776  1733  20 2A 2A 2A
1777  1737  2A
1778  1738  1B 3D 30 3B       MENT76 DATA 1BH,'=','0',';'
1779  173C  2A 2A 2A 20              DEFM '*** MEMORIA NO DISPONIBLE ***'
1780  1740  4D 45 4D 4F
1781  1744  52 49 41 20
1782  1748  4E 4F 20 44
1783  174C  49 53 50 4F
1784  1750  4E 49 42 4C
1785  1754  45 20 2A 2A
1786  1758  2A
1787  1759  1B 27 1B 3D       MENT77 DATA 1BH,'''',1BH,'=',',',',',1BH,'G','1'
1788  175D  2C 2E 1B 47
1789  1761  31
1790  1762  1B 28 1B 26              DATA 1BH,'(',1BH,'&',09H,09H,09H
1791  1766  09 09 09
1792  1769  1B 27 1B 3D       MENT79 DATA 1BH,'''',1BH,'=',',',',',1BH,'G','1'
```

```
    1793  176D  2C 2E 1B 47
    1794  1771  31
    1795  1772  1B 28 1B 26          DATA 1BH,'(',1BH,'&',09H,09H,09H
    1796  1776  09 09 09
    1797  1779  1B 3D 30 35          DATA 1BH,'=','0','5',1BH,'''',1BH,'G','1'
    1798  177D  1B 27 1B 47
    1799  1781  31
    1800  1782  1B 26 41 44          DATA 1BH,'&','A','D',0DH,0AH ;  -
    1801  1786  0D 0A
    1802  1788  1B 3D 30 3B      MENT8 DATA 1BH,'=','0',';'
    1803  178C  2A 20 4C 4F          DEFM '* NO FUNCIONA LA COMPUTADORA *' \
    1804  1790  20 46 55 4E
    1805  1794  43 49 4F 4E
    1806  1798  41 20 4C 41
    1807  179C  20 43 4F 4D
    1808  17A0  50 55 54 41
    1809  17A4  44 4F 52 41
    1810  17A8  20 2A
    1811  17AA  1B 27 1B 3D          DEFB 1BH,'''',1BH,'='          *** AGREGADO ***
    1812  17AE  2D 49 1B 47          DEFB '-','I',1BH,'G'                  '
    1813  17B2  31 1B 28 58          DEFB '1',1BH,'(','X'                  '
    1814  17B6  1B 26                DEFB 1BH,'&'                          '
    1815  17B8  20 21 52 45          DEFB ' ','!','R','E'
    1816  17BC  0A 20 2C 48          DEFB 0AH,20H,2CH,48H,4FH
    1817  17C0  4F
    1818  17C1  1B 27 1B 29      MENT7A DEFB 1BH,'''',1BH,')',1BH,'&'
    1819  17C5  1B 26
    1820  17C7  1B 3D 30 3B          DEFB 1BH,'=','0',';'
    1821  17CB  20 2A 2A 2A          DEFM ' *** ARCHIVO NO LOCALIZADO ***'
    1822  17CF  20 41 52 43
    1823  17D3  48 49 56 4F
    1824  17D7  20 4E 4F 20
    1825  17DB  4C 4F 43 41
    1826  17DF  4C 49 5A 41
    1827  17E3  44 4F 20 2A
    1828  17E7  2A 2A
    1829  17C9                      END

    ASSEMBLER ERRORS =    0
OK,
```