TI–74 BASICALC™
Technical Data Manual

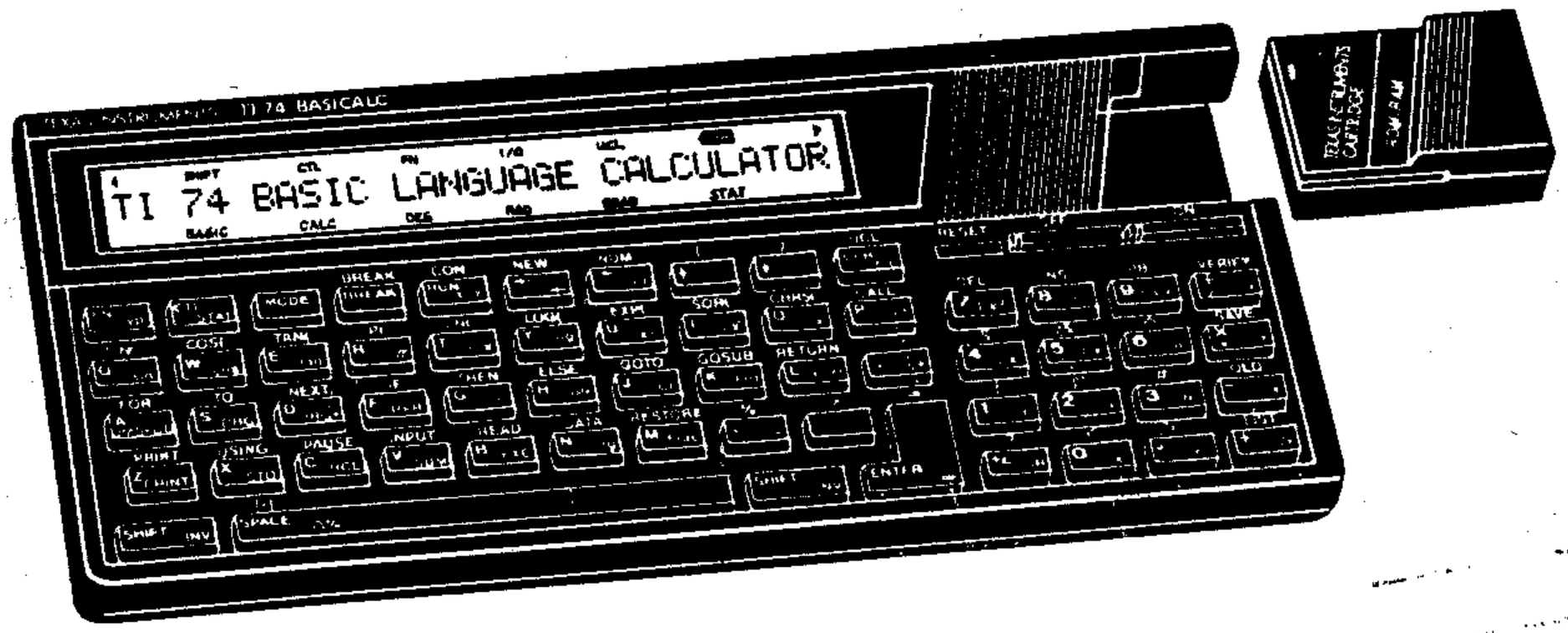


TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Texas Instruments makes no warranty, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose regarding this literature or any information derived therefrom and makes such materials available solely on an "as-is" basis. In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability to Texas Instruments, regardless of the form of action, shall not exceed the purchase price of this book. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the user of this material by any other party.

TI-74 BASICALC
BASIC LANGUAGE/SCIENTIFIC CALCULATOR

GENERAL DESCRIPTION

The TI-74 BASICALC (TM) combines advanced BASIC language programmability and scientific calculator functions in a battery operated portable unit with system expansion capability.

BASIC Programmability

The BASIC mode provides access to a full featured, high level programming language. BASIC on the TI-74 has been implemented in close compliance with ANSI minimal standards for BASIC language computers. This means that the TI-74's BASIC language is similar to the BASIC language implementations on larger computers rather than the reduced capabilities which are usually associated with calculator-type BASIC machines.

In addition to being designed with ANSI standards in mind, many enhancements have also been specifically tailored for use on a portable calculator. Numeric or BASIC expressions entered without line numbers are executed in an immediate mode, whereas expressions entered with line numbers become entries into the BASIC program. This ability to switch quickly from program entry to immediate execution makes the product more versatile.

The TI-74 has 8K CONSTANT MEMORY (TM) RAM for medium-sized applications. The unit's storage capacity can be doubled with the addition of an 8K CONSTANT MEMORY (TM) cartridge. Programs or data can be stored off-line with the CI-7 Audio Cassette Interface. The TI-74 system also includes the optional PC-324, a 24 column thermal printer for applications requiring hard copy of program listings or results.

Scientific Calculator

In the calculator mode, the TI-74 offers 70 scientific functions. alphanumeric error messages, and 13 digit accuracy which spans a numeric range of $\pm$9.9999999999999E127. Features include logarithmic, hyperbolic and trigonometric functions, 2 variable statistics, and probability calculations. Ten user registers are available for intermediate results or constants.

Ease of Use

The TI-74 console contains a cartridge port to augment standard memory. An 8K RAM cartridge can be software configured to double the amount of memory available for programs/data or configured as an 8K mass storage device. Another option is the use of 32K ROM cartridges which contain application programs designed to solve

specific problems. Various application ROM cartridges are available. The TI-74 also has EPROM cartridge capability for low volume applications.

SOFTWARE SYSTEM

System RAM Layout

The 8K system RAM is arranged in several blocks. These are shown in Figure 1 on the following page.

The system reserved area contains all system pointers, keyboard and display buffers, the BASIC buffer for program line compression, and system information used in verifying RAM content correctness.

The calculator memories section of RAM is used by the system calculator mode as storage for the values associated with each of the ten calculator registers. These registers are also used by the TI-74 statistics mode.

The user function key definitions are stored next in RAM. These definitions consist of strings of ASCII characters, up to 255 bytes long (from the keyboard, this length is limited to the 80 byte capacity of the keyboard buffer). These definitions are associated with the ten numeric-digit keys on the TI-74 keyboard.

The variable name storage is used for keeping the names for program variables so that the program can be listed. It is also used to differentiate string variables from numeric variables in the pre-run program-checking procedure.

The floating-point stack is used to temporarily store information such as intermediate calculation results and run-time context information. Context information can consist of BASIC return addresses for subroutines and subprograms, or error handling information.

The free or dynamic-memory pool is used by run-time program-support routines that need blocks of memory to perform their function. Such functions include symbol tables, variable value storage, I/O buffers, peripheral I/O control blocks, and temporary string values.

The user program area consists of a program header followed by the "tokenized" program image. This image is a compressed form of the program that the user entered. It is "tokenized" so that large programs can be stored by the TI-74 in a minimum amount of RAM.

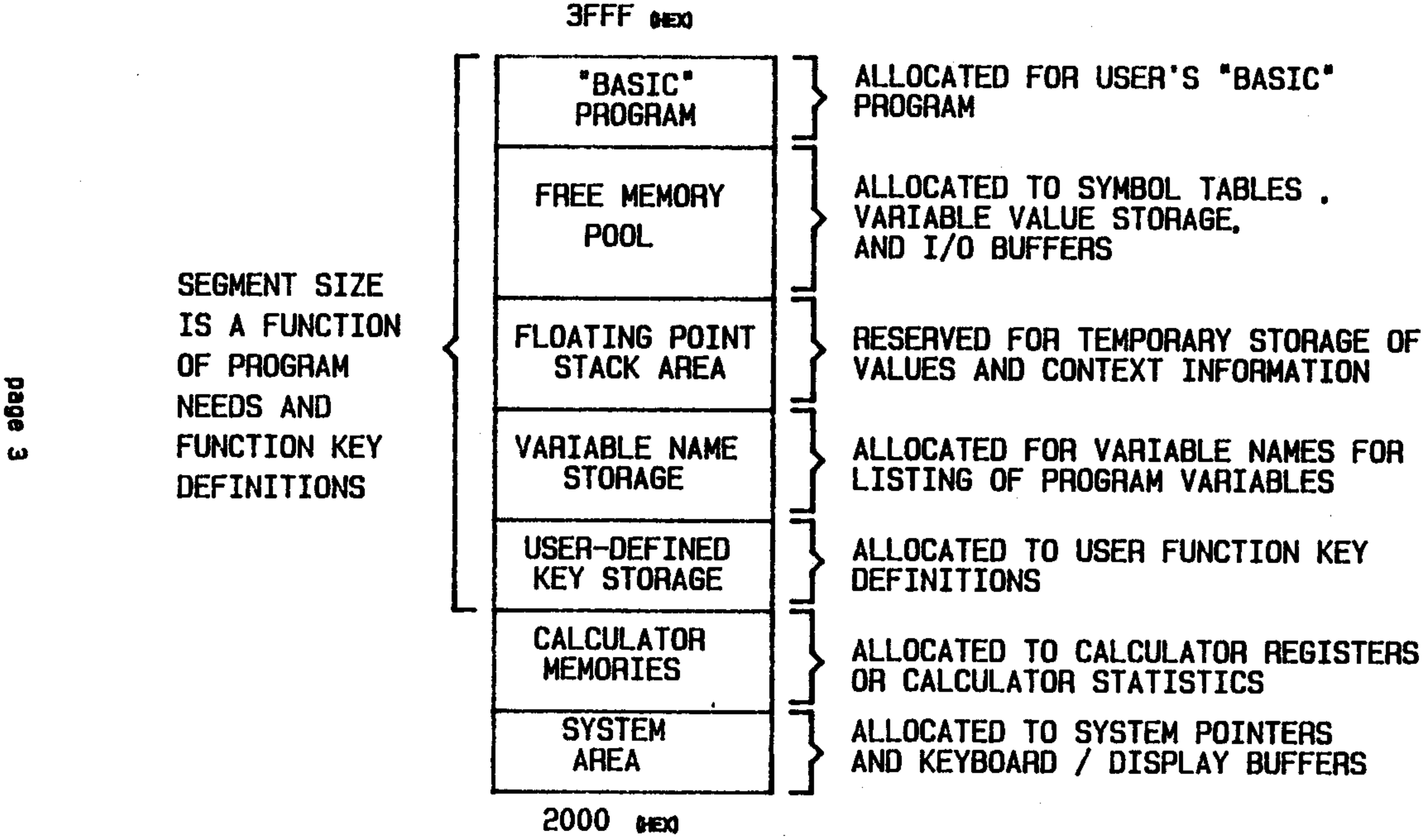


Figure 1. System RAM Organization

Floating-Point Number Representation

The TI-74 system represents floating point numbers using eight bytes for each number. The lowest addressed byte of each number contains a biased exponent value, while the remaining seven bytes contain the 14 BCD (binary-coded decimal) digits which form the mantissa. The number representation used is called "Radix-100".

Radix-100 Format

Radix-100 is a base 100 number representation. A single Radix-100 digit ranges in value from 0 to 99 in base 10. The seven bytes of the mantissa are seven Radix-100 digits. The decimal point is assumed to always be located after the most significant Radix-100 digit. Since this digit may have values from 1 to 99, the most significant BCD digit is sometimes zero. Therefore, numbers are stored in either 13 or 14 digits of decimal precision. Since each Radix-100 digit is made up of two BCD digits, only the exponent requires conversion for output in BCD format.

In Radix-100 format, the exponent value ranges from >0 to >7B (throughout this manual, hexadecimal number values are indicated by the > prefix). Removing the >40 offset and converting to decimal produces an exponent ranging from -64 to +63. These are still exponents of 100 rather than 10. Converting to exponents of 10 and allowing for a possible leading zero in the BCD mantissa produces an actual exponent range of -128 to +127 in BCD. Some examples of Radix-100 format are presented in Table 1.

Table 1. Radix-100 Examples

BCD Number	Radix-100 representation Exp	Mantissa
0.1	>39	10, 00, 00, 00, 00, 00, 00
1	>40	01, 00, 00, 00, 00, 00, 00
10	>40	10, 00, 00, 00, 00, 00, 00
100	>41	01, 00, 00, 00, 00, 00, 00
PI	>40	03, 14, 15, 92, 65, 35, 90
-PI	>BF	03, 14, 15, 92, 65, 35, 90
-9.456 × 10 -35	>D1	94, 56, 00, 00, 00, 00, 00
9.456 × 10 -35	>2E	94, 56, 00, 00, 00, 00, 00
-9.456 × 10 -35	>AE	94, 56, 00, 00, 00, 00, 00
9.456 × 10 -35	>51	94, 56, 00, 00, 00, 00, 00

TI-74 Numeric Range

The TI-74 system has a numeric range from -9.999999999999E+127 through -1.E-128; zero; and +1.E-128 through +9.999999999999E+127. The biased exponent values of +63 to -64 provide a decimal exponent range +127 to -128.

HARDWARE DESIGN

The TI-74 is a compact, self-contained unit consisting of a CPU, built-in functions on ROM, 8K RAM, a keyboard, a liquid crystal display, and connectors for expansion of memory and addition of peripherals.

Hardware Block Diagram

The hardware block diagram for the TI-74 console is illustrated in Figure 2. The console electrical design incorporates an 8-bit TMS70C46 CMOS microcomputer, 32Kx8 CMOS ROM, 8Kx8 CMOS RAM, components for multiplexed drive of the 31-character liquid crystal display, a 62 key alphanumeric keyboard, and power supply circuitry.

System Microprocessor - TMS70C46

The TI-74 uses the TMS70C46 microprocessor which is a member of the TMS7000 family. The TMS70C46 is functionally upward compatible with its TMS7040 Counterpart. In addition, the TMS70C46 contains functionality not available in the TMS7040. For complete information on the TMS7040 and this family of microprocessors, you may obtain purchase information about the TMS7000 Family Data Manual from the following address:

Texas Instruments
Data Book Marketing
P.O. Box 117692, M/S 54
Carrollton, TX 75011-7692

Or telephone: (214) 242-0864

The TMS70C46 CPU is an enhanced version of Texas Instruments' TMS7000 family of 8-bit microcomputers. The modular architecture and topology of the TMS7000 CPU enabled TI designers to customize the device to the application. Several new features and functions are integrated into the basic chip design, resulting in lower component count, lower manufacturing costs, and increased reliability.

The TMS7000 provides flexibility in defining the I/O configuration to meet memory requirements for each application. Designs requiring only the on-chip 4K byte ROM and on-chip 128 byte RAM can configure the device in the single chip mode with all I/O pins available for general purpose input/output. The TI-74 design configures the part in the full expansion mode, with the I/O pin definitions as shown in Figure 3. In the full expansion mode, the TMS70C46 can directly access 64K bytes of external memory. This mode dedicates 16 I/O pins for memory address bus, 8 I/O pins for memory data bus, and 5 I/O pins for memory control, leaving 11 pins remaining for general purpose I/O. In the full expansion mode, the basic TMS7000 architecture allows interfacing to memory

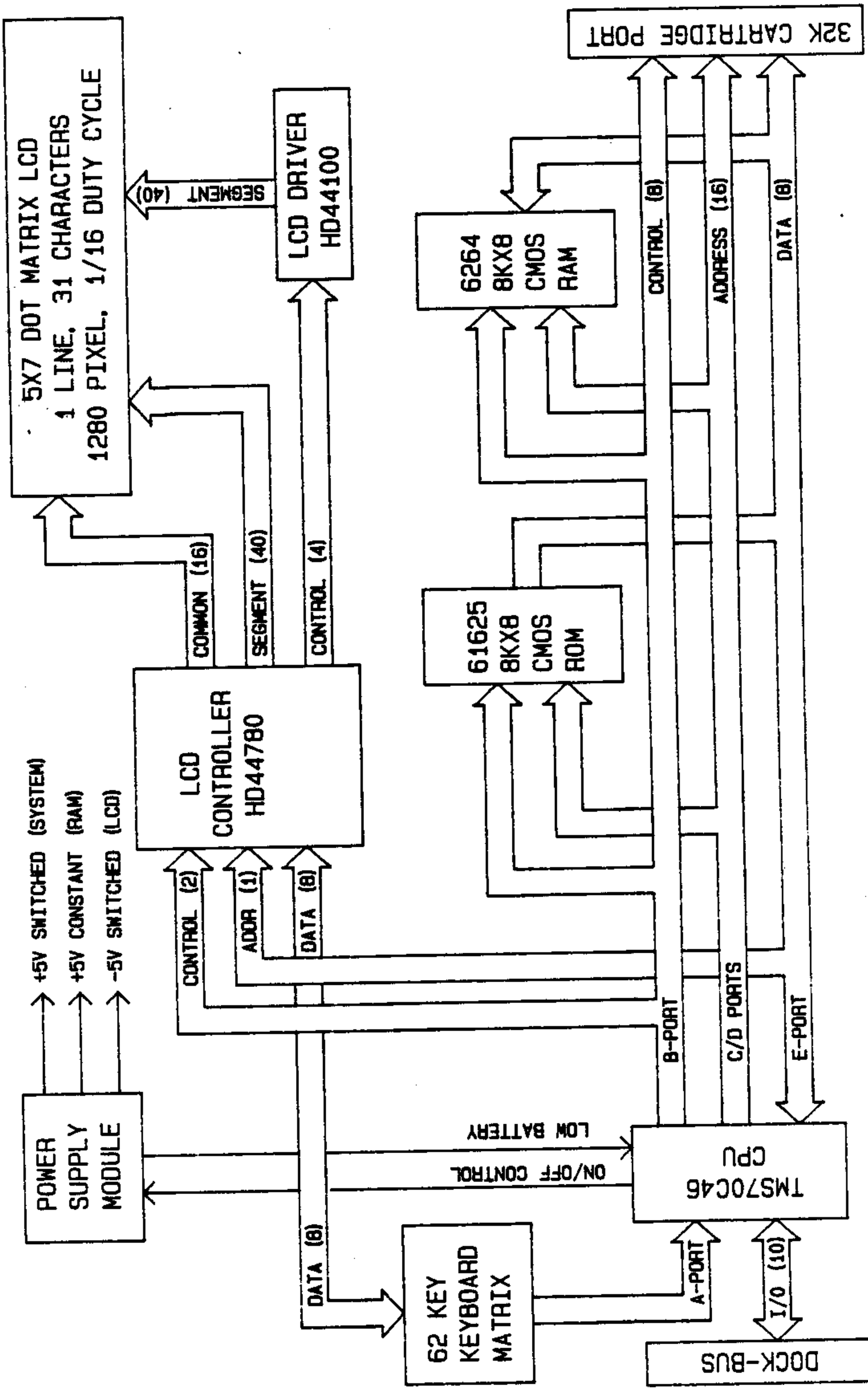


Figure 2. Hardware Block Diagram

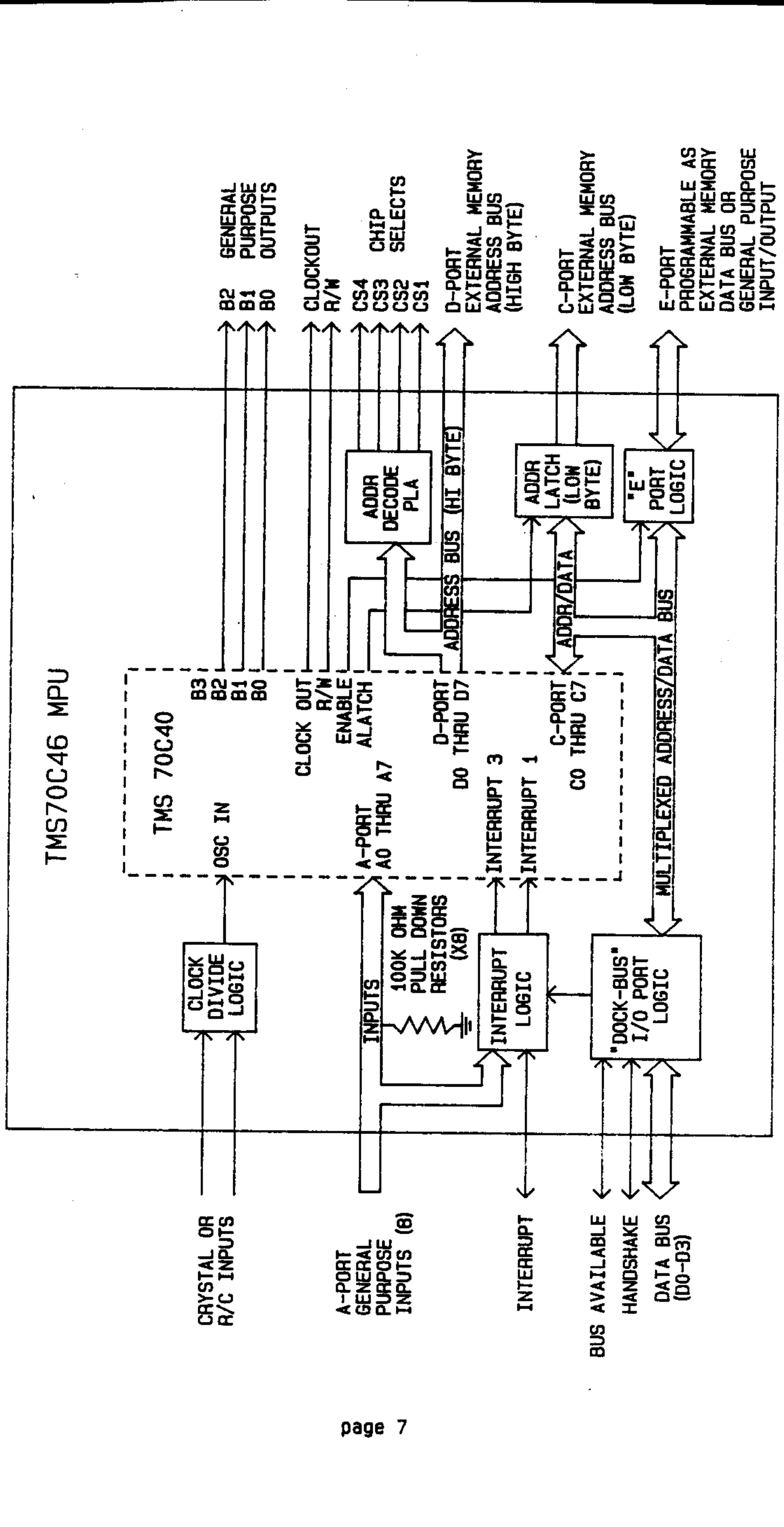


Figure 3. TMS70C46 Functional Diagram

via a multiplexed address/data bus, requiring external latching of the least significant 8-bits of the address. The TMS70C46 includes an additional 8-bit port and logic to provide separate address and data buses. The CPU is packaged in a 54 pin Y-shrink DIP (Dual In-line Package).

The TMS70C46 also includes an 8-bit timer/counter with a 5-bit prescale register. The timer reduces the software overhead associated with keyboard scanning, contact de-bounce and automatic power-down functions and is used during peripheral communication for time-out when the selected peripheral is not attached.

The TMS7000 is equipped with a standard instruction set which is adequate to support the TI-74 application. Instruction types include arithmetic, logical, comparison/test, branch/call and trap. BCD and 2's complement numerical representations are supported. Register, immediate, direct, relative, indexed, and indirect addressing modes are available.

System Memory

The system memory map is depicted in Figure 4. System memory is allocated as follows:

128 byte segment starting at location >0000 for TMS70C46 on-chip read/write memory.

128 byte segment starting at location >0100 for TMS70C46 I/O control registers used in initializing and configuring the device, operating the timer, etc.

4K byte segment starting at location >1000 for TI-74 I/O devices, including the LCD controller. All I/O is memory mapped.

8K byte segment starting at location >2000 for additional read/write memory (RAM) external to the TMS70C46.

32K byte segment starting at location >4000 for cartridge port memory expansion. The cartridge port allows the TI-74 memory capacity to be supplemented with application ROMs or additional read/write memory.

8K byte segment starting at location >C000 for additional ROM external to the TMS70C46. Code for calculator functions and BASIC interpreter reside in this ROM.

Unused 4K byte segment starting at location >E000.

4K byte segment starting at location >F000 for the TMS70C46 on-chip mask programmable ROM. Residing in this ROM are algorithms for system initialization at power-up, floating-

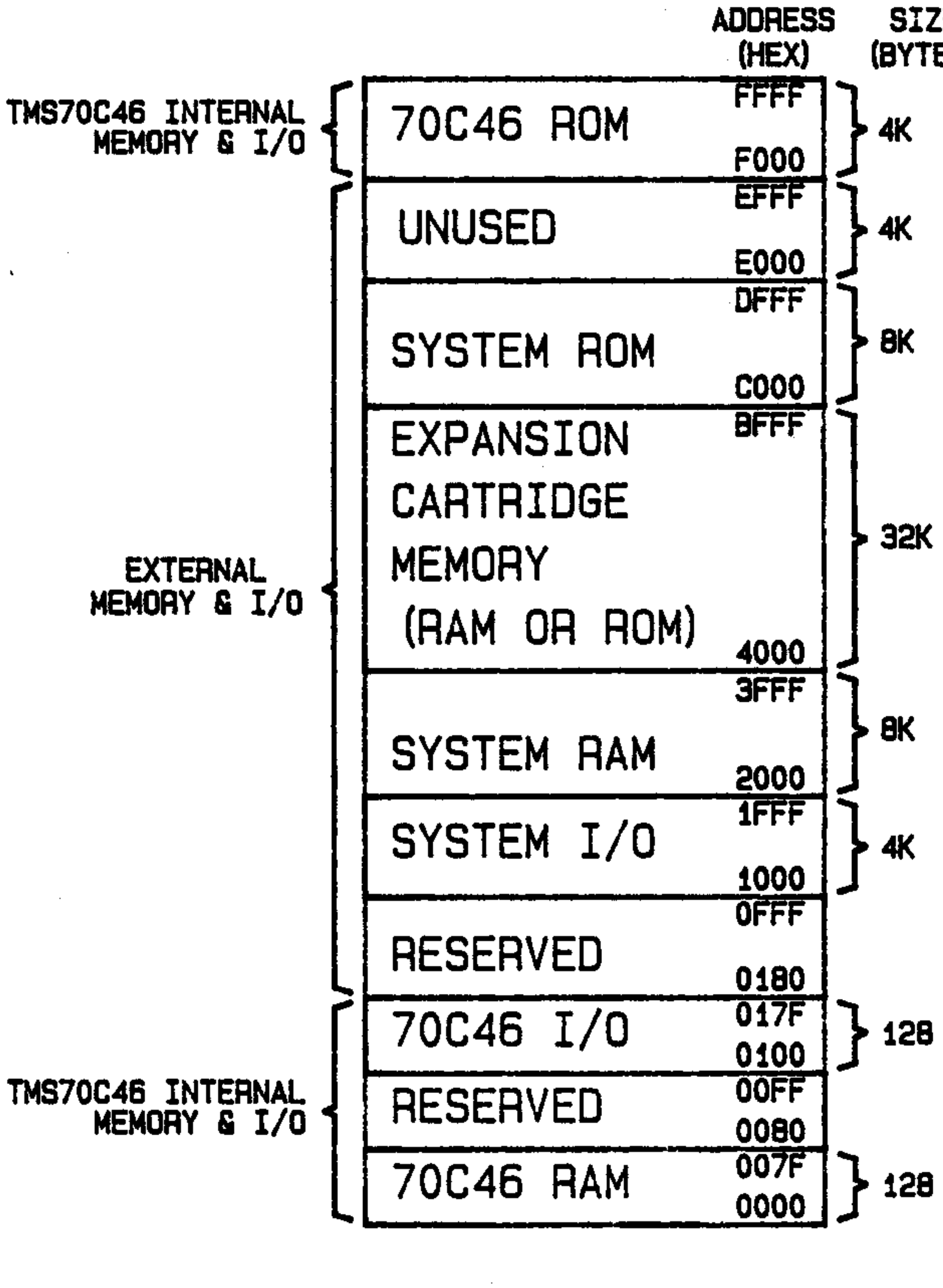


Figure 4. TI-74 Memory Map

point math, and system I/O drivers (including the keyboard scan routine).

Enhancements incorporated in the TMS70C46 include on-chip address decoding. The TMS70C46 contains a mask-programmable logic array (PLA) which enables the designer to define 4 chip select signals based upon decoding of the 5 most significant address bits. The on-chip address decode allows access to the complete 64K byte address space without the need for external address-decode logic. In the TI-74 console, chip selects are generated for the LCD controller, 8K RAM, 32K cartridge port, and 32K system ROM.

Algorithms for calculator functions and BASIC interpreter are stored in the HN61256 CMOS ROM. A special technique is used to address this 32K byte ROM because system memory requirements are in excess of the 64K bytes directly accessible by the CPU. The HN61256 occupies only 8K bytes of the CPU address space. Two general-purpose I/O lines of the TMS70C46 are programmed as outputs and form the most significant bits of the ROM's address bus. Selection of one of the four 8K pages via the two control lines is managed by system software residing in the 4K ROM of the CPU.

Product design goals mandate quick calculation times and a fast BASIC interpreter without sacrificing battery life. The HN61256 features an extremely low power dissipation of 5 microwatts standby and 7.5 milliwatts operating, at the expense of access time which can be as much a 4 microseconds. This presents a problem because the TMS7000 series microcomputers have no provisions for interfacing to slower memory (wait states). The TI-74 design incorporates a custom clock divider circuit on the TMS70C46 bar which enables the CPU to operate at a reduced speed when accessing slower memory. The enhancement includes a software selectable divide ratio and mask programmable PLA which defines the address range for which the CPU operates at a reduced clock rate.

The 128 bytes of CPU read/write memory is augmented with 8K bytes via an HM6264 static CMOS RAM. Discounting system software overhead, approximately 7.5K bytes are available for user programs or data. The HM6264 interfaces directly to the CPU address, data, and control buses; no additional logic is required.

Most applications require this memory to be nonvolatile; the data must be retained when the unit is turned off. The HM6264 is powered by a regulated supply voltage which is maintained when the console is powered down. Memory integrity is verified by system software which computes and compares a checksum when the unit is turned on.

To extend the range of useful applications, TI-74 memory can be augmented via the cartridge port. In addition to supply voltages, the CPU address bus, data bus, and control bus signals are

available at a 30 position card-edge connector for direct connection to a plug-in expansion cartridge. Cartridge port connections are shown in Table 2.

Table 2. Cartridge Port Connections

Signal	Description	Pin #
A7	Address line 7	1
A6	Address line 6	2
A13	Address line 13	3
A5	Address line 5	4
A4	Address line 4	5
A15	Address line 15	6
VDD	+5 Volts	7
A3	Address line 3	8
GND	System common ground	9
D2	Data line 2	10
A2	Address line 2	11
A1	Address line 1	12
D1	Data line 1	13
D0	Data line 0	14
A0	Address line 0	15
D7	Data line 7	16
A11	Address line 11	17
A10	Address line 10	18
D3	Data line 3	19
A12	Address line 12	20
D4	Data line 4	21
GND	System common ground	22
D5	Data line 5	23
VDD	+5 Volts for RAM retention	24
R/-W	Read / Write control	25
D6	Data line 6	26
A8	Address line 8	27
A9	Address line 9	28
A14	Address line 14	29
-CS2	Cartridge port chip select	30

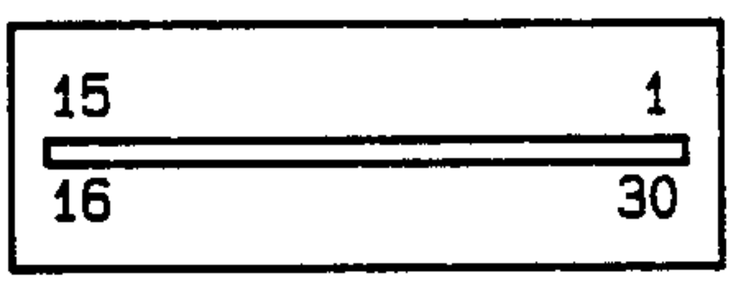


Figure 5. Relative Pin Position Looking Into The Cartridge Port

ROM expansion cartridges contain a single HN61256 32K byte CMOS ROM. The modules are intended to offer a variety of application-specific software including Pascal, statistics and mathematics libraries.

RAM expansion cartridges double the amount of read/write memory available for user programs or data to approximately 15.5K bytes. The cartridge houses a single HM6264 8K CMOS RAM and a 3 volt Lithium battery. When the cartridge is separated from the console, the battery powers the RAM, retaining data for as long as 5 years. The RAM cartridge can also provide a convenient, cost effective solution for off-line mass storage of programs.

Display System

The TI-74 display system includes the liquid crystal display (LCD), HD44780 LCD controller and HD44100 LCD driver. The LCD is arranged as 1 line of 31 characters with 14 annunciators. Each character is formed from a 5x7 dot matrix. A total of 1280 individual pixel elements are required to generate characters, underscore and annunciators. The display is designed for 1/5 bias, 1/16 duty cycle multiplexed drive, with 16 column drive connections and 80 segment drive connections.

All multiplexed drive functions are provided by the HD44780 and HD44100 chip set. The HD44780 controller is a special LSI (Large-Scale Integration) microcomputer dedicated to the task of driving multiplexed dot matrix displays. The HD44780 interfaces to the TMS70C46 CPU external memory bus via 8 data lines, a read/write line, 1 address line, and a chip select line. The HD44780 and TMS70C46 clocks are not synchronized, the 270KHz clock rate of the controller is based upon the LCD characteristics. During data transfers, the CPU clock rate is reduced to ensure reliable communication with the controller.

The CPU sends character data to the controller in 8-bit ASCII format. Display data is maintained by the HD44780 in an 80 byte display data RAM buffer. The controller contains a character generator ROM capable of generating bit patterns for 192 characters. Custom characters may be defined by initializing the character generator RAM. The HD44780 controller includes all the logic and timing circuits required to convert ASCII character codes to serial bit patterns for on/off selection of individual display pixel elements. The HD44100 LCD driver expands the segment drive capability from 40 segments to 80 segments.

The controller accepts high level commands from the CPU to establish display format, length, and cursor positioning. On command, the controller is capable of shifting the position of characters in the display.

The LCD controller and driver chip set form a complete system solution for driving multiplexed displays, minimizing the component count and freeing the CPU for other tasks.

Keyboard

The TI-74 console is populated with 62 keys arranged in an 8x8 matrix (ON and RESET are not considered as part of the matrix). The processor strobes one of 8 rows and decodes key actuations by reading the 8 keyboard column inputs. The TMS70C46 external memory data bus ("E" port) is used to strobe the keyboard rows. Keyboard column lines are connected to the TMS70C46 "A" port inputs. The keyboard scan algorithm resides in the TMS70C46 on-chip ROM because the memory data bus is used to strobe the keyboard. External memory cycles cannot be executed during scanning operations.

During periods of inactivity, when the processor is not running a BASIC program or performing a calculation, the CPU places itself in a low-power (idle) mode. Any interrupt releases the idle mode, enabling the CPU to resume normal instruction execution. The TMS70C46 contains logic which is capable of generating a maskable interrupt upon detection of any key actuation. Placing the processor in a low-power mode while waiting for a key input results in a substantial increase in the battery life of the product.

The "OFF" key is included in the keyboard matrix. When the CPU detects actuation of this key, several software housekeeping tasks including clearing pending operations, closing files, and calculating/storing the RAM checksum are completed prior to actual power down. When the housekeeping activities are completed, a TMS70C46 output control line is toggled to turn off the power supply.

The "ON" key connects directly to the power supply circuitry. Actuation of the "ON" key momentarily activates the power supply. After the CPU completes power-up reset and software initialization sequences, an output control line is toggled to latch the power supply on until subsequent actuation of the "OFF" key or automatic power down.

Since the TI-74 software environment is controlled by the user, a "RESET" key is available to provide a hardware reset of the console and peripherals. The "RESET" key connects directly to the CPU reset input.

five positions on the 64 position matrix are not occupied. Contact closures in these matrix positions activate either self-test program sequences or signature-analysis routines to facilitate manufacture and checkout of the product.

The key contact system consists of interleaved, gold-plated etch runs on a PCB which are shorted by carbon buttons mounted in an elastomeric sheet. The elastomeric sheet/carbon button system results in reliable keyboard operation in excess of 500,000 key actuations.

Power Supply

As illustrated in the power supply block diagram (Figure 6), the TI-74 can operate from 4 "AAA" alkaline batteries or 6.3 volts D.C. supplied to pin 2 of the I/O port. Battery voltage is also provided to pin 1 of the I/O port to power the CI-7 Cassette Interface Cable. Series diodes are used to protect the power input, output, and batteries against reverse current. Voltage to maintain constant memory in RAM and proper I/O bus levels is routed unswitched to these circuits.

Pressing the ON key biases on a bipolar transistor circuit to apply power to the switching regulator, which in turn applies power to the negative voltage converter. The -5 volt output gates on FET Q1, latching on the switch circuit.

To assure the integrity of memory and bus operations when the TI-74 is turned off, the power-down sequence is controlled by the microprocessor. The OFF key functions as a part of the normal key matrix. When the OFF key is detected, pending operations are completed before the microprocessor commands the power supply switch circuit to turn off by taking output B2 (pin 5) to a LOW (logic level 0). This action results in the removal of the gate voltage on FET Q1 which switches the current to the regulator circuit off.

Regulation of the VDD supply (+5 volt) is accomplished with a high degree of efficiency using an RC-4193 switching regulator. Because this device is designed specifically for use in battery operated instruments, a minimum number of external components are needed for the regulator circuit. A low-battery sense provides an output to the CPU for control of the low-battery indicator on the display. The level sensed as "low" is set to approximately 4.5 volts.

The negative voltage converter circuit, mentioned in the discussion of the power-up sequence, consists of a 7660-type CMOS voltage converter and two 10uf electrolytic capacitors. In addition to being used to latch the switch circuit on, The -5 volt output is supplied to a resistive voltage divider to create the bias-voltage levels required by the display system.

To insure that the power supply outputs reach proper levels before the CPU begins to execute instructions, a power-up reset circuit is included. VDD is connected to the base bias network of a bipolar transistor through a series capacitor so that the transistor conducts and pulls RESET LOW during the charge time of the capacitor. The RC combination is selected to maintain RESET LOW for approximately 10 ms (more than twice the time required for VDD to reach maximum).

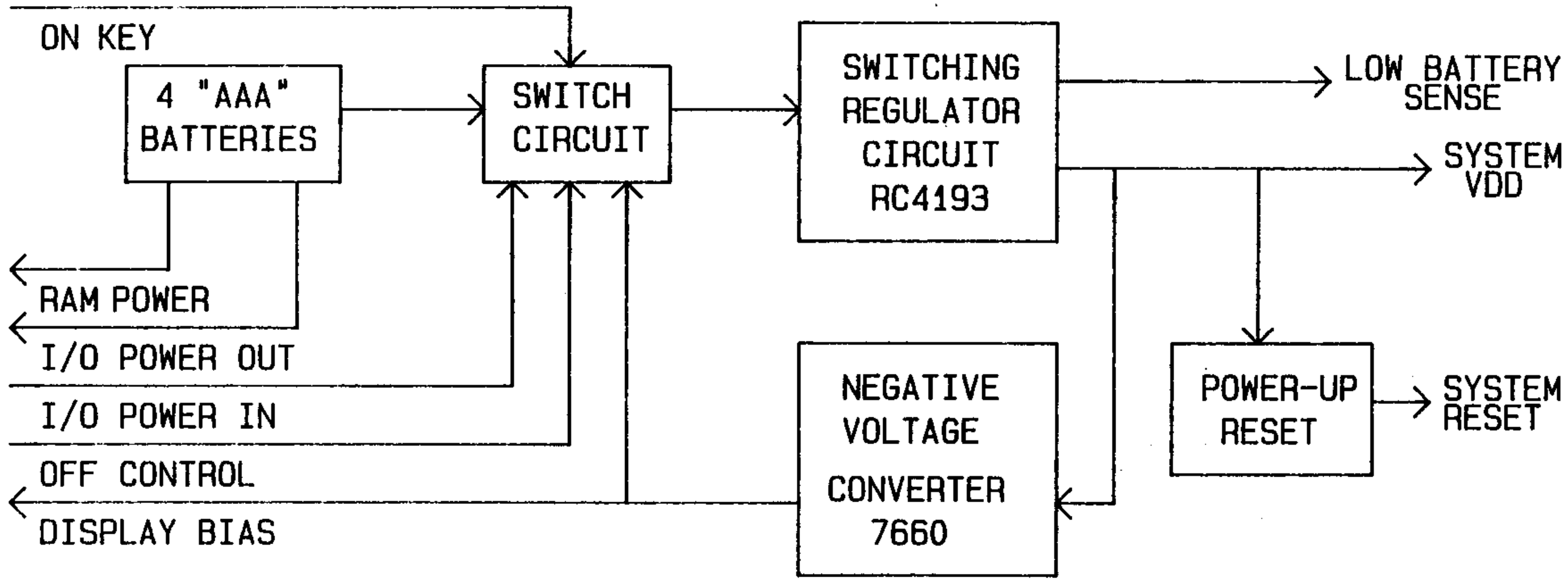


Figure 6. TI-74 Power Supply Block Diagram

Packaging

The electronics are mounted on two printed circuit boards within the console. The keyboard/display PCB includes the key contacts, LCD and LCD driver/controller IC's. The display is secured to the PCB by a metal bracket. Electrical connection to the PCB is established with two zebra strips running the length of the glass.

The CPU printed circuit board contains the TMS70C46, 8K RAM, 32K ROM, power supply circuit, I/O port expansion connector, and memory expansion connector. Electrical connection between the two printed circuit boards is accomplished with a flat ribbon cable.

Summary

The TI-74 hardware design is the result of incorporating some of the techniques used in fully integrated single chip calculators, with the capability of an 8 bit microprocessor. The TMS7000 modular architecture allowed the design of custom I/O circuitry to integrate as many product features as possible while still maintaining enough flexibility for future calculator designs. This design approach results in a system architecture which is both powerful and flexible and at the same time very cost effective.

I/O EXPANSION

I/O expansion is accomplished through the DOCK-BUS intelligent peripheral bus.

General Features

The bus is designed to provide data transmission between a calculator and peripherals. It is organized in a master-slave arrangement with the controlling calculator as the master and the peripherals as slaves. In this manner, the calculator on the bus acts to control data activity. Normal communication will be initiated by a command message from the calculator to a peripheral. The peripheral responds to the calculator with a data or status message to signal completion of the command.

Transmissions on the bus are defined in the context of a "message frame" which consists of a command message from the master and a response message from the slave. This message concept requires that each peripheral be intelligent enough to decode information and bus status and be able to follow protocol. Each peripheral must contain a microcomputer for the bus interface and control functions.

Signal Descriptions

A variety of peripherals can be interfaced to the TI-74 via the DOCK-BUS connector located at the rear of the console. Table 3 shows the signals available on the bus and Figure 7 shows the relative position of the pins in the connector.

Table 3. I/O Pin Description

Signal	Description	Pin #
PO	System Power Distribution Line - Output	1
PI	System Power Distribution Line - Input	2
D0	Data Bit - Least Significant Bit	3
D1	Data Bit	4
D2	Data Bit	5
D3	Data Bit - Most Significant Bit	6
HSK	Handshake - I/O Timing Control Line	7
BAV	Bus Available - I/O Traffic Control Line	8
RESET	System Reset Line	9
GND	Common Ground Line	10

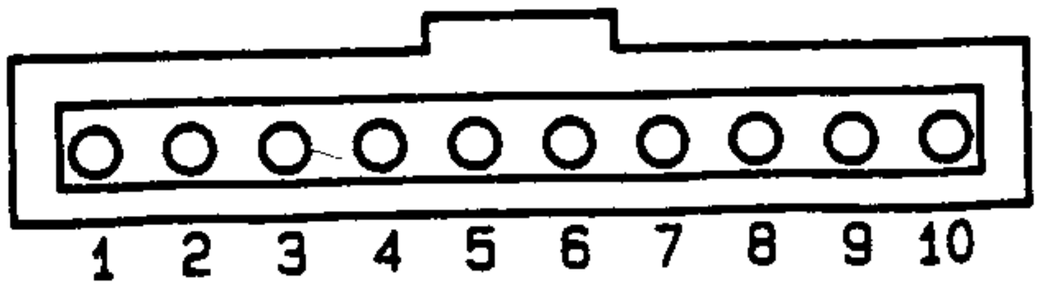


Figure 7. I/O Port Relative Pin Position

All peripherals are connected in parallel on the bus. The speed of transmission of the data bus is controlled by the handshake line (HSK). The bus available (BAV) signal is used to designate the beginning of a command message from the master device.

D0-D3 - Four I/O Data Bits

These four bits are used to send data on the DOCK-BUS one nibble at a time. The lower nibble of the eight bit byte will be sent first, followed by the most significant nibble.

GND - Floating Reference Ground Line

A common ground line is run to all the peripherals in order to reference all voltages equally to the same point. This is a signal reference and should not be used as a power line or be tied to earth-ground.

P0, PI - System Power Distribution

These lines can be used to distribute power for the system. P0 is used to supply +6 volts to system components from the TI-74 batteries. PI is used to power the TI-74 from a peripheral. By utilizing P0 or PI, the entire system can be powered from one source.

RESET - System Reset

This line can be used to distribute reset signals from one system component to another.

HSK - Handshake, I/O Timing Control Line

The handshake (HSK) line is used by both the master and the peripherals to signal the fact that data is on the bus. HSK is an open-drain, CMOS output in order to allow it to pulled LOW by any device on the bus. A device will pull HSK LOW to signal to the other devices on the bus that data is available from that device which may be read by the other devices on the bus.

HSK will be held LOW by the receiving devices until they have latched the four bits of data on the bus. If the transmitter is slower than than the receivers, the transmitter may be the device that dictates the bus speed.

When a device is not interested in the data being transmitted. it may disable itself from the bus and wait for the next message frame (denoted by a BAV transition from HIGH to LOW). Non-active devices need not participate in the handshake activity.

BAV - Bus Available, I/O Traffic Control Line

Each message on the I/O lines consists of two parts; the command message and the response message. Whenever a message transfer is in progress, bus available (BAV) will be held LOW by the master until the message is complete. This tells the other peripherals that the bus is in use.

Handshake Timing Parameters

A timing diagram for the handshake line is shown in Figure 8. Table 4 lists the specifications for timing of pertinent events.

Table 4. Handshake Timing Specifications

Handshake Timing Parameters for Master Devices

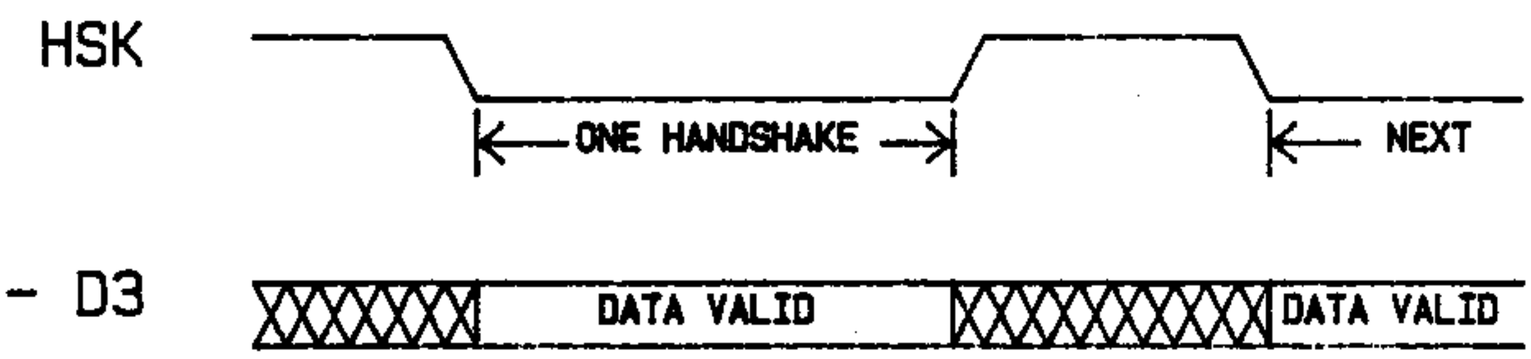
EVENT	MIN.(usec)	MAX.(usec)
HSK LOW TO DATA VALID	-	0.5
HSK LOW (XMIT) TO HSK LOW (BUS)	-	3
HSK LOW (BUS) TO HSK LOW (RCV)	-	12
HSK LOW (XMIT) TO HSK HIGH (XMIT)	8	-
HSK HIGH TO DATA HIGH	0	2.5 *
HSK HIGH TO HSK LOW	8	20.000 **

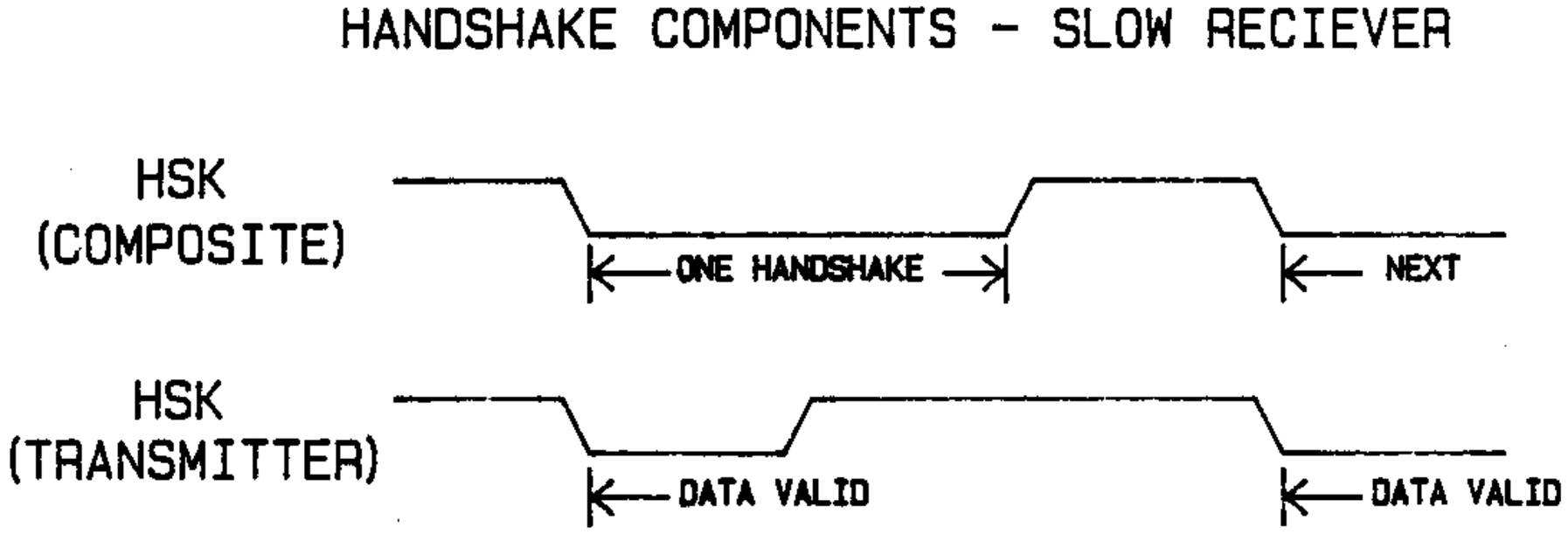
Handshake Timing Parameters for Slave Devices

EVENT	MIN.(usec)	MAX.(usec)
HSK LOW TO DATA VALID	-	0.5
HSK LOW (XMIT) TO HSK LOW (BUS)	-	3
HSK LOW (BUS) TO HSK LOW (RCV)	-	5
HSK LOW (XMIT) TO HSK HIGH (XMIT)	15	-
HSK HIGH TO DATA HIGH	0	2.5 *
HSK HIGH TO HSK LOW	15	20,000 **

* Because data is output via open-drain buffers and HSK going HIGH causes ones to be written to the data output latches, data on the bus will be set HIGH by the rising edge of HSK.

** Within a message frame (when BAV is held LOW) HSK has 20 ms to go LOW or a bus timeout (error) condition results.





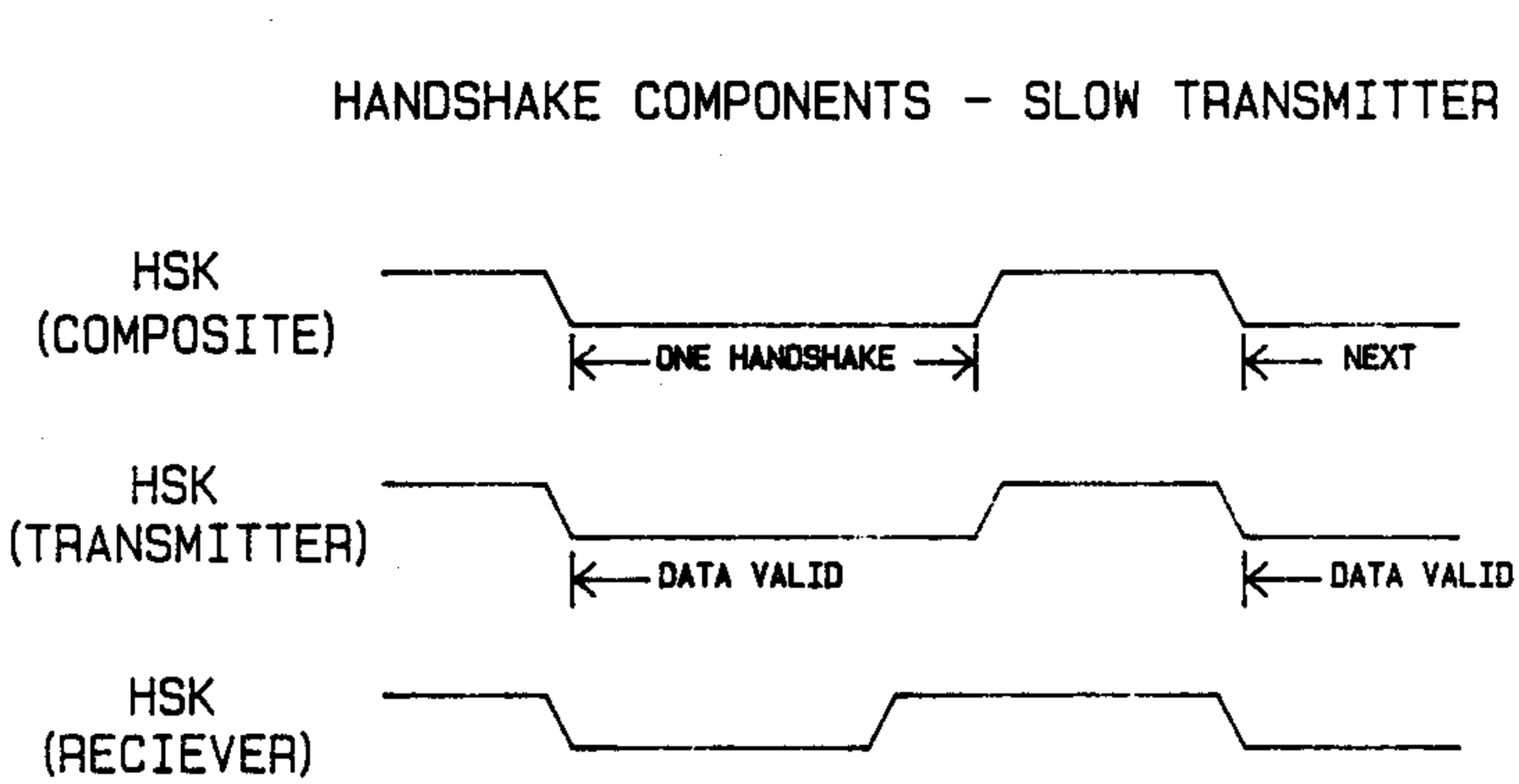


Figure 8. Handshake Timing Diagram

Bus-Available Timing Parameters

A timing diagram for BAV is shown in Figure 9. Table 5 lists the specifications for the timing of pertinent events.

Table 5. Bus-Available Timing Specifications

EVENT	MIN.(usec)	MAX.(usec)
BAV LOW TO HSK LOW	5	20,000
HSK HIGH TO BAV HIGH	1	-
END OF COMMAND TO START OF RESPONSE	10	-
BAV LOW (SLAVE) TO BAV HIGH (SLAVE)	*	*
BAV HIGH TO BAV LOW (SLAVE)	2000 *	-

* These two timing parameters occur only when a slave peripheral polls the master (by pulling BAV LOW) at the same time the master BAV goes LOW. The slave will then continue to hold BAV LOW until it recieves the first HSK signal from the master. The slave will then release BAV and wait 2 ms after the master has released BAV to poll again.

Figure 9. Bus-Available Timing Diagram

Data Transfer Order

All data and overhead information is sent in increments of one byte. As bytes are transmitted, the least significant nibble is placed on the bus first, followed by the most significant nibble. Whenever 16 bit (two byte) fields are sent, the least significant byte is transmitted first.

BUS PROTOCOL DESCRIPTION

As mentioned earlier, the data bus transmits command and response messages in the context of a message frame. In general, the transmission of one command message from the master device will cause a response message to be transmitted back from the slave device selected.

Each message contains overhead information to indicate such things as the slave device selected, the command code for the task to be preformed, and the data length. The BAV signal specifies the start of a message frame. When the master device starts a message frame, it first pulls the BAV line LOW. The command message from the master then follows. The falling edge of BAV alerts all slave devices to look for the two-nibble device code which is always transmitted first in the command message (again, least significant nibble first). The BAV signal does not return to the HIGH level until the message frame is complete.

The first two nibbles of the command message always contain the device code of the slave device to be addressed. All devices on the bus will read this number and test for a match. Each device must therefore have a unique device code. After the device code has been sent, all devices except the one selected will ignore all further data in the message. Device code >00 is reserved to be recognized by all devices, but only the RESET and NULL commands are valid. All other commands for device code >00 should be ignored by the peripherals. The peripheral hardware must be designed so that non-selected devices will not participate in the handshake sequence until the next falling edge of BAV. In this way the bus will operate at the maximum data rate of the two "talking" devices once the device code has been transmitted.

Any device may extend the time to process data or wait for an operation to complete by holding HSK LOW until it is ready to start the next operation. Whenever HSK is high during a message, it must go LOW within 20 ms or the reciever may time out (time-outs are not required for peripheral slave devices).

The DOCK-BUS message format is intended to be flexible enough not to limit the types of peripherals which may be designed for the system. The following discussion of bus protocol includes all the features which are supported in BASIC. However, no inference may be made from any feature of bus protocol about the plans of Texas Instruments for future development of peripheral devices.

Command Message

Communication between the calculator and a peripheral consists of a command message and a response message (in some cases there will be no response to the command). Table 6 lists the data contained in a command message and a description of each segment is made in the following paragraphs.

Table 6. Command Message Format

FIELD	BYTES
DEVICE CODE	1
COMMAND CODE	1
LOGICAL UNIT NUMBER	1
RECORD NUMBER	2
BUFFER LENGTH	2
DATA LENGTH	2
DATA	VARIABLE

Note: All data structures shown in this document relate to the format used by the TI-74 standard I/O subroutines. This is not the only format possible, but must be adhered to if BASIC is to be used to access peripheral devices.

Device Code

Since the device code is one byte in length, 255 unique codes are avalable. The PC-324 Printer has been assigned device code 12. Several other device codes have been reserved for possible new product development. The details of these assignments are not presented here since new device-code reservations may be made at any time. If a peripheral is to be designed for a unique or personal application, the designer need only take care not to assign duplicate codes for any system components. Production of peripheral devices for sale or widespread use will require that the designer apply for a third-party product development license from Texas Instruments.

Command Code

The command-code field tells the slave device the nature of the operation to be performed. Table 7 lists the standard code assignments. It may be noted that several of these commands are not supported by TI-74 BASIC. However, any command code, 0 to 255, may be output to the DOCK-BUS by using the I/O subprogram described in the TI-74 Programming Reference Guide.

Table 7. Command Codes (in hexadecimal)

CODE	FUNCTION
>00	OPEN
>01	CLOSE
>02	DELETE OPEN FILE
>03	READ DATA
>04	WRITE DATA
>05	RESTORE FILE
>06	DELETE
>07	RETURN STATUS
>08	YOU ARE THE MASTER
>0C	VERIFY READ/WRITE OPERATION
>0D	FORMAT AND CERTIFY MEDIA
>10	TRANSMIT BREAK
>11	PROTECT/UNPROTECT FILE
>12	READ SECTORS
>13	WRITE SECTORS
>14	MODIFY FILE NAME
>20	RETURN TO COMMAND MODE
>FE	NULL OPERATION
>FF	BUS RESET

Certain peripherals may extend this list for device dependent features. An example of this type of extension may be found in the TI-74 Programming Reference Guide under "Using Optional Accessories; Controlling the Printer from BASIC".

Logical Unit Number

The Logical Unit Number (LUNO) is reserved for devices which may contain separately addressable segments on one physical unit (e.g. files on disks). Each currently open file on a device must have a unique non-zero code in this byte so the device may continue to relate the commands to the proper files. The OPEN command provides both the LUNO and the file name. The LUNO may range from 1 through 255. Note that the LUNO need not be checked by units that do not support multiple files.

Record Number

The Record number is reserved for devices which support relative-record (random-access) files or for devices which extend the standard command-code set. It is ignored when a file is opened as a sequential file. This field must be maintained by the application software for compatibility with random-access devices. It is zeroed before the OPEN and RESTORE operations for normal access to devices. It should also be incremented by the application program after successful READ, WRITE, and VERIFY operations. The first record of a file is record 0. Peripherals

that do not support random-access files may ignore this field. From TI-74 BASIC, the value for this field is loaded from the REC clause of the various data-handling statements.

Buffer Length

This field indicates the size of the data buffer for receiving data from a peripheral during the current bus operation. It is used by the master's IOS (internal operating system) to check that the length of the returned data does not exceed the buffer size. If the master's IOS determines that more data is being recieved than can be put into the buffer, the operation will be aborted and a buffer-length error wil result. The data returned by the peripheral in the response message must not exceed the length specified here. This length is exclusive of the data length and return status bytes which form part of the response message. The status byte and data length replace their old fields in the PAB (Peripheral Access Block - a contiguous block of RAM used by the I/O subroutines).

Data Length

The data length field gives the number of bytes of data which follow in the data field.

Data

This field contains the data to be written to the peripheral device. The use of the data depends on the command code. If the data length field is zero, the data field is not present.

Response Message

The response message is outlined in table 8, followed by a description of each field.

Table 8. Response Message Format

FIELD	BYTES
DATA LENGTH	2
DATA	VARIABLE
OPERATION STATUS	1

Data Length

This field specifies the number of bytes of data which follow in the data field.

Data

This field contains the data to be returned to the master device; for example, on a read-data operation. If the data length field is zero then this field will be omitted. Whenever the true data length cannot be determined at the time the length is sent, the data field will be padded with trailing blanks (>20) for display-type files, and with trailing zeros for internal-type files.

Operation Status

The final segment of the response message is the operation-status field; containing the status of the current operation. Table 9 lists the assigned response codes in both hexadecimal and decimal.

Table 9. Operation Status Assignments

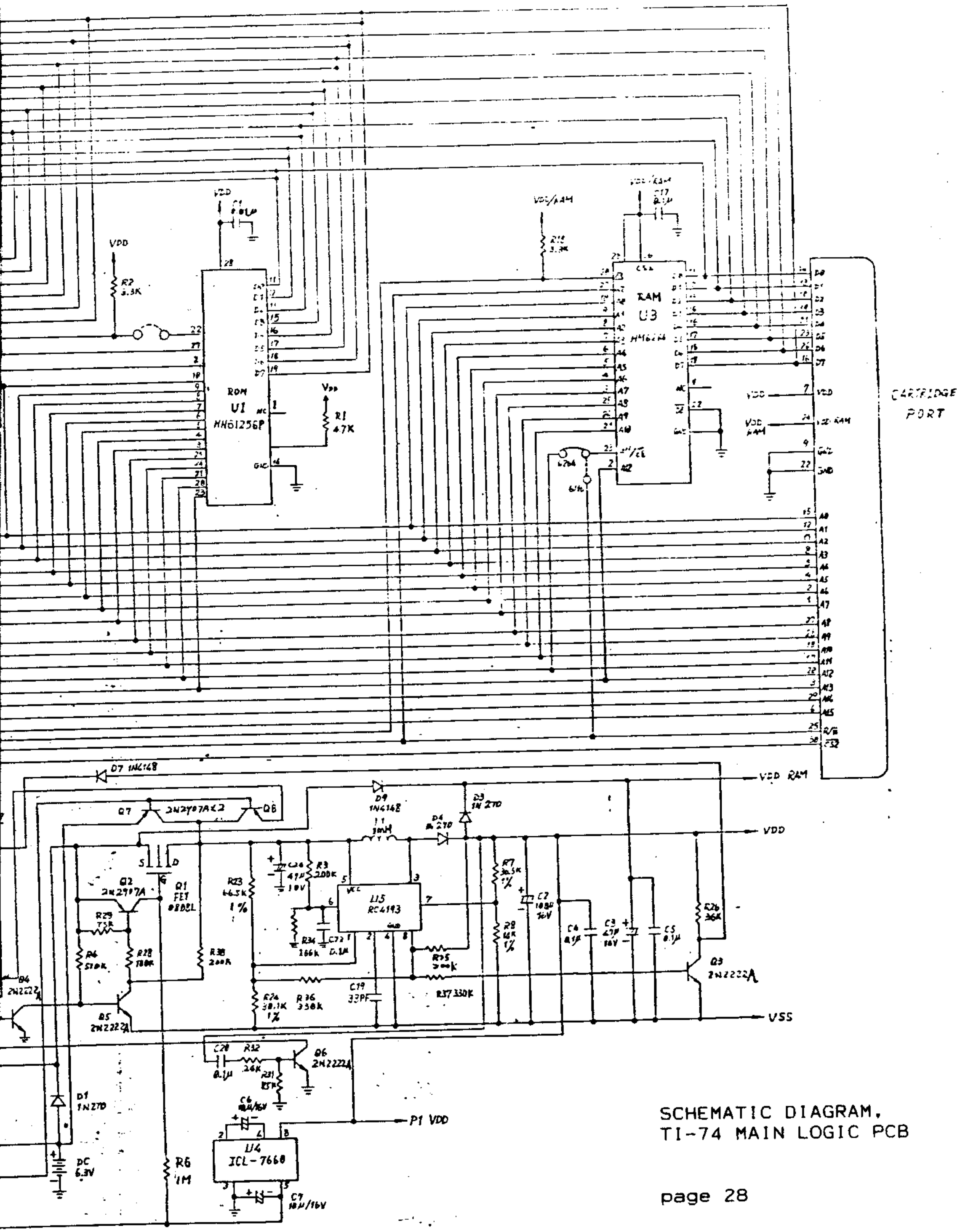
HEX	DEC	INTERPRETATION
>00	0	NORMAL OPERATION COMPLETION
>01	1	DEVICE/FILE OPTION ERROR
>02	2	ATTRIBUTE ERROR
>03	3	FILE/DEVICE NOT FOUND ERROR
>04	4	FILE/DEVICE NOT OPEN ERROR
>05	5	FILE/DEVICE ALREADY OPEN
>06	6	DEVICE ERROR
>07	7	EOF ERROR
>08	8	DATA/FILE TOO LONG ERROR
>09	9	WRITE PROTECT ERROR
>0A	10	(NOT REQUESTING SERVICE)
>0B	11	DIRECTORY FULL ERROR
>0C	12	BUFFER SIZE ERROR
>0D	13	UNSUPPORTED COMMAND ERROR
>0E	14	FILE NOT OPENED FOR WRITE
>0F	15	FILE NOT OPENED FOR READ
>10	16	DATA ERROR (CHECKSUM FAILURE IN DEVICE)
>11	17	FILE ORGANIZATION (RELATIVE / SEQUENTIAL) INCORRECT OR NOT SUPPORTED
>12	18	IMPROPER FILE PROTECTION INFORMATION SPECIFIED
>13	19	APPEND MODE NOT SUPPORTED
>14	20	OUTPUT MODE NOT SUPPORTED
>15	21	INPUT MODE NOT SUPPORTED
>16	22	UPDATE MODE NOT SUPPORTED
>17	23	FILE TYPE (INTERNAL / DISPLAY) INCORRECT OR NOT SUPPORTED
>18	24	VERIFY ERROR
>19	25	LOW BATTERIES IN PERIPHERAL
>1A	26	UNINITIALIZED MEDIA
>27	39	PERIPHERAL BUS ERROR (TIMING ERROR)

Table 9 is continued on the next page.

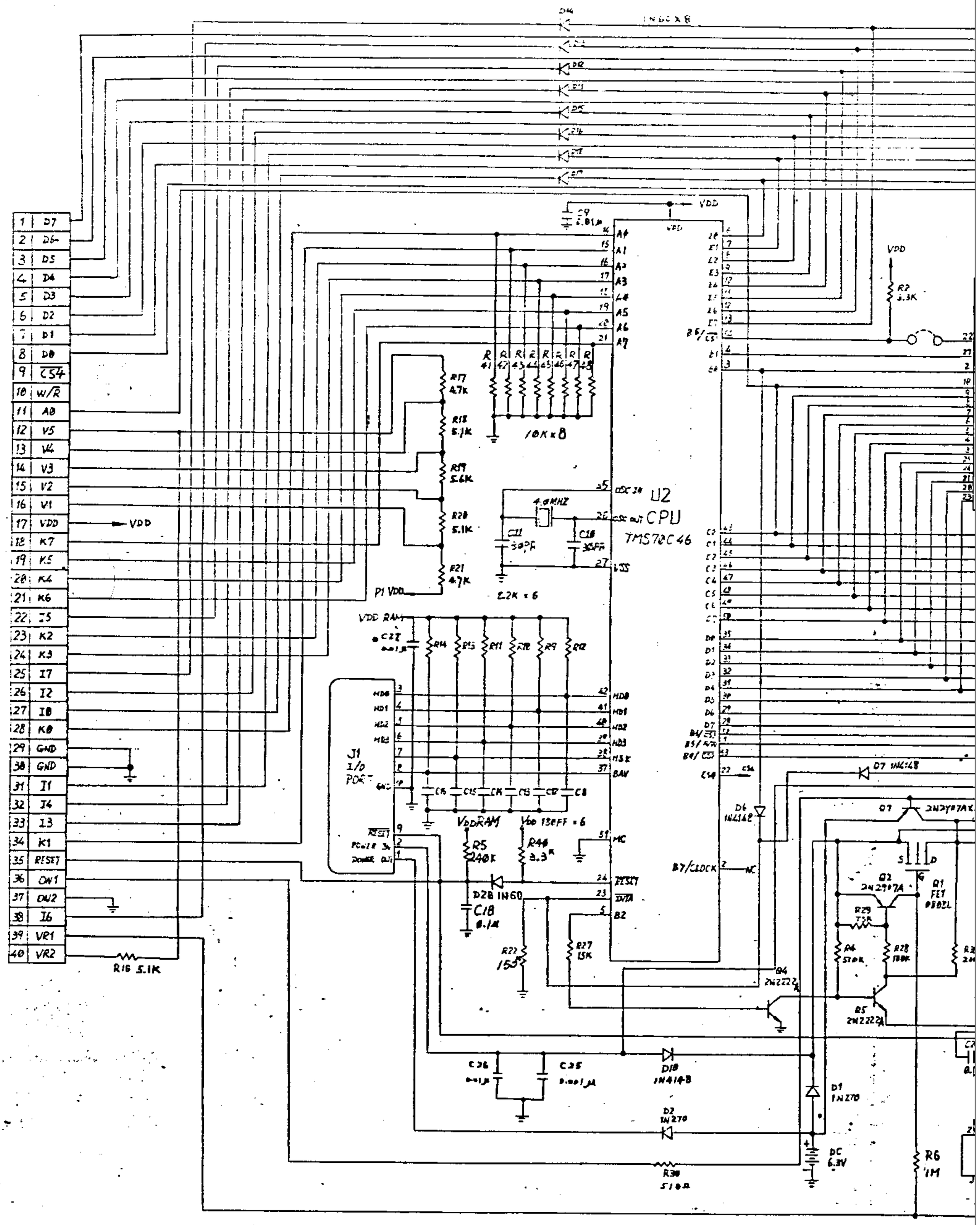
Table 9. Operation Status Assignments (continued)

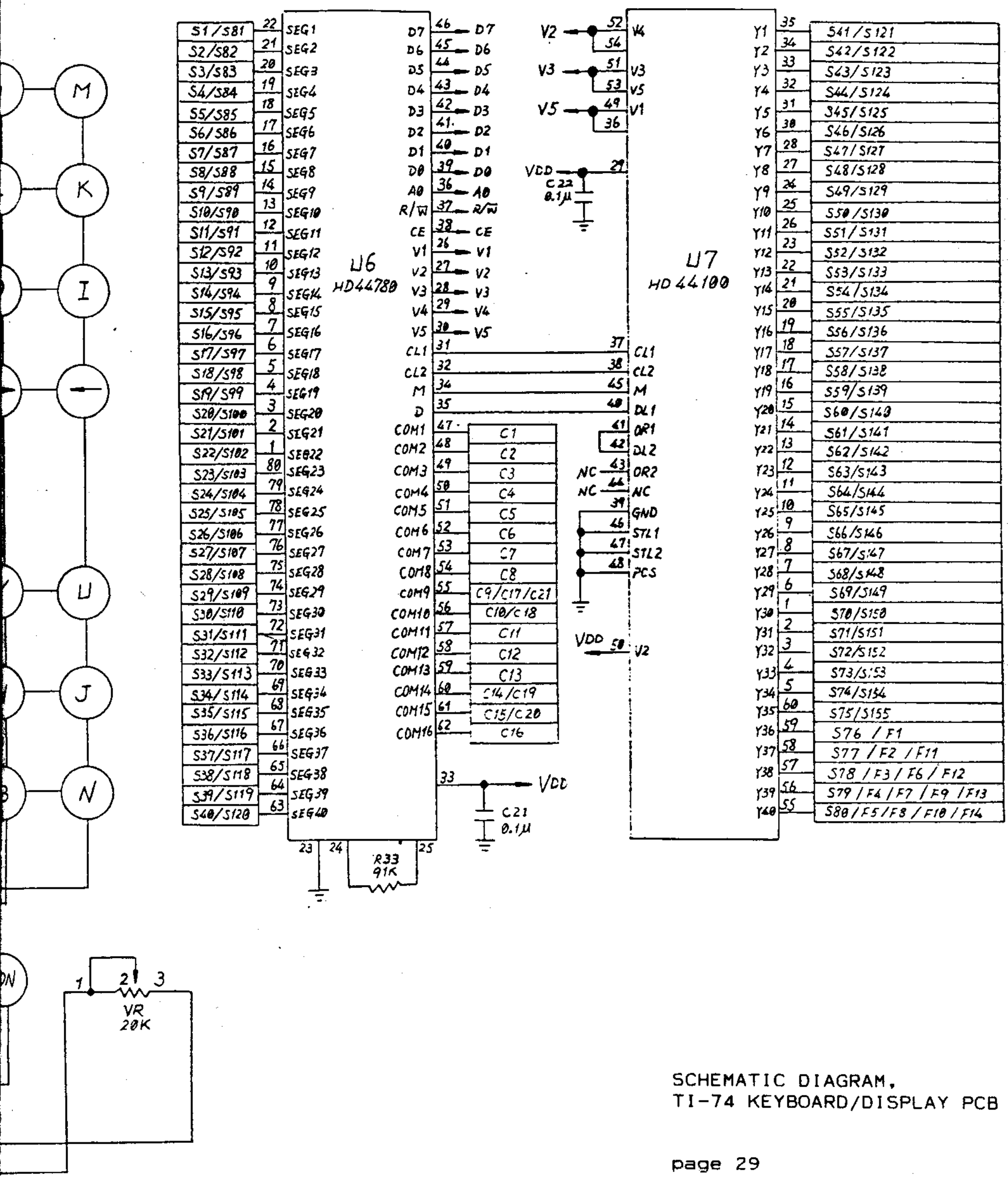
HEX	DEC	INTERPRETATION
>1C	28	FILE IS DELETE PROTECTED
>1D	29	CARTRIDGE NOT INSTALLED AS MASS STORAGE
>1E	30	RESTORE NOT ALLOWED IN THIS MODE
>1F	31	INVALID FILE NAME
>20	32	MEDIA FULL
>21	33	ATTEMPTED TO EXCEED MAXIMUM NUMBER OF LUNOS
>22	34	INVALID DATA (TOO SHORT OR INCORRECT CONTENTS)
>23	35	FILE NAME ALREADY EXISTS
>24	36	RECORD TYPE (FIXED / VARIABLE) INCORRECT OR NOT SUPPORTED
>FE	254	ILLEGAL IN SLAVE MODE
>FF	255	BUS TIME OUT ERROR

Error codes >50 - >EF are reserved for device dependent errors.



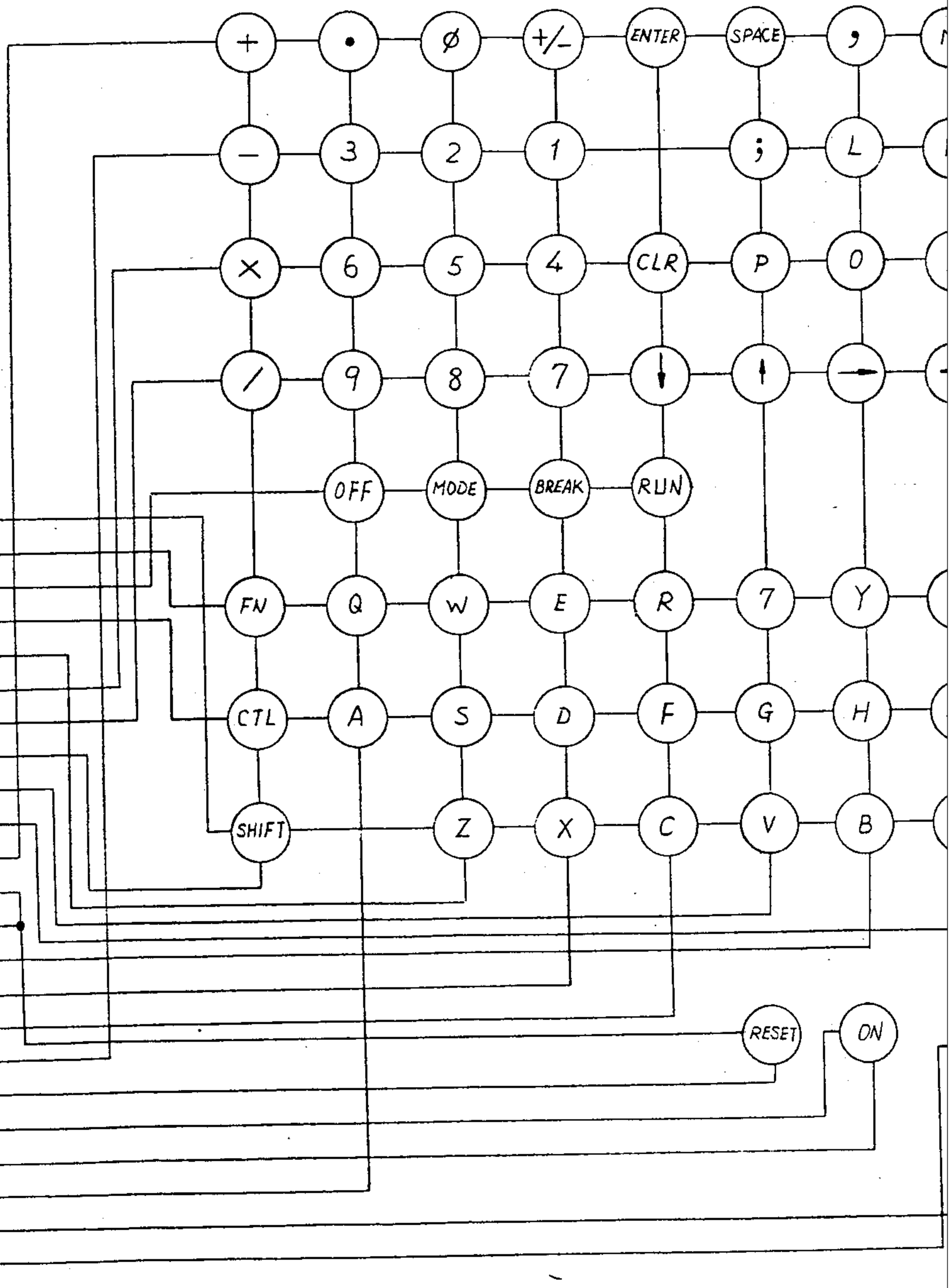
SCHEMATIC DIAGRAM,
TI-74 MAIN LOGIC PCB

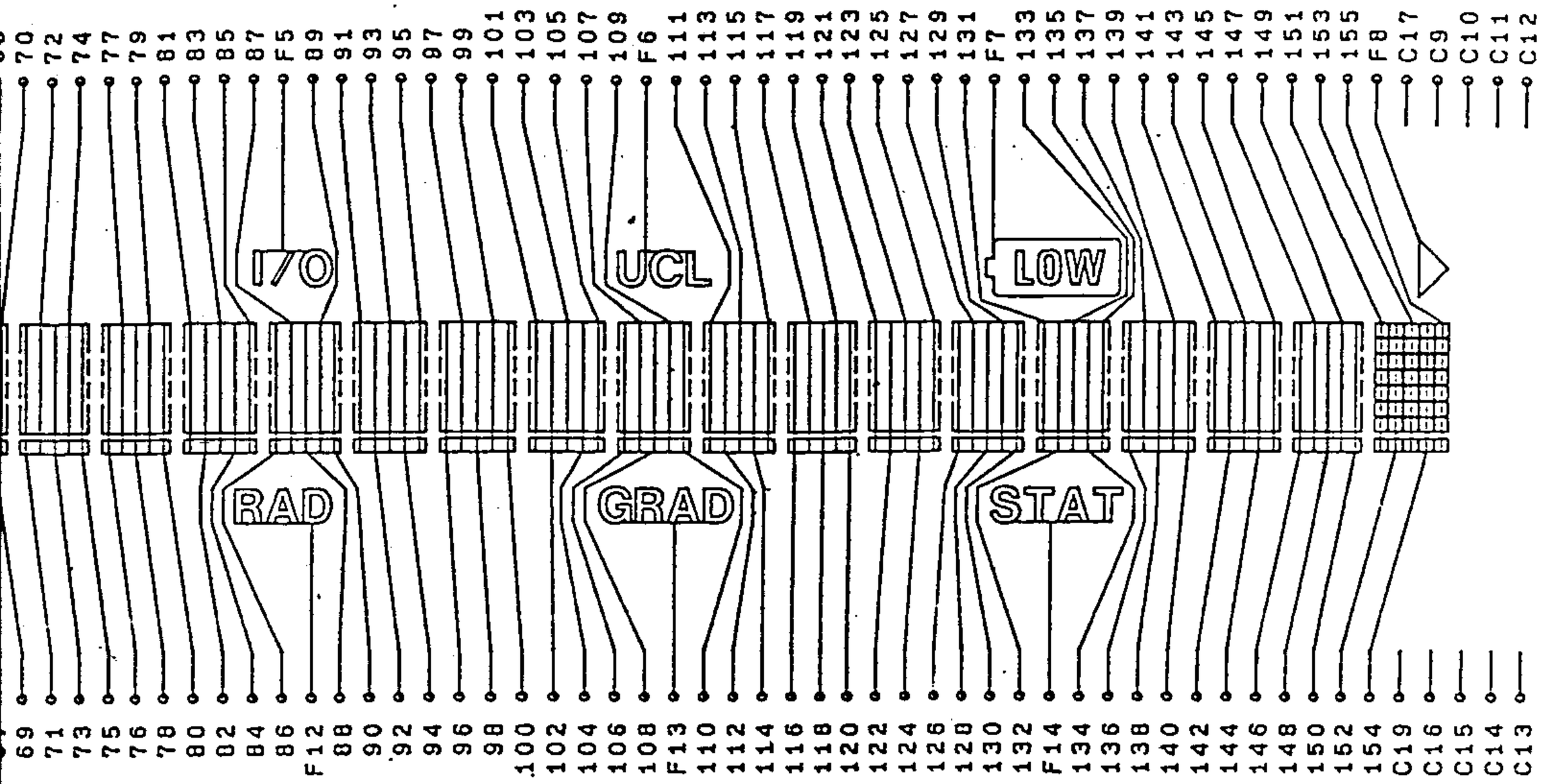

INDEX R
VDD
C9 0.01μ
U2 CPU TMS70C46
4.0MHZ
C11 30pF
C10 30pF
10K x 8
2.2K x 6
VDD RAM
C22 0.1μ
R14 R13 R11 R10 R9 R12
J1 I/O PORT
C16 C15 C14 C13 C12 C8
VDDRAM
Vdd 150FF x 6
R5 240K
R44 3.3K
D28 IN60
C18 0.1μ
R22 15K
R27 15K
B7/CLOCK
Q4 2N2222
Q2 2N2907A
Q1 FET BS62L
R29 33K
R4 510K
R28 180K
R5 2N2222
C26 0.1μ
C25 0.001μ
D18 IN414-B
D2 IN270
D1 IN270
DC 6.3V
R30 510Ω
R6 1M
D7 IN4148
D6 IN4148
Q7 2N2907AK
R16 5.1K
P1 VDD
R17 47K
R18 5.1K
R19 5.6K
R20 5.1K
R21 47K
R41 R42 R43 R44 R45 R46 R47 R48
D1 D2 D3 D4 D5 D6 D7 D8
INDEX R
VDD
R2 3.3K
C54
D7 IN4148
D6 IN4148
Q7 2N2907AK
Q2 2N2907A
Q1 FET BS62L
R29 33K
R4 510K
R28 180K
R5 2N2222
Q4 2N2222
D1 IN270
D2 IN270
DC 6.3V
R6 1M
R30 510Ω
D18 IN414-B
C26 0.1μ
C25 0.001μ
R22 15K
R27 15K
D28 IN60
C18 0.1μ
R5 240K
R44 3.3K
R16 5.1K
P1 VDD
VDD RAM
VDDRAM
Vdd 150FF x 6
B7/CLOCK
1 D7
2 D6
3 D5
4 D4
5 D3
6 D2
7 D1
8 D0
9 C54
10 W/R
11 A0
12 V5
13 V4
14 V3
15 V2
16 V1
17 VDD
18 K7
19 K5
20 K4
21 K6
22 I5
23 K2
24 K3
25 I7
26 I2
27 I0
28 K0
29 GND
30 GND
31 I1
32 I4
33 I3
34 K1
35 RESET
36 ON1
37 ON2
38 I6
39 VR1
40 VR2



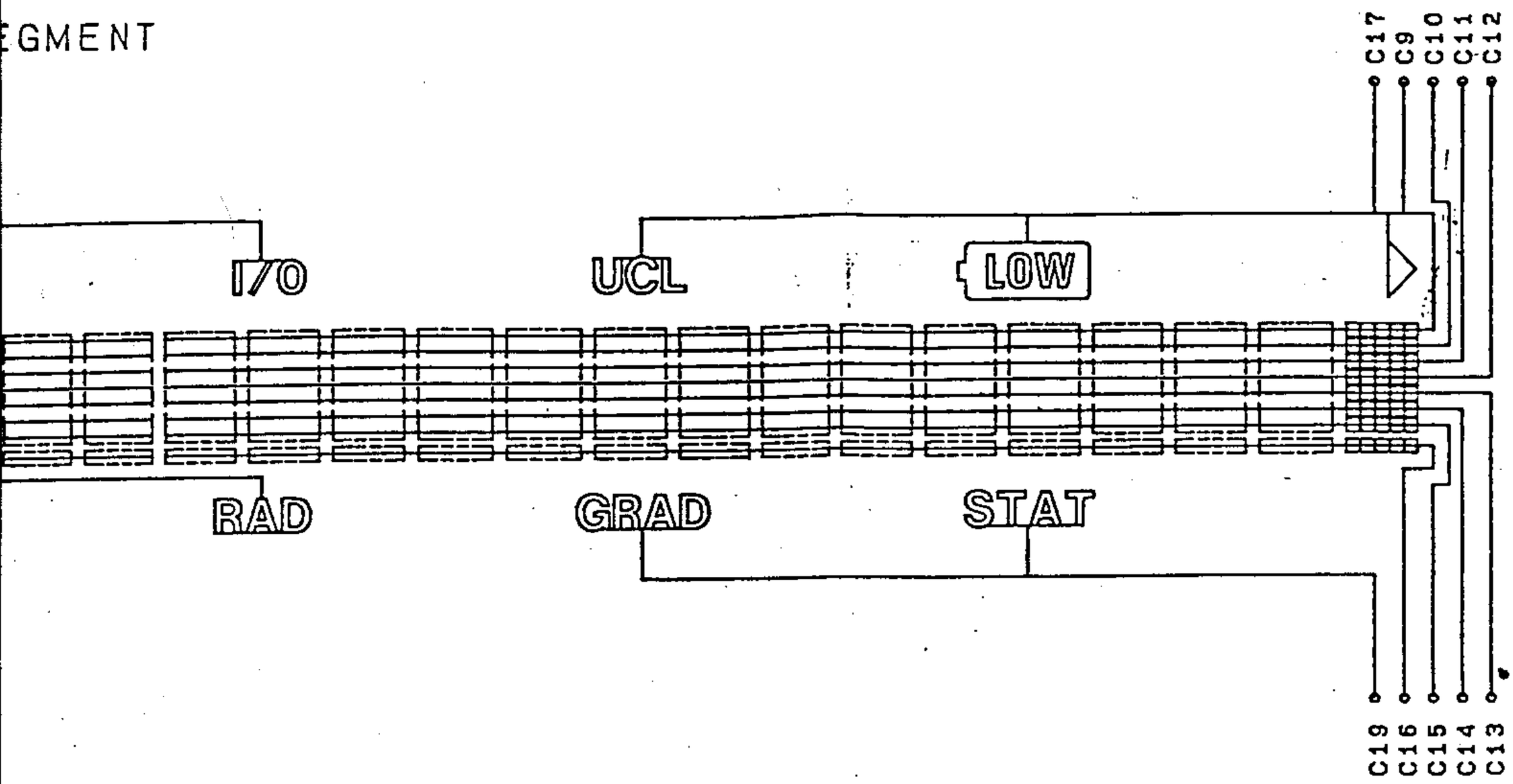
SCHEMATIC DIAGRAM,
TI-74 KEYBOARD/DISPLAY PCB

1	D7
2	D6
3	D5
4	D4
5	D3
6	D2
7	D1
8	D0
9	$\overline{CE}$
10	W/$\overline{R}$
11	A0
12	V5
13	V4
14	V3
15	V2
16	V1
17	VDD
18	K7
19	K5
20	K4
21	K6.
22	I5
23	K2
24	K3
25	I7
26	I2
27	I0
28	K0
29	GND
30	GND
31	I1
32	I4
33	I3
34	K1
35	$\overline{RESET}$
36	ON1
37	ON2
38	I6
39	VR1
40	VR2



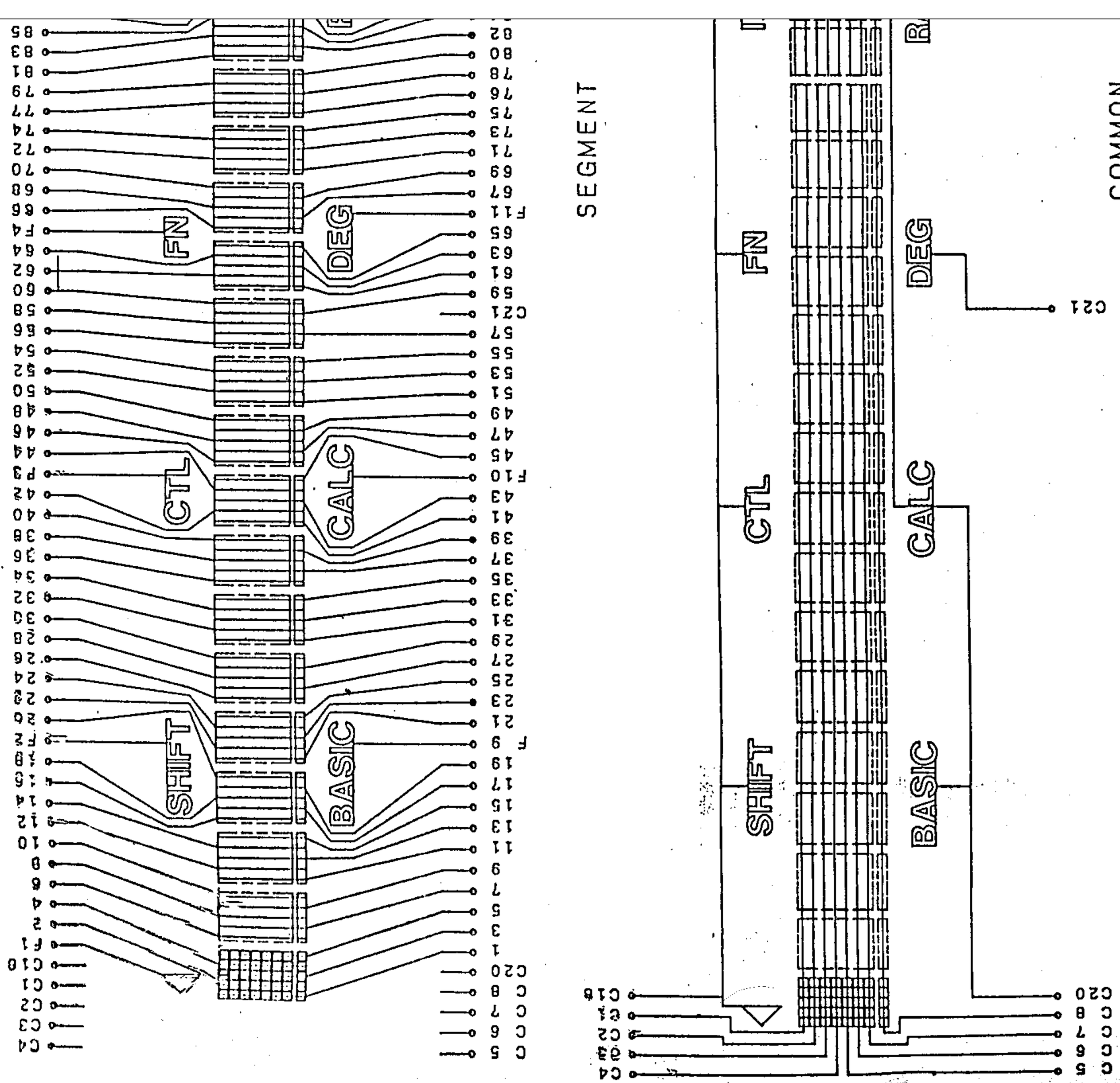


EGMENT



COMMON

SCHEMATIC DIAGRAM,
TI-74 DISPLAY


SEGMENT
COMMON
FN
DEG
CTL
CALC
SHIFT
BASIC
F11
C21
F10
F9
C20
C19
C18
C17
C16
C15
C14
C13
C12
C11
C10
C9
C8
C7
C6
C5
C4
C3
C2
C1
F1
F2
F3
F4